

Normalization

อ. พรอนันต์ เอี่ยมจรชัย
สาขาวิชาเทคโนโลยีสารสนเทศ

คำศัพท์ที่เกี่ยวข้อง

- FD (Functionally Dependent or Functional Dependency)
- Full FD (Full Functional Dependency)
- Partial Dependency
- Transitive Dependency
- Multivalued Dependency

FD (Functional Dependency)

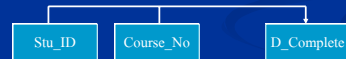
- ความสัมพันธ์ระหว่างค่าของ Attribute แบบฟังก์ชัน
- Attribute หนึ่งหรือมากกว่าหนึ่ง Attribute ประกอบกันสามารถระบุค่าของ Attribute อื่นๆ ในทิวเปิล (Tuple) หนึ่งได้ชัดเจน
- แบ่งได้เป็น 2 รูปแบบ คือ
 - แบบ 1 : 1 ค่าของ Attribute ทางด้านซ้าย 1 ค่า บ่งบอกค่าของ Attribute ทางขวา 1 ค่า
 - แบบ m : 1 ค่าของ Attribute ทางด้านซ้ายตั้งแต่ 1 ค่าขึ้นไป บ่งบอกค่าของ Attribute ทางขวา 1 ค่า

ตัวอย่าง Functional Dependency

- แบบ 1 : 1



- แบบ m : 1



- ความสัมพันธ์ที่เกิดขึ้นมักถูกนิยามด้วยรูปแบบทางคณิตศาสตร์

Determinant-attributes ———> Dependency-attributes

Determinant-attribute

คือ Attribute ที่ระบุค่าแล้วสามารถแสดงค่าของ Dependency-attribute ได้

เช่น ความสัมพันธ์ระหว่าง รหัสพนักงานและชื่อพนักงาน

เมื่อระบุค่าของรหัสพนักงาน (Emp_ID) จะสามารถทราบชื่อพนักงาน (Emp_Name) ที่มีรหัสตามทีระบุได้



Functional Dependency : นิยาม

กำหนดให้ x และ y เป็น Attribute ของ relation R

- y มีฟังก์ชันขึ้นกับ x (y เป็น FD กับ x) ก็ต่อเมื่อถ้า 2 ทิวเปิลใน R มีค่าของ x ตรงกันแล้ว ทั้งสองทิวเปิลจะต้องมีค่าของ y ตรงกัน

- เช่น relation R มี attribute x, y จะได้ว่า

$R.x \longrightarrow R.y$

ความหมาย : x เป็นตัวกำหนดค่า (determine) ของ y หรือค่าของ y ขึ้นอยู่กับค่าของ x (y depends on x หรือ y เป็น FD กับ x)

Functional Dependency : รูปแบบการเขียน

FDs : determinant-attribute \longrightarrow dependency-attribute

FDs : Student_ID \longrightarrow Student_Name

Student_ID \longrightarrow Student_Name

Functional Dependency : ประเภท

- Functional Dependency ที่เกิดขึ้นจากความสัมพันธ์ระหว่าง Determinant และ Dependency อย่างละ 1 ค่า



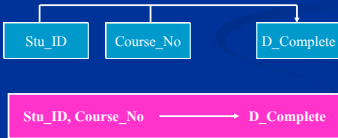
- Functional Dependency ที่เกิดขึ้นจากความสัมพันธ์ระหว่าง Determinant 1 ค่า กับ Dependency หลายค่า



- Functional Dependency ที่มีความสัมพันธ์ 2 ทาง โดยที่ Determinant และ Dependency สามารถระบุค่าของอีกฝ่ายหนึ่งได้



- Functional Dependency ที่ใช้ Determinant มากกว่า 1 ค่าเพื่ออ้างอิงถึง Dependency



Full Functional Dependency : นิยาม

Attribute y ของ relation R จะเป็น Full FD บน Attribute x ของ relation R

ถ้า y เป็น FD กับ x และไม่เป็น FD กับ Subset ใดๆ ของ x

หรือ เป็น FD ที่มี Determinant ที่มีขนาดเล็กที่สุด และสามารถระบุถึง Dependency ได้

Full Functional Dependency : ตัวอย่าง

D1 : Perdon_ID \longrightarrow Address
 D2 : Perdon_ID, Person_name \longrightarrow Address

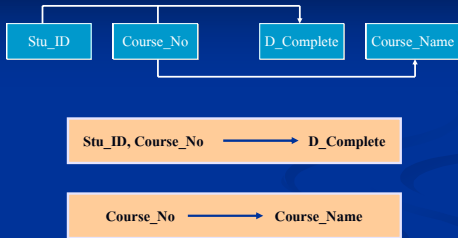
D3 : Product_line, Item_no \longrightarrow Use_qty
 D4 : Product_line, Item_no, Manager \longrightarrow Use_qty

สรุป D1 และ D3 เป็น Full Functional Dependency

Partial Dependency

- เนื่องจาก relation หนึ่งที่มี Determinant มากกว่า 1 ตัว (มี Primary key มากกว่า 1 Attributes) เป็นคีย์ผสม (Composite key)
- ความสัมพันธ์ระหว่างค่าของ Attribute **แบบบางส่วน** เกิดขึ้นเมื่อ Attribute ที่เป็น Determinant บางตัวของ PK สามารถระบุค่าของ Attributes อื่นๆ ที่ไม่ใช่ PK ของ relation ได้ (Non-key attribute)
- ความสัมพันธ์แบบนี้จะทำให้ **เกิดปัญหา** ในเรื่องความซ้ำซ้อน และการปรับปรุงข้อมูล

ตัวอย่าง



204204 DBASE DES & DEV

13

Transitive Dependency

- เป็นความสัมพันธ์ระหว่าง Attributes
- เมื่อ Attributes อื่นๆ ที่ไม่ใช่ Primary key (Non-key Attribute) สามารถระบุค่าของ Attribute อื่นๆ ในทิวเพิล (Tuple) ได้
- ความสัมพันธ์ในลักษณะดังกล่าว เรียกว่า

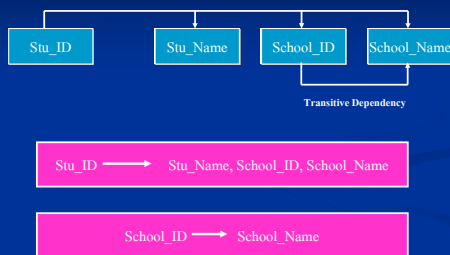
“ความสัมพันธ์ระหว่างค่าของ Attribute แบบ Transitive”



204204 DBASE DES & DEV

14

ตัวอย่าง



204204 DBASE DES & DEV

15

Multivalued Dependency

- ความสัมพันธ์นี้จะเกิดขึ้นกับ relation ที่ประกอบด้วย Attribute อย่างน้อย 3 Attribute และเป็น relation ที่ Attribute หนึ่งสามารถระบุค่าของ Attribute อื่นๆ ใน relation ได้มากกว่าหนึ่งค่า
- ความสัมพันธ์ในลักษณะดังกล่าว เรียกว่า

“ความสัมพันธ์ในการระบุค่าของ Attribute แบบหลายค่า”

204204 DBASE DES & DEV

16

หลักการของ Multivalued Dependency

- สัญลักษณ์ที่ใช้คือ $A \twoheadrightarrow B$
- Attribute A ค่าหนึ่งจะเป็นตัวกำหนดกลุ่มของค่า Attribute B
 - เมื่อทิวเพิล 2 ทิวเพิลในความสัมพันธ์ R มีค่า A เดียวกัน ไม่จำเป็นต้องมีค่า B เหมือนกัน แต่
 - ค่าของ B จะต้องอยู่ในกลุ่มของค่า B ที่ถูกกำหนดโดย A
- การเปลี่ยนแปลงค่าใน Attribute C จะไม่มีผลกระทบต่อค่า B

204204 DBASE DES & DEV

17

- ทิวเพิล 2 ทิวเพิลในความสัมพันธ์ R ที่มีค่า B เหมือนกันไม่จำเป็นต้องมีค่า A เดียวกัน
- ค่าของ Attribute C สองค่าที่มีความสัมพันธ์กับค่า A เดียวกัน จะต้องสัมพันธ์กับค่าของ B ในกลุ่มเดียวกัน และเป็นกลุ่มที่ถูกกำหนดโดยค่า A นั้นๆ

ค่าของ Determinant 1 ค่าสามารถระบุค่าของ Attribute ที่ทำหน้าที่เป็น Dependency ได้ตั้งแต่ 2 Attribute ขึ้นไป ซึ่งอยู่ในรูปของชุดข้อมูล

$Employee\# \twoheadrightarrow Department\#, Project\#$

204204 DBASE DES & DEV

18

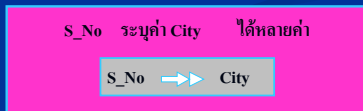
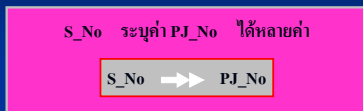
ตัวอย่าง

- ความสัมพันธ์ R ประกอบด้วย Attribute ดังนี้
 - Attribute S_No หมายถึง รหัสผู้ผลิต
 - Attribute PJ_No หมายถึง รหัสโครงการที่ส่งผลิต
 - Attribute City หมายถึง ชื่อจังหวัดที่ผู้ผลิตตั้งอยู่
- ข้อสมมติฐานของความสัมพันธ์ของข้อมูล คือ
 - ผู้ผลิตแต่ละรายสามารถรับงานได้มากกว่า 1 โครงการ
 - ผู้ผลิตแต่ละรายมีโรงงานตั้งอยู่ในจังหวัดต่างๆ ได้มากกว่า 1 จังหวัด

ความสัมพันธ์ R

S_No	PJ_No	City
S1	PJ01	Bangkok
S1	PJ01	Samuthprakarn
S1	PJ02	Bangkok
S1	PJ02	Samuthprakarn
S2	PJ03	Rayong
S2	PJ03	Chonburi
S2	PJ04	Rayong
S2	PJ04	Chonburi

ความสัมพันธ์ R เป็นรีเลชันที่มีความสัมพันธ์ของ Attribute ในการระบุค่าได้มากกว่า 1 ค่า คือ



- relation R เกิดความสัมพันธ์ขึ้นในลักษณะการระบุค่าของ Attribute แบบหลายค่าระหว่าง Attribute S_No กับ PJ_No และ S_No กับ City
- Attribute PJ_No และ City ไม่มีความสัมพันธ์ต่อกัน แต่
- ปรากฏอยู่ใน relation เดียวกัน ทำให้ค่าของ City ปรากฏซ้ำซ้อนอยู่ในหลายทิวเพิล เพื่อให้ข้อมูลของ relation ครบถ้วน
- ก่อให้เกิดปัญหาในการเพิ่ม ลบ หรือปรับปรุงข้อมูลได้

Trivial FD's

- Functional Dependency เป็น Trivial FD's ที่คือเมื่อ Attribute ทางด้านขวายู่ใน Attribute ทางด้านซ้าย

ตัวอย่าง :

```

    SUPPLIER (S_NAME, ADDRESS, ITEM, PRICE)
    S_NAME → ADDRESS
    S_NAME, ITEM → PRICE
    
```



Normalization

- E.F. Codd เป็นผู้คิดแนวคิดการทำ relation ให้อยู่ในรูปแบบบรรทัดฐาน (Normalization Process)
- เป็นกระบวนการที่นำเค้าร่างของ relation มาทำให้อยู่ในรูปแบบที่เป็นบรรทัดฐาน (Normal Form)
- เป็นกระบวนการเพื่อพัฒนาการเชื่อมต่อของข้อมูล เพื่อแก้ปัญหาของ relation
- เพื่อให้แน่ใจว่าการออกแบบเค้าร่างของ relation เป็นการออกแบบที่เหมาะสม

- ข้อมูลที่ใช้ในการทำ Normalization นั้นจะเอามาจาก E-R Model ซึ่งได้มาจาก การวิเคราะห์ระบบ (การออกแบบในระดับแนวคิด) โดยแสดงให้เห็น
 - ข้อมูล (Entity) ที่เกิดขึ้นในระบบ
 - Attributes ของข้อมูล
 - Relation ระหว่างข้อมูล



204204 DBASE DES & DEV 25

Normalization : วัตถุประสงค์

- วัตถุประสงค์ในการจัดเก็บข้อมูล
 - การทำ relation ให้เป็นบรรทัดฐาน เป็นการลดความซ้ำซ้อนของข้อมูลใน relation
 - ทำให้ลดเนื้อที่ในการจัดเก็บข้อมูล / ลดที่ว่างที่ใช้ในการเก็บข้อมูล
- วัตถุประสงค์เรื่องข้อมูล ไม่ถูกต้อง
 - เนื่องจากข้อมูลใน relation หนึ่งจะมีข้อมูลไม่ซ้ำกัน
 - เมื่อมีการปรับปรุงข้อมูลก็จะปรับปรุงทุกที่นั่นๆ ครั้งเดียว ไม่ต้องปรับปรุงหลายแห่ง
 - ลดโอกาสที่จะเกิดความผิดพลาดในการปรับปรุงข้อมูล ไม่ครบถ้วน

204204 DBASE DES & DEV 26

- วัตถุประสงค์การเกิดอนอร์มัลไล (Anomalies) ในการเพิ่ม ปรับปรุง และลบข้อมูล
 - ช่วยแก้ปัญหาที่อาจเกิดขึ้นจากการปรับปรุงข้อมูลไม่ครบ
 - ช่วยแก้ปัญหาที่อาจเกิดขึ้นจากการเพิ่มข้อมูล
 - ช่วยแก้ปัญหาที่อาจเกิดขึ้นจากการลบข้อมูล
 - ช่วยแก้ปัญหาในเรื่องข้อมูลสูญหายจากฐานข้อมูล
- เพิ่มความคงทนแก่โครงสร้างฐานข้อมูล

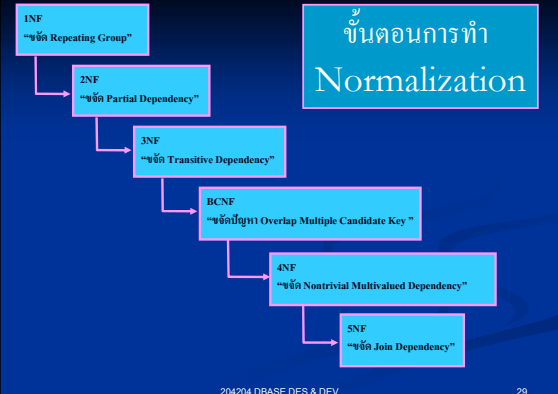
204204 DBASE DES & DEV 27

Normalization : รูปแบบ

- โดยทั่วไปการทำ Normalization มีด้วยกันทั้งสิ้น 6 รูปแบบ คือ
 - First Normal Form (1NF)
 - Second Normal Form (2NF)
 - Third Normal Form (3NF)
 - Boyce-Codd Normal Form (BCNF)
 - Forth Normal Form (4NF)
 - Fifth Normal Form (5NF)

204204 DBASE DES & DEV 28

ขั้นตอนการทำ Normalization



```

    graph TD
      1NF["1NF  
"ขจัด Repeating Group""] --> 2NF["2NF  
"ขจัด Partial Dependency""]
      2NF --> 3NF["3NF  
"ขจัด Transitive Dependency""]
      3NF --> BCNF["BCNF  
"ขจัดปัญหา Overlap Multiple Candidate Key""]
      BCNF --> 4NF["4NF  
"ขจัด Nontrivial Multivalued Dependency""]
      4NF --> 5NF["5NF  
"ขจัด Join Dependency""]
  
```

204204 DBASE DES & DEV 29

First Normal Form : 1NF

- Relation ที่จะอยู่ในรูปแบบ 1NF ก็ต่อเมื่อ relation นั้นๆ ไม่มี Attribute ของทุพหุคูณที่มีค่าของข้อมูลหลายค่า (No Repeating Group)
- ค่าของ Attribute ต่างๆ ในแต่ละทุพหุคูณ มีค่าของข้อมูลเพียงค่าเดียว (Atomic Value)



204204 DBASE DES & DEV 30

ตารางที่มีลักษณะข้อมูลหลายค่าในฟิลด์ใดฟิลด์หนึ่ง (Repeating Group)

S_No	S_Name	City	P_No	Qty
S1	คิม	กรุงเทพฯ	P1	300
			P2	200
			P3	400
			P4	200
S2	ต้อย	ปทุมธานี	P1	300
			P2	400
S3	น้อย	ปทุมธานี	P2	200
S4	แดง	กรุงเทพฯ	P2	200
			P4	300

ปรับเปลี่ยนข้อมูลในตารางไปให้อยู่ในรูปแบบ 1NF

S_No	S_Name	City	P_No	Qty
S1	คิม	กรุงเทพฯ	P1	300
S1	คิม	กรุงเทพฯ	P2	200
S1	คิม	กรุงเทพฯ	P3	400
S1	คิม	กรุงเทพฯ	P4	200
S2	ต้อย	ปทุมธานี	P1	300
S2	ต้อย	ปทุมธานี	P2	400
S3	น้อย	ปทุมธานี	P2	200
S4	แดง	กรุงเทพฯ	P2	200
S4	แดง	กรุงเทพฯ	P4	300

Second Normal Form : 2NF

- Relation ที่จะอยู่ในรูปของ 2NF ก็ต่อเมื่อ
 - Relation นั้นๆ อยู่ในรูปแบบบรรทัดฐานขั้นที่ 1 (1NF)
 - Attribute จะต้องไม่มีความสัมพันธ์แบบ Partial Dependency

กล่าวคือ

- Attribute ทุก Attribute ที่ไม่ได้เป็นคีย์หลัก (PK) จะต้องมีความสัมพันธ์ระหว่างค่าของ Attribute แบบที่อิงกับคีย์หลัก (Full FD) หรือ
- ค่าของ Attribute ที่ไม่ได้เป็นคีย์หลักจะสามารถระบุค่าโดย Attribute ที่เป็นคีย์หลัก หรือ Attribute ทั้งหมดที่ประกอบกันเป็นคีย์หลักในกรณีที่คีย์หลักเป็นคีย์ผสม

- การทำ Normalization ในระดับ 1NF

- เป็นการจัด Attribute หรือกลุ่ม Attribute ที่ซ้ำกัน

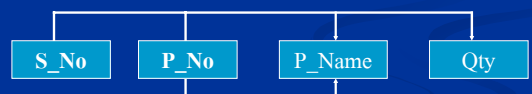
- ความผิดพลาดที่เกิดขึ้นกับข้อมูลในระดับ 1NF

- จากการเพิ่มข้อมูล (Insert Anomaly) อาจทำให้ไม่สามารถเพิ่มข้อมูลบางอย่างไม่ได้ หรือเพิ่มข้อมูลแล้วเกิดความขัดแย้งกับข้อมูลเดิม
- จากการลบข้อมูล (Delete Anomaly) ถ้าต้องลบข้อมูลบางส่วน จะส่งผลให้ลบข้อมูลอื่นๆ ออกไปด้วยโดยไม่ได้ตั้งใจ
- จากการแก้ไขข้อมูล (Update Anomaly) จะต้องแก้ไขทุกแถว ผู้ใช้งานฐานข้อมูลจะต้อง แก้ไขข้อมูลมากกว่า 1 แถว

Relation Order1 ประกอบด้วย Attribute ดังนี้

S_No	P_No	P_Name	Qty
S1	P1	เพชร	100
S1	P2	ทับทิม	200
S1	P3	ไพลิน	300
S2	P1	เพชร	300
S2	P3	ไพลิน	400
S3	P4	พลอย	100

- Attribute รหัสผู้ผลิต (S_No) และรหัสสินค้า (P_No) ประกอบกันเป็นคีย์หลัก (PK) ที่สามารถระบุค่าของชื่อสินค้า (P_Name) และจำนวนสินค้าที่ถูกจัดส่ง (Qty) ได้
- Attribute รหัสสินค้า (P_Name) สามารถระบุค่าของชื่อสินค้า (P_Name) ได้อีกด้วย



- เพราะฉะนั้น relation Order1 จึงไม่ได้อยู่ในรูปแบบ 2NF ดังนั้น
 ต้องทำการแตก relation เพื่อลดปัญหาความซ้ำซ้อนของข้อมูล
 - Order1 (S_No, P_No, Qty) มี S_No และ P_No เป็น PK
 - Product (P_No, P_Name) มี P_No เป็น PK

S_No	P_No	Qty

Product

↓

P_No	P_Name

Order1

↑

204204 DBASE DES & DEV 37

Third Normal Form : 3NF

- Relation ที่จะอยู่ในรูปของ 3NF ก็คือเมื่อ
 - Relation นั้นๆ อยู่ในแบบบรรทัดฐานขั้นที่ 2 (2NF)
 - Attribute ที่ไม่ได้เป็นคีย์หลัก (PK) ไม่มีคุณสมบัติในการกำหนดค่าของ Attribute อื่นที่ไม่ใช่คีย์หลัก
 - Attribute จะต้องไม่มีความสัมพันธ์แบบ Transitive Dependency

204204 DBASE DES & DEV 38

Relation Supplier1 ประกอบด้วย Attribute ดังนี้

S_No	S_Name	City	Rating
S1	ดิม	กรุงเทพฯ	2
S2	ด้อย	ปทุมธานี	3
S3	น้อย	ปทุมธานี	3
S4	แดง	กรุงเทพฯ	2
S5	จุ่ม	สมุทรปราการ	1

204204 DBASE DES & DEV 39

- นอกจากนี้ยังพบว่าความสัมพันธ์ระหว่าง Attribute ที่ไม่ได้เป็น PK คือ Attribute City และ Rating
 - โดย Attribute City สามารถระบุค่าการจัดอันดับของผู้ผลิต ซึ่งเป็นความสัมพันธ์ระหว่างค่าของ Attribute แบบ Transitive Dependency
 - หาก relation ใดมีคุณสมบัติดังกล่าวจะทำให้เกิดความผิดพลาดในการเพิ่ม ลบ หรือปรับปรุงข้อมูลได้ เช่น
 - ความผิดพลาดที่เกิดจากการปรับปรุงข้อมูล (Update Anomaly)
 - ความผิดพลาดที่เกิดจากการลบข้อมูล (Delete Anomaly)

204204 DBASE DES & DEV 40

- Relation Supplier1 อยู่ในรูปแบบ 2NF
 - ไม่มีค่าของข้อมูลซ้ำกัน
 - ค่าของ Attribute ที่ไม่ได้เป็นคีย์หลักอื่นๆ สามารถระบุค่าได้โดยค่าของ Attribute ที่เป็นคีย์หลัก

ตัวอย่าง : กรณีที่ทราบรหัสผู้ผลิต คือ S1 จะสามารถทราบว่าผู้ผลิตชื่ออะไร อยู่ที่จังหวัดใด และอยู่ในจังหวัดที่ถูกรจัดอันดับให้เป็นอันดับที่เท่าไร

- Relation นี้ไม่มีคีย์คู่แข่ง เพราะว่ามีคีย์หลักที่มีคุณสมบัติเป็นคีย์หลักเช่นเดียวกับรหัสผู้ผลิต ถึงแม้ว่าชื่อจังหวัดสามารถที่จะระบุการจัดอันดับของผู้ผลิต แต่ไม่สามารถระบุชื่อของผู้ผลิตได้ชัดเจน

ตัวอย่าง : กรณีที่ทราบชื่อจังหวัด คือ กรุงเทพฯ ชื่อของผู้ผลิตอาจเป็น ดิม หรือแดง ก็ได้ จึงไม่สามารถกำหนดชัดเจนได้ว่าผู้ผลิตชื่ออะไร

204204 DBASE DES & DEV 41

- การแก้ไขปัญหา Attribute City ที่สามารถระบุค่า Attribute Rating ได้ นั้นจะต้องทำการแตก relation Supplier1 เป็น 2 relation (Decomposition)
 - โดยการแยก Attribute ที่ถูกกำหนดค่ากับ Attribute ที่เป็นตัวกำหนดค่า (Determinant) ออกเป็น relation ใหม่ และ
 - กำหนดให้ Attribute ที่เป็นตัวกำหนดค่าเป็น PK ของ relation ใหม่
 - relation ใหม่ประกอบด้วย
 - Supplier (S_No, S_Name, City) มี S_No เป็น PK และมี City เป็น FK
 - City (City, Rating) มี City เป็น PK

204204 DBASE DES & DEV 42

■ วิธีการแตก relation (Decomposition) มีขั้นตอนดังนี้

- นำ Attribute ที่ถูกกำหนดค่า กับ Attribute ที่เป็นตัวกำหนดค่า แยกออกเป็น relation ใหม่
- กำหนดให้ Attribute ที่เป็นตัวกำหนดค่าเป็น PK ของ relation ใหม่

ข้อควรระวัง :

ในบางกรณีผู้ออกแบบเค้าร่างของฐานข้อมูลพยายามแตก relation ที่มีจำนวน Attribute มากๆ ออกเป็น relation ต่างๆ มากมายนั้น อาจก่อให้เกิดปัญหาการแตกฐาน Relation ที่ไม่เหมาะสมได้ (Bad Decomposition) ได้

ตัวอย่าง Bad Decomposition

■ กรณีที่แตก relation Supplier1 ออกเป็น

- Supplier (S_No, S_Name, City) มี S_No เป็น PK
- Supplier2 (S_No, Rating) มี S_No เป็น PK

S_No	S_Name	City
S1	ดีม	กรุงเทพฯ
S2	ดีอ	ปทุมธานี
S3	น้อ	ปทุมธานี
S4	แดง	กรุงเทพฯ
S5	จัน	สมุทรปราการ

S_No	Rating
S1	2
S2	3
S3	3
S4	2
S5	1

■ จากตัวอย่างจะพบว่า การแตก relation ดังกล่าวอาจก่อให้เกิดความผิดพลาดในการเพิ่ม ลบ หรือแก้ไขข้อมูลได้ **เนื่องจาก**

- การเพิ่มจังหวัดและการจัดอันดับจะทำได้ จนกว่าจะมีผู้ผลิตรายใดรายหนึ่งอยู่ที่จังหวัดที่จะเพิ่มชื่อและการจัดอันดับนั้น
- Relation Supplier และ Supplier2 เป็น relation ที่ไม่มีความเป็นอิสระต่อกัน เพราะ Attribute City เป็นตัวกำหนด Attribute Rating นั้นเอง



Boyce / Codd Normal Form : BCNF

■ Relation ที่จะอยู่ในรูปของ BCNF ก็ต่อเมื่อ

- Relation นั้นๆ อยู่ในรูปแบบบรรทัดฐานขั้นที่ 3 (3NF)
- ไม่มี Attribute ใดๆ ใน relation ที่สามารถระบุค่าของ Attribute ที่เป็นคีย์หลัก (PK) หรือส่วนหนึ่งส่วนใดของคีย์หลัก (ในกรณีที่มีคีย์หลักเป็นคีย์ผสม) ได้

โดยทั่วไปรูปของ BCNF จะอยู่ในรูปของ 3NF แต่ไม่จำเป็นเสมอไป ที่รูปของ 3NF จะอยู่ในรูปของ BCNF เนื่องจาก

- รูปแบบ BCNF เป็นรูปแบบที่ขยายขอบเขตของรูปแบบ 3NF ให้มีความเหมาะสมยิ่งขึ้น
- รูปแบบของ Relation ที่จะต้องผ่านการทำให้เป็น BCNF จะต้องมีความสัมพันธ์ ดังนี้
 - เป็น relation ที่มีคีย์คู่แข่งหลายคีย์ (Multiple Candidate Key) โดยที่
 - คีย์คู่แข่งเป็นคีย์ผสม (Composite Key) และ
 - คีย์คู่แข่งนั้นมีบางส่วนซ้ำซ้อนกัน (Overlapped) (มี attribute บางตัวร่วมกัน)

Relation Supplier3 ประกอบด้วย Attribute ดังนี้

S_No	S_Name	P_No	Qty
S1	ดีม	P1	300
S1	ดีม	P2	200
S1	ดีม	P3	400
S1	ดีม	P4	200
S2	ดีอ	P1	300
S2	ดีอ	P2	400
S3	น้อ	P2	200
S4	แดง	P2	200
S4	แดง	P4	300
S4	แดง	P5	400

หมายเหตุ :

ชื่อของผู้ผลิต (S_Name) เป็นสำเนาซ้ำกัน และมีความสัมพันธ์เป็น PK ได้เช่นกัน

- Relation Supplier3 เป็น relation ที่มีปัญหา **เพราะ**
 - มีคีย์คู่แข่งเป็นคีย์ผสม และมีความซ้ำซ้อนกัน
 - คีย์คู่แข่งที่มีคุณสมบัติเป็น PK ของ relation อาจจะเป็น Attribute S_No และ P_No หรือ Attribute S_Name และ P_No ก็ได้
- ในกรณีที่มีชื่อผู้ผลิตไม่ซ้ำกัน relation นี้ไม่ได้อยู่ในรูปแบบ BCNF **เพราะ**
 - เมื่อเลือกคีย์คู่แข่งใดเป็น PK แล้ว คีย์คู่แข่งที่ไม่ได้ถูกเลือกจะยังปรากฏซ้ำซ้อนอยู่ใน relation นี้ และมีคุณสมบัติในการระบุค่าของ Attribute ที่เป็น PK

204204 DBASE DES & DEV 49

- กรณีที่เกิด Attribute S_No และ P_No เป็น PK จะเกิดปัญหา
 - Attribute S_Name จะมีคุณสมบัติในการระบุค่าของ Attribute S_No

- Relation นี้ยังเป็น relation ที่ประกอบด้วยข้อมูลที่ซ้ำซ้อนกัน ซึ่งจะก่อให้เกิดความผิดพลาดจากการเพิ่ม แก้ไขหรือปรับปรุงข้อมูล

เช่น การเปลี่ยนชื่อของผู้ผลิต S1 จะต้องแก้ไขข้อมูลในทุกฟิลด์ต่างๆ ซึ่งอาจเกิดความผิดพลาดได้ (แก้ไขข้อมูลไม่ครบถ้วน)

204204 DBASE DES & DEV 50

แนวทางแก้ไข

- การแตก relation Supplier3 ออกเป็น relation ใหม่
- โดยการแยก Attribute ที่สามารถระบุค่าของคีย์คู่แข่งแยกเป็นอีก relation หนึ่ง ซึ่งอาจทำได้ 2 กรณี คือ
 - กรณีที่หนึ่ง Supplier4 (S_No, S_Name) มี S_No, S_Name เป็น PK
Order2 (S_No, P_No, Qty) มี S_No, P_No เป็น PK
 - กรณีที่สอง Supplier4 (S_No, S_Name) มี S_No, S_Name เป็น PK
Order2 (S_Name, P_No, Qty) มี S_Name, P_No เป็น PK

204204 DBASE DES & DEV 51

สรุป :

- รูปแบบ BCNF เป็นรูปแบบที่ขยายขอบเขตของรูปแบบ 3NF โดยมี การตรวจสอบ relation ว่ามี Attribute อื่นที่ไม่ใช่ PK แต่สามารถระบุค่าของ PK ได้ ดังนี้

204204 DBASE DES & DEV 52

Forth Normal Form : 4NF

- Relation ที่จะอยู่ในรูปของ 4NF ก็ต่อเมื่อ
 - Relation นั้นๆ อยู่ในรูปแบบ BCNF
 - เป็น relation ที่ไม่มีความสัมพันธ์ในการระบุค่าของ Attribute แบบหลายค่า โดยที่ Attribute ที่ถูกระบุค่าหลายค่าเหล่านี้ไม่มีความสัมพันธ์กัน
 - Attribute จะต้องมีไม่มีความสัมพันธ์แบบ Independently Multivalued Dependency

Relation ที่อยู่ในรูปแบบ 4NF จะต้องมี Attribute อย่างน้อย 3 Attribute

204204 DBASE DES & DEV 53

Relation SPJC ประกอบด้วย Attribute ดังนี้

S_No	PJ_No	City
S1	PJ01	กรุงเทพฯ
	PJ01	ปทุมธานี
	PJ02	กรุงเทพฯ
S2	PJ02	ปทุมธานี
	PJ03	สมุทรปราการ
	PJ03	อุซฮา
	PJ04	สมุทรปราการ
	PJ04	อุซฮา
S2	PJ05	สมุทรปราการ
	PJ05	อุซฮา

Relation SPJC ประกอบด้วย

- Attribute รหัสผู้ผลิต (S_No)
- Attribute รหัสโครงการ (PJ_No)
- Attribute ชื่อจังหวัดที่โรงงานของผู้ผลิตตั้งอยู่ (City)

ข้อกำหนด :
ผู้ผลิตแต่ละรายสามารถผลิตให้กับโครงการต่างๆ ได้มากกว่า 1 โครงการ และ ผู้ผลิตแต่ละรายมีโรงงานตั้งอยู่ในจังหวัดต่างๆ ได้มากกว่า 1 จังหวัด

204204 DBASE DES & DEV 54

ทำให้ relation SPJC อยู่ในรูปแบบ BCNF โดยให้ทุก Attribute ประกอบกันเป็น PK

S_No	PJ_No	City
S1	PJ01	กรุงเทพฯ
	PJ01	ปทุมธานี
	PJ02	กรุงเทพฯ
S2	PJ02	ปทุมธานี
	PJ03	สมุทรปราการ
	PJ03	อยุธยา
	PJ04	สมุทรปราการ
	PJ04	อยุธยา
	PJ05	สมุทรปราการ
	PJ05	อยุธยา

S_No	PJ_No	City
S1	PJ01	กรุงเทพฯ
S1	PJ01	ปทุมธานี
S1	PJ02	กรุงเทพฯ
S1	PJ02	ปทุมธานี
S1	PJ03	สมุทรปราการ
S2	PJ03	อยุธยา
S2	PJ04	สมุทรปราการ
S2	PJ04	อยุธยา
S2	PJ05	สมุทรปราการ
S2	PJ05	อยุธยา

204204 DBASE DES & DEV 55

- Relation SPJC อาจเกิดปัญหาความผิดพลาดจากการเพิ่ม ลบ หรือปรับปรุงข้อมูลได้ เช่น
 - ผู้สมัครหีส S2 ชายโรงงานจาก สมุทรปราการ ไปอยู่จังหวัดอื่น ก็ต้องทำการแก้ไขข้อมูลในทุพื้นที่ต่าง ๆ หรือ
 - กรณีที่มีการเก็บข้อมูลชื่อจังหวัดที่ตั้งของผู้ผลิตคนใหม่ที่ยังไม่เคยรับงานโครงการใดเลยจะทำได้
- แนวทางการแก้ไขปัญหา
 - การแตก relation SPJC ออกเป็น 2 relation
 - SPJ (S_No, PJ_No) มี S_No, PJ_No เป็น PK
 - SC (S_No, City) มี S_No, City เป็น PK

204204 DBASE DES & DEV 56

- Relation SPJC ประกอบด้วย
 - S_No ซึ่งมีความสัมพันธ์ในการระบุค่า PJ_No และ City ได้หลายค่า
- Relation SPJC อยู่ในรูปแบบ BCNF
 - เมื่อมี Attribute ทุก Attribute ประกอบกันเป็น PK
 - S_No มีความสัมพันธ์ในการระบุค่า PJ_No แบบหลายค่า (S_No → PJ_No)
 - S_No มีความสัมพันธ์ในการระบุค่า City แบบหลายค่า (S_No → City)
- คีย์หลักที่ Attribute ทั้งสองไม่มีความสัมพันธ์กันแต่มาอยู่ใน relation เดียวกันจะก่อให้เกิดความซ้ำซ้อนของข้อมูลขึ้นได้ **เพราะว่า** ค่าของชื่อจังหวัดจะปรากฏใน relation ซ้ำกัน เพื่อให้ข้อมูลในแต่ละทุเท็มีความสมบูรณ์
 - แม้ว่า relation SPJC จะอยู่ในรูปแบบ BCNF โดยที่ทุก Attribute ประกอบกันเป็น PK แต่ relation ที่ยังมีปัญหาแฝงอยู่เช่นกัน

204204 DBASE DES & DEV 57

SPJC		
S_No	PJ_No	City
S1	PJ01	กรุงเทพฯ
S1	PJ01	ปทุมธานี
S1	PJ02	กรุงเทพฯ
S1	PJ02	ปทุมธานี
S2	PJ03	สมุทรปราการ
S2	PJ03	อยุธยา
S2	PJ04	สมุทรปราการ
S2	PJ04	อยุธยา
S2	PJ05	สมุทรปราการ
S2	PJ05	อยุธยา

SPJ	
S_No	PJ_No
S1	PJ01
S1	PJ02
S2	PJ03
S2	PJ04
S2	PJ05

SC	
S_No	City
S1	กรุงเทพฯ
S1	ปทุมธานี
S2	สมุทรปราการ
S2	อยุธยา

204204 DBASE DES & DEV 58

Fifth Normal Form : 5NF

- ชื่ออื่นของรูปแบบบรรทัดฐานขั้นที่ 5 (5NF) คือ
 - Project-Join Normal Form (PJ/NF)
- Relation ที่จะอยู่ในรูปของ 5NF ก็ต่อเมื่อ
 - Relation นั้นๆ อยู่ในรูปแบบบรรทัดฐานขั้นที่ 4 (4NF)
 - ไม่มี Symmetric Constraint
 - หากมีการแตก relation ออกเป็น relationย่อย (Projection)
 - เมื่อทำการเชื่อมโยง relation ย่อยทั้งหมด (Join) จะไม่ก่อให้เกิดข้อมูลใหม่ที่ไม่มีเหมือน relation เดิม (Spurious Tuple)

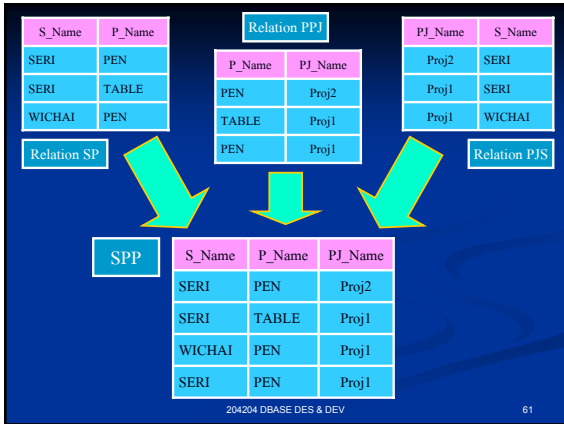
204204 DBASE DES & DEV 59

ตัวอย่างที่ 1 relation SPP ประกอบด้วย Attribute ต่างๆ ดังนี้

- Attribute ชื่อผู้ผลิต (S_Name)
- Attribute ชื่อสินค้า (P_Name)
- Attribute ชื่อโครงการ (PJ_Name)
- โดยที่ Attribute ทั้งสามเป็น PK

SPP		
S_Name	P_Name	PJ_Name
SERI	PEN	Proj2
SERI	TABLE	Proj1
WICHAJ	PEN	Proj1
SERI	PEN	Proj1

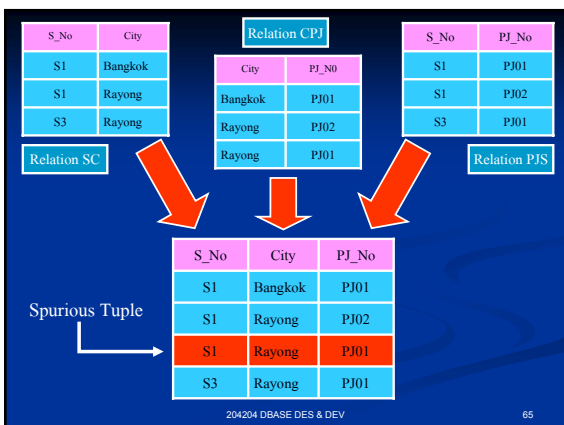
204204 DBASE DES & DEV 60



- Relation SPP อยู่ในรูปแบบ 4NF ประกอบด้วย Attribute ทั้งสามประกอบกันเป็น PK แต่ประสบปัญหาในการปรับปรุงแก้ไขข้อมูล เช่น
 - การลบข้อมูลของทุกเทปที่ 3 (WICHAI, PEN, Proj1) ส่งผลให้ข้อมูลของชื่อผู้ผลิตที่ชื่อ "WICHAI" หายไปจากฐานข้อมูล หรือ
 - การเปลี่ยนชื่อสินค้าใดสินค้าหนึ่ง จะต้องทำการปรับปรุงข้อมูลหลายเทป
 - แนวทางแก้ไขปัญหาคือ
 - การแตก relation SPP ออกเป็น 3 relation ย่อย (Projection)
 - แต่ละ relation ย่อยประกอบด้วย Attribute แต่ละคู่เป็น PK
- 204204 DBASE DES & DEV 62

- เมื่อแตก relation SPP ออกเป็น relation SP, PPJ และ PJS แล้ว หากนำ relation ทั้งสามมาเชื่อมโยงกัน (Join) จะมีข้อมูลเหมือนใน relation เดิม (Symmetric Constraint) คือ ไม่มีลักษณะของทูเปิลที่เกินมา เรียกว่า Spurious Tuple
 - เมื่อแตก relation ออกมาแล้วทำการเชื่อมโยง relation ย่อยอีกครั้ง หากไม่มีข้อมูลที่แตกต่างไปจาก relation เดิม ก็สามารถจะแตก relation นั้นได้
 - เมื่อแตก relation ออกมาแล้วทำการเชื่อมโยง relation ย่อยอีกครั้ง แล้วข้อมูลแตกต่างไปจาก relation เดิม หรือเกิดข้อมูลไม่เหมือนกับ relation เดิม ก็ไม่ควรแตก relation นั้น ให้ถือว่า relation เดิมอยู่ในรูปแบบ 5NF
- 204204 DBASE DES & DEV 63

- ตัวอย่างที่ 2 relation SCPJ ประกอบด้วย Attribute ต่างๆ ดังนี้
- Attribute รหัสผู้ผลิต (S_No)
 - Attribute ชื่อจังหวัดของผู้ผลิต (City)
 - Attribute รหัสโครงการ (PJ_No)
 - โดยที่ Attribute ทั้งสามเป็น PK
- SCPJ
- | S_No | City | PJ_No |
|------|---------|-------|
| S1 | Bangkok | PJ01 |
| S1 | Rayong | PJ02 |
| S3 | Rayong | PJ01 |
- 204204 DBASE DES & DEV 64



- Relation SCPJ อยู่ในรูปแบบ 4NF ประกอบด้วย Attribute ทั้งสามประกอบกันเป็น PK แต่มีปัญหาคืออาจเกิดความคิดในการเพิ่ม ลบ หรือปรับปรุงข้อมูล
 - แนวทางแก้ไขปัญหาคือ
 - การแตก relation SPP ออกเป็น 3 relation ย่อย (Projection) ประกอบด้วย Relation SC, CPJ และ PJC
 - แต่ปัญหาที่เกิดขึ้น คือ เมื่อนำ relation ทั้งสามมาเชื่อมโยงกัน จะมีข้อมูลเกินมา (Spurious Tuple) คือ S1 Rayong PJ01 ซึ่งไม่มีใน relation เดิม
 - กรณีที่เกิดปัญหาในลักษณะนี้ก็ไม่ควรจะแตก relation นั้น
- 204204 DBASE DES & DEV 66

ประเด็นที่ควรคำนึงถึง ในการทำให้เป็นรูปแบบบรรทัดฐาน (Normal Form)

- การแตก relation มากเกินไป (Overnormalization)
- การคืนอร์มอลไลเซชัน (Denormalization)

204204 DBASE DES & DEV

67

การแตก relation มากเกินไป (Overnormalization)

- วัตถุประสงค์ของการทำให้เป็นรูปแบบบรรทัดฐาน คือ
 - เพื่อลดปัญหาในด้านความซ้ำซ้อนของข้อมูล
 - เพื่อลดปัญหาในเรื่องการเพิ่ม การลบ หรือการปรับปรุงข้อมูล
- โดยทั่วไปการออกแบบในระดับแนวคิด ผู้ออกแบบจะพยายามวิเคราะห์ relation ให้อยู่ในรูปแบบ 3NF
- กรณีที่เกิดปัญหาต่างๆ ที่จำเป็นคือทำต่อไปถึงรูปแบบ BCNF, 4NF และ 5NF (เกิดขึ้นน้อยมากในทางปฏิบัติ)

204204 DBASE DES & DEV

68

- ด้วยเหตุผลดังกล่าว ผู้ออกแบบไม่ควรพยายามที่จะแตก relation มากเกินความจำเป็น (Overnormalization) **เพราะ**
 - การแตก relation ออกเป็น relation ย่อยมากเกินไปมีผลต่อประสิทธิภาพในการทำงานของฐานข้อมูล
- เช่น ในการค้นคืนข้อมูลจะต้องใช้เวลามากกว่าเดิม เป็นต้น



204204 DBASE DES & DEV

69

การคืนอร์มอลไลเซชัน (Denormalization)

- กรณีที่บาง relation ถูกออกแบบโดยการไม่ทำให้อยู่ในรูปแบบบรรทัดฐานที่เป็นไปตามกฎเกณฑ์ที่กำหนดไว้
- เช่น relation นั้นควรจะปรับให้อยู่ในรูปแบบ 3NF แต่หยุดอยู่ที่เชิงรูปแบบ 2NF เป็นต้น
- อาจเป็นเพราะเหตุผลในเรื่องของประสิทธิภาพในการเรียกดู หรือ การค้นคืนข้อมูล และยอมให้เกิดความซ้ำซ้อนของข้อมูลได้

204204 DBASE DES & DEV

70

- การคืนอร์มอลไลเซชันอาจก่อให้เกิดปัญหาความซ้ำซ้อนของข้อมูลเกิดขึ้นได้
 - ควรมีการระบุสาเหตุ และวิธีการในการปรับปรุงข้อมูลในโปรแกรมประยุกต์ใช้งาน เพื่อป้องกันไม่ให้เกิดปัญหาข้อมูลไม่ถูกต้อง
 - ถ้าข้อมูลใน relation นั้นๆ ส่วนใหญ่จะเป็นการเรียกดูข้อมูล (Select) มากกว่าการเพิ่ม ลบ หรือปรับปรุงข้อมูล เพื่อเพิ่มประสิทธิภาพในการทำงานของฐานข้อมูล และไม่มีปัญหาด้านความไม่ถูกต้องของข้อมูลที่ซ้ำซ้อนกันได้

204204 DBASE DES & DEV

71



204204 DBASE DES & DEV

72