

Logical Database Design

01418221 Fundamentals of Database Systems

การออกแบบฐานข้อมูลเชิงสัมพันธ์

- การศึกษาแนวคิดของการออกแบบฐานข้อมูลเชิงสัมพันธ์ เพื่อจะช่วยให้ในการสร้างแบบจำลองสำหรับฐานข้อมูลเชิงสัมพันธ์ โดยอ้างอิงหลักเกณฑ์ในการกำหนดตารางคอลัมน์และสร้างความสัมพันธ์ระหว่างตารางเพื่อลดความซ้ำซ้อนของข้อมูล
 - การสร้างแบบจำลองจากปัญหาจริงเป็นตารางที่สัมพันธ์กัน
 - การระบุถึงปัญหาและการลดความซ้ำซ้อนของข้อมูลที่จัดเก็บ
 - การระบุการอ้างอิง และการรวมตัวกันในการออกแบบฐานข้อมูลเชิงสัมพันธ์
 - การปรับแต่งตารางในฐานข้อมูลเชิงสัมพันธ์เพื่อให้การออกแบบมีความเหมาะสมมากที่สุด

ปัญหาการซ้ำซ้อนของข้อมูล

- ความซ้ำซ้อนข้อมูลหมายถึงข้อมูลประเภทเดียวกันที่มีการจัดเก็บมากกว่าหนึ่งชุดข้อมูล
 - หมายถึงการจัดเก็บภายในตารางเดียวกันของฐานข้อมูล
 - การซ้ำซ้อนข้อมูลนี้เกิดจากการออกแบบฐานข้อมูลเชิงสัมพันธ์ที่ไม่เหมาะสม
- เกิดความผิดปกติของข้อมูลในฐานข้อมูล (Anomalies) ดังต่อไปนี้
 - ความผิดปกติจากการปรับปรุงแก้ไขข้อมูล (Update Anomalies)
 - ความผิดปกติจากการลบข้อมูล (Delete Anomalies)
 - ความผิดปกติจากการเพิ่มข้อมูล (Insert Anomalies)

ความผิดปกติจากการแก้ไขปรับปรุงข้อมูล (Update Anomalies)

- ความผิดปกติจากการปรับปรุงแก้ไขข้อมูลจะเกิดขึ้นเมื่อรายการข้อมูลซ้ำกันตั้งแต่หนึ่งรายการขึ้นไปต้องการแก้ไขปรับปรุงข้อมูล ยกตัวอย่าง
 - สมบัติย้ายที่อยู่จาก กรุงเทพฯ ไป ชลบุรี จะต้องมีการปรับปรุงข้อมูลเดียวกัน ซ้ำกันหลายที่

StudentNum	CourseNum	Student Name	Address	Course
S21	9201	สมบัติ	กรุงเทพฯ	บัญชีเบื้องต้น
S21	9267	สมบัติ	กรุงเทพฯ	ฟิสิกส์
S24	9267	รัตนา	ระยอง	ฟิสิกส์
S30	9201	วิชา	เชียงใหม่	คอมพิวเตอร์เบื้องต้น
S30	9322	วิชา	เชียงใหม่	คณิตศาสตร์

ความผิดปกติจากการลบข้อมูล (Update Anomalies)

- ความผิดปกติจากการลบข้อมูลจะเกิดขึ้นเมื่อข้อมูลในบางแอททริบิวต์จะหายไปเมื่อมีการลบข้อมูลส่วนอื่นออกไป ยกตัวอย่าง
 - สมบัติเป็นนิสิตคนสุดท้ายที่เรียนวิชาบัญชีเบื้องต้น เมื่อสมบัติถอนวิชาบัญชีเบื้องต้นออกแล้ว จะทำให้รายการลงทะเบียนของสมบัติในวิชานี้จะถูกลบออก จะส่งผลให้ข้อมูลวิชาบัญชีจะหายไปด้วย

StudentNum	CourseNum	Student Name	Address	Course
S21	9201	สมบัติ	กรุงเทพฯ	บัญชีเบื้องต้น
S21	9267	สมบัติ	กรุงเทพฯ	ฟิสิกส์
S24	9267	รัตนา	ระยอง	ฟิสิกส์
S30	9201	วิชา	เชียงใหม่	คอมพิวเตอร์เบื้องต้น
S30	9322	วิชา	เชียงใหม่	คณิตศาสตร์

ความผิดปกติจากการเพิ่มข้อมูล (Insert Anomalies)

- ความผิดปกติจากการเพิ่มข้อมูลจะเกิดขึ้นเมื่อต้องการเพิ่มข้อมูลในบางแอททริบิวต์แต่ไม่สามารถทำได้ เพราะข้อมูลบางส่วนเป็นค่าว่างไม่ได้ ยกตัวอย่าง
 - เมื่อต้องการเพิ่มรายวิชา 9330 ฐานข้อมูลในระบบลงทะเบียน แต่ไม่สามารถทำได้ เพราะ ไม่มีข้อมูลนิสิตที่จะลงทะเบียนเรียน โดยที่ StudentNum เป็นคีย์หลักของตาราง ซึ่งเป็นค่าว่างไม่ได้

StudentNum	CourseNum	Student Name	Address	Course
S21	9201	สมบัติ	กรุงเทพฯ	บัญชีเบื้องต้น
S21	9267	สมบัติ	กรุงเทพฯ	ฟิสิกส์
S24	9267	รัตนา	ระยอง	ฟิสิกส์
S30	9201	วิชา	เชียงใหม่	คอมพิวเตอร์เบื้องต้น
S30	9322	วิชา	เชียงใหม่	คณิตศาสตร์
	9330			ฐานข้อมูล

การกระจายความสัมพันธ์ (Decompositions)

- การกระจายความสัมพันธ์ในการออกแบบฐานข้อมูลเชิงสัมพันธ์แสดงถึงการแบ่งโครงร่าง (Schema) ที่เกี่ยวข้องให้มีความสัมพันธ์ที่มีขนาดเล็กลง
 - มีความซับซ้อนที่น้อยลง เพื่อหลีกเลี่ยงความซ้ำซ้อน
 - สามารถสืบค้นได้อย่างถูกต้อง โดยสร้างความสัมพันธ์ที่มีขนาดเล็กสำหรับข้อมูลกลุ่มต่างๆ

StudentNum	Student Name	Address
S21	สมบัติ	กรุงเทพฯ
S24	รัตนา	ระยอง
S30	วิชา	เชียงใหม่

StudentNum	CourseNum
S21	9201
S21	9267
S24	9267
S30	9201
S30	9322

CourseNum	Course
9201	บัญชีเบื้องต้น
9267	ฟิสิกส์
9201	คอมพิวเตอร์เบื้องต้น
9322	คณิตศาสตร์

Functional Dependencies

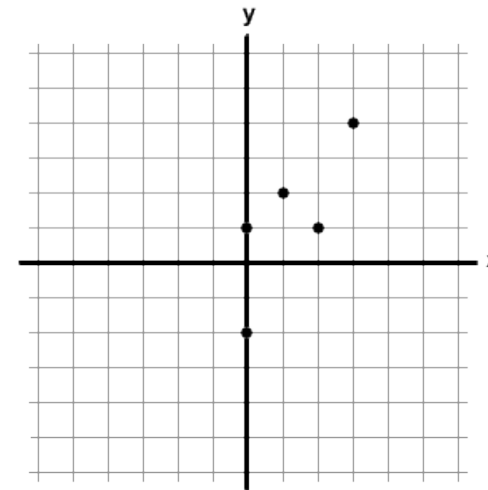
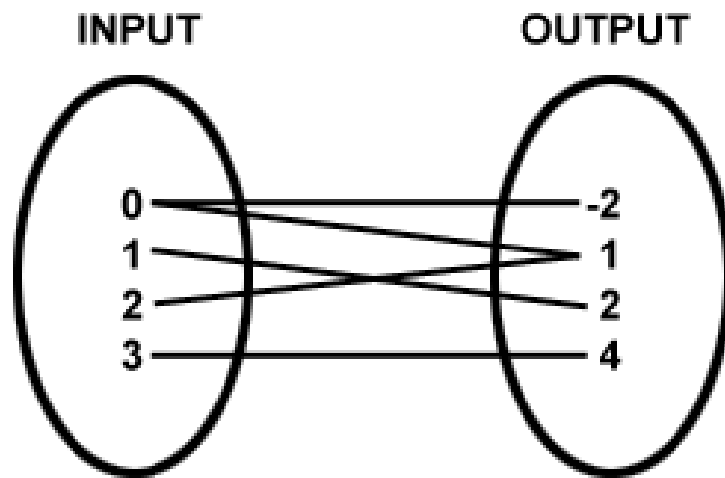
- Functional Dependencies (FD) เป็นชนิดของความสัมพันธ์เพื่อสร้างเงื่อนไขให้เกิดความถูกต้องของข้อมูล โดยนำแนวคิดของการมี Super key มาใช้ โดยกำหนดให้มีการพึ่งพาระหว่าง subset ของ แอตทริบิวต์ของความสัมพันธ์ที่กำหนดไว้ คำจำกัดความของ functional dependency ได้ให้นิยามไว้ดังนี้
- นิยาม: กำหนด schema เชิงสัมพันธ์ R กับ subset ของแอตทริบิวต์ของ A และ B กล่าวได้ว่ามี functional dependency $A \rightarrow B$ ปรากฏอยู่ ถ้าค่าใน A แสดงถึงค่าที่ B สามารถอ้างอิงถึงได้ในทุกอินสแตนซ์ r ของความสัมพันธ์ R

Functional Dependencies

- FD นำหลักการซูเปอร์คีย์มาใช้งานเนื่องจาก แอตทริบิวต์ ในเซต A จะเป็นตัวกำหนดค่าของแอตทริบิวต์ในเซต B
 - สำหรับความสัมพันธ์ R ที่กำหนด นอกจากนี้ การอ้างอิงฟังก์ชัน $A \rightarrow B$ มีอยู่เมื่อค่าของ B จะขึ้นอยู่กับ A อย่างเป็นฟังก์ชัน เนื่องจากใกล้เคียงกับแนวคิดของฟังก์ชัน FD เชื่อมโยงความสัมพันธ์ของเซตแอตทริบิวต์ A ไปยังเซต B
- Functional Dependency หมายถึง “A กำหนด B”, “B คือขึ้นอยู่กับ A” หรือ “A สัมพันธ์กับ B” และกล่าวได้ว่า “ $A \rightarrow B$ ”

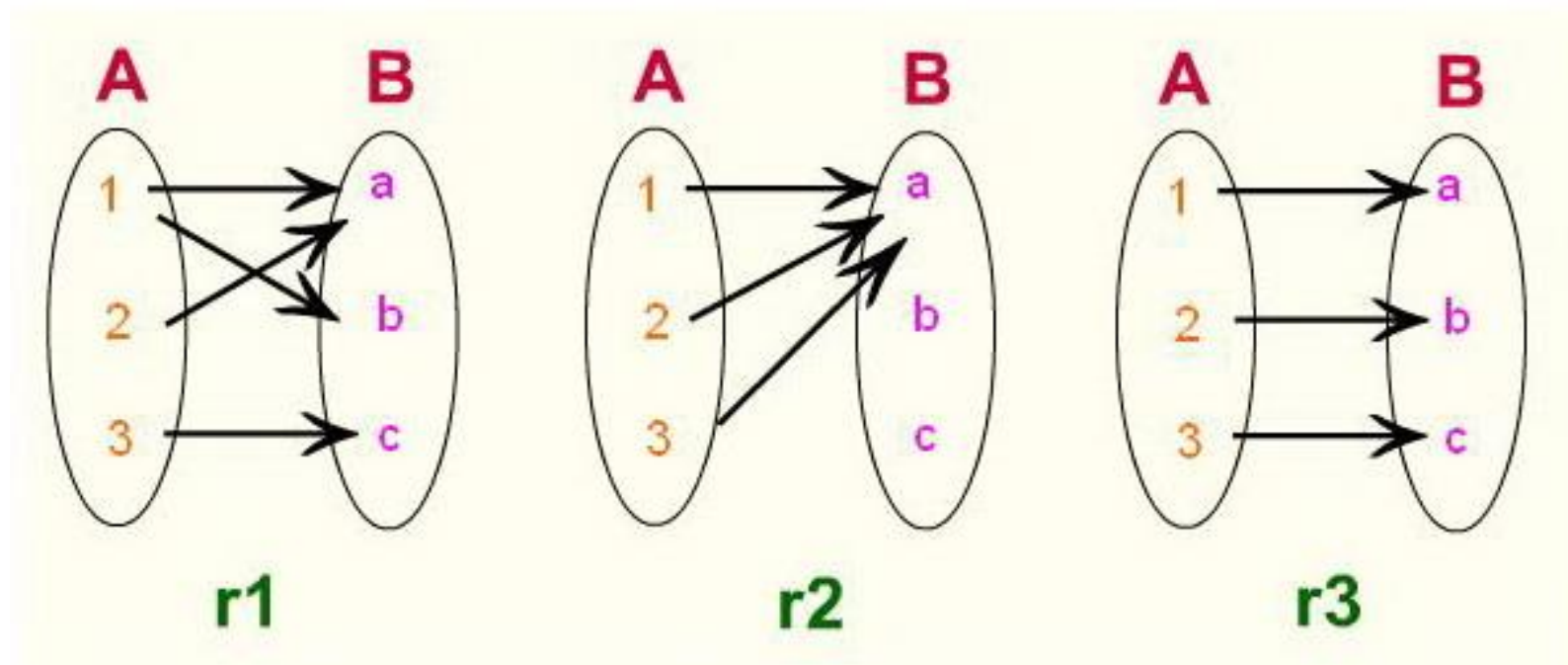
ความสัมพันธ์ (Relation)

- ความสัมพันธ์ (relation) เป็นความสัมพันธ์ระหว่างการเชื่อมโยงระหว่างสมาชิกภายในเซตสองเซต โดยที่
 - โดเมน (Domain) ของความสัมพันธ์ r คือ เซตที่มีสมาชิกตัวหน้าของทุกคู่อันดับในความสัมพันธ์ r ซึ่งเป็นรายการข้อมูลนำเข้า
 - เรนจ์ (range) หรือโคโดเมน (codomain) ของความสัมพันธ์ r คือ เซตที่มีสมาชิกตัวหลังของทุกคู่อันดับในความสัมพันธ์ r ซึ่งเป็นรายการผลลัพธ์



ฟังก์ชัน (Function)

- ฟังก์ชัน (function) เป็นความสัมพันธ์ที่แต่ละค่าของข้อมูลนำเข้าจะมีผลลัพธ์เป็นค่าเดียวเท่านั้น
- เหตุผลหลักของการ**ไม่อนุญาตให้มีผลลัพธ์หลายค่ากับค่านำเข้าหนึ่งค่า**คือ เมื่อเราใช้ฟังก์ชันเดิมกับค่านำเข้าใดค่าหนึ่งแล้วหลายครั้ง เช่นถ้าค่าโดเมน a หรือ b มีค่าเท่ากัน ($a = b$) ควรจะได้ค่าผลลัพธ์ $f(a) = f(b)$ เสมอ



ฟังก์ชัน (Function)

- จงพิจารณาว่า ความสัมพันธ์ใดต่อไปนี้เป็นฟังก์ชัน

1. $r_1 = \{(1,3), (2,5), (3,7), (4,9)\}$

2. $r_2 = \{(1,-1), (1,1), (2,4)\}$

3. $r_3 = \{(2,0), (4,0), (6,0)\}$

r_1 และ r_3 เป็นฟังก์ชัน

4. แต่ r_2 ไม่เป็นฟังก์ชัน เนื่องจาก ถ้า $x = 1$ จะได้ $y = -1$ และ $y = 1$

Functional Dependencies

- คุณสมบัติของ FD
 - เป็นความสัมพันธ์แบบหนึ่งต่อหนึ่ง (1-to-1 Relationship)
 - เป็นจริงในทุกกรณี (Hold For All Time)
 - เป็น Non Trivial
- FD Trivial คือ แอตทริบิวต์ทางขวาหรือแอตทริบิวต์ที่ถูกกำหนดเป็นสับเซต (Subset) ของแอตทริบิวต์ที่อยู่ทางซ้ายซึ่งเป็นแอตทริบิวต์ที่เป็นผู้กำหนด (Determinant)

Functional Dependencies

StudentNum	Student Name	Address	Major
S21	สมบัติ	กรุงเทพฯ	CS
S24	รัตนา	ระยอง	IT
S30	วิชา	เชียงใหม่	CS
S32	สมบัติ	ระยอง	IT

เซตต่อไปนี้ของ FDs ยังคงเป็นจริง

$\{\text{StudentNum} \rightarrow \text{StudentName},$

$\text{StudentNum} \rightarrow \text{Address},$

$\text{StudentNum} \rightarrow \text{Major},$

$\text{StudentNum} \rightarrow \text{StudentName} \rightarrow \text{Address},$

$\text{StudentNum} \rightarrow \text{StudentName} \rightarrow \text{Major}\}$

ตัวอย่าง Trivial FD :

$(\text{StudentNum} \rightarrow \text{StudentName}) \rightarrow \text{StudentName}$

$\text{StudentNum} \rightarrow \text{StudentNum}$

โดยทั่วไป functional dependency $A \rightarrow B$ ในกรณีที่ B คือ subset ของ A หมายถึง B บรรจุอยู่ใน A (ด้านขวามือจะเป็นส่วนหนึ่งของทางด้านซ้ายมือ) ในการออกแบบฐานข้อมูลเชิงสัมพันธ์โดยทั่วไปเราสนใจ non-trivial FDs ที่ช่วยกำหนดความถูกต้องของเงื่อนไขความสัมพันธ์

Functional Dependencies

STUDENT_ID	STUDENT	RANK	COLLEGE
0001	Ria Sinha	6	Fergusson
0002	Vivek Kaul	15	PICT
0003	George Smith	9	IIT
0004	Will Brown	1	IIT

- จงเขียน Functional Dependencies ที่เป็นจริง

Full FD และ Partial FD

- พิจารณารีเลชัน R ที่ประกอบด้วยแอตทริบิวต์ A และ B โดยที่ $A \rightarrow B$ แล้ว
- แอตทริบิวต์ B เป็น Full FD บนแอตทริบิวต์ A ถ้าทุกแอตทริบิวต์ที่ไม่ใช่คีย์ (B) ขึ้นอยู่กับคีย์ทั้งหมด (A)
 - ไม่สามารถนำบางแอตทริบิวต์ออกจาก A ได้ เพราะจะทำให้คุณสมบัติ DB หายไป
- แอตทริบิวต์ B เป็น Partial FD บนแอตทริบิวต์ A ถ้าทุกแอตทริบิวต์ที่ไม่ใช่คีย์ (B) ขึ้นอยู่กับคีย์บางส่วน (A)
 - สามารถนำบางแอตทริบิวต์ออกจาก A ได้ โดยที่ $A \rightarrow B$ ยังคงอยู่เหมือนเดิม

StudentNum	Student Name	Address	Major
S21	สมบัติ	กรุงเทพฯ	CS
S24	รัตนา	ระยอง	IT
S30	วิชา	เชียงใหม่	CS
S32	สมบัติ	ระยอง	IT

Full FD:

StudentNum \rightarrow StudentName

StudentNum \rightarrow Address

StudentNum \rightarrow Major

Partial FD:

StudentNum, StudentName \rightarrow Address

StudentNum, StudentName \rightarrow Major

Transitive Dependency

- พิจารณา รีเลชัน R ที่ประกอบด้วยแอตทริบิวต์ A, B และ C ถ้า $A \rightarrow B$ และ $B \rightarrow C$ ดังนั้นแอตทริบิวต์ C เป็น Transitive Dependency บนแอตทริบิวต์ A หรือกล่าวได้ว่า รีเลชัน R มี Transitive Dependency ถ้า
 - แอตทริบิวต์ที่ไม่ใช่คีย์ของรีเลชัน (Non Key Attribute) ขึ้นอยู่กับ แอตทริบิวต์ที่ไม่ใช่คีย์ของรีเลชัน

STUDENTS

StudentNum	MajorCode	MajorName
S21	CS	วิทยาการคอมพิวเตอร์
S24	IT	เทคโนโลยีสารสนเทศ
S30	CS	วิทยาการคอมพิวเตอร์
S32	IT	เทคโนโลยีสารสนเทศ

$\text{StudentNum} \rightarrow \text{MajorCode}$

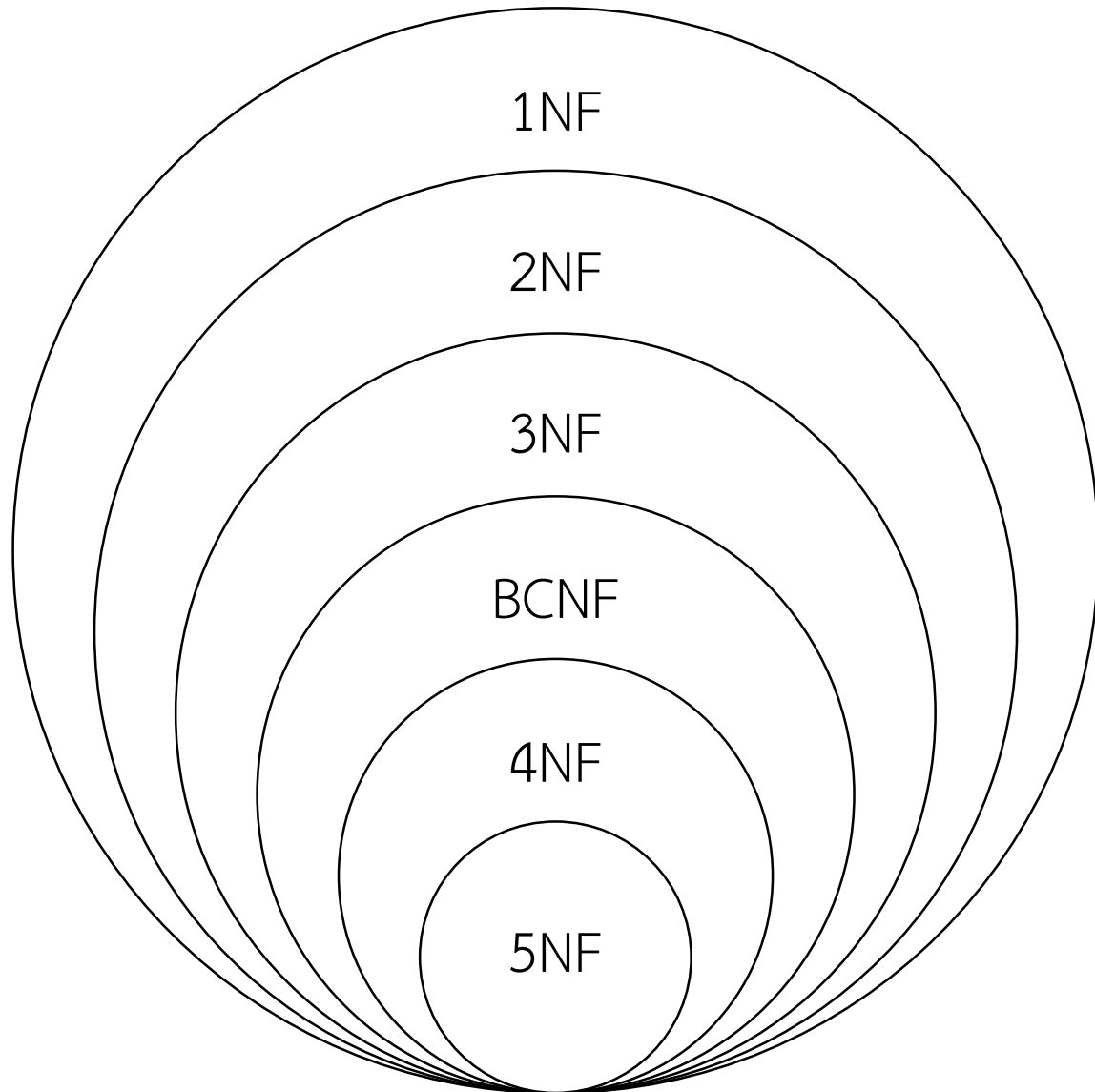
$\text{MajorCode} \rightarrow \text{MajorName}$

ดังนั้น STUDENTS มี Transitive Dependency

กฎการอนุมานสำหรับ FD (Inference Rules)

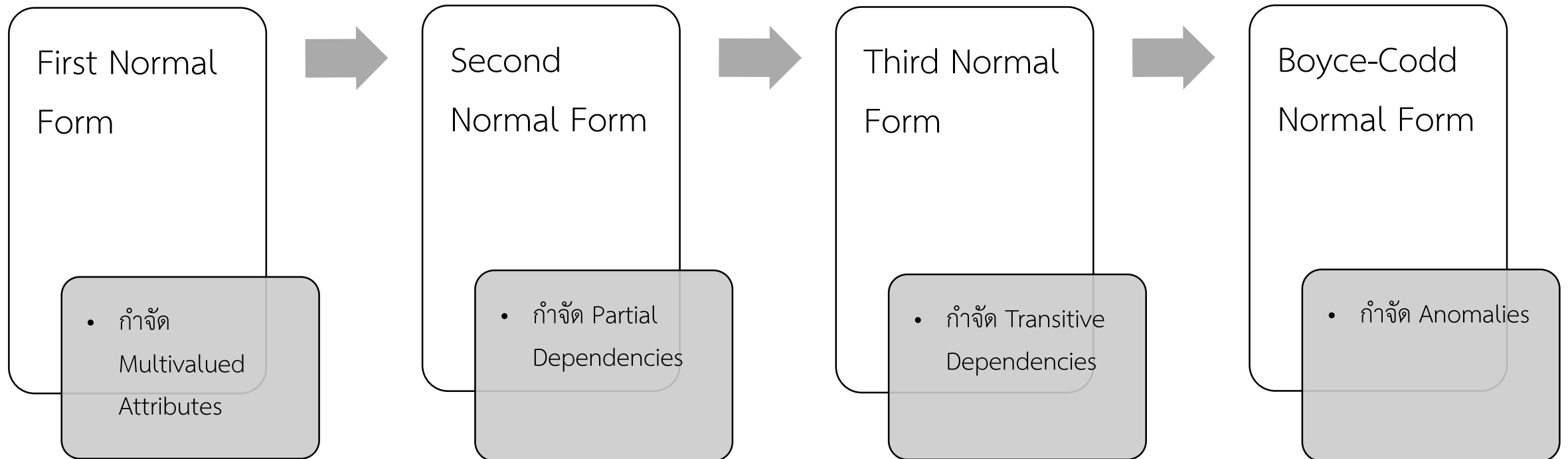
- กฎการอนุมาน (Inference Rules) หรือกฎอาร์มสตรอง (Armstrong's Axioms) เป็นกฎที่บ่งชี้การได้มาของสมบัติปิดของเซตการขึ้นอยู่กับ (Derive Closure S+) Inference Rules ประกอบด้วย 3 กฎ ดังนี้
 - Reflexivity ถ้า B เป็น subset ของแอตทริบิวต์ในเซต A แล้ว $A \rightarrow B$
 - Augmentation ถ้า $A \rightarrow B$ และ C เป็นแอตทริบิวต์อื่น แล้ว $AC \rightarrow BC$
 - Transitivity ถ้า $A \rightarrow B$ และ $B \rightarrow C$ ดังนั้น $A \rightarrow C$
- นอกจากนี้ มีกฎประยุกต์อีก 4 ข้อคือ
 - Self-determination $A \rightarrow A$
 - Decomposition ถ้า $A \rightarrow B, C$ แล้ว $A \rightarrow B$ และ $A \rightarrow C$
 - Union ถ้า $A \rightarrow B$ และ $B \rightarrow C$ แล้ว $A \rightarrow B, C$
 - Composition ถ้า $A \rightarrow B$ และ $C \rightarrow D$ แล้ว $A, C \rightarrow B, D$

การนอร์มัลไลส์ (Normalization)



- กระบวนการ Normalization เป็นกระบวนการในการออกแบบฐานข้อมูลเชิงสัมพันธ์เพื่อให้มีการปรับปรุงเปลี่ยนแปลงแบบแผนเชิงสัมพันธ์ให้อยู่ในรูปแบบที่เหมาะสม คือ
- **การกำจัดความซ้ำซ้อนในความสัมพันธ์** และลดปัญหาที่จะตามมา ได้แก่ ความผิดปกติอันเนื่องมาจากการเพิ่ม ลบ และปรับปรุง ข้อมูล เป็นต้น

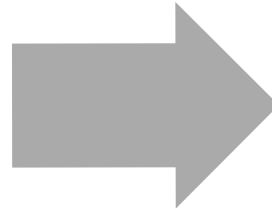
การนอร์มัลไลซ์ (Normalization)



การนอร์มัลไลส์ (Normalization)

Fourth Normal Form

- กำจัด Multivalued Dependencies



Fifth Normal Form

- กำจัด Anomalies อื่น ๆ

นอร์มัลฟอร์มระดับที่ 1 (First Normal Form (1NF))

- ความสัมพันธ์นั้นอยู่ในรูปของ First Normal Form โดยแอตทริบิวต์ของทั้งหมด มีโดเมนที่ไม่สามารถแบ่งออกเป็นส่วนๆ ได้ หรือ การทำให้เป็นหน่วยที่เล็กที่สุด การทำนอร์มัลฟอร์มระดับ 1NF จะต้องกำจัด multi-valued Attribute ของรีเลชันออกไป ด้วยวิธีการดังต่อไปนี้

1. การเพิ่มข้อมูลให้เต็มทุกช่องเซลล์ภายในรีเลชัน

FD: StudentNum, CourseNum -> StudentName, Course Day, Time, LecturerCode, LecturerName

StudentNum	StudentName	CourseNum	Course	Day	Time	LecturerCode	LecturerName
S21	สมบัติ	9201	บัญชี	อังคาร	9.00-12.00	L01	รัชนิกร
		9267	ฟิสิกส์	พฤหัสบดี	13.00-16.00	L02	สามารถ
S24	รัตนา	9267	ฟิสิกส์	พฤหัสบดี	13.00-16.00	L02	สามารถ
		9201	คอมพิวเตอร์	ศุกร์	9.00-12.00	L03	มานะ
S30	วิชา	9201	คอมพิวเตอร์	ศุกร์	9.00-12.00	L03	มานะ

StudentNum	StudentName	CourseNum	Course	Day	Time	LecturerCode	LecturerName
S21	สมบัติ	9201	บัญชี	อังคาร	9.00-12.00	L01	รัชนิกร
S21	สมบัติ	9267	ฟิสิกส์	พฤหัสบดี	13.00-16.00	L02	สามารถ
S24	รัตนา	9267	ฟิสิกส์	พฤหัสบดี	13.00-16.00	L02	สามารถ
S24	รัตนา	9201	คอมพิวเตอร์	ศุกร์	9.00-12.00	L03	มานะ
S30	วิชา	9201	คอมพิวเตอร์	ศุกร์	9.00-12.00	L03	มานะ

นอร์มัลฟอร์มระดับที่ 1 (First Normal Form (1NF))

2. การแยกตาราง เพื่อกำจัด Multi-valued Attribute ในรีเลชันออกไป จะทำให้เพิ่มรีเลชันจำนวนมากขึ้น
- ความสัมพันธ์จากตารางดังกล่าวยังไม่ได้อยู่ในรูปของ 1NF เนื่องจากตารางนี้ยังประกอบด้วยแอตทริบิวต์ Actors ที่มีค่าแบบ Multi-valued ได้ เมื่อต้องการแปลงเป็นความสัมพันธ์ 1NF เรากระจายตารางเพิ่มเติมลงในตาราง Movie Table และ Cast Table

Movie_Title	Year	Type	Director	Director_DOB	Actors
Notting Hill	1999	Romantic	Roger M	05/06/1956	Hugh G Rhys I
Lagaan	2000	Drama	Ashutosh G	15/02/1968	Aamir K Gracy S

Cast Table

Movie Table

Movie_Title	Year	Type	Director	Director_DOB
Notting Hill	1999	Romantic	Roger M	05/06/1956
Lagaan	2000	Drama	Ashutosh G	15/02/1968

Movie_Title	Year	Actors
Notting Hill	1999	Hugh G
Notting Hill	1999	Rhys I
Lagaan	2000	Aamir K
Lagaan	2000	Gracy S

นอร์มัลฟอร์มระดับที่ 2 (Second Normal Form (2NF))

- ความสัมพันธ์ของตารางจะถือว่าอยู่ในรูปของ Second Normal Form เมื่อตาราง
 - อยู่ใน 1NF
 - **ไม่มี nonkey attribute ที่ขึ้นอยู่กับบางส่วนของ candidate key** แต่ขึ้นอยู่กับทั้งหมดของ candidate key จากข้อกำหนดข้างต้น ความสัมพันธ์ที่มีแอตทริบิวต์เดียวเป็น candidate key จะถือว่าอยู่ในรูป 2NF เสมอ

1. กำหนดแอตทริบิวต์ที่เป็น Key attribute ทั้งหมด

- StudentNum
- CourseNum
- StudentNum, CourseNum

2. กำหนด FD ทั้งหมดในรีเลชันที่มีแอตทริบิวต์ใน

ข้อที่ 1 เป็น Determinant (ตัวกำหนดหรือระบุค่า)

FD1: StudentNum \rightarrow StudentName

FD2: CourseNum \rightarrow Course

FD3: StudentNum, CourseNum \rightarrow

Day, Time, LecturerCode, LecturerName

StudentNum	StudentName	CourseNum	Course	Day	Time	LecturerCode	LecturerName
S21	สมบัติ	9201	บัญชี	อังคาร	9.00-12.00	L01	รัชนิกร
S21	สมบัติ	9267	ฟิสิกส์	พฤหัสบดี	13.00-16.00	L02	สามารถ
S24	รัตนา	9267	ฟิสิกส์	พฤหัสบดี	13.00-16.00	L02	สามารถ
S24	รัตนา	9201	คอมพิวเตอร์	ศุกร์	9.00-12.00	L03	มานะ
S30	วิชา	9201	คอมพิวเตอร์	ศุกร์	9.00-12.00	L03	มานะ

นอร์มัลฟอร์มระดับที่ 2 (Second Normal Form (2NF))

StudentNum	StudentName	CourseNum	Course	Day	Time	LecturerCode	LecturerName
S21	สมบัติ	9201	บัญชี	อังคาร	9.00-12.00	L01	รัชนี้กร
S21	สมบัติ	9267	ฟิสิกส์	พฤหัสบดี	13.00-16.00	L02	สามารถ
S24	รัตนา	9267	ฟิสิกส์	พฤหัสบดี	13.00-16.00	L02	สามารถ
S24	รัตนา	9201	คอมพิวเตอร์	ศุกร์	9.00-12.00	L03	มานะ
S30	วิชา	9201	คอมพิวเตอร์	ศุกร์	9.00-12.00	L03	มานะ

StudentNum	StudentName
S21	สมบัติ
S24	รัตนา
S30	วิชา

CourseNum	Course
9201	บัญชี
9267	ฟิสิกส์
9201	คอมพิวเตอร์

StudentNum	CourseNum	Day	Time	LecturerCode	LecturerName
S21	9201	อังคาร	9.00-12.00	L01	รัชนี้กร
S21	9267	พฤหัสบดี	13.00-16.00	L02	สามารถ
S24	9267	พฤหัสบดี	13.00-16.00	L02	สามารถ
S24	9201	ศุกร์	9.00-12.00	L03	มานะ
S30	9201	ศุกร์	9.00-12.00	L03	มานะ

นอร์มัลฟอร์มระดับที่ 3 (Third Normal Form (3NF))

- ความสัมพันธ์ที่อยู่ในรูปของ Third Normal Form หมายถึงตารางที่
 - อยู่ในรูปของ 2NF และ
 - **ไม่มี nonkey attribute ที่ขึ้นกับ candidate transitively key** ที่มีแอตทริบิวต์ทุกตัวขึ้นโดยตรงกับ primary key และไม่ผ่าน transitive relation ยกตัวอย่างเช่น
 - แอตทริบิวต์ Z อาจขึ้นกับ non-key attribute Y และ Y ขึ้นกับ primary key X ตามลำดับ
โดยอ้างถึงกฎความสัมพันธ์แบบ Transitivity หมายถึง เมื่อ $X \rightarrow Y$ และ $Y \rightarrow Z$ แล้ว $X \rightarrow Z$
- เมื่อพิจารณาตัวอย่างก่อนหน้านี้

FD	เหตุผล
1: StudentNum \rightarrow StudentName	ไม่มี transitive dependency เพราะมี StudentName ตัวเดียวที่เป็น Non-key attributes
2: CourseNum \rightarrow Course	ไม่มี transitive dependency เพราะมี Course ตัวเดียวที่เป็น Non-key attributes
3: StudentNum, CourseNum \rightarrow Day, Time, LecturerCode, LecturerName แยกเรเลชันระดับ 3 NF เป็น 3.1 : LecturerCode \rightarrow LecturerName 3.2 : StudentNum, CourseNum \rightarrow Day, Time, LecturerCode	มี tr: ansitive dependency เพราะ LecturerCode เป็นตัว Determinant ของ LecturerName LecturerCode \rightarrow LecturerName

นอร์มัลฟอร์มระดับที่ 3 (Third Normal Form (3NF))

StudentNum	CourseNum	Day	Time	LecturerCode	LecturerName
S21	9201	อังคาร	9.00-12.00	L01	รัชนิกร
S21	9267	พฤหัสบดี	13.00-16.00	L02	สามารถ
S24	9267	พฤหัสบดี	13.00-16.00	L02	สามารถ
S24	9201	ศุกร์	9.00-12.00	L03	มานะ
S30	9201	ศุกร์	9.00-12.00	L03	มานะ

StudentNum	StudentName
S21	สมบัติ
S24	รัตนา
S30	วิชชา

CourseNum	Course
9201	บัญชี
9267	ฟิสิกส์
9201	คอมพิวเตอร์

StudentNum	CourseNum	Day	Time	LecturerCode
S21	9201	อังคาร	9.00-12.00	L01
S21	9267	พฤหัสบดี	13.00-16.00	L02
S24	9267	พฤหัสบดี	13.00-16.00	L02
S24	9201	ศุกร์	9.00-12.00	L03
S30	9201	ศุกร์	9.00-12.00	L03

LecturerCode	LecturerName
L01	รัชนิกร
L02	สามารถ
L03	มานะ

นอร์มัลฟอร์ม Boyce Codd Normal Form (BCNF)

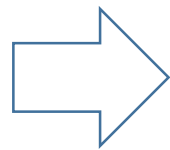
- ลักษณะของนอร์มัลฟอร์มแบบ Boyce-Codd Normal Form (BCNF) จะเป็น 3NF แบบเข้มงวด ที่นำไปใช้ต่อความสัมพันธ์ ที่อาจมีการ overlapping ของ candidate keys ความสัมพันธ์นี้เรียกว่าเป็น BoyceCodd normal form
 - ถ้าเป็น 3NF และทุก non-trivial FD สำหรับความสัมพันธ์นี้มี candidate key เป็นดีเทอร์มิแนนต์ กล่าวคือ สำหรับทุก $X \rightarrow Y$, X คือ candidate key
- แปลงรีเลชันที่เป็นผลลัพธ์ของ 3NF ให้เป็น BCNF โดยกำหนดให้อาจารย์แต่ละคนสอน 1 รายวิชา
 - FD1: StudentNum \rightarrow StudentName เป็น BCNF
 - FD2: CourseNum \rightarrow Course เป็น BCNF
 - FD3.1 : LecturerCode \rightarrow LecturerName เป็น BCNF
 - FD3.2 : StudentNum, CourseNum \rightarrow Day, Time, LecturerCode ไม่เป็น BCNF
- แต่เนื่องจากโจทย์กำหนดให้อาจารย์แต่ละคนสอน 1 รายวิชา ทำให้ได้ LecturerCode \rightarrow CourseNum
- ทำให้ FD3.2 ไม่เป็น BCNF แปลง FD3.2 เป็น BCNF
 - FD3.2.1 : StudentNum, LecturerCode \rightarrow Day, Time
 - FD3.2.2 : LecturerCode \rightarrow CourseNum

นอร์มัลฟอร์มระดับที่ 4 (Fourth Normal Form (4NF))

- รีเลชันที่อยู่ใน 4NF จะต้องเป็นรีเลชันที่อยู่ใน 3NF และ ไม่มี Multi-valued Dependency (MVD)
 - การแปลงรีเลชันใด ๆ ให้อยู่ในรูปนอร์มัลฟอร์มระดับที่ 4 จะต้องกำจัด Multi-valued Dependency ออกจากรีเลชัน
- กำหนดรีเลชัน CLASS ซึ่งประกอบด้วยแอตทริบิวต์ STUDENT_ID หมายถึงรหัสนิสิต COURSE หมายถึงชื่อวิชา และ HOBBY หมายถึงงานอดิเรก

จากรีเลชันแสดงว่านิสิตหนึ่งคนมีวิชาเรียนได้มากกว่าหนึ่งวิชา และนิสิตแต่ละคนจะมีงานอดิเรกได้มากกว่าหนึ่งงาน ทำให้รีเลชันมีลักษณะไม่เป็น 4NF เพราะมี Multi-valued Dependency เราสามารถแยกออกเป็นสองรีเลชัน คือ

STUDENT_ID	COURSE	HOBBY
21	Computer	Dancing
21	Math	Singing
34	Chemistry	Dancing
74	Biology	Cricket
59	Physics	Hocke



STUDENT_ID	COURSE
21	Computer
21	Math
34	Chemistry
74	Biology
59	Physics

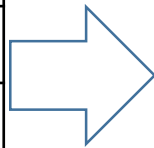
STUDENT_ID	HOBBY
21	Dancing
21	Singing
34	Dancing
74	Cricket
59	Hocke

นอร์มัลฟอร์มระดับที่ 5 (Fifth Normal Form (5NF))

- รีเลชันที่อยู่ในระดับ 5NF จะต้องเป็นรีเลชันที่อยู่ในระดับ 4NF และไม่มี Joint Dependency (JD)
 - การแปลงรีเลชันใด ๆ ให้อยู่ในรูปนอร์มัลฟอร์มระดับที่ 5 จะต้องกำจัด Joint Dependency ออกจากรีเลชัน

จอห์นเรียนทั้งวิชาคอมพิวเตอร์และคณิตศาสตร์สำหรับภาคเรียนที่ 1 แต่เขาไม่ได้เรียนวิชาคณิตศาสตร์สำหรับภาคเรียนที่ 2
ในกรณีนี้การรวมกันของฟิลด์เหล่านี้ทั้งหมดเป็นสิ่งจำเป็นในการระบุข้อมูลที่ถูกต้อง

SUBJECT	LECTURER	SEMESTER
Computer	Anshika	Semester 1
Computer	John	Semester 1
Math	John	Semester 1
Math	Akash	Semester 2
Chemistry	Praveen	Semester 1



SEMESTER	SUBJECT
Semester 1	Computer
Semester 1	Math
Semester 1	Chemistry
Semester 2	Math

SUBJECT	LECTURER
Computer	Anshika
Computer	John
Math	John
Math	Akash

SEMESTER	LECTURER
Semester 1	Anshika
Semester 1	John
Semester 1	John
Semester 2	Akash
Semester 1	Praveen