

บทที่ 7

การปรับบรรทัดฐานข้อมูล (Normalization)

1. การปรับบรรทัดฐานข้อมูล หรือ นอร์มอลไลซ์เซชัน

ในอดีตการเก็บรวบรวมข้อมูลของระบบงาน ส่วนมากจะเป็นการเก็บรายละเอียดของข้อมูลทุกอย่างไว้ด้วยกันหมด อาจจะมีอยู่ในหลายรูปแบบ เช่นอยู่ในรูปแบบของเอกสาร หรือรายงานต่าง ๆ ซึ่งจะมีปัญหาการซ้ำซ้อนของข้อมูล ทำให้ข้อมูลมีขนาดใหญ่เกินความจำเป็นสิ้นเปลืองเนื้อที่ในการเก็บข้อมูล และยากต่อการปรับปรุงข้อมูล ไม่ว่าจะเป็นการเพิ่ม หรือการลดข้อมูล ในปี ค.ศ.1917 Dr. Edgar Frank Codd เป็นคนแรก ที่ได้คิดหาวิธีการหรือแนวทางที่ใช้ในการตรวจสอบ และแก้ปัญหาดังกล่าวโดยการนำเค้าร่างของตารางหรือรีเลชัน มาทำให้อยู่ในรูปแบบที่เป็นรูปแบบบรรทัดฐาน (Normal Form) หรืออีกนัยหนึ่งคือเป็นการแปลงข้อมูลที่มีโครงสร้างแบบตาราง จากรูปแบบที่มีความซ้ำซ้อนให้อยู่ในรูปแบบที่ใช้งานได้ง่าย ทำให้การเพิ่มข้อมูล การลบข้อมูล หรือการแก้ไขข้อมูลที่อยู่ในรีเลชัน ไม่เกิดความผิดพลาดขึ้น โดยการทำให้ข้อมูลในแต่ละรีเลชันอยู่ในรูปแบบที่เป็นหน่วยที่เล็กที่สุดที่ไม่สามารถแตกออกเป็นหน่วยย่อย ๆ ได้อีก แต่ยังคงความสัมพันธ์ระหว่างข้อมูลในรีเลชันต่าง ๆ ไว้ตามหลักการที่กำหนดไว้ในแบบจำลองฐานข้อมูลเชิงสัมพันธ์ เพราะการออกแบบฐานข้อมูลที่ดีจะต้องไม่มีความซ้ำซ้อนในการจัดเก็บข้อมูล

Codd ได้คิดค้นรูปแบบบรรทัดฐานดังกล่าว และให้ชื่อว่ารูปแบบบรรทัดฐานระดับที่ 1 รูปแบบบรรทัดฐานระดับที่ 2 และ รูปแบบบรรทัดฐานระดับที่ 3 ต่อมาเขาได้พบว่ารูปแบบบรรทัดฐานระดับที่ 3 มีข้อจำกัดบางอย่าง ในปี ค.ศ. 1974 Codd และ R. Boyce จึงได้ร่วมกันพัฒนารูปแบบบรรทัดฐานระดับที่ 3 ขึ้นมาใหม่ ซึ่งในปัจจุบันเรียกกันว่า รูปแบบบรรทัดฐานบอยซ์-คอดด์ (Boyce-codd Normal Form หรือ BCNF) สำหรับรูปแบบบรรทัดฐานระดับที่ 4 และรูปแบบบรรทัดฐานระดับที่ 5 นั้นคิดค้นโดย Ronald Fagin ในปี ค.ศ.1977 และ 1979 ตามลำดับ

การออกแบบจำลองฐานข้อมูลเชิงสัมพันธ์ สามารถออกแบบได้ 2 ทาง คือ 1) ผ่านขั้นตอนการออกแบบจำลองข้อมูลเชิงสัมพันธ์ และ 2) ไม่ผ่านขั้นตอนการออกแบบจำลองข้อมูล คือนักวิเคราะห์และออกแบบระบบที่มีประสบการณ์สูง สามารถที่จะออกแบบจำลองฐานข้อมูลเชิงสัมพันธ์ได้โดยตรง โดยไม่ต้องผ่านการออกแบบจำลองมาก่อน อาศัยเพียงข้อมูลปฐมภูมิ (Primary Data) เช่น รายงาน หรือแบบฟอร์มต่าง ๆ เป็นข้อมูลเบื้องต้นในการออกแบบโมเดลฐานข้อมูลเชิงสัมพันธ์

1.1 ความหมายของการปรับบรรทัดฐานข้อมูล

มีผู้ให้ความหมายของ การปรับบรรทัดฐานข้อมูล ไว้ดังนี้ เช่น พุทธศิริ แสงตระกูล (2539) กล่าวว่า การปรับบรรทัดฐานข้อมูล “เป็นวิธีการที่ใช้ วิเคราะห์และจัดโครงสร้างของฐานข้อมูลใหม่ โดยพยายามลดความซ้ำซ้อนของโครงสร้างฐานข้อมูล เพื่อให้ได้ฐานข้อมูลที่เสถียรภาพ ซึ่งวิธีการทำคือปรับโครงสร้างของฐานข้อมูลให้อยู่ในรูปแบบอีร์มีลระดับต่าง ๆ ”

กิตติ ภัคดีวัฒนะกุล และ จำลอง ครูอุตสาหะ กล่าวว่า (2544) การปรับบรรทัดฐานข้อมูล “เป็นวิธีการที่ใช้ในการตรวจสอบและแก้ไขปัญหาด้านความซ้ำซ้อนของข้อมูล โดยการดำเนินการให้ข้อมูลในแต่ละรีเลชันอยู่ในรูปที่เป็นหน่วยที่เล็กที่สุด ไม่สามารถแตกออกเป็นหน่วยย่อย ๆ ได้อีก แต่ยังคงความสัมพันธ์ระหว่างข้อมูลในรีเลชันต่าง ๆ ไว้ตามหลักการที่กำหนดไว้ในแบบจำลองฐาน ข้อมูลเชิงสัมพันธ์”

ศิริลักษณ์ โรจนกิจอำนวย (2542) กล่าวว่า “เป็นกระบวนการที่ใช้ในการทดสอบการออกแบบรีเลชันตามเกณฑ์ของขั้นตอนต่าง ๆ ในการทำให้เป็นบรรทัดฐาน เป็นการพิจารณาว่าคีย์หลักหรือคีย์คู่แข่งสามารถระบุค่าของแอททริบิวต์อื่น ๆ ของทูเพิล”

จากความหมายของการปรับบรรทัดฐานข้อมูลดังที่ได้กล่าวมาข้างบนพอจะสรุปได้ว่า การปรับบรรทัดฐานข้อมูล เป็นกระบวนการที่ผู้ออกแบบฐานข้อมูลจะนำมาใช้ในการปรับปรุงหรือแปลงข้อมูลที่อยู่ในรูปแบบซ้ำซ้อน ให้อยู่ในรูปแบบบรรทัดฐาน โดยกำจัดความซ้ำซ้อนของข้อมูลในรีเลชันออกไป ให้อยู่ในรูปแบบที่เหมาะสม ง่ายกับการใช้งานในระบบฐานข้อมูล และก่อให้เกิดปัญหาน้อยที่สุด โดยการกระจายรีเลชันที่มีโครงสร้างซ้ำซ้อนออก เป็นรีเลชันย่อย ๆ มีโครงสร้างแบบง่าย ๆ มีรูปแบบกระชับรัด และข้อมูลอยู่ในรูปแบบบรรทัดฐานสามารถนำไปใช้งานได้มีประสิทธิภาพ และเป็นการลดเนื้อที่ในการจัดเก็บข้อมูลด้วยการปรับบรรทัดฐานข้อมูล จึงมีความสำคัญต่อการออกแบบระบบของฐานข้อมูลมาก ฐานข้อมูลที่ดีที่สามารถจัดการข้อมูลได้อย่างมีประสิทธิภาพ จะต้องถูกออกแบบโดยผ่านกระบวนการปรับบรรทัดฐานข้อมูลมาก่อนเสมอ

1.2 วัตถุประสงค์และประโยชน์ของการปรับบรรทัดฐานข้อมูล

วัตถุประสงค์ สำคัญของการปรับบรรทัดฐานข้อมูล คือ

- 1) ลดเนื้อที่ในการจัดเก็บข้อมูล การปรับบรรทัดฐานข้อมูลเป็นการลดความซ้ำซ้อนของข้อมูลในรีเลชัน (Data Redundancy) ซึ่งจะทำให้ประหยัดเนื้อที่ในการจัดเก็บข้อมูล
- 2) ลดปัญหาความไม่ตรงกันของข้อมูล (Data Inconsistency) ลดปัญหาข้อมูลที่ไม่ถูกต้องหรือความไม่ตรงกันของข้อมูลในฐานข้อมูลให้เหลือน้อยที่สุด เพราะหลังจากการปรับบรรทัดฐานข้อมูลแล้ว ข้อมูลในรีเลชันจะไม่ซ้ำกัน โอกาสที่จะเกิดความผิดพลาดในการปรับปรุงข้อมูลก็จะไม่เกิดขึ้น

3) ได้รูปแบบที่เป็นบรรทัดฐาน การปรับบรรทัดฐานข้อมูลทำให้การออกแบบฐานข้อมูลเชิงสัมพันธ์ให้อยู่ในรูปแบบที่เป็นบรรทัดฐาน และเพิ่มความคงทนให้กับโครงสร้างของฐานข้อมูล การค้นหาข้อมูลทำได้ง่าย และสามารถปรับปรุงข้อมูลได้จากแหล่งข้อมูลเพียงแหล่งเดียว

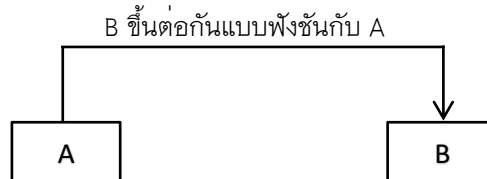
ประโยชน์หลัก ๆ ของการปรับบรรทัดฐานข้อมูล

- 1) เป็นเครื่องมือช่วยในการออกแบบจำลองฐานข้อมูลแบบเชิงสัมพันธ์ให้อยู่ในรูปแบบที่เป็นบรรทัดฐาน
- 2) ทำให้ทราบว่ารีเลชันที่ถูกออกแบบมานั้นอยู่ในรูปแบบบรรทัดฐานหรือไม่ และจะก่อให้เกิดปัญหาอะไรบ้าง? และจะมีวิธีแก้ไขปัญหานั้นอย่างไร?
- 3) เมื่อทำการปรับบรรทัดฐานข้อมูลรีเลชันที่มีปัญหาแล้ว จะสามารถรับประกันได้ว่ารีเลชันนั้นจะไม่มีปัญหาอีกหรือถ้ามีปัญหาก็จะมีน้อยลง

2. การขึ้นต่อกันแบบฟังก์ชัน (Functional Dependency)

การขึ้นต่อกันแบบฟังก์ชัน หมายถึงความสัมพันธ์ ระหว่างค่าแอททริบิวต์ที่อยู่ในรีเลชันเดียวกันในรูปของฟังก์ชัน โดยที่แอททริบิวต์จำนวนหนึ่งหรือกลุ่มของแอททริบิวต์ที่เมื่อประกอบกันแล้ว สามารถระบุค่าของแอททริบิวต์อื่น ๆ ในทูเปิลเดียวกันของรีเลชันได้ แอททริบิวต์ หรือกลุ่มของแอททริบิวต์ ที่เป็นคีย์หรือตัวระบุค่า หรือ ตัวเลือก ของแอททริบิวต์อื่น เรียกกันว่า “Determinant” ส่วนแอททริบิวต์ที่ถูกระบุค่า จะเรียกกันว่า “Dependent” ในทางทฤษฎีจะใช้สัญลักษณ์ลูกศร (\rightarrow) ในการบอกถึงการขึ้นต่อกัน

“Determinant” \rightarrow “Dependent”



สมมติให้ A และ B เป็นแอททริบิวต์ ในรีเลชัน R (รีเลชัน R ประกอบด้วย แอททริบิวต์ A และ B หรือเขียนแทนด้วย R (A , B) A และ B อาจเป็นแอททริบิวต์เดียว หรือเป็นกลุ่มของแอททริบิวต์ก็ได้

- ถ้าแอททริบิวต์ B มีการขึ้นต่อกันเชิงฟังก์ชันกับแอททริบิวต์ A จะเขียนเป็นสัญลักษณ์ ได้ดังนี้ : $A \rightarrow B$ นั่นคือ A เป็น “Determinant” ของ B หรือ B ขึ้นกับ A นั้นเอง หรืออาจจะกล่าวได้ว่า แต่ละค่าของ A สามารถระบุค่า B ได้เพียง 1 ค่า

2.1 ลักษณะของฟังก์ชันการขึ้นต่อกัน แบ่งออกได้หลายแบบ ดังนี้

1) กรณี “Determinant” และ “Dependent” มีเพียงอย่างละ 1 แอททริบิวต์ (การขึ้นต่อกันแบบฟังก์ชันที่เกิดขึ้นจากความสัมพันธ์ระหว่าง “Determinant” และ “Dependent” อย่างละ 1 แอททริบิวต์) เช่น ความสัมพันธ์ระหว่างหมายเลขบัตรประชาชน และชื่อเจ้าของบัตร ความสัมพันธ์ระหว่างรหัสนิสิตและอาจารย์ที่ปรึกษา ซึ่งเขียนได้หลายแบบ ดังนี้

- รหัสนิสิต → ชื่ออาจารย์ที่ปรึกษา หรือ
- ชื่ออาจารย์ที่ปรึกษา → รหัสนิสิต หรือ
- ชื่ออาจารย์ที่ปรึกษา ↔ รหัสนิสิต

อาจารย์ที่ปรึกษา

รหัสนิสิต	อาจารย์ที่ปรึกษา
10058475	วิภา ใจดี
10058480	มาลินี เย็นใจ
10058485	ทวีศักดิ์ เกิดเพชร

2) กรณีที่ “Determinant” 1 แอททริบิวต์ มี “Dependent” มากกว่า 1 แอททริบิวต์ (การขึ้นต่อกันแบบฟังก์ชันที่เกิดขึ้นจากความสัมพันธ์ระหว่าง “Determinant” 1 ค่า และ “Dependent” หลายค่า) เช่น รหัสนิสิต และข้อมูลที่ปรากฏอยู่บนบัตรประจำตัวนิสิต

อธิบาย : รหัสนิสิต เป็น “Determinant” ส่วน ข้อมูลเป็น “Dependent” เมื่อระบุค่าของรหัสนิสิตจะทำให้ทราบชื่อของนิสิต และข้อมูลต่าง ๆ ของนิสิตนั่นเอง

นิสิต

รหัสนิสิต	ชื่อนิสิต	คณะวิชา	ภูมิลำเนา	หมู่เลือด
10058475	นนุช ใจงาม	พยาบาล	เชียงราย	A
10258480	สมหมาย รักงาน	นิติศาสตร์	อุดรธานี	AB
10058485	นารี สุขใจ	พยาบาล	ชัยภูมิ	B
10659590	สมหญิง เอื้อเพื่อ	วิทยาศาสตร์	พะเยา	O
10759100	วิภา เอื้องคำ	ICT	ยะลา	O
10759201	ประสิทธิ์ ผลดี	ศิลปศาสตร์	เชียงใหม่	A

รื้อเลชันอาจารย์ที่ปรึกษา มีความสัมพันธ์ระหว่างแอมทริบิวท์แบบการขึ้นต่อกัน ดังนี้

- รหัสนิสิต → ชื่อนิสิต
- รหัสนิสิต → คณะวิชา
- รหัสนิสิต → ภูมิลำเนา
- รหัสนิสิต → ชื่อนิสิต , คณะวิชา , ภูมิลำเนา, หมู่เลือด

อธิบาย : รหัสนิสิต เป็น “Determinant” ส่วน ชื่อนิสิต, คณะวิชา, ภูมิลำเนา, หมู่เลือด เป็น “Dependent” เมื่อระบุค่าของรหัสนิสิตจะทำให้ทราบชื่อนิสิต, คณะวิชา, ภูมิลำเนา และ หมู่เลือดของนิสิตได้

3) กรณีที่ “Determinant” และ “Dependent” ต่างก็มีความสัมพันธ์ซึ่งกันและกัน ทั้ง “Determinant” และ “Dependent” ต่างก็สามารถทำหน้าที่ของแต่ละฝ่ายได้ (ฟังก์ชันการขึ้นต่อกันที่มีความสัมพันธ์สองทาง) เช่น ความสัมพันธ์ระหว่าง รหัสนิสิต และ หมายเลขบัตรประชาชน ที่ต่างก็ไม่มีข้อมูลซ้ำกันเลย และต่างก็สามารถทำหน้าที่เป็น “Determinant” ได้เช่นเดียวกัน ดังนี้

- รหัสนิสิต → หมายเลขบัตรประชาชน
- หมายเลขบัตรประชาชน → รหัสนิสิต
- รหัสนิสิต \leftrightarrow หมายเลขบัตรประชาชน

4) กรณีที่ “Determinant” มีมากกว่า 1 แอมทริบิวท์ และ “Dependent” มี 1 แอมทริบิวท์หรือมากกว่า 1 แอมทริบิวท์ (การขึ้นต่อกันแบบฟังก์ชัน ที่ต้องใช้ “Determinant” มากกว่า 1 แอมทริบิวท์ เพื่ออ้างอิง “Dependent”)

การลงทะเบียน

รหัสวิชา	หมู่เรียน	ผู้สอน	ห้องเรียน
005101	111	บรรพต สุดสงวน	CE 101
012102	121	มาลินี เย็นใจ	CE 411
150100	130	สุดใจ กลิ่นหอม	CE 222
106225	100	ทวีศักดิ์ เกิดเพชร	CE 301

รีเลชันอาจารย์ที่ปรึกษา มีความสัมพันธ์ระหว่างแบบการขึ้นต่อกัน ดังนี้

- รหัสวิชา, หมู่เรียน, ห้องเรียน → ชื่อผู้สอน

2.2 รูปแบบการขึ้นต่อกัน

รูปแบบการขึ้นต่อกันหรือความสัมพันธ์ระหว่างแอททริบิวต์ แบ่งออกเป็น 3 รูปแบบ

1) รูปแบบการขึ้นต่อกันแบบทั้งหมด (Fully Functional Dependency) เป็นรูปแบบของการขึ้นต่อกันที่แอททริบิวต์ที่ไม่ใช่คีย์หลักของรีเลชัน มีการขึ้นต่อแอททริบิวต์หรือกลุ่มของแอททริบิวต์ที่ทำหน้าที่เป็นคีย์หลักของรีเลชัน เป็นรูปแบบการขึ้นต่อกันที่แอททริบิวต์ที่เป็น “Determinant” มีขนาดเล็กที่สุดและสามารถระบุค่าแอททริบิวต์อื่นที่เป็น “Dependent” ได้ชัดเจน (แอททริบิวต์ด้านที่เป็นคีย์ สามารถใช้ระบุค่าของแอททริบิวต์ ที่เป็น “Determinant” จะทำให้ทราบค่าของแอททริบิวต์ที่เป็น “Dependent” ได้) เช่น

- รหัสลูกค้า → ชื่อลูกค้า

2) รูปแบบการขึ้นต่อกันบางส่วน (Partial Dependency) ความสัมพันธ์ระหว่างแอททริบิวต์แบบบางส่วนนี้ เกิดขึ้นเมื่อรีเลชันหนึ่ง ๆ มีคีย์หลักเป็นคีย์ผสมซึ่งประกอบไปด้วยแอททริบิวต์หลาย ๆ แอททริบิวต์รวมกัน และแอททริบิวต์บางส่วนของคีย์หลักสามารถที่จะไประบุค่าของแอททริบิวต์อื่นในทูเพิลเดียวกันที่ไม่ใช่คีย์หลัก (Non-key attribute) ของรีเลชันได้นั่นคือ หากทราบค่าใดค่าหนึ่งของบางแอททริบิวต์ที่เป็น “Determinant” ก็จะทำให้ทราบค่าแอททริบิวต์ อื่น ๆ ที่เป็น “Dependent” ได้

การลงทะเบียน

รหัสวิชา	หมู่เรียน	ผู้สอน	ห้องเรียน
005101	111	บรรพต สูดสงวน	CE 101
012102	121	มาลินี เย็นใจ	CE 411
150100	130	สุดใจ กลิ่นหอม	CE 222
106225	100	ทวีศักดิ์ เกิดเพชร	CE 301

ภาพที่ 7.1 รีเลชันที่มีรูปแบบการขึ้นต่อกันบางส่วน (Partial Dependency)

จากภาพจะเห็นว่าแอททริบิวต์หมู่เรียนจะขึ้นอยู่กับคีย์รหัสวิชา ในขณะที่แอททริบิวต์ชื่อผู้สอนจะขึ้นอยู่กับคีย์ห้องเรียน ข้อมูลที่อยู่ในรีเลชันเดียวกัน แต่ไม่ได้ขึ้นอยู่กับคีย์ใดคีย์หนึ่งทั้งหมด แต่จะขึ้นอยู่กับคีย์ใดคีย์หนึ่งเพียงบางส่วนเท่านั้น

3) รูปแบบการขึ้นต่อกันแบบส่งผ่าน (Transitive Dependency) เป็นรูปแบบการขึ้นต่อกันที่แอททริบิวต์อื่นที่ไม่ใช่คีย์หลักของรีเลชัน (Non-key attribute) แต่สามารถที่จะระบุค่าแอททริบิวต์อื่น ๆ ในทิวเพิลเดียวกันของรีเลชันได้ (เพราะตามปกติแล้วแอททริบิวต์ที่มีคุณสมบัติเป็นคีย์หลักจะสามารถระบุค่าของทุกแอททริบิวต์ในแต่ละทิวเพิลได้) เช่น

- รหัสนิสิต → รหัสสาขาวิชา, ชื่อสาขาวิชา
- รหัสสาขาวิชา → ชื่อสาขาวิชา

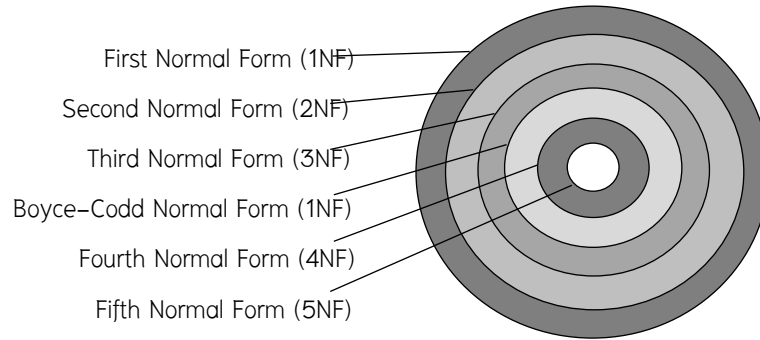
รีเลชัน : รหัสนิสิต เป็นคีย์หลักที่สามารถระบุ รหัสสาขาวิชาและ ชื่อสาขาวิชาของนิสิต แต่ในขณะเดียวกัน รหัสสาขาวิชา ซึ่งไม่ใช่คีย์หลักของรีเลชัน ก็สามารถระบุชื่อสาขาวิชานิสิตได้

3. รูปแบบบรรทัดฐานข้อมูล (Normal Forms)

จากระบบงานเดิมๆ ส่วนมากจะพบว่า การเก็บรวบรวมข้อมูลที่เกี่ยวข้องกับการทำงานจะมีการเก็บรายละเอียดของข้อมูลทุกอย่างไว้ด้วยกันหมด ทำให้มีกลุ่มข้อมูลที่ซ้ำซ้อนกัน และมีรูปแบบที่ซับซ้อน รีเลชันที่มีลักษณะแบบนี้เรียกกันว่า “รีเลชันยังไม่ได้ปรับบรรทัดฐานข้อมูล” ((Un-Normalized relation)

การปรับบรรทัดฐานข้อมูล จะเริ่มจากรายงานของผู้ใช้ข้อมูล (มีข้อมูลอะไรบ้าง ? ข้อมูลแต่ละตัวมีความสัมพันธ์กันอย่างไร ? ข้อจำกัดของข้อมูลมีอะไรบ้าง ? ฯลฯ) แล้วทำการวิเคราะห์ตามความต้องการของผู้ใช้ นำข้อมูลที่ผ่านการวิเคราะห์ มาแปลงเป็นตารางตามแนวคิดของ Data Model โดยทำการปรับปรุงโครงสร้างฐานข้อมูลให้มีความซ้ำซ้อนกันน้อยที่สุด

Codd เป็นคนแรกที่ได้พัฒนารูปแบบบรรทัดฐานข้อมูล โดยทำการจัดระบบข้อมูลแยกออกเป็นตาราง เก็บข้อมูลทุกอย่างให้อยู่ในตารางเดียวกัน การจัดระบบแบบนี้เป็นการดำเนินงานอย่างเป็นขั้นตอน เป็นลำดับตามปัญหาที่เกิดขึ้น การปรับบรรทัดฐานข้อมูลแบ่งออกได้เป็น 3 รูปแบบตามแนวคิดของ Codd แต่ละระดับของรูปแบบบรรทัดฐานข้อมูล จะมีวัตถุประสงค์ในการแก้ปัญหาของรีเลชันที่แตกต่างกันออกไป รีเลชันที่ผ่านกระบวนการปรับบรรทัดฐานในระดับที่สูงขึ้น ก็จะมีรูปแบบที่เป็นบรรทัดฐานมากขึ้น ปัญหาต่างๆ ที่เกิดขึ้นก็ลดน้อยลง



ภาพที่ 7.2 Domain / key Normal Form (DK/NF)

3.1 ชนิดของรูปแบบบรรทัดฐานข้อมูล

การปรับรูปแบบบรรทัดฐานข้อมูล เป็นเทคนิคในการออกแบบรีเลชันอาคัยพื้นฐานของ คีย์หลัก คีย์คู่แข่ง และฟังก์ชันการขึ้นต่อกันระหว่างแอททริบิวต์ หรือเป็นกระบวนการที่ใช้ในการ ปรับเค้าร่างของรีเลชัน (Relational schema) โดยการแตกรีเลชันออกเป็นรีเลชันย่อยๆ ให้อยู่ใน รูปแบบของบรรทัดฐาน ที่เหมาะสมเหมาะสำหรับการใช้งานนั่นเอง

ตามปกติแล้ว การปรับรูปแบบบรรทัดฐานข้อมูล จะเริ่มจากแบบจำลอง E-R ก่อน แล้วจึงพิจารณาการปรับรูปแบบบรรทัดฐานข้อมูลแต่ละระดับ โดยเริ่มจากรูปแบบบรรทัด ฐานข้อมูลระดับที่ 1 ซึ่งนับได้ว่าเป็นการปรับบรรทัดฐานข้อมูลระดับแรกสุด มีหลักการปรับ คือ ถ้าในรีเลชันมีกลุ่มของข้อมูลซ้ำกัน ให้แยกข้อมูลนั้นออกต่างหากให้เป็นแต่ละทูเพิล และ กำหนดคีย์หลักให้กับรีเลชันนั้น

1) รูปแบบบรรทัดฐานข้อมูลระดับที่ 1 (First Normal Form หรือ 1 NF)

(กำจัดกลุ่มของข้อมูลที่มีความซ้ำซ้อน หรือ repeating group)

เงื่อนไขของรูปแบบบรรทัดฐานข้อมูลระดับที่ 1

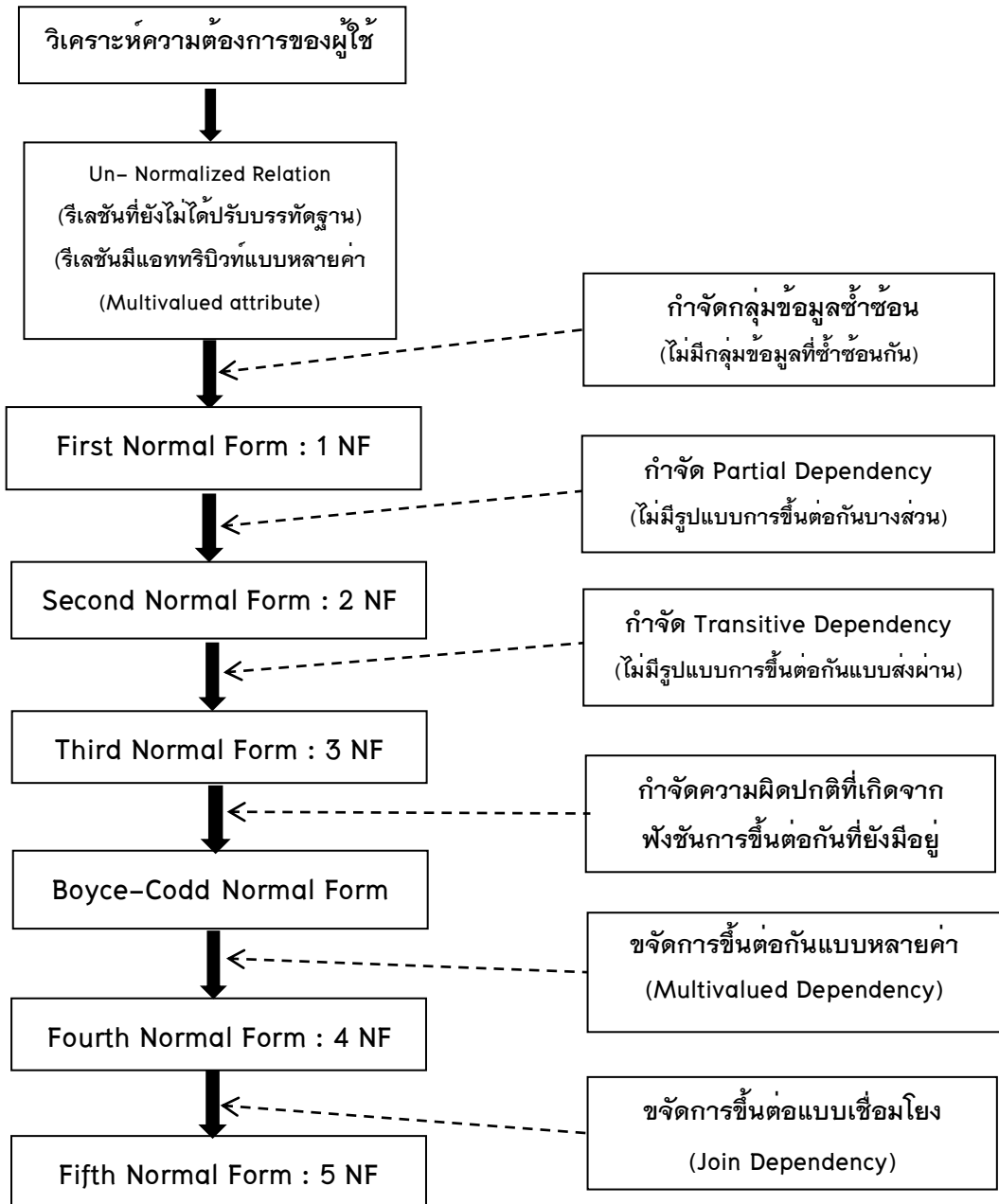
- ค่าของแอททริบิวต์ต่างๆ ในแต่ละทูเพิลจะมีค่าของข้อมูลเพียงค่าเดียว
- ไม่มีกลุ่มข้อมูลซ้ำอยู่ในรีเลชัน

หลักการปรับรูปแบบบรรทัดฐานข้อมูลระดับที่ 1

- แยกแอททริบิวต์ที่เป็นกลุ่มข้อมูลซ้ำไปสร้างเป็นรีเลชันใหม่อีกหนึ่งรีเลชัน
- นำคีย์ที่อยู่ในรีเลชันเดิมมารวมสร้างเป็นคีย์หลักของรีเลชันใหม่
- แยกแอททริบิวต์ที่ไม่มีกลุ่มข้อมูลซ้ำออกเป็นรีเลชันใหม่ และหาคีย์หลัก ให้กับรีเลชันใหม่

คุณสมบัติของรีเลชันรูปแบบบรรทัดฐานข้อมูลระดับที่ 1

- เป็นรีเลชันที่มีคีย์หลักของรีเลชัน และจะต้องไม่มีค่า “Null”
- ไม่มีกลุ่มข้อมูลซ้ำอยู่ในรีเลชัน
- ค่าของแอททริบิวต์ต่างๆ ในแต่ละทิวเพิลจะต้องมีค่าของข้อมูลเพียงค่าเดียว
- ทุก ๆ แอททริบิวต์จะต้องถูกระบุได้ด้วยคีย์หลัก



ภาพที่ 7.3 แสดงขั้นตอนการปรับบรรทัดฐานข้อมูล

ตัวอย่างที่ 1 : รีเลชันที่ยังไม่ได้ปรับบรรทัดฐานข้อมูล : รีเลชันสินค้า

รหัสสินค้า	สีสินค้า	ราคาสินค้า
101	สีแดง สีเขียว	15
102	สีเหลือง	24
103	สีเขียว	17
104	สีเหลือง สีน้ำเงิน	9

จากรีเลชันนี้จะเห็นว่าแอททริบิวต์สีสินค้า ยังมีหลายค่าคือแถวที่ 1 และ แถวที่ 4 มี 2 ค่า คือ “สีแดง สีเขียว” และ “สีเหลือง สีน้ำเงิน” ตามลำดับ การปรับบรรทัดฐานข้อมูลให้เป็นรูปแบบระดับที่ 1 กระทำได้โดยแยกรีเลชัน สินค้า ออกเป็น 2 รีเลชันย่อย คือ รีเลชัน สี และ รีเลชัน ราคา ดังภาพที่ 7.4

รีเลชัน สี		รีเลชัน ราคา	
รหัสสินค้า	สีสินค้า	รหัสสินค้า	ราคาสินค้า
101	สีแดง	101	15
101	สีเขียว	102	24
102	สีเหลือง	103	17
103	สีเขียว	104	9
104	สีน้ำเงิน		

ภาพที่ 7.4 รูปแบบบรรทัดฐานข้อมูลระดับที่ 1

รีเลชันทั้ง 2 อยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 1 แต่ละรีเลชันจะไม่มีกลุ่มข้อมูลซ้ำอยู่ในรีเลชัน และค่าของแอททริบิวต์ในแต่ละหมู่ฟิลด์จะมีค่าของข้อมูลเพียงค่าเดียวเท่านั้น

ตัวอย่างที่ 2 : รีเลชันที่ยังไม่ได้ปรับบรรทัดฐานข้อมูล : รีเลชันทักษะความรู้บุคลากร

ชื่อ	นามสกุล	ความรู้ด้าน IT
ปิติ	ใจดี	JAVA C++ PHP
อุรศรี	ปัญหา	JAVA PHP
ลีลา	ใจดี	JAVA C++

ปรับให้อยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 1



ชื่อ	นามสกุล	ความรู้ด้าน IT
ปิติ	ใจดี	JAVA
ปิติ	ใจดี	C++
ปิติ	ใจดี	PHP
อูรศรี	ปัญหา	JAVA
อูรศรี	ปัญหา	PHP
ลีลา	ใจดี	C++
ลีลา	ใจดี	JAVA

ภาพที่ 7.5 รูปแบบบรรทัดฐานข้อมูลระดับที่ 1

ข้อเสียของรูปแบบบรรทัดฐานข้อมูลระดับที่ 1

- สิ้นเปลืองเวลาในการปรับปรุงข้อมูล เพราะจะต้องทำซ้ำหลายแห่ง
- การลบข้อมูลออกจากรีเลชันอาจทำให้ข้อมูลบางส่วนสูญหายไปจากฐานข้อมูลได้

2) รูปแบบบรรทัดฐานข้อมูลระดับที่ 2 (Second Normal Form :หรือ 2 NF)

(กำจัดการขึ้นต่อกันบางส่วน หรือ partial dependency)

รูปแบบบรรทัดฐานข้อมูลระดับที่ 2 จะเกี่ยวข้องกับเรื่องของความสัมพันธ์ระหว่างคีย์หลักกับแอททริบิวต์อื่น ๆ ที่ไม่ได้เป็นส่วนหนึ่งของคีย์หลัก เป็นการตรวจสอบแก้ไขปัญหาคือความซ้ำซ้อนของข้อมูลที่ปรากฏในรีเลชันที่มีคุณสมบัติอยู่ในรูปแบบบรรทัดฐานของข้อมูลระดับที่ 1 โดยให้ความสำคัญเรื่องของคีย์หลักและความสัมพันธ์ระหว่างแอททริบิวต์โดยเฉพาะอย่างยิ่งความสัมพันธ์ระหว่างแอททริบิวต์แบบทั้งหมด

เงื่อนไขของรูปแบบบรรทัดฐานข้อมูลระดับที่ 2

- รีเลชันต้องอยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 1 มาก่อน
- แอททริบิวต์ทุกตัวที่ไม่ใช่คีย์หลัก จะต้องมีความสัมพันธ์กับแอททริบิวต์ที่เป็นคีย์หลักแบบทั้งหมด
- ไม่มีแอททริบิวต์ที่มีการขึ้นต่อกันบางส่วนของคีย์หลัก

หลักการปรับรูปแบบบรรทัดฐานข้อมูลระดับที่ 2

- ตัดแอททริบิวต์ที่มีการขึ้นต่อกันแบบบางส่วนของรีเลชันออกไปไว้ในรีเลชันใหม่ ให้คงแอททริบิวต์ที่ขึ้นกับทุกส่วนของคีย์หลักไว้ในรีเลชันเดิม
- สร้างรีเลชันใหม่ โดยดึงแอททริบิวต์ที่ขึ้นกับบางส่วนของคีย์หลัก และกำหนดคีย์หลักของรีเลชันจากแอททริบิวต์ที่เป็นส่วนประกอบของรีเลชันที่แอททริบิวต์เหล่านี้มีรูปแบบการขึ้นต่อกัน

ตัวอย่าง : รีเลชันที่ยังไม่ได้ปรับบรรทัดฐานข้อมูล

รหัสอาจารย์	วิชาที่สอน	ตำแหน่งวิชาการ	คุณวุฒิการศึกษา	อายุ
111	Botany	Assist.Prof	M.Sc	35
111	Physiology	Assist.Prof	M.Sc	35
222	English	Assoc.Prof	Ph.D	40
333	Mathematics	Prof	Ph.D	43
333	Physics	Prof	Ph.D	43

คีย์คู่แข่ง : {รหัสอาจารย์, วิชาที่สอน}

Non prime attribute (Non-key attributes) : อาจารย์_ตำแหน่งวิชาการ ,
อาจารย์_คุณวุฒิการศึกษา, อาจารย์_อายุ

รีเลชันข้างบนอยู่ในรูปแบบบรรทัดฐานระดับที่ 1 เนื่องจากทุกรีเลชัน เป็นรีเลชันที่มีคีย์หลัก และเป็นรีเลชันที่ไม่มีกลุ่มข้อมูลซ้ำ แต่ยังไม่อยู่ในรูปแบบบรรทัดฐานระดับที่ 2 เพราะว่ามี Non prime attribute หรือ แอททริบิวต์ที่ไม่เป็นส่วนหนึ่งของคีย์หลักคือ อาจารย์_ตำแหน่งวิชาการ, อาจารย์_คุณวุฒิการศึกษา, อาจารย์_อายุ ยังขึ้นอยู่กับ รหัสอาจารย์ ซึ่งเป็นคีย์หลักของรีเลชัน

ดังนั้นในการปรับให้เป็นรูปแบบบรรทัดฐานข้อมูลระดับที่ 2 จะต้องแยกออกเป็น 2 ตารางโดยแยกแอททริบิวต์ที่มีการขึ้นต่อกันแบบบางส่วนของรีเลชันออกไปไว้ใน รีเลชันใหม่ และให้คงแอททริบิวต์ที่ขึ้นกับทุกส่วนของคีย์หลักไว้ในรีเลชันเดิม

อาจารย์_วิชาที่สอน		อาจารย์_อายุ	
รหัสอาจารย์	วิชาที่สอน	รหัสอาจารย์	อายุ
111	Botany	111	35
111	Physiology	222	40
222	English	333	43
333	Mathematics		
333	Physics		

ภาพที่ 7.7 รีเลชันอยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 2

3) รูปแบบบรรทัดฐานข้อมูลระดับที่ 3 (Third Normal Form หรือ 3NF)

(กำจัดการขึ้นต่อกันของแอททริบิวต์ที่ไม่ใช่คีย์หลัก หรือ transitive dependency)

รูปแบบบรรทัดฐานข้อมูลระดับที่ 3 เป็นกระบวนการที่พยายามขจัดสภาพของรูปแบบการขึ้นต่อกันแบบส่งผ่าน หรือการที่ไม่มีแอททริบิวต์ที่ไม่ใช่คีย์หลักสามารถระบุค่าแอททริบิวต์ที่ไม่ใช่คีย์หลักอื่นได้

เงื่อนไขของรูปแบบบรรทัดฐานข้อมูลระดับที่ 3

- รีเลชันต้องอยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 2 มาก่อน
- ทุกแอททริบิวต์จะขึ้นอยู่กับคีย์หลักเท่านั้น
- รีเลชันต้องไม่มีความสัมพันธ์ระหว่างแอททริบิวต์แบบส่งผ่าน

หลักการปรับปรุงแบบบรรทัดฐานข้อมูลระดับที่ 3

- แยกแอททริบิวต์ที่มีความสัมพันธ์แบบส่งผ่าน (แอททริบิวต์ที่มีปัญหา) ออกไปสร้างเป็นรีเลชันใหม่
- กำหนดคีย์หลักโดยเลือกเอาแอททริบิวต์ที่สามารถกำหนดค่าของแอททริบิวต์อื่น ๆ ได้
- ในรีเลชันเดิมให้คงแอททริบิวต์ที่สามารถเลือกค่าแอททริบิวต์ที่ไม่ใช่คีย์หลักไว้เป็นคีย์นอก เพื่อใช้ในการเชื่อมโยงกับรีเลชันที่สร้างขึ้นใหม่

จากรีเลชัน การทำงานของพนักงาน (ข้างล่าง) จะพบว่ามีการขึ้นต่อกันแบบส่งผ่าน คือ แอททริบิวต์ชื่อพนักงาน และแอททริบิวต์ รหัสตำแหน่ง จะขึ้นอยู่กับคีย์รหัสพนักงาน ในขณะที่แอททริบิวต์เงินประจำตำแหน่งของพนักงาน จะขึ้นอยู่กับแอททริบิวต์รหัสตำแหน่งซึ่งไม่ใช่คีย์หลักอีกต่อหนึ่งทำให้มีการขึ้นต่อกันแบบส่งผ่านเกิดขึ้นในรีเลชันนี้

เมื่อต้องการปรับปรุงบรรทัดฐานข้อมูล ให้อยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 3 กระทำได้โดยการแตกออกเป็นรีเลชันย่อยที่ไม่มีการขึ้นต่อกันแบบส่งผ่าน (แยกแอททริบิวต์ที่ไม่ใช่คีย์หลัก หรือส่วนใดส่วนหนึ่งของคีย์หลัก ที่ก่อให้เกิดปัญหาออกมา) จะได้รีเลชันสองรีเลชันคือ คือรีเลชันพนักงาน และรีเลชันรหัสตำแหน่ง ซึ่งอยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 3 ดังภาพ

ตัวอย่าง : รีเลชันที่ยังไม่ได้ปรับบรรทัดฐานข้อมูล

รีเลชัน “การทำงานของพนักงาน” ประกอบด้วย รหัสพนักงาน, ชื่อ-สกุล, รหัสตำแหน่ง, เงินประจำตำแหน่ง

รหัสพนักงาน	ชื่อ-สกุล	รหัสตำแหน่ง	เงินประจำตำแหน่ง
1001	มานะ ใจงาม	105	5,000
1032	นารี สุขใจ	101	2,000
1022	เมตตา กรุณา	103	3,000
2010	ดีจริง ขยัน	104	3,000

ภาพที่ 7.8 รีเลชันที่มีรูปแบบการขึ้นต่อกันแบบส่งผ่าน (Transitive Dependency)

รีเลชันพนักงาน

<u>รหัสพนักงาน</u>	ชื่อ-สกุล
1001	มานะ ใจงาม
1032	นารี สุขใจ
1022	เมตตา กรุณา
2010	ดีจริง ขยัน

รีเลชันตำแหน่งงาน

<u>รหัสตำแหน่ง</u>	เงินประจำตำแหน่ง
105	5,000
101	2,000
103	3,000
104	3,000

ภาพที่ 7.9 รีเลชันที่อยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 3

ตัวอย่าง : รีเลชันที่ยังไม่ได้ปรับบรรทัดฐานข้อมูล

รีเลชัน “ประวัติพนักงาน” ประกอบด้วย รหัสพนักงาน ชื่อ-สกุล ที่อยู่ ฯลฯ

Emp_id	Emp_name	Emp_zip	Emp_province	Emp_city	Emp_district
1001	สมศักดิ์ ใจดี	56000	พะเยา	จุน	ดอกคำใต้
1002	นารี สดสวย	65000	พิษณุโลก	เมือง	ในเมือง
1006	มีนา เมตตา	46000	เชียงราย	แม่สาย	แม่จัน
1101	มานะ ชัยน	34000	กรุงเทพ	ลาดพร้าว	จตุจักร
1201	สุภาพ ดียิ่ง	12000	อุดรธานี	เมือง	ประจักษ์

ซูเปอร์คีย์ (Super keys) หรือกลุ่มของแอททริบิวต์ที่สามารถระบุความเป็นเอกลักษณ์ของแต่ละแถวเพื่อให้แตกต่างกันในตารางได้ เช่น : {Emp_id}, {Emp_id, Emp_name}, {Emp_id, Emp_name, Emp_zip}..... ฯลฯ

คีย์คู่แข่ง (Candidate Keys) หรือซูเปอร์คีย์ที่มีขนาดเล็กที่สุด ที่ไม่มีแอททริบิวต์อื่นเป็นเซตย่อย มารวมกันเพื่อให้เกิดเป็นเอกลักษณ์ในตารางนั้น ๆ : {Emp_id}

Non-key attributes หรือ แอททริบิวต์ที่ไม่ได้เป็นส่วนหนึ่งของคีย์คู่แข่งใด ๆ เลย คือทุกแอททริบิวต์ยกเว้นแอททริบิวต์ Emp_id ที่เป็น key attribute

ปรับรูปบรรทัดฐานข้อมูลให้อยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 3 จะได้รีเลชันสองรีเลชันคือ คือรีเลชันพนักงาน และรีเลชันที่อยู่พนักงาน

รีเลชันพนักงาน			รีเลชันที่อยู่พนักงาน			
Emp_id	Emp_name	Emp_zip	Emp_zip	Emp_province	Emp_city	Emp_district
1001	สมศักดิ์ ใจดี	56000	56000	พะเยา	จุน	ห้วยข้าวก่ำ
1002	นารี สดสวย	65000	65000	พิษณุโลก	เมือง	ในเมือง
1006	มีนา เมตตา	46000	46000	เชียงราย	แม่สาย	แม่จัน
1101	มานะ ชัยน	34000	34000	กรุงเทพ	ลาดพร้าว	จตุจักร
1201	สุภาพ ดียิ่ง	12000	12000	อุดรธานี	เมือง	ประจักษ์

ภาพที่ 7.10 รีเลชันที่อยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 3

4) รูปแบบบรรทัดฐานข้อมูลของบอยส์และคอดด์ (Boyce–Codd Normal Form)

ตามปกติแล้วในทางปฏิบัติ การปรับรูปแบบบรรทัดฐานข้อมูลจนรีเลชันมีคุณสมบัติอยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 3 ก็จะสามารถที่จะขจัดปัญหาความซ้ำซ้อนของข้อมูลได้จนเกือบหมด แต่ก็อาจจะมีปัญหาความซ้ำซ้อนของข้อมูลได้อีก แม้จะพบได้ค่อนข้างน้อยก็ตาม เพื่อให้รีเลชันมีความถูกต้อง และรัดกุมมากยิ่งขึ้น อาจจำเป็นต้องทำให้รีเลชันให้อยู่ในรูปแบบบรรทัดฐานข้อมูลที่ขยายขอบเขตมาจากรูปแบบบรรทัดฐานระดับที่ 3 เราสามารถกำหนดนิยามของรีเลชันที่อยู่ในรูปแบบบรรทัดฐานข้อมูลของบอยส์และคอดด์ก็ต่อเมื่อรีเลชันมีคุณสมบัติตามเงื่อนไขดังต่อไปนี้

เงื่อนไขของรูปแบบบรรทัดฐานข้อมูลของบอยส์ – คอดด์

- รีเลชันต้องอยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 3 อยู่แล้ว
- ทุกแอททริบิวต์ที่เป็น “Determinant” ในรีเลชันนั้นต้องเป็นคีย์หลัก
- ไม่มีแอททริบิวต์ใดในรีเลชัน ที่สามารถระบุค่าของแอททริบิวต์ที่เป็นคีย์หลัก หรือส่วนใดส่วนหนึ่งของแอททริบิวต์ที่ประกอบเป็นคีย์หลักได้

หลักการปรับรูปแบบบรรทัดฐานข้อมูลของบอยส์ – คอดด์

- คัดลอกแอททริบิวต์ที่เป็น “Determinant” ซึ่งไม่ใช่คีย์คู่แข่งออกมาเป็นรีเลชันใหม่ และให้เป็นคีย์หลักของรีเลชันใหม่
- ดึงแอททริบิวต์ที่ขึ้นกับแอททริบิวต์ที่เป็นตัวกำหนดค่านั้นออกมาอยู่ในรีเลชันใหม่

โดยทั่วไปการปรับรูปแบบบรรทัดฐานข้อมูลแบบบอยส์-คอดด์ รีเลชันจะต้องอยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 3 อยู่แล้ว แต่รูปแบบระดับที่ 3 ไม่จำเป็นต้องเป็นรูปแบบบอยส์-คอดด์เสมอไป เพราะรูปแบบนี้เป็นรูปแบบที่ขยายเพิ่มเติมจากรูปแบบ ระดับที่ 3 ให้มีความเหมาะสมยิ่งขึ้น เช่น รีเลชันลงทะเบียน (ข้างล่าง) อยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 3 แล้ว แต่แอททริบิวต์รหัสวิชา และผลการเรียนจะขึ้นอยู่กับคีย์นิสิต และคีย์ผู้สอน ในขณะเดียวกันคีย์รหัสผู้สอนก็ขึ้นอยู่กักรหัสวิชา ถ้าหากต้องการเปลี่ยนแปลงผู้สอนในวิชา 101 จะต้องมีการเปลี่ยนแปลงถึง 2 ทูเปิล คือทูเปิลที่ 2 และที่ 5 ซึ่งผลลัพธ์ที่ได้อาจจะทำให้เกิดความผิดพลาดถ้าทำการแก้ไขไม่ครบถ้วน และถ้านิสิตรหัส 1022 ถอนการลงทะเบียนวิชา 103 หากเราลบข้อมูลนี้ ข้อมูลของผู้ที่สอนวิชานี้ (241) จะหายไปจากระบบ

ลงทะเบียน

รหัสนิสิต	รหัสผู้สอน	รหัสวิชา	ผลการเรียน
1001	331	105	C
1001	241	101	A
1022	241	103	C
2010	555	104	B
2010	241	101	A

ภาพที่ 7.11 รีเลชันที่อยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 3
(แต่ไม่อยู่รูปแบบบรรทัดฐานข้อมูลของบอยล์-คอตต์)

ผู้สอน

ผลการเรียน

รหัสวิชา	รหัสผู้สอน
105	331
103	241
104	555
101	241

รหัสนิสิต	รหัสผู้สอน	ผลการเรียน
1001	331	C
1001	241	A
1022	241	C
2010	555	B
2010	241	A

ภาพที่ 7.12 รีเลชันที่ปรับบรรทัดฐานให้อยู่ในรูปแบบบรรทัดฐานข้อมูลบอยล์-คอตต์

เราสามารถทำการแตกตารางออกมาให้อยู่ในรูปของบอยล์คอตต์นอร์มัลฟอร์มได้ โดยการแยกแอตทริบิวต์รหัสวิชา และรหัสผู้สอนซึ่งขึ้นอยู่กับแอตทริบิวต์รหัสวิชาออกมาเป็นอีกหนึ่งรีเลชัน และแยกแอตทริบิวต์ รหัสนิสิต รหัสผู้สอน และผลการเรียนออกมาเป็นอีกหนึ่งรีเลชัน

5) รูปแบบบรรทัดฐานข้อมูลระดับที่ 4 (Forth Normal Form : 4 NF)

ถ้าความสัมพันธ์ระหว่างแอตทริบิวต์ในรีเลชันเป็นความสัมพันธ์แบบเชิงกลุ่มหรือมีการขึ้นต่อกันแบบกลุ่ม ซึ่งเป็นคุณสมบัติของการนำรีเลชันย่อยที่เกิดจากการแยกรีเลชันมารวมกันแล้วได้ข้อมูลเช่นเดียวกับรีเลชันเดิม จะทำให้เกิดปัญหาความซ้ำซ้อนของข้อมูลได้เช่นเดียวกัน และปัญหานี้ไม่สามารถแก้ไขได้โดยการปรับรูปแบบบรรทัดฐานระดับที่ 3 หรือแบบ บอยล์-คอตต์ เพื่อแก้ปัญหาดังกล่าว จึงได้มีการพัฒนารูปแบบบรรทัดฐานข้อมูลระดับที่ 4 ขึ้นมา

เงื่อนไขของรูปแบบบรรทัดฐานข้อมูลระดับที่ 4

- รีเลชันต้องอยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 3 หรือรูปแบบบรรทัดฐานข้อมูล บอยส์-คอตต์ มาก่อน
- เป็นรีเลชันที่ไม่มีความสัมพันธ์ในการระบุค่าของแอททริบิวต์แบบหลายค่า (multivalued dependency) หรือรีเลชันจะต้องไม่มีการขึ้นต่อกันแบบกลุ่ม

หลักการปรับรูปแบบบรรทัดฐานข้อมูลระดับที่ 4

- แยกรีเลชันออกเป็นสองรีเลชัน แต่ละรีเลชันเก็บข้อมูลที่ขึ้นต่อกัน

ตัวอย่าง : รีเลชัน นิสิต-ลงทะเบียน-ชมรม (ภาพที่ 7.13) ซึ่งอยู่ในรูปแบบบรรทัดฐานข้อมูลระดับบอยส์-คอตต์ แล้ว แต่จะพบว่าลักษณะของข้อมูลในรีเลชันนี้มีการขึ้นต่อกันแบบเชิงกลุ่ม ทำให้รีเลชันนี้ยังคงมีปัญหาเกี่ยวกับการจัดการข้อมูล เช่น

นิสิต-ลงทะเบียน-ชมรม : (รหัสนิสิต, วิชาลงทะเบียน, ชมรม)

รหัสนิสิต	วิชาลงทะเบียน	ชมรมที่สนใจ
59-10001	Mathematics	Badminton
59-10001	Physics	Badminton
59-10001	Mathematics	Tennis
59-10001	Physics	Tennis
59-20011	Operating System	Badminton
59-20011	Operating System	Swimming
59-30111	English	Volleyball

ภาพที่ 7.13 รีเลชันอยู่ในระดับบอยส์-คอตต์ แต่ยังพบว่าข้อมูลมีการขึ้นต่อกันแบบเชิงกลุ่ม (ตารางที่ยังไม่ได้ปรับบรรทัดฐานข้อมูลเป็นรูปแบบบรรทัดฐานข้อมูลระดับที่ 4)

- การเพิ่มข้อมูล : สมมติว่าถ้านิสิตรหัส “59-10001” มีชมรมที่สนใจเพิ่มขึ้นมาอีก คือ “Volleyball” ในการเก็บข้อมูลจำเป็นต้องมีการเพิ่มแถวเข้าไปอีก 2 แถว ตามจำนวนวิชาที่นิสิตคนนี้ลงทะเบียน ดังภาพที่ 7.14

นิสิต-ลงทะเบียน-ชมรม : (รหัสนิสิต,วิชาลงทะเบียน,ชมรม)

รหัสนิสิต	วิชาลงทะเบียน	ชมรมที่สนใจ
59-10001	Mathematics	Badminton
59-10001	Physics	Badminton
59-10001	Mathematics	Tennis
59-10001	Physics	Tennis
59-10001	Mathematics	Volleyball
59-10001	Physics	Volleyball
59-20011	Operating System	Badminton
59-20011	Operating System	Swimming
59-30111	English	Volleyball

ภาพที่ 7.14 การเพิ่มชมรม “Valleyball” ของนิสิตรหัส 59-10001 เข้าไปในรีเลชัน

● **การลบข้อมูล** : ปัญหานี้เกิดจากลักษณะของข้อมูลในรีเลชันนี้มีการขึ้นต่อกันแบบเชิงกลุ่ม เช่นนิสิตรหัส 59-10001 ลงทะเบียนเรียน 2 วิชา คือ Mathematics และ Physics ดังภาพที่ 7.14 แต่ชมรมที่นิสิตคนนี้นั้นสนใจ มี 2 ชมรม คือชมรม Badminton และ Tennis ทำให้ข้อมูลไม่สอดคล้องกับข้อมูลในภาพที่ 7.15

● **การเปลี่ยนแปลงการเก็บข้อมูล** เช่น นิสิต รหัส “59-10001” มีการเก็บข้อมูลนิสิตใหม่เพียงแค่ 2 แถว ดังภาพที่ 7.15 จะทำให้เข้าใจได้ว่า นิสิตคนนี้จะสนใจชมรม “Tennis” ก็ต่อเมื่อเขาลงทะเบียนวิชา “Mathematics” และวิชา “Physics” เท่านั้น ซึ่งจะเห็นว่าเป็นรูปแบบการเก็บข้อมูลที่ไม่ถูก ต้อง เพราะทำให้เกิดการตีความหมายของข้อมูลผิดไปได้เนื่องจากวิชาที่ลงทะเบียนกับชมรมที่นิสิตสนใจนั้นเป็นอิสระต่อกันหรือไม่ขึ้นต่อกัน

รหัสนิสิต	วิชาลงทะเบียน	ชมรมที่สนใจ
59-10001	Mathematics	Tennis
59-10001	Physics	Tennis
59-20011	Operating System	Badminton
59-20011	Operating System	Swimming
59-30111	English	Volleyball

ภาพที่ 7.15 แสดงการเก็บข้อมูลนิสิตที่ทำให้เกิดการเข้าใจผิดได้

รีเลชันในภาพที่ 7.15 จะเป็นรีเลชันที่มีรูปแบบการขึ้นต่อกันเชิงกลุ่ม โดยมีแอททริบิวต์ รหัสนิสิต ที่ใช้ในการเลือกกลุ่มข้อมูลของ วิชาที่ลงทะเบียน เขียนแทนด้วยสัญลักษณ์ :

รหัสนิสิต →→ วิชาที่ลงทะเบียน

นอกจากนี้ แอททริบิวต์ รหัสนิสิต ก็ยังใช้ในการเลือกกลุ่มข้อมูลของ ชมรม ซึ่งเขียนแทนด้วยสัญลักษณ์ :

รหัสนิสิต →→ ชมรมที่สนใจ

แต่ วิชาที่ลงทะเบียน และ ชมรม จะมีความเป็นอิสระต่อกัน

จากนิยามของรูปแบบบรรทัดฐานข้อมูลระดับที่ 4 ที่ว่า “ รีเลชันใดจะอยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 4 รีเลชันนั้นต้องอยู่ในรูปแบบบรรทัดฐานข้อมูลแบบ บอยล์-คอดด์ มาก่อน และข้อมูลจะต้องไม่มีการขึ้นต่อกันเชิงกลุ่มภายในรีเลชันนั้น ”

เพื่อขจัดปัญหาที่เกิดขึ้น จำเป็นจะต้องจัดการขึ้นต่อกันเชิงกลุ่มนี้ ออกไปจากรีเลชัน เพื่อให้รีเลชันนั้นอยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 4 โดยทำการแยกแอททริบิวต์ที่มีการขึ้นต่อกันเชิงกลุ่ม ออกเป็นรีเลชันใหม่และกำหนดให้แอททริบิวต์ทุกตัวในรีเลชันทำหน้าที่เป็นคีย์หลักของรีเลชัน นั่นคือรีเลชัน “นิสิต-ลงทะเบียน-ชมรม” ในภาพที่ 7.15 จะถูกแยกออก เป็นสองรีเลชันคือ รีเลชัน “นิสิต-ลงทะเบียน” และรีเลชัน “นิสิต-ชมรม” โดยแต่ละรีเลชันจะเก็บข้อมูลที่มีการขึ้นต่อกัน ดังภาพที่ 7.16 ซึ่งเป็นรีเลชันที่อยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 4 เรียบร้อยแล้ว

นิสิต-ลงทะเบียน

คีย์หลัก: รหัสนิสิต, วิชาลงทะเบียน

รหัสนิสิต	วิชาลงทะเบียน
59-10001	Mathematics
59-10001	Physics
59-20011	Operating System
59-30111	English

นิสิต-ชมรม

คีย์หลัก: รหัสนิสิต, ชมรม

รหัสนิสิต	ชมรมที่สนใจ
59-10001	Tennis
59-10001	Tennis
59-10001	Volleyball
59-20011	Badminton
59-20011	Swimming
59-30111	Volleyball

ภาพที่ 7.16 : รีเลชันที่อยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 4 โดยจัดการขึ้นต่อกันเชิงกลุ่ม

6) รูปแบบบรรทัดฐานข้อมูลระดับที่ 5 (Fifth Normal Form : 5 NF)

รูปแบบบรรทัดฐานข้อมูลระดับที่ 5 หรือ Project-Join Normal Form (PJ/NF) เป็นรูปแบบบรรทัดฐานข้อมูลที่ค่อนข้างจะเกิดขึ้นยาก

เงื่อนไขของรูปแบบบรรทัดฐานข้อมูลระดับที่ 5

- รีเลชันจะต้องมีคุณสมบัติรูปแบบบรรทัดฐานข้อมูลระดับที่ 4 มาก่อน
- รีเลชันจะต้องมีคุณสมบัติJoin Dependency (Join Dependency)”
(เมื่อแตกรีเลชันออกเป็นรีเลชันย่อย ๆ แล้วสามารถนำรีเลชันย่อย ๆ นั้นมารวมกัน (join) แล้วได้ค่าหรือข้อมูลเช่นเดียวกับรีเลชันเดิม)

หลักการแปลงเป็นรูปแบบบรรทัดฐานข้อมูลระดับที่ 5

- แยกรีเลชันเดิมออกเป็นรีเลชันย่อย ๆ
- กำหนดคีย์หลักให้กับแต่ละรีเลชันย่อย ๆ นั้น

ตัวอย่าง : รีเลชันที่ยังไม่ได้ปรับรูปแบบบรรทัดฐานข้อมูล : รีเลชันการขาย

ชื่อลูกค้า	ชื่อสินค้า	ชื่อตัวแทนขาย
A-Minimart	Coke	นารี
A-Minimart	Pepsi	นารี
B-Minimart	Coke	วิชา
B-Minimart	Slippers	วิชา
Books Store	Pen	สุดา

ภาพที่ 7.18 รีเลชันก่อนผ่านกระบวนการปรับรูปแบบบรรทัดฐานข้อมูลในระดับที่ 4

ปัญหาจากการเพิ่ม การแก้ไข และการลบข้อมูล ในตารางข้างบน เช่น

- ถ้ามีการแก้ไขชื่อสินค้า จะต้องแก้หลายรายการ และหลายแถว

- ถ้ามีการลบข้อมูลชื่อสินค้าออกไปจากรีเลชัน จะทำให้ข้อมูลชื่อลูกค้าที่อยู่ในแถวเดียวกันหายไปด้วย

ในการแก้ปัญหาดังกล่าว จะต้องทำให้รีเลชันให้อยู่ในรูปแบบบรรทัดฐานข้อมูลในระดับที่ 4 ซึ่งกระทำได้โดยแตกรีเลชันให้เป็นรีเลชันย่อย ๆ 3 รีเลชัน ดังนี้

รีเลชันย่อยที่ 1		รีเลชันย่อยที่ 2		รีเลชันย่อยที่ 3	
ชื่อลูกค้า	ตัวแทนขาย	ชื่อลูกค้า	ชื่อสินค้า	ตัวแทนขาย	ชื่อสินค้า
A-Minimart	นารี	A-Minimart	Coke	นารี	Coke
B-Minimart	วิชา	A-Minimart	Pepsi	นารี	Pepsi
Books Store	สุดา	B-Minimart	Coke	วิชา	Coke
		B-Minimart	Slippers	วิชา	Slippers
		Books Store	Pens	สุดา	Pens

ภาพที่ 7.19 รีเลชันที่อยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 4

กระบวนการปรับรีเลชันให้อยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 5 กระทำได้ดังนี้
ขั้นตอนที่ 1 นำรีเลชันย่อยที่ 2 และรีเลชันย่อยที่ 3 มา join รีเลชันกัน ได้ผลลัพธ์ดังนี้

ชื่อลูกค้า	ชื่อสินค้า	ชื่อตัวแทนขาย
A-Minimart	Coke	นารี
A-Minimart	Pepsi	นารี
A-Minimart	Coke	วิชา
B-Minimart	Coke	วิชา
B-Minimart	Coke	นารี
B-Minimart	Slippers	วิชา
Books Store	Pen	สุดา

ภาพที่ 7.20 ผลลัพธ์จากการ join รีเลชันที่ 2 และ 3

ขั้นตอนที่ 2 นำรีเลชันที่เป็นผลลัพธ์จากขั้นตอนที่ 1 และรีเลชันย่อยที่ 1 มา join รีเลชันกัน ได้ผลลัพธ์ดังนี้

ชื่อลูกค้า	ชื่อสินค้า	ชื่อตัวแทนขาย
A-Minimart	Coke	นารี

A-Minimart	Pepsi	নারী
B-Minimart	Coke	বিশা
B-Minimart	Slippers	বিশা
Books Store	Pen	সুতা

ภาพที่ 7.21 ผลลัพธ์จากการ join รีเลชันจากขั้นตอนที่ 1 กับรีเลชันย่อยที่ 1

ตัวอย่างรีเลชันวิชาประจำภาคเรียน อยู่ในรูปบรรทัดฐานข้อมูลระดับที่ 4 แล้ว เพราะแอททริบิวต์ภาคเรียนเป็นตัวกำหนดแอททริบิวต์รายวิชาหลายค่า และแอททริบิวต์รายวิชาที่เป็นตัวกำหนดแอททริบิวต์ชั้นเรียนหลายค่าเช่นกัน เมื่อทำการทดสอบคุณสมบัติของรูปแบบบรรทัดฐานข้อมูลระดับที่ 5 ของรีเลชันวิชาประจำภาคเรียน โดยแตกรีเลชันดังกล่าวออกเป็นรีเลชันย่อย ๆ 3 รีเลชัน คือ รีเลชัน ภาคเรียน รีเลชันวิชา_ชั้นเรียน และ รีเลชันภาคเรียน_ชั้นเรียน เมื่อทำการรวมทั้งสามรีเลชันกลับไปเป็นหนึ่งรีเลชันอีกครั้งหนึ่งจะได้จำนวนข้อมูลเท่ากับกับรีเลชันเดิมก่อนที่จะมีแตกเป็นรีเลชันย่อยทุกประการ นั่นคือรีเลชันดังกล่าวมีคุณสมบัติ *join ดีเฟนเดนซี* และอยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 5 แล้ว

ตัวอย่าง : รีเลชันที่ยังไม่ได้ปรับรูปแบบบรรทัดฐานข้อมูล

วิชาประจำภาคเรียน

ภาคเรียน	วิชา	ชั้นเรียน
1/2558	คณิตศาสตร์	A002
1/2558	ฟิสิกส์	A001
2/2558	คณิตศาสตร์	A001
1/2558	คณิตศาสตร์	A001



ภาพที่ 7.22 การปรับรีเลชันให้อยู่ในรูปแบบบรรทัดฐานข้อมูลระดับที่ 5

จะเห็นได้ว่า ยิ่งมีการปรับบรรทัดฐานข้อมูลในระดับที่สูงขึ้น ก็ยิ่งจะมีการแบ่งแยก ตารางย่อย ๆ มากขึ้นตามไปด้วย ทำให้เสียเวลาในการค้นหาข้อมูล เพราะต้องทำการเชื่อมโยง ข้อมูลในหลาย ๆ ตารางเข้าด้วยกัน ดังนั้นการปรับตารางหรือรีเลชันด้วยวิธีการปรับบรรทัด ฐานควรเลือกปรับในระดับที่พอดีและเหมาะสมกับการใช้งาน ในแต่ละงาน
