บทที่ 3

ทฤษฎี

ระบบสารสนเทศเพื่อการจัดการ (MIS)

ระบบสารสนเทศเพื่อการจัดการนี้หมายถึงระบบสารสนเทศซึ่งช่วยในการประมวลผลรายการเปลี่ยนแปลงและช่วยในการตัดสินใจของผู้บริหาร ตามประวัติได้มีการใช้และวิจัยเกี่ยวกับระบบสารสนเทศเพื่อการจัดการ มาตั้งแต่ปี พ.ศ.2513 ในบางครั้งอาจมีการเรียกชื่อที่แตกต่างกันออกไปบ้างเช่น information services, information systems ฯลฯ แต่ถึงอย่างไรก็หมายถึงระบบที่รวมสิ่งต่างๆเข้าด้วยกัน (integrated system),ระบบผู้ใช้เครื่อง (user-machine system) ,ระบบที่ใช้คอมพิวเตอร์เป็นพื้นฐาน(computer-based system) และมีระบบช่วยวางแผนและตัดสินใจ(support,planning and decision)

1. ระบบที่รวมสิ่งต่างๆเข้าด้วยกัน ก็คือระบบที่จะรวมข้อมูลและแฟ้มข้อมูลต่างๆเข้าด้วยกัน นำมาใช้ร่วมกันโดยผู้ใช้หรือโปรแกรมประยุกต์มากกว่า 1 ขึ้นไป นอกจากนั้นข้อมูลของผู้ใช้หลายคนสามารถรวมกันและปรากฏอยู่ในโปรแกรมอันเดียวกันได้ การรวมข้อมูลดังกล่าวจะช่วยแก้ปัญหาความไม่ตรงกัน ข้อผิดพลาดและความซ้ำซ้อนจากการเก็บข้อมูลแบบต่างคนต่างเก็บได้ และสิ่งที่ควรจะต้องคำนึงถึงสำหรับระบบที่รวมสิ่งต่างๆเข้าด้วยกัน ก็คือ ความปลอดภัยของข้อมูล จึงต้องมีการกำหนดสิทธิในการเข้าถึงข้อมูลให้ผู้ใช้แต่ละคนด้วย

2. ระบบผู้ใช้เครื่อง โดยทั่วไปผู้ใช้ระบบระบบสารสนเทศเพื่อการจัดการ ก็คือผู้ที่จะใส่ข้อมูลเข้าเครื่อง(ป้อนข้อมูลให้ระบบ) และใช้ประโยชน์จากเอาต์พุตที่ได้จากระบบผ่านทาง คีย์บอร์ด,หน้าจอ,เมนูฯ นั่นเอง ดังนั้นการติดต่อระหว่างผู้ใช้กับเครื่องจะเป็นจุดที่ผู้ออกแบบระบบสารสนเทศเพื่อการจัดการ จะต้องคำนึงถึงอย่างมาก โดยจะต้องพิจารณาทั้งลักษณะและความสามารถของผู้ใช้ กล่าวกันว่ามีหลายสิ่งที่ช่วยทำให้ผู้ใช้ยอมรับระบบด้วยความสะดวกใจ เช่น การใช้ภาษาหรือกระบวนการที่ง่ายต่อการศึกษา, ทั้งฮาร์ดแวร์และซอฟต์แวร์ให้ผลตอบแทนที่ทันต่อการตัดสินใจ ฉะนั้นผู้ออกแบบระบบสารสนเทศเพื่อการจัดการ ต้องมีความรู้ทั้งเทคโนโลยีทางคอมพิวเตอร์และรู้ถึงความต้องการในระบบธุรกิจที่เราออกแบบ

3. ระบบที่ใช้คอมพิวเตอร์เป็นพื้นฐาน ระบบสารสนเทศเพื่อการจัดการจะดีหรือไม่นั้นขึ้นกับความเข้ากันได้ของทั้งฮาร์ดแวร์และซอฟต์แวร์ รวมทั้งกำลังของเครื่องที่เราเลือกใช้ด้วย ระบบสารสนเทศเพื่อการจัดการเป็นระบบที่เชื่อมต่อหรือรวมโปรแกรมประยุกต์ต่างๆเข้าด้วยกัน

โดยผ่านฐานข้อมูล(database) ดังนั้นผู้ออกแบบและพัฒนาระบบสารสนเทศเพื่อการจัดการ จะต้องมีความรู้ทางคอมพิวเตอร์และการใช้คอมพิวเตอร์ในการประมวลข่าวสารต่างๆ จนสามารถที่จะเลือกทั้งฮาร์ดแวร์, ซอฟต์แวร์, ระบบจัดการฐานข้อมูล(DBMS เป็นซอฟต์แวร์ที่ใช้อำนวยความสะดวกแก่ระบบฐานข้อมูล ในการกำหนดลักษณะของข้อมูล รายละเอียด ความสัมพันธ์สำหรับการจัดการระบบฐานข้อมูล), งานที่จะนำมาใช้เครื่องและบุคลากรสำหรับระบบสารสนเทศเพื่อการจัดการได้อย่างถูกต้อง โดยการออกแบบจะต้องคำนึงถึงค่าใช้จ่ายและการจัดการที่จะเปลี่ยนแปลงไปอีกในอนาคต

4. ระบบช่วยวางแผนและตัดสินใจ มีการนำแผนภาพเพื่อตัดสินใจ (decision-models) มาใช้รวมไปถึงการใช้ระบบช่วยการตัดสินใจ(DDS) ซึ่งระบบช่วยการตัดสินใจนี้มีใช้มาตั้งแต่ปี ค.ศ.1970 ช่วยผู้บริหารใจการตัดสินใจโดยการช่วยวางหนทางเพื่อไปสู่จุดหมายปลายทางและเลือกหนทางที่ดีที่สุดไว้ให้

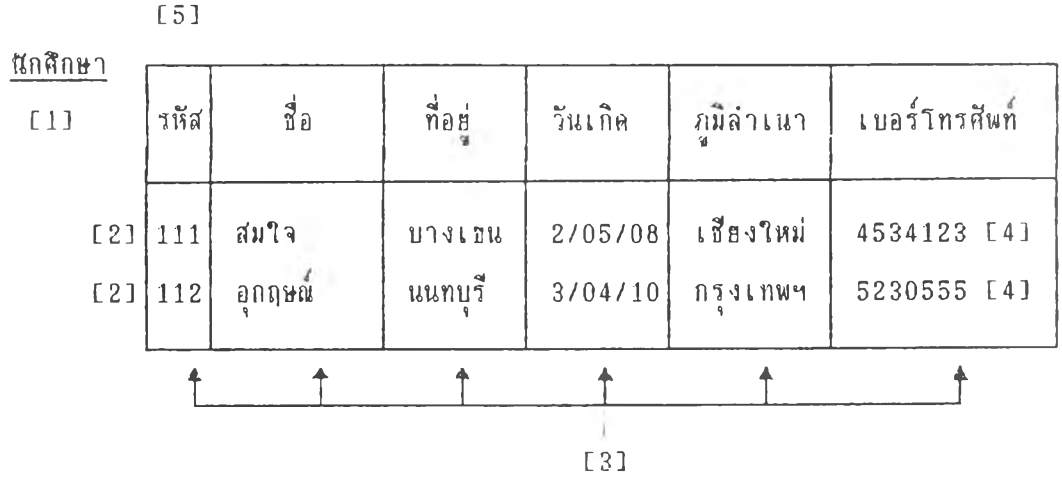
หลักการนอร์มัลไลเซชัน (Normalization)

นอร์มัลไลเซชันเป็นกระบวนการเพื่อพัฒนาการเชื่อมต่อของข้อมูลเพื่อตอบคำถามที่ว่าการออกแบบฐานข้อมูลทั้งทางตรรกะและทางกายภาพที่ได้ออกมาใช้ได้หรือยัง ก่อนจะอธิบายถึงวิธีทำนอร์มัลไลเซชัน เราควรทำความเข้าใจถึงคำศัพท์ต่างๆที่เกี่ยวข้อง อันได้แก่ ความสัมพันธ์และคีย์ของความสัมพันธ์ก่อน

ความสัมพันธ์ (relation) คือกลุ่มของทูเปิล (เทียบได้กับระเบียนข้อมูล) โดยมากแสดงในรูปแบบของตาราง แต่ละแถวของตารางก็คือ 1 ทูเปิลนั่นเอง ซึ่งแต่ละทูเปิลจะมีค่าต่างๆบรรจุอยู่ ดังแสดงในรูปที่ 3.1 ในความสัมพันธ์จะประกอบด้วย ชื่อความสัมพันธ์ (เอนติตี) [1], ทูเปิล (tuples) หรือโรล (rows) [2], แอตตริบิว [3], ค่าของแอตตริบิว [4], คีย์ของความสัมพันธ์ [5]

คีย์ คือแอตตริบิวตั้งแต่ 1 ตัวขึ้นไป ซึ่งจะช่วยระบุ ค้นหา ทูเปิล ที่ต้องการในความสัมพันธ์ คีย์ตัวหนึ่งจะระบุถึงทูเปิลหนึ่งเท่านั้น ดังรูปที่ 3.1 คีย์ของความสัมพันธ์คือรหัสนักศึกษา

- คีย์หลัก (primary key) คือ แอตตริบิวซึ่งทุกแอตตริบิวในความสัมพันธ์ต้องขึ้นกับมัน(dependence) หรือกลุ่มของแอตตริบิวที่เป็นหนึ่งเดียว (unique) และทุกทูเปิลที่จะเพิ่มเข้าในความสัมพันธ์อย่างน้อยแอตตริบิวนี้ต้องมีค่า
- คีย์ประกอบ (compound key) คือ คีย์ที่ประกอบด้วยแอตตริบิวมากกว่า 1 ตัวขึ้น
- คีย์รอง (alternate key) คือ แอตตริบิวที่มีคุณสมบัติเหมือนคีย์หลักทุกประการแต่ไม่ได้รับเลือกเป็นคีย์หลัก
- คีย์เป็นได้ (candidate key) คือ แอตตริบิวที่มีคุณสมบัติที่สามารถเป็นคีย์หลักได้



รูปที่ 3.1 ความสัมพันธ์นักศึกษา

การนอร์มัลไลเซชันแบ่งออกเป็นหลายระดับ ได้แก่ แบบไม่นอร์มัลไลซ์, นอร์มัลฟอร์มระดับที่ 1 (1NF), ..., นอร์มัลฟอร์มระดับที่ 5 (5NF) ซึ่งถ้าความสัมพันธ์ใดเป็นได้ถึง 5NF เรียกว่าเป็นการนอร์มัลไลที่สมบูรณ์ (fully normalized) ซึ่งจะเป็นความสัมพันธ์ที่ง่ายต่อการทำงาน พื้นฐานความคิดของการทำนอร์มัลไลเซชันจะพิจารณาการขึ้นต่อกันของแอตตริบิว ดังนี้

ก. การขึ้นต่อกันทั้งหมด (Functional Dependency) ถ้าในความสัมพันธ์ R มีแอตตริบิว A และ B จะกล่าวว่าแอตตริบิว B เป็นการขึ้นต่อกันทั้งหมดกับแอตตริบิว A ก็ต่อเมื่อแต่ละค่าของ A ในความสัมพันธ์ R จะสัมพันธ์กับแอตตริบิว B เพียงหนึ่งค่าเท่านั้น และเราจะเรียก A ว่าดีเทอร์มิแนน(determinant) กล่าวคือทุกทูเปิลในความสัมพันธ์ R ที่มีค่าของ B ตรงกันจะต้องมีค่าของ A ตรงกันด้วย เขียนแทนด้วย A ------> B และจะกล่าวว่าแอตตริบิว A ขึ้นต่อกันทั้งหมดกับคีย์ประกอบ[K1,K2] หรือเขียนแทนด้วย [K1,K2 ==> A]

เมื่อ K1,K2 -----> A และ K1 --/--> A

และ K2 --/--> A

ข. การไม่ขึ้นตรงกับคีย์หลัก (Transitively dependency) ถ้าในความสัมพันธ์ R มีคีย์หลักคือ K และแอตตริบิว A และ B จะกล่าวว่าแอตตริบิว B ไม่ขึ้นตรงกับคีย์หลัก

เมื่อ K --------> A และ A ----/---> K

และ A --------> B

ค. การขึ้นต่อกันหลายค่า (Multivalued Dependency) ในความสัมพันธ์ R มีแอตตริบิว A,B,C เราจะกล่าวว่า แอตตริบิว B เป็นการขึ้นต่อกันหลายค่าบน A ก็ต่อเมื่อเซทของค่าของแอตตริบิว B ในความสัมพันธ์ R ที่มีความสัมพันธ์ตรงกับคู่ของ <A,C> นั้น จะไม่ขึ้นกับค่าของ C เขียนแทนด้วย A ------>> B และจะมีลักษณะที่เห็นได้ดังนี้

- แอตตริบิว A ค่าหนึ่งจะเป็นตัวกำหนดกลุ่มของค่าแอตตริบิว B กล่าวคือ เมื่อ 2 ทูเปิลในความสัมพันธ์ R มีค่า A เดียวกัน ไม่จำเป็นต้องมีค่า B เหมือนกัน แต่ค่าของ B จะต้องอยู่ในกลุ่มของค่า B ที่ถูกกำหนดโดย A

- การเปลี่ยนแปลงค่าใดๆในแอตตริบิว C จะไม่มีผลกระทบต่อค่า B

- 2 ทูเปิลในความสัมพันธ์ R ที่มีค่า B เหมือนกันไม่จำเป็นต้องมีค่า A เดียวกัน

- ค่าของแอตตริบิว C 2 ค่าที่สัมพันธ์กับค่า A เดียวกัน จะต้องสัมพันธ์กับค่าของ B ในกลุ่มเดียวกันและเป็นกลุ่มที่ถูกกำหนดโดยค่า A นั้นๆ

1. ความสัมพันธ์ไม่นอร์มัลไลซ์ (Unnormalized relations) คือความสัมพันธ์ที่มีค่าของแอตตริบิวเป็นกลุ่ม เวลาจะเพิ่มทูเปิลหรือคอลัมภ์ก็จะมีแอตตริบิวหลายค่า ดังเช่นเอนติตีนักศึกษาในรูปที่ 3.2

เอนติตีนักศึกษา

รหัสนักศึกษา	ชื่อนักศึกษา	รหัสวิชาที่เรียน	ชื่อวิชา	รหัสอ.ที่ปรึกษา	ชื่ออ.ที่ปรึกษา
1111111	สำรวย	171161,171162, ...	แคล1,แคล2	123456789012	อ.มยุรี

รูปที่ 3.2 ความสัมพันธ์แบบไม่นอร์มัลไลซ์

ข้อเสียของความสัมพันธ์แบบนี้คือ ยากแก่การดูแลเพราะจำนวนค่าของแอตตริบิวในกลุ่มไม่คงที่ และจะพบสภาพที่มีความซ้ำซ้อนของข้อมูลจำนวนมาก และการผิดพลาดของข้อมูลที่เกิดจากการเพิ่ม, แก้ไขและลบข้อมูลก็จะมีตามมาด้วย

2. ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 1 (First normal form : 1NF) คือความสัมพันธ์ที่ทุกทูเปิลในความสัมพันธ์จะมีค่าของแอตตริบิวเป็นค่าเดียวโดดๆ ไม่เป็นกลุ่ม หรืออาจไม่มีค่าเลยก็ได้ (เป็น null) ยกเว้นแอตตริบิวที่เป็นคีย์หลัก หรือกล่าวอีกแบบหนึ่งได้ว่า แอตตริบิวที่ไม่ใช่คีย์จะต้องขึ้นต่อกันทั้งหมดกับคีย์ ดังในรูปที่ 3.3

วิธีการทำความสัมพันธ์ให้อยู่ใน 1NF ก็คือจะต้องกำจัดทูเปิลที่มีแอตตริบิวเป็นกลุ่มนั่นเอง(flatten) ซึ่งจะทำให้การดูแลข้อมูลสะดวกและง่ายขึ้นแต่อย่างไรก็ตามนอร์มัลไลเซชันระดับนี้ก็ยังคงมีความซ้ำซ้อนของข้อมูล และอาจเกิดข้อผิดพลาดจากการกระทำต่างๆขึ้นได้เช่นเดิม

เอนติตีนักศึกษา

รหัสนักศึกษา	ชื่อนักศึกษา	รหัสวิชาที่เรียน	ชื่อวิชา	รหัสอ.ที่ปรึกษา	ชื่ออ.ที่ปรึกษา
1111111	สำรวย	171161	แคล1	1234567890112	อ.มยุรี
1111111	สำรวย	171162	แคล2	1234567890112	อ.มยุรี
2222222	อาสา	171813	ฟิสิกส์	1234567890112	อ.มยุรี

รูปที่ 3.3 ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 1

3. ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 2 (Second normal form : 2NF) คือ ความสัมพันธ์ที่อยู่ในระดับที่ 1 และทุกแอตตริบิวที่ไม่ใช่คีย์หลักต้องขึ้นตรงอย่างสมบูรณ์กับคีย์หลัก หรือกล่าวได้ว่านอร์มัลฟอร์มระดับที่ 2 จะต้องไม่มีแอตตริบิวใดขึ้นกับบางส่วนของคีย์หลัก ดังในรูปที่ 3.4 (ก) และ (ข) จะพิจารณาได้โดยดูคีย์หลักที่เป็นคีย์ประกอบ ถ้าแอตตริบิวแต่ละอันในคีย์หลักมีความเกี่ยวข้องกันเป็นแบบ (M:M) จะพบว่าไม่อยู่ใน 2NF

เอนติตีนักศึกษา

รหัสนักศึกษา	ชื่อนักศึกษา	รหัสอ.ที่ปรึกษา	ชื่ออ.ที่ปรึกษา
1111111	สำรวย	1234567890112	อ.มยุรี
2222222	อาสา	1234567890112	อ.มยุรี

(ก)

เอนติตีรายวิชา

รหัสรายวิชา	ชื่อวิชา
171161	แคล 1
171162	แคล 2
171813	ฟิสิกส์

(ข)

รูปที่ 3.4 ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 2 (ก) ได้จากความสัมพันธ์ในรูป 3.3 แล้วตัดแอตตริบิวรายวิชาออกเนื่องจากไม่ขึ้นกับรหัสนักศึกษา ได้ความสัมพันธ์เพิ่มใหม่ดัง (ข)

วิธีทำให้อยู่ใน 2NF คือให้แยกเอนติตีที่มีแอตตริบิวที่ขึ้นกับส่วนหนึ่งของคีย์อยู่ออกเป็นเอนติตีใหม่ ซึ่งจะสามารถตัดปัญหาความซ้ำซ้อนในบางจุดลงได้ แต่ก็ยังคงมีความซ้ำซ้อนอยู่

4. ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 3 (Third normal form : 3NF) คือ ความสัมพันธ์ที่อยู่ในระดับที่ 2 และแอตตริบิวที่ไม่ใช่คีย์จะไม่มีการขึ้นต่อกัน กล่าวคือ ต้องไม่มีแอตตริบิวที่ไม่ขึ้นตรงกับคีย์หลัก จากเอนติตีนักศึกษาในรูปที่ 3.4 (ก) แยกใหม่ได้ดังรูปที่ 3.5

เอนติตีนักศึกษา

รหัสนักศึกษา	ชื่อนักศึกษา	รหัส อ.ที่ปรึกษา
1111111	สำรวย	1234567890112
2222222	อาสา	1234567890112

เอนติตีอาจารย์ที่ปรึกษา

รหัส อ.ที่ปรึกษา	ชื่ออาจารย์
1234567890112	อ.มยุรี

รูปที่ 3.5 ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 3

5. บอยด์คอดนอร์มัลฟอร์ม (Boyce/Codd normal form : BCNF) คือความสัมพันธ์ที่อยู่ใน 3 NF และทุกแอตตริบิวที่เป็นดีเทอร์มิแนนจะต้องเป็นคีย์เป็นได้ กล่าวคือ ทุกแอตตริบิวต้องขึ้นกับทั้งหมดของคีย์เป็นได้ ไม่ใช่เพียงบางส่วน จะสังเกตได้ในเอนติตีที่มีคีย์เป็นได้เป็นคีย์ประกอบและมีการกับซ้อนกันของคีย์นั้นๆ ดังในรูปที่ 3.6 เอนติตีงานที่มอบหมาย จะเห็นว่าที่ปรึกษาคนหนึ่งจะรับผิดชอบงานเพียงงานเดียวเท่านั้น ดังนั้นจึงจัดว่าแอตตริบิวชื่อที่ปรึกษาเป็นดีเทอร์มิแนนของรหัสงานแต่ไม่ได้เป็นส่วนหนึ่งของคีย์เป็นได้ ดังนั้นจึงยังไม่อยู่ในบอยด์คอดนอร์มัลฟอร์ม จึงต้องแตกออกเป็น 2 เอนติตีดังรูปที่ 3.7 (ก)และ(ข)

เอนติตีงานที่มอบหมาย

รหัสงาน	ชื่อผู้ร่วมงาน	ชื่อที่ปรึกษา
101	สาคร	สมศรี
101	ดารณี	สมศรี
101	เจนจิรา	ธาณี
112	สาคร	เกรียงไกร

รูปที่ 3.6 เอนติตีที่ไม่อยู่ใน BCNF

เอนติตีสมาชิกผู้ทำงาน

ชื่อที่ปรึกษา	ชื่อผู้ร่วมงาน
สมศรี	สาคร
สมศรี	ดารณี
ชาญี	เจนจิรา
เกรียงไกร	สาคร

เอนติตีงานกับที่ปรึกษา

รหัสงาน	ชื่อผู้ร่วมงาน
101	สาคร
101	ดารณี
101	เจนจิรา
112	สาคร

รูปที่ 3.7 ความสัมพันธ์แบบบอยด์คอดนอร์มัลฟอร์ม

6. ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 4 (Fourth normal form : 4NF) คือ ความสัมพันธ์ที่อยู่ใน 3NF และไม่มีการขึ้นต่อกันแบบหลายค่าอยู่ในความสัมพันธ์ ดังรูปที่ 3.8(ก)

เอนติตีบุคคลากร

บุคลากร	ภาษา	งานวิจัย
1111	ไทย	002
1111	จีน	002
1111	ไทย	004
1111	จีน	004

(ก)

เอนติตีภาษาที่บุคคลากรใช้ได้

บุคลากร	ภาษาที่ใช้
1111	ไทย
1111	จีน

(ข)

เอนติตีงานวิจัยของบุคคลากร

บุคลากร	งานวิจัย
1111	002
1111	004

(ค)

รูปที่ 3.8 การทำให้อยู่ในรูปนอร์มัลฟอร์มระดับที่ 4

แอตตริบิวภาษาขึ้นต่อกันหลายค่ากับรหัสบุคลากรทำให้ไม่จัดเป็น 4NF จะต้องทำการแยกเป็น 2 เอนติตีดังรูป 3.8 (ข)และ(ค)

7. ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 5 (Fifth normal form : 5NF) คือ ความสัมพันธ์ที่ไม่สามารถนำไปสร้างความสัมพันธ์ใหม่โดยการเชื่อมระหว่าง 2 ความสัมพันธ์ ด้วยคีย์ต่างกันได้ โดยปกติมักจะเป็นปัญหาสำหรับคีย์หลักที่เป็นคีย์ประกอบ เป็นขั้นที่พิจารณาได้ยาก จะเกิดระเบียนข้อมูลใหม่ที่ไม่อยู่จริงขึ้นมาเมื่อนำเอนติตีมารวมกัน

ฐานข้อมูล

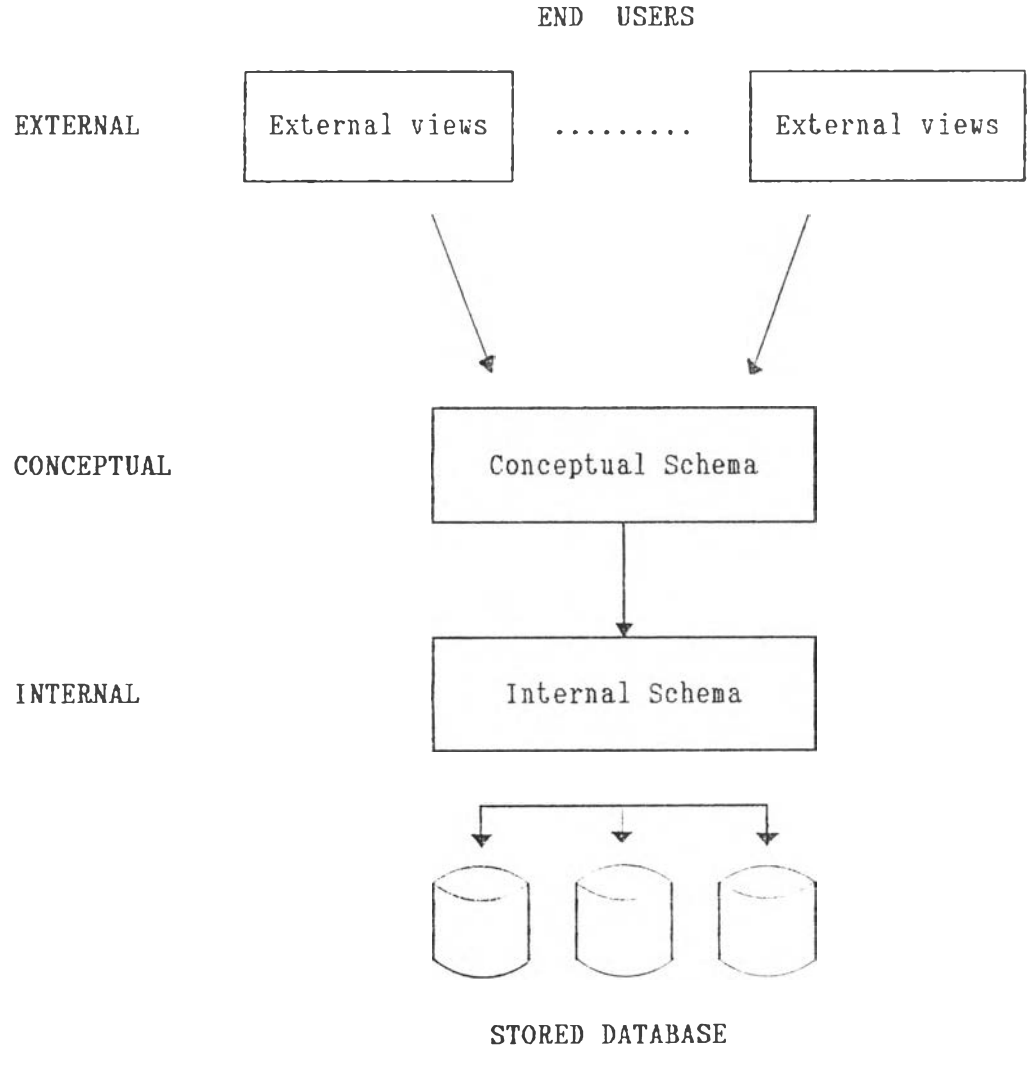
ฐานข้อมูลคือที่เก็บรวบรวมกลุ่มข้อมูลของระบบไว้ด้วยกัน เพื่อให้บุคคลและงานประยุกต์ต่างๆสามารถมาดำเนินการอย่างใดอย่างหนึ่ง(ดึง,เพิ่ม,ลบ,แก้ไข)กับข้อมูลนั้นๆได้ มีเรื่องต่างๆที่ควรรู้ดังนี้

1. สถาปัตยกรรมของฐานข้อมูล (Database Architecture)

ได้มีการกำหนดมาตรฐานขึ้นโดย ANSI(American National Standards Institute) ในปี 1975 เรียกกันว่าสถาปัตยกรรมสามระดับ (Three-level Architecture) การอธิบายรายละเอียดของระดับต่างๆจะไม่ได้เกี่ยวข้องกับข้อมูลที่เก็บในฐานข้อมูล แต่จะอธิบายโครงสร้างของฐานข้อมูลโดยรวมเพื่อให้สามารถแยกได้ระหว่างฐานข้อมูลเชิงกายภาพ และสิ่งที่ผู้ใช้มองเห็น การพิจารณาเพื่อให้สามารถแยกได้นั้นก็เนื่องมาจากการที่ผู้ใช้แต่ละคนอาจมองข้อมูลตัวเดียวกันต่างกัน ดังนั้นถ้ามีการเปลี่ยนแปลงโครงสร้างภายในก็ไม่ควรให้กระทบกับมุมมองของผู้ใช้ สถาปัตยกรรมสามระดับดังกล่าวได้แก่

1.1. แบบแผนฐานข้อมูลภายนอก(External schemas) จะเป็นระดับที่ใกล้กับผู้ใช้มากที่สุด เป็นสิ่งที่ผู้ใช้คิดเกี่ยวกับข้อมูล จะอธิบายถึงวิว(view)ที่ผู้ใช้สนใจ ข้อมูลที่เก็บจริงอาจมีมากกว่าที่ผู้ใช้ต้องการและข้อมูลตัวเดียวกันผู้ใช้อาจมองไม่เหมือนกัน เช่น ข้อมูลวันที่ (ผู้ใช้คนหนึ่งอาจมองเป็น วัน/เดือน/ปี อีกคนมองเป็น เดือน/วัน/ปี ก็ได้) นอกจากนั้นสิ่งที่ผู้ใช้มองเห็นอาจไม่ได้เก็บจริงในเครื่องแต่ได้จากการคำนวณออกมา ส่วนนี้เองจะถูกแปลโดยระบบจัดการฐานข้อมูลเก็บไว้ในพจนานุกรมข้อมูล

1.2. แบบแผนฐานข้อมูลภายใน (Internal schemas) พิจารณาการจัดการระบบ การเก็บข้อมูลจริง อธิบายฐานข้อมูลในการเก็บทางกายภาพจริงๆ มองข้อมูลโดยมุมมองของระบบจัดการฐานข้อมูล ใช้โครงสร้างข้อมูล (data structure) และการจัดการแฟ้มข้อมูล (file organization) ในการอธิบาย และจะทำงานรวมกับระบบปฏิบัติการ (operating system) ในการเก็บข้อมูลลงที่เก็บข้อมูลสำรอง



รูปที่ 3.9 สถาปัตยกรรมของฐานข้อมูล

1.3. แบบแผนฐานข้อมูลเชิงมโนภาพ (Conceptual schemas) จะเป็นตัวที่ใช้เชื่อมระหว่างแบบแผนฐานข้อมูลภายนอกกับภายใน อธิบายฐานข้อมูลในรายละเอียดโดยรวมทั้งหมดเพื่อเชื่อมกับสิ่งที่ผู้ใช้มอง รูปแบบข้อมูล ความสัมพันธ์ เงื่อนไขต่างๆ รวมถึงความปลอดภัยและความถูกต้องของข้อมูลจะถูกเก็บไว้ด้วย แต่จะไม่ลงลึกถึงการเก็บในเครื่อง

นอกจากนั้นสถาปัตยกรรมทั้งสามระดับดังกล่าวยังมีความเป็นอิสระของข้อมูลที่เห็นได้คือ แบบแผนฐานข้อมูลภายนอกจะมีรูปแบบที่คงที่ไม่เปลี่ยนไปแม้จะมีการเปลี่ยนแบบแผนฐานข้อมูลเชิงมโนภาพ เช่น มีการเพิ่มรูปแบบข้อมูลใหม่หรือมีความสัมพันธ์ใหม่ๆเพิ่มขึ้น เรียกว่าความเป็นอิสระทางตรรก (logical data independence) และแบบแผนฐานข้อมูลเชิงมโนภาพก็เช่นกันจะมีรูปแบบที่คงที่แม้จะมีการเปลี่ยนแปลงทางกายภาพ เช่น เปลี่ยนวิธีการดึงข้อมูล (access method) หรือลำดับของข้อมูลที่เก็บอยู่จริงเปลี่ยนไป เรียกว่าความเป็นอิสระทาง

กายภาพ (physical data independence) อย่างไรก็ตามโดยทั่วไปแล้วระบบจัดการฐานข้อมูลไม่ได้แบ่งสถาปัตยกรรมทั้ง 3 ระดับจากกันโดยเด็ดขาด ในการทำงานส่วนมากจะให้ผู้ใช้ระบุถึงระดับภายนอกเท่านั้น แล้วระบบจัดการฐานข้อมูลจะเป็นตัวเปลี่ยนการร้องขอหรือการระบุนั้นๆให้เป็นระดับมโนภาพหรือภายในเองแล้วจึงประมวลผลข้อมูลออกมาให้

2. วงจรระบบฐานข้อมูล (Database system Life cycle)

คือช่วงเวลาระหว่างการออกแบบ สร้าง และใช้จนถูกแทนที่ด้วยระบบใหม่ที่สมบูรณ์ มักจะต้องกินเวลาหลายปี ประกอบด้วยขั้นตอนต่างๆดังนี้

2.1. ข้อกำหนดระบบ (System definition) การศึกษาระบบงานปัจจุบัน รายงาน, งานประยุกต์ต่างๆ รวมถึงซอฟแวร์ที่ใช้ การกำหนดขอบเขตและจุดมุ่งหมายของระบบฐานข้อมูลอันใหม่

2.2. การออกแบบ (Design) จะประกอบไปด้วยการออกแบบทางตรรกและการออกแบบทางกายภาพ(จะอธิบายรายละเอียดภายหลัง) การออกแบบที่ดีจะต้องได้ระบบที่ง่ายต่อการเปลี่ยนแปลงหรือขยายตัวของระบบในอนาคต ผลที่ได้สุดท้ายจากขั้นนี้คือ การออกแบบทางตรรกและทางกายภาพที่พัฒนาบนระบบจัดการฐานข้อมูลที่ได้เลือกแล้ว

2.3. การติดตั้งระบบ (Implemantation) เป็นกระบวนการบรรจุข้อกำหนดของฐานข้อมูลทั้ง 3 ระดับลงในฐานข้อมูลผ่านระบบจัดการฐานข้อมูลที่เลือกแล้วนั้นเอง และรวมไปถึงการสร้างต้นแบบด้วย

2.4. การบรรจุหรือการเปลี่ยนแปลงข้อมูล (Loading or data conversion) เพื่อนำข้อมูลเข้าสู่ระบบใหม่ซึ่งจะนำเข้าโดยตรงหรืออาจแปลงแฟ้มข้อมูลจากระบบเดิมก่อนเข้าระบบใหม่

2.5. การแปลงงานประยุกต์ (Application conversion) การดัดแปลงงานประยุกต์ที่มีอยู่เดิมให้เข้าสู่ระบบใหม่ และสร้างงานประยุกต์ใหม่ขึ้นมาด้วย

2.7. การทดสอบและตรวจสอบความถูกต้อง (Testing and validation) ของระบบใหม่

2.8. การปฏิบัติการ (Operation) เป็นขั้นตอนการรวมงานประยุกต์ทุกงานเข้าด้วยกัน แล้วจัดการเกี่ยวกับสิทธิและขอบเขตการทำงานของผู้ใช้แต่ละคน กำหนดใครจะเป็นคนทำการกู้ข้อมูลและทำแฟ้มข้อมูลสำรอง

2.9. การตักเตือนและบำรุงรักษา (Monitoring and maintanance) ดูแลการเติบโตและขยายตัวที่จะเกิดขึ้นกับข้อมูลและซอฟแวร์

ขั้นตอนการออกแบบ ติดตั้งและนำข้อมูลเข้าเป็นจุดสำคัญที่สุดในวงจร ขั้นตอนต่างๆเหล่านี้ ในบางครั้งก็แยกกันโดยเด็ดขาด บางครั้งก็อาจทับซ้อนกันได้ ในที่นี้จะอธิบายรายละเอียดเฉพาะการออกแบบเท่านั้น

3. การออกแบบฐานข้อมูล

กระบวนการในการออกแบบทางตรรกและทางกายภาพของฐานข้อมูลจะประกอบด้วยงาน 2 ส่วน คือ การออกแบบในส่วนของเนื้อหาข้อมูลและโครงสร้างของฐานข้อมูล (content and structure of database) อีกส่วนคือ การออกแบบในเรื่องการประมวลผลฐานข้อมูลและซอฟต์แวร์ของงานประยุกต์ (database processing and software application) จุดประสงค์ของการออกแบบก็คือ

- เตรียมข้อมูลที่จำเป็นเพื่อสนองตอบตามความต้องการของผู้ใช้งาน และงานประยุกต์ที่เข้ามา
- เตรียมโครงสร้างข้อมูลที่เป็นธรรมชาติและง่ายต่อการเข้าใจมากที่สุด
- เพื่อรองรับต่อความต้องการในการประมวลผล จัดเก็บข่าวสาร อย่างมีประสิทธิภาพและใช้เวลาในการตอบสนองที่เหมาะสม

การออกแบบงานทั้งสองส่วนนั้นมักจะทำไปด้วยกันแยกจากกันได้ยากมาก เพราะในบางครั้งเราจะสามารถระบุเนื้อหาข้อมูลได้จากการศึกษางานประยุกต์ หรืออาจกำหนดงานประยุกต์โดยอ้างถึงโครงสร้างของฐานข้อมูล

การออกแบบฐานข้อมูลแบ่งงานออกได้เป็นหลายช่วง และไม่จำเป็นต้องทำโดยเรียงลำดับ อาจทำหลายช่วงควบคู่กันไปก็ได้ ประกอบด้วย 6 ช่วงด้วยกัน คือ

- การเก็บรวบรวมและวิเคราะห์ข้อมูล (Requirements collection and analysis)
- การออกแบบฐานข้อมูลเชิงมโนภาพ (Conceptual database design)
- การเลือกระบบจัดการฐานข้อมูล (Choice of a DBMS)
- การแปลงให้อยู่ในโมเดลของระบบจัดการฐานข้อมูล (Data model mapping บางทีเรียก logical database design)
- การออกแบบฐานข้อมูลทางกายภาพ (Physical database design)
- การติดตั้งและนำฐานข้อมูลมาใช้จริง (Database implementation)

3.1. การเก็บรวบรวมและวิเคราะห์ข้อมูล

การออกแบบฐานข้อมูลให้มีประสิทธิภาพนั้น เราจะต้องคำนึงถึงความคาดหวังและจุดประสงค์ของผู้ใช้ฐานข้อมูลนั้นๆให้ละเอียดเท่าที่จะเป็นไปได้ โดยจะเก็บรวบรวมสิ่งเหล่านั้นไว้เป็นข้อมูลของระบบที่จะนำไปออกแบบ การเก็บรวบรวมและวิเคราะห์นี้จะต้องพิจารณาทั้งข้อมูลที่มีใช้อยู่ในปัจจุบันและข้อมูลที่ต้องการใช้เพิ่มเติม รวมถึงข้อมูลที่จะมีเกิดขึ้นได้ในอนาคต เป็นขั้นตอนที่ใช้เวลามากประกอบด้วยหลักเกณฑ์ต่างๆที่ผู้ออกแบบต้องทำ ดังนี้

- จัดกลุ่มผู้ใช้และงาน โดยให้มีการตั้งคีย์เฉพาะกลุ่ม
- ทำการศึกษาและตรวจสอบโปรแกรมประยุกต์ รวมถึงรายงาน แบบ

ฟอร์ม หรือผังงานต่างๆ โดยละเอียด

- วิเคราะห์เกี่ยวกับสิ่งแวดล้อมที่ต้องจัดการ, การประมวลผลที่ต้องทำ, แผนงานทั้งในปัจจุบันและอนาคต รวมถึงชนิดของรายการเปลี่ยนแปลงและความถี่ในการเกิด ดูทิศทางของข่าวสารที่เข้าและออกจากระบบ คูอินต์พุต-เอาต์พุต ที่แต่ละรายการเปลี่ยนแปลงกำหนดไว้ เพื่อให้ทราบถึงความถี่และปริมาณของข้อมูลในฐานข้อมูลที่จะถูกเรียกใช้หรือเปลี่ยนแปลง

- จดบันทึกการสอบถามและสัมภาษณ์ โดยจะต้องถามบุคลากรทุกระดับ โดยอาจใช้แบบสอบถามหรือจะถามโดยตรงเลยก็ได้ ควรถามถึงสิทธิของผู้ใช้แต่ละคน ,คีย์ในการดึงข้อมูลของแฟ้มข้อมูลในระบบเก่า ,ชื่อข้อมูล ,รายละเอียดและแหล่งที่มา กล่าวคือ ถามสิ่งต่างๆที่จะต้องนำมาเก็บในระบบพจนานุกรมข้อมูล คำถามอย่างหนึ่งที่สำคัญคือ ข้อมูลนั้นๆ ยังจำเป็นต้องใช้ในระบบหรือไม่ และมีข้อมูลใดที่ยังขาดอยู่ต้องการเพิ่มไปในระบบใหม่นี้บ้าง

เมื่อผู้ออกแบบได้ข้อมูลต่างๆตามที่ต้องการแล้วให้ทำการเปลี่ยนแปลงให้อยู่ในรูปแบบที่แน่นอนเข้าใจได้ทั่วไป โดยใช้เทคนิคต่างๆเช่น DFD, SADT HIPO ฯลฯ

3.2. การออกแบบฐานข้อมูลเชิงมโนภาพ

ผู้ออกแบบสามารถพัฒนารายละเอียดของข้อมูลด้วยการใช้โมเดลข้อมูลเชิงตรรก (logical data model) โดยมีการกำหนดเอนติตี รีเลชันชิป และแอตตริบิว การออกแบบช่วงนี้จะประกอบด้วยกิจกรรม 2 อย่างคือ

- การออกแบบแบบแผนฐานข้อมูลเชิงมโนภาพ จะทำการตรวจสอบความต้องการในการใช้ข้อมูลที่ได้มาจากข้อ 3.1 และผลิตแบบแผนฐานข้อมูลเชิงมโนภาพโดยใช้โมเดลข้อมูลที่เป็นอิสระจากระบบจัดการฐานข้อมูล

- การออกแบบรายการการเปลี่ยนแปลง (transaction design) เป็นการตรวจสอบงานประยุกต์ของฐานข้อมูล ซึ่งได้รับการวิเคราะห์มาแล้วจากข้อที่ 3.1 และผลิตข้อกำหนดรายละเอียดสำหรับงานที่จะเข้ามา

แต่อย่างไรก็ดีโมเดลข้อมูลที่ได้นี้ยังไม่สามารถนำไปใช้ติดตั้งฐานข้อมูลได้โดยตรง อาจมีการเปลี่ยนแปลงเพื่อให้ใช้ได้ต่อไป

3.2.1. การออกแบบแบบแผนฐานข้อมูลเชิงมโนภาพ

เป็นการออกแบบที่ใช้โมเดลข้อมูลที่เป็นอิสระจากระบบจัดการฐานข้อมูล ซึ่งจะยังไม่สามารถนำไปติดตั้งฐานข้อมูลได้จริง แต่อย่างไรก็ยังมีข้อดีอยู่บ้าง คือ

ก. เมื่อไม่มีการระบุชนิดของระบบจัดการฐานข้อมูล ทำให้เข้าใจโครงสร้างของข้อมูล ความหมาย ความสัมพันธ์ภายในและกฎข้อบังคับต่างๆได้อย่างเป็นอิสระ เพราะระบบจัดการฐานข้อมูลจะมีคุณสมบัติหรือข้อจำกัดพิเศษของแต่ละอัน ซึ่งอาจไปมีผลบังคับต่อการออกแบบ

ข. แบบแผนฐานข้อมูลเชิงมโนภาพ เป็นระดับที่มีเสถียรภาพ

มากที่สุดในฐานข้อมูล การเลือกชนิดของระบบจัดการฐานข้อมูลหรือเปลี่ยนแปลงรายการออกแบบในภายหลังจึงไม่ควรจะมีผลมากระทบกับโมเดลส่วนนี้

ค. ความเข้าใจในแบบแผนฐานข้อมูลเชิงมโนภาพเป็นสิ่งที่สำคัญสำหรับผู้ใช้ฐานข้อมูลและผู้ออกแบบงานประยุกต์ และการใช้โมเดลเชิงตรรกนี้จะละเอียดลึกซึ้งเป็นธรรมชาติมากกว่าโมเดลของแต่ละชนิดของระบบจัดการฐานข้อมูล (ได้แก่ โมเดลเชิงความสัมพันธ์ ,โมเดลเครือข่าย ,โมเดลลำดับชั้น)

ง. แผนภาพที่ใช้อธิบายรายละเอียดของแบบแผนฐานข้อมูลเชิงมโนภาพสามารถอธิบายถึงการติดต่อสื่อสารระหว่างผู้ใช้งานฐานข้อมูล, โปรแกรมเมอร์, และผู้วิเคราะห์ระบบได้ดีที่สุด

การออกแบบนี้มีวิถีทางในการออกแบบอยู่ 2 วิถีทางโดยพิจารณาจากวิธีการและลำดับในการรวมความต้องการหรือมุมมองของผู้ใช้ต่างๆเข้าด้วยกัน ได้แก่

ก. วิถีทางการออกแบบแบบแผนส่วนกลาง (centralized schema design approach) เป็นการรวมความต้องการของงานและผู้ใช้ที่ต่างๆกันที่เก็บรวบรวมได้จากข้อ 3.1 เข้าด้วยกันก่อนจะทำการออกแบบ กล่าวคือรวบรวมข้อมูลทั้งหมดเข้าด้วยกันก่อนจึงจะออกแบบโมเดล ซึ่งค่อนข้างยากเนื่องจากจะตกลงกันไม่ได้ ดังนั้นในการออกแบบลักษณะนี้จึงกำหนดให้ผู้บริหารฐานข้อมูลเป็นผู้ตัดสินใจ ความสำเร็จจึงขึ้นกับความสามารถและประสบการณ์ของกลุ่มผู้บริหารฐานข้อมูล

ข. วิถีทางการรวมมุมมอง (view integration approach) การออกแบบขั้นแรกจะออกแบบโมเดลแยกกันไปของแต่ละคนหรือแต่ละงานโดยผู้ใช้ของงานนั้นๆเป็นผู้ทำ จากนั้นจึงนำโมเดลที่ได้นั้นมารวมกันโดยผู้บริหารฐานข้อมูล

ความแตกต่างที่เห็นได้ชัดก็คือ ในวิถีทางแรกการตัดสินใจและการออกแบบจะตกอยู่กับผู้บริหารฐานข้อมูล ส่วนวิถีทางที่สองผู้ออกแบบแต่ละกลุ่มจะออกแบบโมเดลเชิงมโนภาพของตน แล้วผู้บริหารฐานข้อมูลจึงมารวบรวมอีกที สำหรับวิถีทางแรกจะยุ่งยากและเป็นภาระกับผู้บริหารฐานข้อมูล จึงเริ่มมานิยมวิถีทางที่สองซึ่งเหมาะมากกับระบบฐานข้อมูลขนาดใหญ่

นอกจากนั้นยังมีกลวิธีในการออกแบบแบบแผนฐานข้อมูลได้หลายทิศทางด้วยกัน คือ

ก. การออกแบบจากสูงสู่ต่ำ (Top-down design) จะเริ่มกับแบบแผนฐานข้อมูลในระดับจินตภาพ โดยกำหนดเอนติตีจำนวนไม่มากนักแล้วกำหนดแอตทริบิวของแต่ละเอนติตีนั้น จากนั้นแตกเอนติตีที่สมควรแตกให้ออกเป็นเอนติตีและรีเลชันชิปที่ย่อยลงมา

ข. การออกแบบจากต่ำสู่สูง (Bottom-up design) เริ่ม

จากการกำหนดภาพพื้นฐานและนำมารวมกัน จากนั้นจึงเพิ่มสิ่งอื่นๆเข้าไป กล่าวคือเริ่มด้วยการกำหนดและให้ความหมายแอตตริบิวจากนั้นจึงรวมเข้าเป็นเอนติตีและรีเลชันชิประหว่างเอนติตี

ค. การออกแบบจากภายในสู่ภายนอก (inside-out design) เป็นการออกแบบจากสูงสู่ต่ำวิธีหนึ่ง ที่จะสนใจพิจารณาในเรื่องที่เห็นชัดเจนก่อน แล้วจึงเพิ่มเอนติตีอื่นที่เกี่ยวข้องและรีเลชันชิปที่สัมพันธ์กับเอนติตีที่มีอยู่เดิม

ง. แบบผสม (mixed) จะดูจากความต้องการของผู้ใช้ที่รวบรวมได้ แล้วบางส่วนจะทำโดยใช้แบบสูงสู่ต่ำ บางอันใช้แบบต่ำสู่สูง จากนั้นจึงนำมารวมกัน

3.2.2. การออกแบบรายการเปลี่ยนแปลง

จะทำการตรวจสอบงานของฐานข้อมูลที่เก็บรวมได้จากข้อที่ 3.1 ซึ่งจะเห็นได้ว่ามีงานประยุกต์ต่างๆมากมาย เพื่อให้ข้อมูลครบถ้วนควรมีการกำหนดลักษณะหน้าที่เฉพาะของงานนั้นๆ

3.3. การเลือกระบบจัดการฐานข้อมูล

การเลือกนี้จะมีตัวประกอบอื่นๆเข้ามามีผลทั้งที่เป็นทางเทคนิคและที่เกี่ยวกับเศรษฐกิจ สิ่งที่ต้องพิจารณาในช่วงนี้นอกจากแบบของระบบจัดการฐานข้อมูลอันได้แก่ ระบบจัดการฐานข้อมูลแบบความสัมพันธ์,แบบเครือข่ายและแบบลำดับชั้นแล้ว ยังต้องพิจารณาโครงสร้างในการเก็บข้อมูล, ทิศทางการเข้าถึงข้อมูลที่ฐานข้อมูลชนิดนั้นๆจะทำได้, ภาษาระดับสูงที่ใช้ได้ ฯ

3.3.1. การพิจารณาในทางเทคนิค จะขออธิบายถึงระบบจัดการฐานข้อมูลแต่ละแบบโดยคร่าวๆ ก่อน

ก. ระบบจัดการฐานข้อมูลแบบความสัมพันธ์ จะแสดงข้อมูลในรูปแบบของตารางที่ประกอบด้วยคอลัมภ์ต่างๆ ซึ่งห้ามมีชื่อซ้ำกันในแต่ละตาราง แต่ละอันจะมีลักษณะเป็นหนึ่งเดียวคือไม่สามารถแยกย่อยออกได้อีก ระบบจัดการฐานข้อมูลชนิดนี้ที่มีในปัจจุบัน เช่น System R , DB2, SQL/DS, INGRES, ORACLE

ข้อดี : เข้าใจง่าย การดึงข้อมูลไม่ต้องทำเป็นขั้นตอน ทฤษฎีที่ใช้พัฒนาระบบจัดการฐานข้อมูลแบบนี้ เป็นที่รู้จักกันดีและง่าย

ข้อเสีย : ประสิทธิภาพจะเปรียบไม่ได้กับระบบจัดการฐานข้อมูลแบบอื่น

ข. ระบบจัดการฐานข้อมูลแบบลำดับชั้น เป็นแบบที่เก่าแก่ที่สุด มีโครงสร้างในการเก็บข้อมูลเป็นแบบต้นไม้ (tree) คือจะแบ่งข้อมูลเป็นลำดับชั้นต่ำลงมาเรื่อยๆ แต่ละอันจะเรียกว่าโน้ค (node) โน้คอันแรกสุดเรียกว่า root นอกจากนั้นโน้คที่มีการแตกย่อยจะเรียกโน้คตัวบนว่าโน้คแม่ (parent node) และเรียกโน้คตัวล่างว่าโน้คลูก (child node) ฐานข้อมูลแบบนี้ก็ เช่น IMS/VS

ข้อดี : ความสัมพันธ์เป็นแบบง่ายๆ การนำข้อมูลมาใช้จึงทำได้ง่ายและมีประสิทธิภาพดี

ข้อเสีย : ไม่สามารถแสดงความสัมพันธ์ชนิด (M:N) ได้

ค. ระบบจัดการฐานข้อมูลแบบเครือข่าย มีออกมาในเวลาใกล้เคียงกับแบบลำดับชั้น อธิบายข้อมูลในรูปแบบโน้ดและตัวเชื่อม แต่ละโน้ดคือข้อมูลหนึ่งระเบียนหรือหนึ่งรูปแบบข้อมูล (data type) และตัวเชื่อมแต่ละอันแทนความสัมพันธ์ระหว่างระเบียนต่างๆ แต่ละโน้ดสามารถมีตัวเชื่อมได้มากกว่าหนึ่งตัว ระบบจัดการฐานข้อมูลที่เป็นแบบนี้ เช่น DBTG

ข้อดี : เหมาะสมกับระบบที่มีความสัมพันธ์ของข้อมูลเป็นแบบ (M:N) ซึ่งระบบโดยทั่วไปจะมีความสัมพันธ์แบบนี้มาก

ข้อเสีย : ค่อนข้างซับซ้อน ผู้เขียนคำสั่งหรือโปรแกรมเพื่อดึงข้อมูลที่ต้องการจะต้องเข้าใจและรู้ตำแหน่งของข้อมูล จึงยุ่งยากกว่าแบบอื่น

ระบบจัดการฐานข้อมูลทั้ง 3 แบบที่กล่าวมาแล้วนี้จะมีวิธีในการเก็บข้อมูล ประโยคดีดีแอล(DDL) และ ดีเอมแอล(DML) ที่แตกต่างกันออกไป และจะมีโมเดลข้อมูลของแต่ละแบบโดยเฉพาะ ซึ่งเราจะต้องนำโมเดลที่ได้จากข้อ 3.2 มาแปลงให้อยู่ในโมเดลแต่ละชนิดในขั้นต่อไป

3.3.2. การพิจารณาในทางเศรษฐกิจ เป็นการพิจารณาถึงค่าใช้จ่ายที่ต้องใช้เมื่อเลือกระบบจัดการฐานข้อมูลชนิดนั้นๆ ค่าใช้จ่ายต่างๆได้แก่

ก. ราคาของซอฟต์แวร์ การเลือกต้องพิจารณาหน้าจอ และวิธีการดึงข้อมูล การทำแฟ้มข้อมูลสำรองต่างๆ ประกอบกับการพิจารณาราคา

ข. ค่าบำรุงรักษาระบบ

ค. ราคาของฮาร์ดแวร์ ต้องพิจารณาร่วมกับชนิดของเทอร์มินัล หน่วยความจำหลัก หน่วยความจำสำรองที่จะได้มา

ง. ค่าใช้จ่ายในการสร้างระบบใหม่และการเปลี่ยนจากระบบเก่าเป็นระบบใหม่

จ. ค่าจ้างบุคคลากร รวมถึงค่าฝึกหัดต่างๆ

3.4. การแปลงให้อยู่ในโมเดลของระบบจัดการฐานข้อมูลแต่ละแบบ

เป็นการสร้างแบบแผนฐานข้อมูลเชิงมโนภาพในรูปโมเดลของระบบจัดการฐานข้อมูลที่เลือกแล้ว โดยแปลงจากโมเดลที่ได้ในข้อ 3.2 จะแบ่งทำเป็น 2 ตอน คือ

ก. แปลงโดยเป็นอิสระจากชนิดของระบบจัดการฐานข้อมูล คือทำการแปลงให้อยู่ในแบบที่เลือก แต่ไม่สนใจข้อจำกัดพิเศษของชนิดที่เลือก

ข. ปรับปรุงให้อยู่ใช้กับระบบจัดการฐานข้อมูลชนิดที่เลือก

ผลที่ได้จากช่วงนี้คือ ประโยคดีดีแอลของระบบจัดการฐานข้อมูลที่เลือกแล้ว

3.5. การออกแบบฐานข้อมูลทางกายภาพ

เป็นกระบวนการในการเลือกโครงสร้างในการเก็บข้อมูล และทิศทางการเข้าหาข้อมูล สำหรับแฟ้มข้อมูลของฐานข้อมูล เพื่อให้ได้มาซึ่งฐานข้อมูลที่มีประสิทธิภาพมากที่สุด แต่ละระบบจัดการฐานข้อมูลจะมีระบบจัดการแฟ้มข้อมูล ,ทิศทางการเข้าถึงข้อมูล ,ดัชนีตัวชี้ ฯลฯ ที่ต่างกันออกไป ซึ่งจะเป็นตัวจำกัดให้เราเลือกสิ่งที่เหมาะสมให้กับแฟ้มข้อมูลของเรา สิ่งที่เราจะพิจารณาเพื่อการออกแบบทางกายภาพ คือ

ก. เวลาในการตอบสนอง (response time) คือช่วงเวลาตั้งแต่การส่งงานเข้าไปถึงเมื่อได้รับผลลัพธ์ที่ต้องการออกมา

ข. การใช้ที่ว่าง (Space utilization) จำนวนที่ว่างของหน่วยเก็บที่จะถูกใช้โดยแฟ้มข้อมูลของฐานข้อมูลและโครงสร้างทิศทางการเข้าถึงข้อมูล

ค. งานที่ได้ออกมา (transaction throughput) จะคิดเป็นค่าเฉลี่ย คำนวณจากจำนวนงานที่สามารถประมวลผลได้โดยระบบจัดการฐานข้อมูลต่อหนึ่งหน่วยเวลา

ผลที่ได้คือ การตกลงใจเบื้องต้นเกี่ยวกับโครงสร้างในการเก็บและวิธีการเข้าถึงข้อมูลของแฟ้มข้อมูลของฐานข้อมูล

4. ความปลอดภัยของฐานข้อมูล (Database Security)

ฐานข้อมูลนั้นก็เป็นที่ทราบกันดีอยู่แล้วว่า เป็นที่เก็บข้อมูลที่มีค่าจำนวนมาก ดังนั้นจึงต้องมีการรักษาความปลอดภัยของข้อมูลเหล่านั้น เพื่อให้บรรลุเป้าหมายคือ

- ป้องกันและขัดขวาง ไม่ให้ผู้ไม่มีสิทธิเข้าไปดึง ,แก้ไข ,เปลี่ยนแปลงข้อมูลหรือปรับโครงสร้างของฐานข้อมูล
- กีดขวางไม่ให้ผู้มีสิทธิขยายวงในการสืบค้นหรือทำความเข้าใจกับข้อมูลในส่วนที่เขาไม่มีสิทธิ
- ทำให้เงินลงทุนและอัตราการเสี่ยงของผู้ที่บุกรุกสูงกว่าประโยชน์ที่จะได้

ซึ่งในการออกแบบผู้ออกแบบจะต้องพิจารณาในเรื่องเหล่านี้ด้วย ผู้บุกรุกนั้นเป็นได้ทั้งบุคคลภายในและภายนอกหน่วยงานโดยสาเหตุของการบุกรุกมีได้ทั้งอุบัติเหตุและโดยตั้งใจ

โดยอุบัติเหตุ เช่น

- เกิดจากข้อจำกัดของการกำหนดสิทธิในการใช้ข้อมูลของระบบจัดการฐานข้อมูลหรือระบบปฏิบัติการ ทำให้ผู้ใช้ที่อยู่ระดับเดียวกับผู้มีสิทธิเข้าไปดึงข้อมูลนั้นโดยไม่ตั้งใจ
- การส่งข่าวสารถึงผู้ใช้ผิดคน ทำให้ผู้ไม่มีสิทธิรู้ข้อมูลนั้น
- ความผิดพลาดของระบบสื่อสารที่มีผลไปเชื่อมผู้ใช้เข้าไปในเครือข่ายของผู้อื่น
- ความผิดพลาดของระบบปฏิบัติการ ทำให้เกิดการเขียนทับหรือส่งแฟ้มข้อมูลผิดให้กับผู้ใช้คนอื่น

โดยตั้งใจ เช่น

- การลอบต่อสายในการสื่อสาร
- การใช้สัญญาณอิเลคโทรนิคเพื่อดึงสัญญาณจากเครื่องอื่น
- แอบดูข้อมูลจากหน้าจอหรือสิ่งพิมพ์ของผู้ใช้อื่นที่มีสิทธิ
- การดึงข้อมูลจากผู้มีสิทธิในระดับต่ำกว่า
- การขโมยสิ่งที่ใช้บันทึกข้อมูล เช่น เทป ออกจากห้องคอมพิวเตอร์

การรักษาความปลอดภัยแบ่งได้เป็นทางกายภาพและทางซอฟแวร์

4.1. ทางกายภาพ ได้แก่การติดตั้งระบบรักษาความปลอดภัยต่างๆ ก่อนที่ผู้บุกรุกจะเข้ามาเปิดเครื่องเราได้ เช่น การมียามรักษาการณ์ มีกุญแจล็อจเครื่อง ให้เซ็นชื่อหรือพิมพ์ลายนิ้วมือ ซึ่งสิ่งเหล่านี้จะอยู่นอกเหนือขอบเขตของผู้ออกแบบ แต่ผู้ออกแบบก็ควรได้แนะนำไว้

4.2. ทางซอฟแวร์ การทำไว้ในโปรแกรม มีกลวิธีต่างๆมากมาย เช่น

ก. การให้อำนาจ (Authorization) คือการตรวจสอบหลักฐานผู้ใช้ว่าเป็นตัวจริงที่มีสิทธิมาร้องขอข้อมูลหรือไม่ มักจะคิดตั้งการรักษาความปลอดภัยไว้ที่ระดับระบบปฏิบัติการ วิธีที่นิยมมากที่สุดคือให้ใส่รหัสผ่าน (password) แต่ถึงอย่างไรก็ไม่ปลอดภัย นักบางแห่งอาจมีการเพิ่มเติมให้ใช้บัตรหรือกุญแจเฉพาะเมื่อจะเข้าสู่ระบบ (login) ในระบบจัดการฐานข้อมูลส่วนมากจะมีระบบการรักษาความปลอดภัยนี้ให้สำหรับผู้ใช้แต่ละคน เพื่อกำหนดผู้มีสิทธิใช้ข้อมูลของตนได้เองเรียกว่า ภาษาในการให้อำนาจ(authorization language) เช่น ใน ORACLE มีคำสั่ง grant มีรูปแบบคือ

```
GRANT { privilege,privilege,...|ALL} ON table
      TO {user,user,...|PUBLIC} [WITH GRANT OPTION];
```

ซึ่งเป็นการให้อำนาจในการกระทำกับตารางนั้นกับผู้ใช้อื่นๆ

ข. การควบคุมการเข้าถึง (Access Control) คือการทำให้แน่ใจว่าข้อมูลหรือสิ่งต่างๆในระบบจะถูกเข้าถึงได้เฉพาะทิศทางที่ต้องผ่านการตรวจสอบการให้อำนาจเท่านั้น ซึ่งวิธีนี้จะทำได้เมื่อมีการติดตั้งการให้อำนาจแล้วเท่านั้น เราอาจทำโดยสร้างเป็นตารางการเข้าถึงข้อมูล (access control matrix) เพื่อให้ง่ายต่อการควบคุม โดยแต่ละคอลัมภ์แทนเป้าหมายในระบบ เช่น ชื่อตารางหรือวิว ส่วนแถวต่างๆแทนรหัสประจำตัวของผู้ใช้หรือกลุ่มผู้ใช้ และค่าในแต่ละช่องที่สัมพันธ์กันก็คืออำนาจในการเข้าถึงที่กำหนดให้ผู้ใช้สำหรับตารางหรือวิวนั้นๆ ได้แก่ อำนาจการลบ เพิ่ม เปลี่ยนแปลง อ่านข้อมูล

ค. วิว การสร้างวิวในระบบจัดการฐานข้อมูลโดยทั่วไปจะมีคำสั่งที่ใช้ในการสร้างวิวจากตารางที่มีในระบบ เพื่อให้ได้ข้อมูลเฉพาะในส่วนที่ผู้ใช้จะมีสิทธิ โดยผู้จัดการฐานข้อมูลจะเป็นคนทำและให้สิทธิในการเข้าถึงวิวนั้นๆให้แก่ผู้ใช้อื่นๆ

ง. การเข้ารหัส (Data Encryption) เป็นการเก็บข้อมูลที่จะใช้ระบบที่เรียกว่า cipher system ที่ประกอบไปด้วยการเข้ารหัสและการถอดรหัส โดยเราจะนำข้อมูลในรูปแบบปกติ (plain text) มาผ่านอัลกอริทึมส์ในการเข้ารหัสซึ่งจะมีการกำหนดคีย์ในการเข้ารหัสด้วย จึงจะได้ข้อมูลในรูปแบบที่เรียกว่าไซเฟอร์เทกซ์ (cipher text) และเราจะเก็บข้อมูลไว้ในระบบในรูปแบบนี้ เวลาจะต้องใช้ก็จะนำไปผ่านอัลกอริทึมส์ในการถอดรหัสซึ่งก็จะมีคีย์สำหรับการถอดรหัสเช่นกัน ดังนั้นถ้ามีผู้บุกรุกมาขโมยข้อมูลไปก็จะใช้ไม่ได้ทันทีต้องทำการถอดรหัสก่อนซึ่งยุ่งยากและอาจเสียเวลามาก อัลกอริทึมส์ในการเข้ารหัสที่เป็นที่นิยมและตั้งเป็นมาตรฐานไว้คือ ดีอีเอส (DES:Data Encrption Standard) ซึ่งสามารถสร้างไซเฟอร์เทกซ์ได้แตกต่างกันมากกว่า 2^{56} แบบ

จ. การลงบันทึกความปลอดภัย (Security logs) เป็นการบันทึกเกี่ยวกับผู้ที่ตั้งใจจะฝ่าฝืนเข้าสู่ระบบ โดยจะบันทึกข้อมูลเหล่านั้นไว้ในแฟ้มลงบันทึก หรือมีการส่งข่าวไปให้ผู้ควบคุมเครื่องหรือผู้บริหารฐานข้อมูลทราบ

ฉ. การทำหลักฐานการตรวจสอบ (Audit Trail) เป็นการเก็บข่าวสารเกี่ยวกับว่าใครมาดึงข้อมูล มาใช้เครื่องเทอร์มินอล เมื่อเวลาใด

โมเดลข้อมูลเชิงตรรก (Logical Data Model)

เป็นโมเดลที่ใช้ในขั้นการออกแบบฐานข้อมูลเชิงมโนภาพ(ข้อ 3.2) โดยจะแสดงข้อมูลในขอบเขตที่ผู้ออกแบบสนใจโดยมีสิ่งที่ต้องกำหนดเป็นพื้นฐานได้แก่ เอนติตี, รีเลชันชิป, แอตตริบิว

เอนติตี คือ สิ่งที่มีอยู่จริง จับต้องได้ หรือเป็นจินตภาพที่แสดงความเป็นหนึ่งเดียว เมื่อกล่าวถึงแล้วทุกคนเข้าใจตรงกัน เช่น นักศึกษา, ชั้นเรียน

รีเลชันชิป คือ ความสัมพันธ์ระหว่างเอนติตีใดเอนติตีหนึ่งกับตัวมันเองหรือเอนติตีอื่น อาจเป็นความสัมพันธ์ที่มากกว่า 2 เอนติตีก็ได้ เช่น อาจารย์ที่ปรึกษาของนักศึกษา

แอตตริบิว คือ กลุ่มของค่าความจริงใดๆ ที่เป็นรายละเอียดของเอนติตี ทำให้เข้าใจเอนติตีได้ลึกซึ้งยิ่งขึ้น และเป็นสิ่งที่ไม่สามารถแตกย่อยลงไปได้อีกโดยไม่เสียความหมายไป เช่น บ้านเลขที่ของนักศึกษา

นอกจากนั้นยังมีการระบุด้วยว่าแอตตริบิวใดเป็นคีย์ กำหนดกฎข้อบังคับต่างๆของเอนติตีและรีเลชันชิป

1. คุณลักษณะของโมเดลข้อมูลทางตรรก

1.1. แสดงได้ด้วยแผนภาพ (Graphical Diagrams) ไม่ว่าจะเป็นเทคนิคโมเดลข้อมูลแบบใดก็ตามจะมีภาษาและรูปภาพทางกราฟฟิกของมันโดยเฉพาะ เพื่อใช้แสดงรายละเอียดข้อมูลทั้งกลุ่มใหญ่และรายละเอียดส่วนย่อย ซึ่งทำให้ง่ายต่อการแปลความ เช่น ใช้วงกลม

หรือสี่เหลี่ยมแทนเอนติตี ใช้เส้นโค้งหรือเส้นตรงแทนรีเลชันชิป

1.2. การแสดงชัดเจนถึงความหมายของข้อมูล (Explicit representation of semantic) มีทางเลือกในการแสดงความหมายของข้อมูล เราอาจใช้สัญญลักษณ์ที่ต่างกันจำนวนมากบ้างน้อยบ้างเพื่อแสดง แต่จุดสำคัญคือแผนภาพที่ได้ออกมาควรง่าย ไม่ซับซ้อน และเห็นความหมายของข้อมูลชัดเจน โดยเฉพาะอย่างยิ่งสัญญลักษณ์หนึ่งๆไม่ควรมีหลายความหมาย

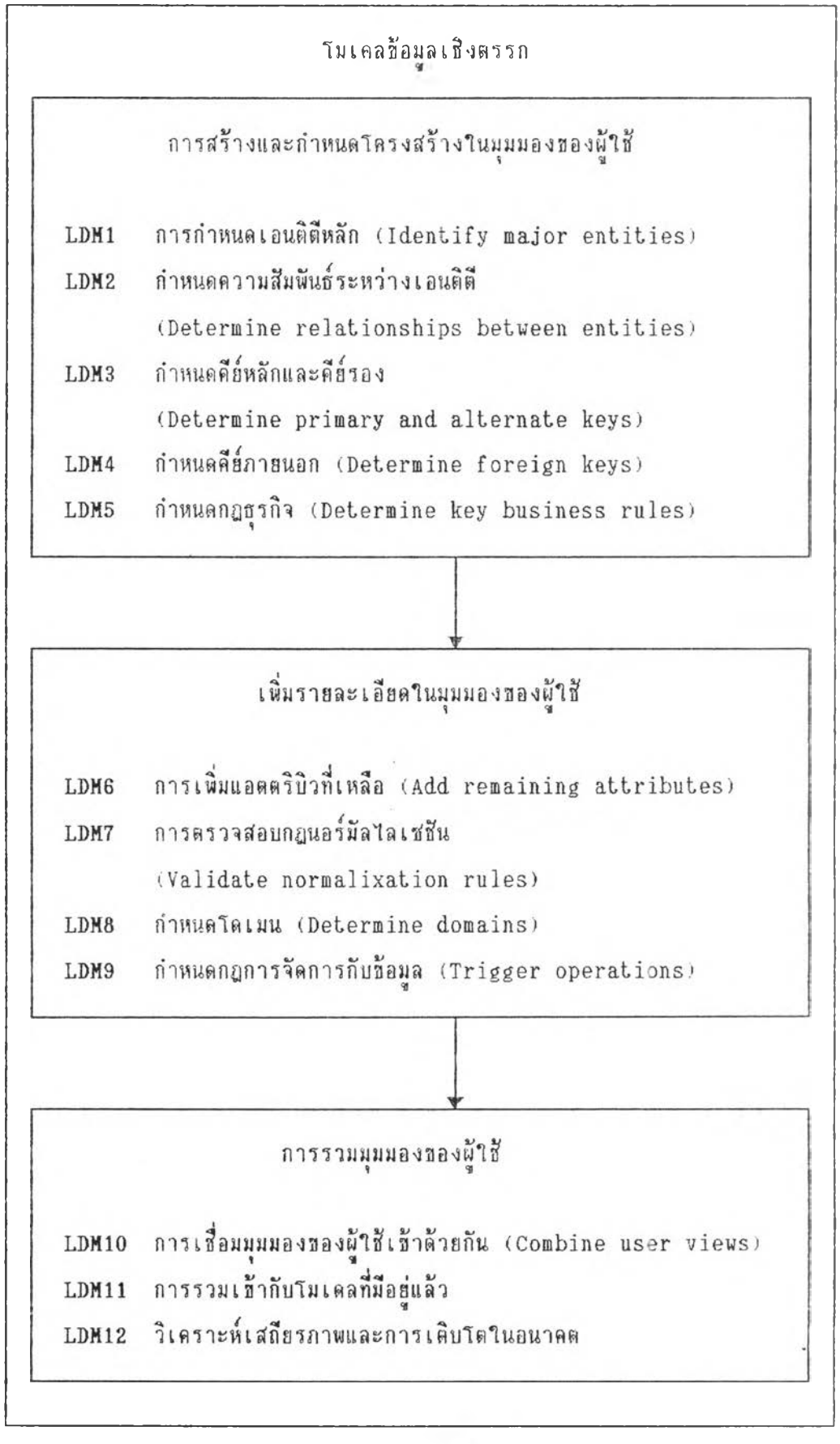
1.3. แสดงรายละเอียดในระดับที่เหมาะสม (Appropriate level of Detail) กล่าวคือโมเดลระดับตรรกจะมีรายละเอียดที่เพียงพอที่จะชี้จุดที่ทำให้เกิดความแตกต่างระหว่างชนิดของข้อมูล รีเลชันชิปและข้อบังคับต่างๆ แต่จะน้อยกว่าโมเดลทางกายภาพ

1.4. เป็นอิสระจากระบบจัดการฐานข้อมูล (DBMS independence) โมเดลที่ได้จากการออกแบบแล้วควรใช้ได้กับระบบฐานข้อมูลหลายแบบ ได้แก่ แบบความสัมพันธ์, แบบลำดับชั้นและแบบเครือข่าย

1.5. ง่ายต่อการศึกษาและใช้งาน (Easy to Learn and Use) ในที่นี้จะต้องง่ายเพียงพอสำหรับผู้ใช้ทุกประเภทจะทำความเข้าใจและนำไปใช้ได้

2. ขั้นตอนการออกแบบโมเดลข้อมูลเชิงตรรก

ในการออกแบบโมเดลข้อมูลเชิงตรรกมีด้วยกันหลายขั้นตอนด้วยกัน สำหรับใน 5 ขั้นตอนแรกจะเป็นการออกแบบทางด้านโครงสร้างพื้นฐานของโมเดล ได้แก่พวก เอนติตี , รีเลชันชิป ,คีย์หลัก ,คีย์สำรอง ,คีย์ภายนอก ,กฎเกณฑ์พื้นฐาน จากนั้นจึงจะเริ่มเพิ่มรายละเอียดในระดับที่ผู้ใช้มองเห็น(user view) และรวมรายละเอียดเหล่านั้นเข้าด้วยกัน จึงจะได้เป็นโมเดลข้อมูลเชิงตรรกที่สมบูรณ์ ขั้นตอนต่างๆสรุปได้ดังรูปที่ 3.10



รูปที่ 3.10 ขั้นตอนการออกแบบโมเดลข้อมูลเชิงตรรก

ขั้นตอนที่ 1 การกำหนดเอนติตีหลัก

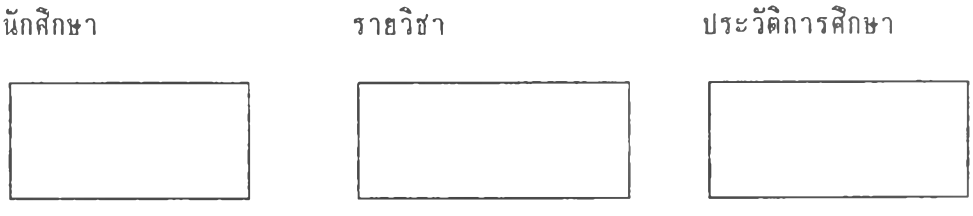
โดยกำหนดชื่อและความหมาย ลงในพจนานุกรมข้อมูล และเขียนลงโมเดลข้อมูลด้วย การตั้งชื่อไม่ควรเกิน 20 ตัวอักษร ดังแสดงในรูปที่ 3.11

การกำหนดเอนติตีนั้นเป็นงานที่ยาก และต้องอาศัยความร่วมมือของผู้ที่เข้าใจระบบที่เราออกแบบ เพื่อคัดเลือกสิ่งที่ถูกต้อง มีความสำคัญและเหมาะสมที่สุดมาเป็นเอนติตี วิธีการอย่างคร่าวๆ ก็คือให้พิจารณาข้อมูลทั้งหมดที่มี และจัดกลุ่มของข้อมูลโดยดูจากค่าและความหมาย ถ้าสามารถรวมกลุ่มกันได้ก็ให้รวมเข้าไว้ในเอนติตีเดียวกัน นอกจากนั้นเรายังสามารถแยกเอนติตีย่อยลงไปอีกได้ โดยเรียกว่าเป็นซับไทป์ (subtype) ของอีกเอนติตีหนึ่ง เช่น ถ้าบอกว่า X เป็นซับไทป์ของ Y จะสังเกตได้ดังนี้

ก. เอนติตี X และเอนติตี Y ต้องแทนสิ่งเดียวกัน

ข. X จะประกอบด้วยสิ่งต่างๆใน Y และส่วนที่เป็นของมันเองด้วย

ค. ทุกๆ X จะประกฏเพียงหนึ่งใน Y (1:1)



รูปที่ 3.11 ตัวอย่างเอนติตีหลัก

ขั้นตอนที่ 2 การกำหนดรีเลชันชิประหว่างเอนติตี

ให้กำหนดชื่อ ความหมาย ทิศทาง และขนาดอัตราส่วนที่เกิดรีเลชันชิปนั้นๆ พร้อมทั้งบันทึกลงพจนานุกรมข้อมูลด้วย สำหรับชื่อก็เช่นเดียวกับชื่อเอนติตีคือไม่ควรเกิน 20 ตัวอักษร สำหรับความหมายถ้าเป็นภาษาอังกฤษให้ใช้คำกริยาในรูปแบบปัจจุบัน (present tense) รีเลชันชิปที่เกิดขึ้นเป็นไปได้ 3 อย่างด้วยกัน คือ

1. รีเลชันชิปที่เป็นเหตุการณ์ที่มีอยู่จริง เช่น ประวัติการศึกษาของนักศึกษา

2. รีเลชันชิปที่เป็นหน้าที่ เช่น นักศึกษาต้องเรียนรายวิชา

3. รีเลชันชิปจากเหตุการณ์ เช่น นักศึกษาลงทะเบียนเรียน

อัตราส่วนและทิศทางของรีเลชันชิปจะเป็นพื้นฐานในการแบ่งประเภทของความสัมพันธ์ สำหรับรีเลชันชิประหว่าง 2 เอนติตี (binary relationship) จะมีด้วยกัน 3 ชนิด คือ

1. หนึ่งต่อหนึ่ง (one-to-one, 1:1 relationship) รีเลชันชิปที่เกิดขึ้นนี้จะมีมากที่สุดแค่หนึ่งเท่านั้น เช่น คณะหนึ่งจะมีคณบดีเพียงคนเดียว และในทางกลับกันอาจารย์

ท่านหนึ่งก็เป็นคณบดีของคณะได้คณะเดียวเท่านั้น ดังนั้นรีเลชันชิปแบบนี้ทำให้เราต้องตัดสินใจว่าจะให้หัวลูกศรไปทางใด ซึ่งเมื่อได้แล้วเอนติตีใดที่หัวลูกศรชี้เข้าหาจะเรียกเอนติตีลูก (child entity) อีกอันก็จะเป็นเอนติตีแม่ (parent entity) ใช้สัญญลักษณ์ (P) ---------> (C) แทน ดังรูปที่ 3.12

รูปที่ 3.12 รีเลชันชิปแบบหนึ่งต่อหนึ่ง

2. หนึ่งต่อหลาย (one-to-many, 1:N relationship) รีเลชันชิปจะเกิดขึ้นได้ตั้งแต่ 0 ครั้งจนถึงหลายๆครั้ง โดยแต่ละอันของเอนติตีลูกจะสัมพันธ์กับเอนติตีแม่ได้หนึ่งค่าเท่านั้น แต่หนึ่งค่าของเอนติตีแม่สัมพันธ์กับเอนติตีลูกได้หลายค่า ใช้สัญญลักษณ์ (P) -------->> (C) แทน เช่น อาจารย์ท่านหนึ่งมีลูกศิษย์ในที่ปรึกษาได้หลายคน แต่นักศึกษาจะมีอาจารย์ที่ปรึกษาได้คนเดียวเท่านั้น ดังรูปที่ 3.13

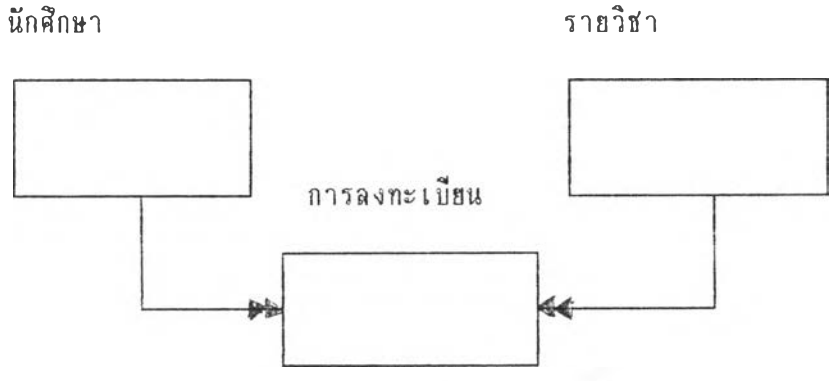
รูปที่ 3.13 รีเลชันชิปแบบหนึ่งต่อหลาย

3. หลายต่อหลาย (many-to-many, M:N relationship) รีเลชันชิปที่เกิดขึ้นได้ตั้งแต่ 0 ครั้งจนถึงหลายๆครั้งในทั้งสองทิศทาง กล่าวคือแต่ละเอนติตีลูกจะสัมพันธ์กับเอนติตีแม่ได้หลายค่า และเอนติตีแม่ก็สัมพันธ์กับเอนติตีลูกได้หลายค่าเช่นกัน ใช้สัญญลักษณ์ (P) <<------>> (C) แทน เช่น นักเรียนแต่ละคนเลือกรายวิชาเรียนได้หลายวิชาและวิชาแต่ละวิชาจะถูกเลือกเรียนโดยนักเรียนได้หลายคน ดังรูปที่ 3.14

หลังจากเราสามารถแบ่งกลุ่มรีเลชันชิประหว่างเอนติตีได้เรียบร้อยแล้ว ความสัมพันธ์แบบ (M:N) เป็นสิ่งที่เราต้องสนใจมากที่สุดเพราะเป็นตัวยุ่งยาก โดยปกติฐานข้อมูลทั่วไปจะไม่สามารถจัดการกับรีเลชันชิปแบบกลุ่มนี้ได้ ดังนั้นเราจะแตกมันออกมาเป็นรีเลชันชิปแบบ (1:N) สองรีเลชันชิป โดยการกำหนดเอนติตีใหม่ขึ้นมาอีกหนึ่งตัวให้มีรีเลชันชิปกับเอนติตี

เดิมแบบ (1:N) เช่น รีเลชันชิประหว่างนักศึกษากับรายวิชาแตกได้โดยเพิ่มเอนติตีการลงทะเบียนขึ้นมา ดังรูปที่ 3.15 จากนั้นให้ใส่ชื่อ ความหมายและรายละเอียดของเอนติตีและรีเลชันชิปที่เกิดขึ้นใหม่นี้ลงในพจนานุกรมข้อมูลด้วย

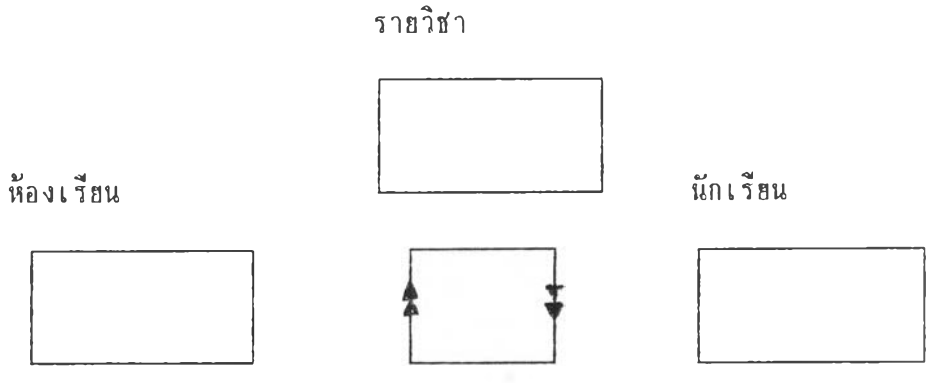
รูปที่ 3.14 รีเลชันชิปแบบหลายต่อหลาย



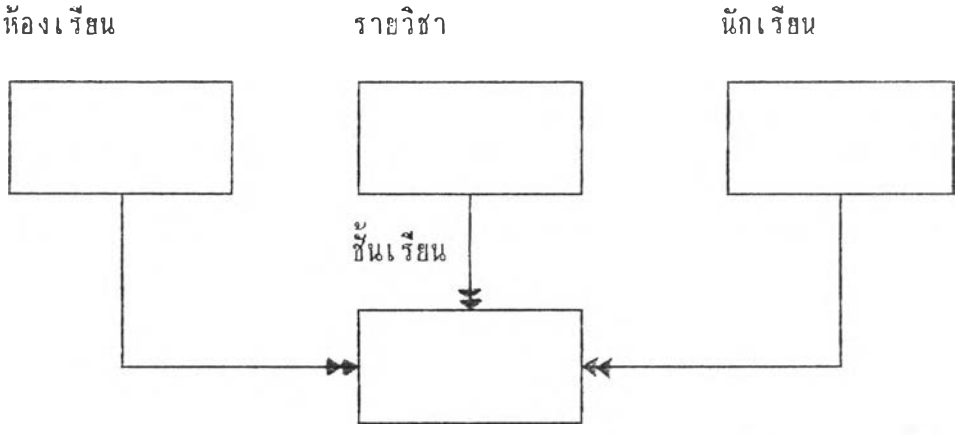
รูปที่ 3.15 รีเลชันชิปแบบ(M:N) ที่ถูกกำหนดขึ้นเป็นเอนติตีใหม่

นอกจากรีเลชันชิประหว่างเอนติตีแบบที่กล่าวมาแล้วนี้ยังมีรีเลชันชิปแบบพิเศษอื่นๆอีก ซึ่งเราจะต้องหาทางจัดการทำให้ง่ายขึ้น ดังนี้

1. รีเลชันชิปแบบซับซ้อน (complex) คือ รีเลชันชิปที่เกิดขึ้นจากเอนติตีตั้งแต่สามตัวมาสัมพันธ์กัน(รูปที่ 3.16 (ก)) เราจะทำให้ง่ายขึ้นโดยสร้างรีเลชันชิปขึ้นเป็นเอนติตีใหม่ ซึ่งจะมีรีเลชันชิประหว่างสองเอนติตีกับเอนติตีเดิม ดังรูปที่ 3.16 (ข)

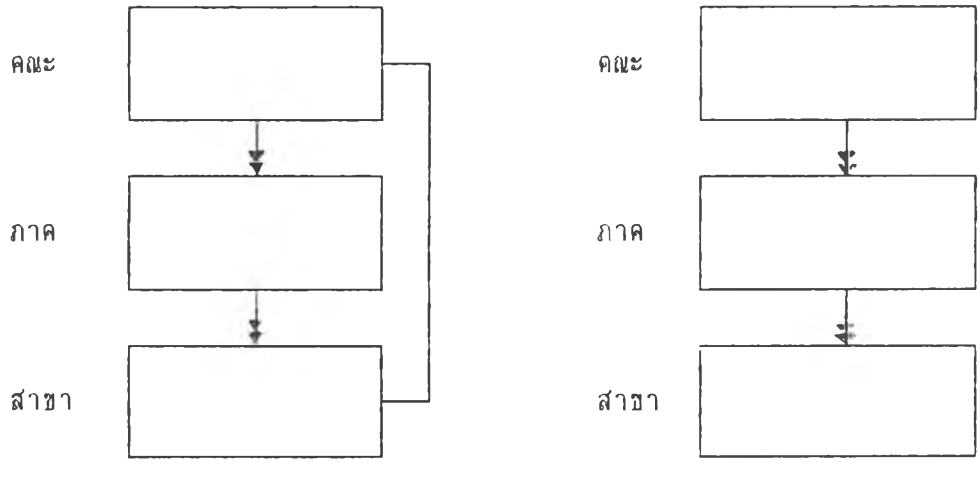


รูปที่ 3.16 (ก) รีเลชันชิปแบบซับซ้อน



รูปที่ 3.16 (ข) รีเลชันชิประหว่างเอนติตีเดิมกับเอนติตีชั้นเรียนที่เพิ่มขึ้น

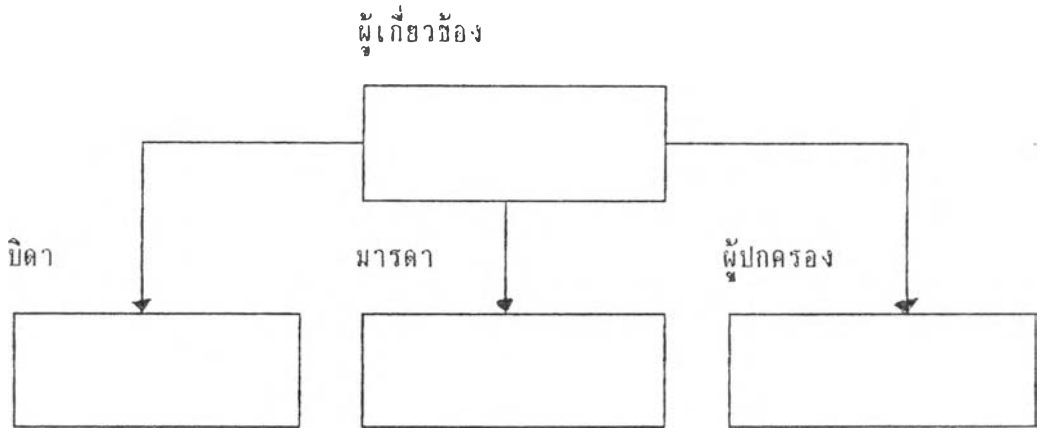
2. รีเลชันชิปแบบเหลือเฟือ (Potentially redundant) คือรีเลชันชิปจากเอนติตีหนึ่งไปยังอีกเอนติตีหนึ่งที่ให้ความหมายได้โดยเส้นทางอื่นที่ตั้งต้นจากเอนติตีเดียวกัน ซึ่งในกรณีนี้เราควรตัดมันทิ้งเสีย เหตุผลก็เนื่องจากรีเลชันชิปแบบนี้ ทำให้โมเดลเราดูยุ่งยาก ทั้งที่ไม่จำเป็น นำไปวิเคราะห์ได้ยากและเมื่อมีรายละเอียดมาเพิ่มในโมเดลใหม่ จะสับสนได้ง่าย



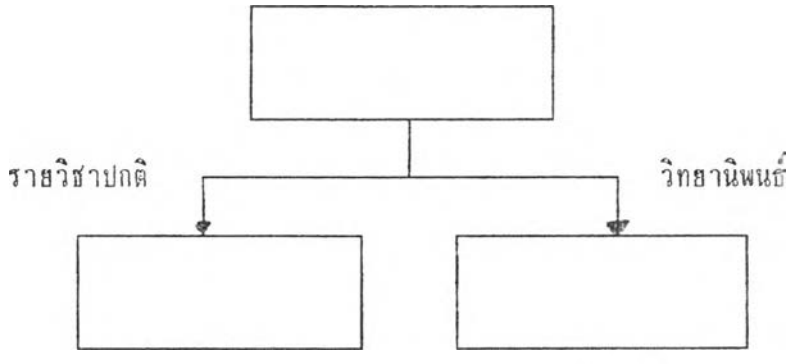
(ก) (ข)

รูปที่ 3.17 (ก) รีเลชันชิปแบบเหลือเฟือที่ถูกตัดออกได้เป็นรูป (ข)

3. รีเลชันชิปแบบลำดับชั้น (Categories or Subtype-supertype) คือมีเอนติตีบางอันที่มีกลุ่มข้อมูลที่เป็นซับไทป์ของตัวเอง ซึ่งมีรายละเอียดโดยพื้นฐานเหมือนกันทุกอย่าง แต่จะเพิ่มขึ้นมาบางอันที่เฉพาะเจาะจงลงไปกว่าเดิม เราจึงต้องสร้างรีเลชันชิปแบบ (1:1) ขึ้นมาระหว่างซูเปอร์ไทป์ (supertype) และซับไทป์ ในกรณีซูเปอร์ไทป์หนึ่งระเบียนสัมพันธ์กับซับไทป์ได้เพียงหนึ่งอันจะเรียกว่าเป็นซูเปอร์ไทป์-แคทิกอรี่

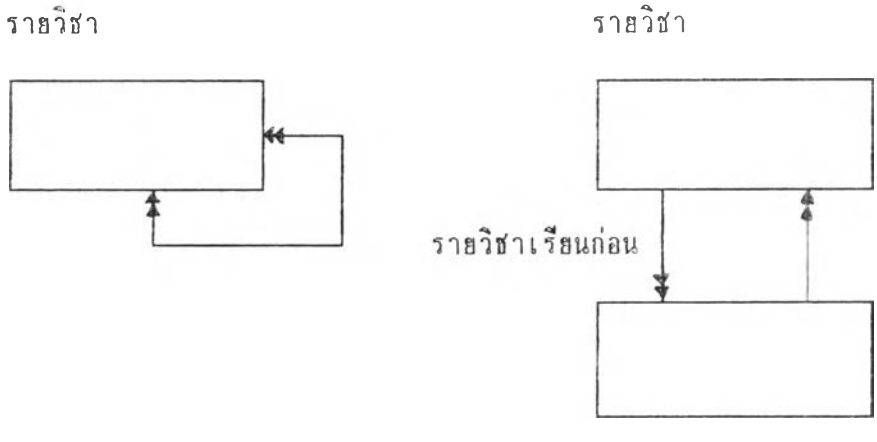


รูปที่ 3.18 (ก) รีเลชันชิปแบบซูเปอร์ไทป์-ซับไทป์



รูปที่ 3.18 (ข) รีเลชันชิปแบบซูเปอร์ไทป์-แคทิกอรี่

4. รีเลชันชิปกับตัวเอง (Bill-of-materials) คือรีเลชันชิปที่เกิดจากเอนติตีมาสัมพันธ์กับตัวมันเองอาจเป็นทั้งแบบ (1:N) หรือ (M:N) ในกรณีเป็น (1:N) ก็คงไว้แบบเดิม แต่ถ้าเป็น (M:N) ต้องทำการกำหนดเอนติตีขึ้นมาใหม่ให้สัมพันธ์กับเอนติตีเดิมแบบ (1:N) สองรีเลชันชิป

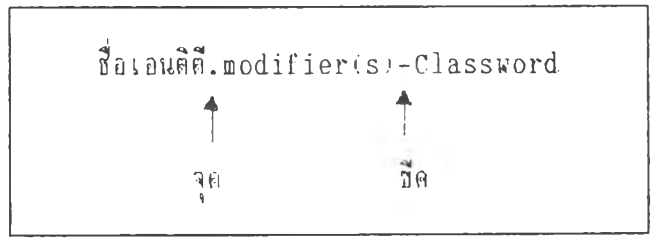


รูปที่ 3.19 รีเลชันชิปกับตัวเอง

ขั้นตอนที่ 3 การกำหนดคีย์หลักและคีย์รอง

คีย์หลักและคีย์รองจะเป็นแอตตริบิวตัวแรกที่เราจะกำหนดในเอนติตี โดยให้กำหนดคีย์หลักของทุกเอนติตีโดยเลือกจากแอตตริบิวที่เป็นคีย์เป็นไปได้ และให้ระบุคีย์รองของทุกเอนติตีด้วย โดยมากก็คือคีย์เป็นไปได้ที่ไม่ได้ถูกเลือกเป็นคีย์หลักนั่นเอง ในกรณีที่คีย์หลักและคีย์รองเป็นคีย์ประกอบ แอตตริบิวหนึ่งอาจเป็นส่วนของคีย์หลักและคีย์รองได้มากกว่าหนึ่งคีย์ สิ่งที่สำคัญอีกอันคือเอนติตีที่เป็นซับไทป์จะต้องมีคีย์หลักอันเดียวกับเอนติตีที่เป็นซูเปอร์ไทป์ของมัน หลังจากกำหนดแล้วให้ตั้งชื่อระบุในโมเดลข้อมูลเชิงตรรกพร้อมทั้งใส่ในพจนานุกรมข้อมูลด้วย

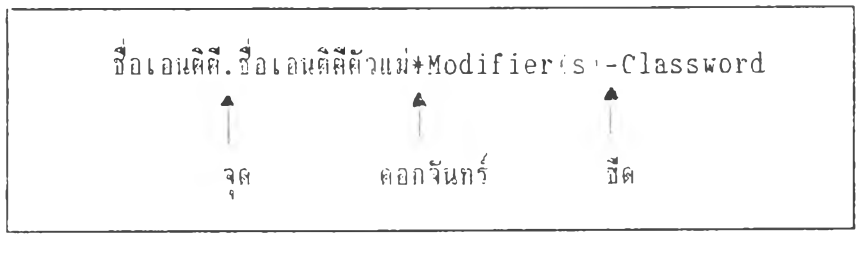
การตั้งชื่อควรกำหนดสั้นๆ ง่ายๆ อาจใช้ชื่อย่อได้ และควรหลีกเลี่ยงการตั้งชื่อแอตตริบิวของสองสิ่งที่ไม่เหมือนกันด้วยชื่อเดียวกัน รูปแบบการกำหนดอาจทำได้ดังนี้



เช่น Employee.last-name

ขั้นตอนที่ 4 การกำหนดคีย์ภายนอก

ให้กำหนดคีย์ภายนอกสำหรับเอนติตีที่มีรีเลชันชิปกันทุกอัน โดยตั้งชื่อและใส่ลงในโมเดลข้อมูลเชิงตรรก พร้อมกับเก็บในพจนานุกรมข้อมูลด้วย คีย์ภายนอกจะถูกกำหนดลงในเอนติตีตัวลูกและมีค่าเท่ากันกับคีย์หลักของเอนติตีตัวแม่ มีกฎการตั้งชื่อแอตตริบิวที่เป็นคีย์ภายนอกดังนี้



เช่น student.race*code

ความสำคัญของคีย์ภายนอก คือ

1. ทำให้เกิดกฎธุรกิจ (Business rules) ท่ามกลางเอนติตีต่างๆ
2. ถึงแม้จะดูเหมือนว่าการกำหนดคีย์ภายนอกเป็นการทำให้ซ้ำซ้อน แต่จริงๆ แล้วมันจะเป็นตัวช่วยในการออกแบบ เพื่อให้เราสามารถตรวจสอบได้ว่าเอนติตีอันไหนเป็นเอนติตีลูกอันไหนเป็นเอนติตีแม่

3. ทำให้ขั้นตอนการออกแบบฐานข้อมูลง่ายขึ้น

แต่ข้อควรระวังคือจะกำหนดให้คีย์รองเป็นคีย์ภายนอกด้วยนั้นไม่ควร เพราะจะทำให้ซับซ้อนและอาจมีค่าเป็นนัลได้ ซึ่งคีย์รองไม่ควรเป็นเช่นนั้น

ข้อสังเกต

1. ในโมเดลข้อมูลเราจะกำหนดคีย์หลักไว้ในส่วนบนเส้นแบ่งภายในเอนติตีและกำหนดคีย์รองและคีย์ภายนอกไว้ใต้เส้นแบ่ง ยกเว้นคีย์ภายนอกที่เป็นส่วนของคีย์หลัก

2. ความสัมพันธ์แบบ (1:1) จะบรรจุคีย์ภายนอกไว้ในเอนติตีลูกเท่านั้น

3. ในกรณีคีย์หลักประกอบด้วยคีย์ภายนอก เอนติตีที่เป็นซับไทป์หรือแคทิกอรี่จะมีคีย์ภายนอกเหมือนซูเปอร์ไทป์หมด

ขั้นตอนที่ 5 การกำหนดกฎธุรกิจของคีย์

เป็นขั้นตอนที่ทำขึ้นเพื่อความสมบูรณ์ของข้อมูล และความถูกต้องตรงกันของค่าของข้อมูล กฎธุรกิจแบ่งได้เป็น 3 ประเภทคือ

1. กฎธุรกิจของคีย์ (Key business rules) เป็นการกำหนดกฎเพื่อความสมบูรณ์ของรีเลชันชิป กฎต่างๆจะมีผลจากการจัดการต่างๆบนรีเลชันชิป อันได้แก่ การเพิ่ม,ลบและแก้ไขข้อมูล เช่น กำหนดผลที่จะเกิดขึ้นจากการเปลี่ยนค่าของคีย์หลัก

2. โดเมน (Domain) เป็นการกำหนดกฎเพื่อความสมบูรณ์ถูกต้องของแอตตริบิว ก็คือการกำหนดข้อบังคับและค่าที่เป็นไปได้สำหรับแอตตริบิวทั้งที่เป็นคีย์และไม่ใช่คีย์

3. กฎการจัดการ (Triggering operation) เป็นการกำหนดผลกระทบจากการเพิ่ม, ลบ, แก้ไขหรือดึงข้อมูลที่เกิดกับเอนติตีอื่นหรือแอตตริบิวอื่นภายในเอนติตีเดียวกัน

สำหรับในขั้นตอนนี้จะพิจารณาเฉพาะกฎธุรกิจของคีย์ก่อน ซึ่งการกำหนดกฎในการเพิ่มข้อมูลนั้นมักจะเป็นผลเมื่อเราทำการเพิ่มข้อมูลในเอนติตีตัวลูก สำหรับกฎในการแก้ไขและลบข้อมูลมันจะเป็นผลเมื่อกระทำกับเอนติตีตัวแม่ ดังมีรูปแบบและรายละเอียดต่างๆกัน ดังนี้

รูปแบบข้อกำหนดในการเพิ่มข้อมูล มี 6 รูปแบบด้วยกัน คือ

ก. แบบขึ้นต่อกัน (Dependent) จะอนุญาติให้เพิ่มข้อมูลในเอนติตีลูกเฉพาะที่มีความสัมพันธ์กับเอนติตีแม่เท่านั้น กล่าวคือค่าคีย์ภายนอกในเอนติตีลูก(ตัวที่อ้างอิงถึงเอนติตีแม่)จะต้องเป็นค่าที่หาได้ในเอนติตีแม่

ข. แบบอัตโนมัติ (Automatic) ในการเพิ่มข้อมูลในเอนติตีลูกถ้าหาค่าในเอนติตีแม่ที่ตรงกันไม่ได้ให้เพิ่มเข้าในเอนติตีแม่ด้วยเลยโดยอัตโนมัติ

ค. แบบกำหนดเป็นนัล (Nullify) เมื่อหาค่าในเอนติตีแม่ที่ตรงกันไม่ได้ให้กำหนดค่าคีย์ภายนอกในเอนติตีลูกเป็นนัลเสีย

ง. แบบมีค่ากำหนดไว้ (Default) จะมีการกำหนดค่าค่าหนึ่งไว้ในกรณีหาค่า

ในเอนติตีแม่ที่ตรงกันไม่ได้ ให้กำหนดคีย์ภายนอกเป็นค่าค่านั้น

จ. แบบมีธรรมเนียม (Customized) จะมีข้อกำหนดไว้ก่อน ดังนั้นจะอนุญาติให้เพิ่มข้อมูลในเอนติตีลูกเมื่อตรวจสอบแล้วตรงกับข้อกำหนดนั้นๆก่อน

ฉ. แบบไม่มีผลใดๆ (No effect) เพิ่มข้อมูลในเอนติตีลูกได้โดยไม่มีเงื่อนไขใดๆ

รูปแบบข้อกำหนดในการลบหรือแก้ไขข้อมูล มี 6 รูปแบบ คือ

ก. แบบมีข้อจำกัด จะอนุญาติให้ลบหรือแก้ไขข้อมูลในเอนติตีแม่ได้ถ้าในเอนติตีลูกไม่มีคีย์ภายนอกที่อ้างมาถึงเอนติตีแม่ระเบียนนั้น

ข. แบบเป็นชั้นๆ ถ้าในเอนติตีลูกมีการอ้างถึงเอนติตีแม่ตัวที่ต้องการจะลบหรือแก้ไขให้ลบหรือแก้ไขทั้งสองเอนติตีเลย

ค. แบบกำหนดเป็นนัล ถ้าในเอนติตีลูกมีการอ้างถึงเอนติตีแม่นั้นให้เปลี่ยนค่าคีย์ภายนอกในเอนติตีลูกเป็นนัล แล้วจึงลบหรือแก้ไขเอนติตีแม่ตามต้องการ

ง. แบบกำหนดค่า ถ้ามีการอ้างถึงเช่นเดียวกับข้อค.ให้เปลี่ยนค่าคีย์ภายนอกนั้นมีค่าตามที่กำหนดไว้

จ. แบบมีธรรมเนียม จะลบหรือแก้ไขข้อมูลในเอนติตีแม่ได้เมื่อพบเงื่อนไขถูกต้อง

ฉ. แบบไม่มีผลใดๆ สามารถลบหรือแก้ไขข้อมูลในเอนติตีแม่โดยไม่ต้องตรวจสอบใดๆเลย

ถึงแม้ว่าจะมีการกำหนดกฎธุรกิจของคีย์แบบกำหนดเป็นนัลไว้ แต่จริงๆแล้วเป็นรูปแบบที่ควรจะหลีกเลี่ยงเพราะจะเป็นปัญหาต่อไปในการทำงาน โดยเฉพาะอย่างยิ่งถ้าคีย์ภายนอกนั้นเป็นส่วนหนึ่งของคีย์หลักจะใช้แบบกำหนดเป็นนัลไม่ได้โดยเด็ดขาด นอกจากนั้นในกรณีเป็นรีเลชันชิปแบบซับไทป์-ซูเปอร์ไทป์ ถ้าจะลบข้อมูลต้องลบจากทั้งเอนติตีแม่และเอนติตีลูก เวลาจะเพิ่มข้อมูลในเอนติตีลูก(หรือเอนติตีซับไทป์)ต้องแน่ใจว่า ยังไม่มีข้อมูลระเบียนใดในเอนติตีซับไทป์มีคีย์ภายนอกที่เหมือนกันอ้างไปยังเอนติตีซูเปอร์ไทป์ เนื่องจากว่ารีเลชันชิปแบบนี้เป็นแบบ(1:1) จึงมีการเชื่อมด้วยคีย์เดียวกันได้เพียงหนึ่งระเบียนเท่านั้น

ขั้นตอนที่ 6 การเพิ่มแอตตริบิวที่เหลือ

ให้ทำการกำหนดแอตตริบิวอื่นๆในเอนติตี โดยตั้งชื่อและเพิ่มในโมเดลข้อมูลเชิงตรรกพร้อมทั้งบันทึกลงในพจนานุกรมข้อมูลด้วย

จากขั้นตอนการทำโมเดลเชิงตรรกที่ผ่านมาแอตตริบิวที่เรารู้จักก็คือ คีย์หลัก,คีย์รองและคีย์ภายนอก ขั้นนี้ก็จะเป็นการกำหนดแอตตริบิวอื่นๆที่ไม่ใช่คีย์เพิ่มเข้าไป โดยแอตตริบิวแต่ละตัวที่จะเพิ่มนั้นต้องขึ้นกับทั้งหมดของคีย์ของเอนติตีนั้นไม่ใช่ขึ้นกับบางส่วนของคีย์ โดยต้องขยายในเอนติตีตัวแม่ไม่ใช่ขยายในเอนติตีที่มีคีย์นั้นเป็นคีย์ภายนอก นอกจากนั้นถ้าเกิดแอตตริบิวดังกล่าวขึ้นกับคีย์หลักทั้งหมดแล้วแต่มีค่ามากกว่าหนึ่งค่า (multivalued) ให้แตกออกเป็นอีก

เอนติตีที่มีความสัมพันธ์กับเอนติตีเดิมแบบ (1:N) จากนั้นให้บันทึกชื่อและรายละเอียดอื่นๆใส่ในโมเดลและพจนานุกรมข้อมูล

การพิจารณาว่าสิ่งใดเป็นแอตตริบิวจริงหรือไม่นั้น บางครั้งยุ่งยากแต่มีข้อสังเกตคือเราจะถือว่าเป็นแอตตริบิวเมื่อไม่มีรายละเอียดใดปลีกย่อย เช่น ห้องเรียน ถ้าไม่ต้องการแยกย่อยลงไปอีก ก็จะเป็นแอตตริบิวหนึ่งของชั้นเรียน แต่ถ้าต้องการรู้ว่าขนาดเท่าไร ตั้งอยู่ที่ไหน ก็ต้องกำหนดเป็นเอนติตี สำหรับแอตตริบิวบางตัวเมื่อพิจารณาแล้วพบว่าจะขยายรีเลชันชิปก็ให้แตกรีเลชันชิปนั้นออกมาตั้งเป็นเอนติตีใหม่ที่มีรีเลชันชิปกับเอนติตีเดิมเป็นแบบ (1:N) สิ่งที่ควรหลีกเลี่ยงคือการใช้รหัสแทนแอตตริบิว แต่ถ้าจำเป็นก็ให้กำหนดรหัสอย่างเป็นอิสระจากกันคือไม่รวมความหมายมากกว่าหนึ่งอย่างไว้ในรหัสเดียวกัน ในกรณีที่แอตตริบิวใดๆสามารถหาได้จากสูตรหรือคำนวณได้จากแอตตริบิวอื่นจะเรียกว่าเป็นดิไรฟ์วแอตตริบิว (Derived Attribute) ให้ระบุตัวดี(d) ในโมเดลข้อมูลด้วย ในกรณีเป็นแฟล็กใส่ไว้เพื่อระบุซับไทป์-ซูเปอร์ไทป์ให้ระบุตัวเอส(s) ในโมเดลเช่นกัน

นอกจากการกำหนดแอตตริบิวแล้วยังให้พิจารณาเชื่อมเอนติตีที่มีลักษณะดังต่อไปนี้เข้าด้วยกัน คือ

- เอนติตีที่มีคีย์หลักเหมือนกันและให้ความหมายอันเดียวกัน
- เอนติตีที่เป็นซับไทป์ด้วยกันและมีแอตตริบิวทุกตัวเหมือนกัน
- เอนติตีที่เป็นซับไทป์-ซูเปอร์ไทป์ ซึ่งซับไทป์เป็นตัวขยายซูเปอร์ไทป์
- เอนติตีที่ไม่มีแอตตริบิวอื่นที่ไม่ใช่คีย์ให้ยุบรวมกันเอนติตีที่มีรีเลชันชิปต่อกันเสีย

ขั้นตอนที่ 7 การตรวจสอบด้วยกฎของนอร์มัลไลเซชัน

ให้ทำการตรวจสอบเอนติตีต่างๆให้อยู่ในกฎนอร์มัลไลเซชันซึ่งประกอบด้วย 1NF, 2NF, 3NF, BCNF, 4NF, 5NF ตามที่ได้กล่าวมาแล้วข้างต้น ประโยชน์ก็คือ

1. ลดที่ว่างที่ต้องใช้ในการเก็บข้อมูล
2. ลดความผิดพลาด ความไม่ตรงกันของข้อมูลในฐานข้อมูล
3. ลดการเกิดอะนอร์มัลไลของการลบและแก้ไขข้อมูล
4. เพิ่มความคงทนแก่โครงสร้างฐานข้อมูล

ในกรณีได้เอนติตีที่เป็นนอมัลไลซ์เซชันที่สมบูรณ์แล้ว สิ่งที่ต้องระวังคือไม่แตกเอนติตีนั้นย่อยลงไปอีก

ขั้นตอนที่ 8 การกำหนดโดเมน

ให้กำหนดโดเมนของแอตตริบิวทุกตัวในเอนติตีแล้วบันทึกในพจนานุกรมข้อมูล

โดเมน : กลุ่มค่าที่ถูกต้องเป็นไปได้สำหรับแอตตริบิวแต่ละตัว อันได้แก่

- ชนิดของข้อมูล(data type) เช่น จำนวนเต็ม, วันที่, ตัวอักษร, ทศนิยม
- ความยาว(length) เช่น 5 หลัก, 35 ตัวอักษร
- รูปแบบข้อมูล(format) เช่น dd/mm/yy(วันที่), ccc-cccc(เบอร์โทรศัพท์)
- ค่าที่อนุญาติ(allowable value) เช่น เป็นได้เฉพาะวันศุกร์ต้นเดือน
- ช่วงของข้อมูลหรือข้อกำหนดอื่นๆ(constraints, range) เช่น 2-50 วัน
- ความหมาย(meaning) อธิบายความหมายของแอตตริบิวนั้นว่าคืออะไร
- ความเป็นหนึ่งเดียว(uniqueness) ต้องมีค่าเป็นหนึ่งเดียว
- การเป็นนัล(null support) อนุญาติให้เป็นนัลได้หรือไม่
- ค่าโดยปริยาย(default value) กำหนดให้มีค่าเป็น 0

ได้มีการกำหนดกฎไว้เป็นพิเศษสำหรับโดเมนของแอตตริบิวบางพวกต่อไปนี้

1. แอตตริบิวของคีย์หลัก จะต้องมีความเป็นหนึ่งเดียว และห้ามมีค่าเป็นนัล ในกรณีคีย์หลักเป็นคีย์ประกอบแอตตริบิวแต่ละตัวที่มาประกอบคีย์หลักไม่จำเป็นต้องเป็นหนึ่งเดียว

2. แอตตริบิวของคีย์รอง มีกฎเหมือนคีย์หลัก เว้นแต่เป็นนัลได้

3. คีย์ภายนอก กลุ่มของโดเมนที่กำหนดให้จะต้องเหมือนกับกลุ่มโดเมนของคีย์หลักในเอนติตีแม่

4. คิไรฟว์แอตตริบิว ต้องมีค่าอนุญาติอยู่ในช่วงของผลลัพธ์ที่ได้จากการผ่านอัลกอริธึมส์ของแอตตริบิวที่เป็นที่มาและมีชนิดของข้อมูลแบบเดียวกันด้วย

5. แอตตริบิวที่เป็นคีย์หลักของเอนติตีซับไทป์ จะต้องมีค่าเป็นซับเซทของคีย์หลักของเอนติตีซูเปอร์ไทป์

ขั้นตอนที่ 9 กำหนดกฎการจัดการกับข้อมูล

เป็นกฎที่มีเพื่อตรวจสอบความถูกต้องเมื่อเราทำการเพิ่ม, ลบ, แก้ไขหรือดึงข้อมูล เราจะพิจารณาผลรวมทั้งที่เกิดกับเอนติตีอื่นและแอตตริบิวอื่นภายในเอนติตีที่เรากระทำด้วย เมื่อกำหนดกฎการจัดการต่างๆแล้วให้เก็บลงในพจนานุกรมข้อมูลโดยมีรูปแบบที่ประกอบด้วย เหตุการณ์ที่ทำ(เช่น เพิ่ม, ลบ ฯ), เอนติตีหรือแอตตริบิวที่เรากระทำด้วย, เงื่อนไขที่กำหนดไว้, การกระทำที่จะต้องเกิดสืบเนื่องจากเหตุการณ์นั้น เราจะกำหนดกฎการจัดการเพื่อ

ก. ความสมบูรณ์ ถูกต้องและเป็นอันหนึ่งอันเดียวกันของค่าของแอตตริบิว

ข. กำหนดกฎการจัดการสำหรับทุกแอตตริบิวที่เป็นตัวต้นของดิไรฟว์แอตตริบิว

ค. กำหนดกฎการจัดการสำหรับซับไทป์-ซูเปอร์ไทป์ โดยถ้าซับไทป์ถูกลบต้องลบซูเปอร์ไทป์ด้วย

ง. กำหนดกฎการจัดการโดยกำหนดด้วยเวลา เช่น ถ้าระเบียนข้อมูลนี้มีอายุเกิน

1 ปีให้ทำการลบทิ้ง

ขั้นตอนต่างๆที่ผ่านมาเราได้ทำการพัฒนาจนได้มุมมองของผู้ใช้ที่มีรายละเอียดสมบูรณ์ ขั้นตอนต่อจากนี้จะเป็นการปรับแต่งมุมมองต่างๆเข้าด้วยกัน เพื่อขจัดส่วนที่ซ้ำซ้อนและแก้ปัญหาความไม่ตรงกันของข้อมูล ซึ่งเป็นขั้นตอนที่ต้องใช้ทั้งประสบการณ์และทักษะในการวิเคราะห์มาก และนับว่าเป็นส่วนที่สำคัญที่สุด ประกอบด้วย

ขั้นตอนที่ 10 การเชื่อมมุมมองผู้ใช้ทุกคนเข้าด้วยกัน

มีจุดประสงค์ก็คือ แสดงมุมมองที่ละเอียดขึ้นด้วยโมเดลข้อมูลที่ซับซ้อน ตัดความซ้ำซ้อนและแก้ปัญหาความไม่ถูกต้องตรงกันของข้อมูล โดยอาจมีการเพิ่มรีเลชันชิปหรือกฎทางธุรกิจใหม่ๆขึ้นด้วย

สิ่งที่จะพิจารณาในการรวมเอนติตี

1. การรวมเอนติตีที่มีคีย์หลักตัวเดียวกันและค่าที่เป็นไปได้ของคีย์หลักเหมือนกัน จะต้องได้เอนติตีใหม่ที่มีแอตตริบิวรวมของสองเอนติตีเดิม

2. ถ้าเอนติตีสองอันมีคีย์หลักเดียวกันแต่ค่าที่เป็นไปได้ของคีย์หลักนั้นเป็นซับเซตของกัน เราจะรวมได้ในรูปแบบของซับไทป์-ซูเปอร์ไทป์ โดยต้องตัดแอตตริบิวที่มีแล้วในเอนติตีซูเปอร์ไทป์ออกจากเอนติตีที่เป็นซับไทป์

3. ถ้าเอนติตีสองอันมีคีย์หลักเดียวกัน แต่มีผลไปกำหนดแอตตริบิวที่ต่างกันบางตัว ให้กำหนดซูเปอร์ไทป์ขึ้นมาอันหนึ่งสัมพันธ์กันสองเอนติตีเดิม

4. การเชื่อมเอนติตีสองตัวที่คีย์หลักของตัวหนึ่งเป็นคีย์รองของอีกตัว จะได้เอนติตีใหม่ที่มีคีย์หลักตามเอนติตีตัวหนึ่งส่วนคีย์หลักของเอนติตีอีกตัวจะกลายเป็นคีย์รองไป และมีแอตตริบิวรวมระหว่างสองเอนติตีเดิม แล้วตัดแอตตริบิวที่ซ้ำซ้อนออกเสียและต้องกำหนดกฎธุรกิจให้ด้วยตามข้อบังคับเก่า เช่น ในกรณีคีย์หลักเดิมที่กลายมาเป็นคีย์สำรองในเอนติตีใหม่ก็ต้องยังคงห้ามเป็นนัล

5. การรวมเอนติตีใดๆก็ตามต้องไม่มีผลไปเปลี่ยนแปลงเอนติตีอื่นที่ไม่ได้เกี่ยวข้อง

สิ่งที่ต้องพิจารณาในการรวมรีเลชันชิป

1. ให้รวมรีเลชันชิประหว่างเอนติตีที่ให้ความหมายเหมือนกันเข้าด้วยกัน โดยถ้าผลทำให้เกิดเป็นรีเลชันชิปแบบ (M:N) จะต้องทำการแตกให้เป็นรีเลชันชิปแบบ (1:N) สองอัน

2. การรวมรีเลชันชิปใดๆต้องไม่ไปกระทบกับรีเลชันชิปอื่นที่ไม่ต้องการการเปลี่ยนแปลง นอกจากจะพิจารณาแล้วว่าควรตัดออกเนื่องจากซ้ำซ้อน หรืออาจเพิ่มขึ้นใหม่เพื่อความเหมาะสม

3. จากการรวมเอนติตีที่มีคีย์หลักเป็นคีย์รองของเอนติตีอีกตัว ให้ตรวจสอบคีย์ภายนอกของเอนติตีอื่นๆที่อ้างมาถึงว่า ได้อ้างถึงคีย์หลักหรือคีย์รองของเอนติตีใหม่ที่ได้จากการ

รวมนั้น ถ้าอ้างถึงคีย์รองต้องทำการเปลี่ยนให้เป็นคีย์หลัก

4. เมื่อรวมมุมมองต่างๆแล้วให้กำหนดกฎธุรกิจของคีย์สำหรับรีเลชันชิปใหม่ด้วย

สิ่งที่ต้องพิจารณาในการรวมแอตตริบิว

1. ให้ทำการรวมแอตตริบิวที่มีความหมายเหมือนกันภายในเอนติตีเดียวกัน และรวมค่าที่เป็นไปได้รวมถึงกฎการจัดการเข้าด้วยกันด้วย และให้พิจารณาค่าที่เป็นไปได้ของแอตตริบิวอื่นได้ด้วยว่าเปลี่ยนไปหรือไม่

2. เมื่อรวมเอนติตีแล้วให้พิจารณาตัดแอตตริบิวที่เป็นไดรฟ์หรือแฟลกซ์ที่ไม่จำเป็นทิ้งเสีย

3. หลังจากได้รวม ตัดหรือเพิ่มรีเลชันชิปแล้ว ให้ทำการนอร์มัลไลซ์อีกครั้งเพื่อตัดสิ่งที่ซ้ำซ้อนออกเสีย

ขั้นตอนที่ 11 การรวมเข้ากับโมเดลข้อมูลที่มีอยู่แล้ว

จะเป็นขั้นที่เกี่ยวข้องกับแบบแผนเชิงมโนภาพ โดยจะรวมโมเดลข้อมูลเชิงตรรกที่ได้กับของที่มีอยู่แล้วเดิม ให้พัฒนาโมเดลใหม่ควบคู่ไปกับการพิจารณากฎเกณฑ์ข้อบังคับของเดิม โดยอาจมีการใช้เอนติตีหรือรีเลชันชิปร่วมกับของเดิม และมีการกำหนดเอนติตีขึ้นมาใหม่ด้วย

ขั้นตอนที่ 12 การวิเคราะห์ความเสถียรภาพและการเติบโตในอนาคต

การออกแบบโมเดลที่ทำผ่านมาจะพิจารณาข้อมูลที่เห็นได้ในปัจจุบันเป็นส่วนมาก สำหรับขั้นนี้ให้พิจารณาถึงสิ่งที่อาจเกิดขึ้นหรือเป็นได้ในอนาคตด้วย เช่น

- อาจมีเอนติตีหรือรีเลชันชิปใหม่เกิดขึ้น ทำให้ต้องเพิ่มคีย์ภายนอกในเอนติตีของเดิม
- รีเลชันชิปแบบ (1:N) อาจกลายเป็น (M:N) ได้
- คีย์หลักอาจเปลี่ยนไปเนื่องจากของเดิมไม่เป็นหนึ่งเดียวแล้ว

สิ่งเหล่านี้เมื่อเราพิจารณาแล้วอาจทำการดัดแปลงโมเดลไว้เผื่อรองรับ หรือจดบันทึกเก็บไว้ก่อนเฉยๆก็ได้