

แผนการสอนประจำบทเรียน

รายชื่ออาจารย์ผู้จัดทำ ชุตินันท์ บุญมาก

รายละเอียดของเนื้อหา

ตอนที่ 3.1 แนวคิดเกี่ยวกับแบบจำลองความสัมพันธ์ระหว่างข้อมูล

เรื่องที่ 3.1.1 Semantic โมเดล

เรื่องที่ 3.1.2 อี-อาร์โมเดล

ตอนที่ 3.2 อี-อาร์โมเดลกับฐานข้อมูลเชิงสัมพันธ์

เรื่องที่ 3.2.1 การออกแบบฐานข้อมูลด้วยอี-อาร์โมเดล

เรื่องที่ 3.2.2 ปัญหาจากการออกแบบฐานข้อมูลด้วยอี-อาร์โมเดล

เรื่องที่ 3.2.3 การแปลงฐานข้อมูลที่ออกแบบด้วยอี-อาร์โมเดลเป็นฐานข้อมูลเชิงสัมพันธ์

แนวคิด

1. แบบจำลองความสัมพันธ์ระหว่างข้อมูลเป็นเครื่องมืออย่างหนึ่งในการออกแบบฐานข้อมูลในระดับแนวคิด โดยแสดงถึงรายละเอียดและความสัมพันธ์ของข้อมูลในลักษณะที่เป็นภาพรวมของฐานข้อมูล ทำให้เป็นประโยชน์ต่อการรวบรวมและวิเคราะห์ข้อมูล การกำหนดโครงสร้าง การเชื่อมโยงข้อมูล และลดความซ้ำซ้อนของข้อมูล
2. อี-อาร์โมเดลเป็นแบบจำลองข้อมูลที่ได้รับการนิยามในงานออกแบบฐานข้อมูล โดยนำเสนอโครงสร้างของฐานข้อมูลในระดับแนวคิดออกมาในรูปของแผนภาพที่มีโครงสร้างง่ายต่อการทำความเข้าใจ ทำให้เห็นภาพรวมของเอนทิตีทั้งหมดและความสัมพันธ์ระหว่างเอนทิตีในระบบฐานข้อมูล ส่วนฐานข้อมูลเชิงสัมพันธ์เป็นฐานข้อมูลที่มีความง่ายต่อการใช้งาน ผู้ใช้สามารถใช้งานได้โดยไม่ต้องทราบเกี่ยวกับการจัดเก็บข้อมูลในระดับกายภาพ การเรียกใช้หรือการเชื่อมโยงข้อมูลก็สามารถทำได้ง่าย

วัตถุประสงค์

หลังจากศึกษาบทเรียนที่ 3 แล้ว นักศึกษาสามารถ

1. บอกแนวคิดเกี่ยวกับแบบจำลองความสัมพันธ์ระหว่างข้อมูลได้
2. บอกความสัมพันธ์ระหว่างอี-อาร์โมเดลและฐานข้อมูลเชิงสัมพันธ์ได้

กิจกรรมการเรียนรู้การสอน

กิจกรรมที่นักศึกษาต้องทำสำหรับการเรียนการสอน ได้แก่

1. ศึกษาเอกสารการสอนตอนที่ 3.1 และตอนที่ 3.2

2. ทำกิจกรรมในแบบฝึกปฏิบัติบทเรียนที่ 3
3. ทำแบบประเมินผลของบทเรียนที่ 3

สื่อการสอน

เอกสารการสอน

1. เอกสารประกอบการสอน

ประเมินผล

1. ประเมินผลจากกิจกรรมที่ทำ
2. ประเมินผลจากคำถามท้ายบทเรียน

ตอนที่ 3.1 แนวคิดเกี่ยวกับแบบจำลองความสัมพันธ์ระหว่างข้อมูล

หัวเรื่อง

เรื่องที่ 3.1.1 Semantic โมเดล

เรื่องที่ 3.1.2 อี-อาร์โมเดล

แนวคิด

1. Semantic โมเดล เป็นแบบจำลองข้อมูลที่คิดค้นขึ้นเพื่อแก้ไขข้อจำกัดของแบบจำลองข้อมูลในยุคแรก ๆ ซึ่งมีเฉพาะรายละเอียดทางด้านโครงสร้างของฐานข้อมูล แต่ขาดรายละเอียดเกี่ยวกับความหมายของข้อมูลภายในฐานข้อมูล ทั้งนี้ ใน Semantic โมเดล มีการให้นิยามคำต่าง ๆ คือ เอนทิตี Property ความสัมพันธ์ และ Subtype
2. อี-อาร์โมเดล เป็นแบบจำลองความสัมพันธ์ระหว่างข้อมูลที่ใช้สัญลักษณ์ต่าง ๆ ในการเขียนแบบจำลองอย่างเป็นขั้นตอน ซึ่งผู้สร้างแบบจำลองต้องนำแนวคิดในการออกแบบฐานข้อมูลมาใช้ร่วมกันด้วย เพื่อให้รายละเอียดของข้อมูลที่จะแปลงเป็นฐานข้อมูลรูปแบบต่าง ๆ ประกอบด้วยข้อมูลที่เหมาะสม โดยแนวคิดที่เกี่ยวข้องกับอี-อาร์โมเดลประกอบด้วย เอนทิตี Property ความสัมพันธ์ Subtype และ Supertype รวมทั้งสัญลักษณ์ที่ใช้ในอี-อาร์โมเดล

วัตถุประสงค์

หลังจากที่ศึกษาตอนที่ 3.1 แล้ว นักศึกษาสามารถ

1. บอกแนวคิดเกี่ยวกับ Semantic โมเดล ได้
2. บอกแนวคิดเกี่ยวกับ อี-อาร์ โมเดล ได้

เรื่องที่ 3.1.1 Semantic Model

ในการออกแบบฐานข้อมูล นิยมใช้แบบจำลองข้อมูล (data model) เพื่อนำเสนอรายละเอียดต่าง ๆ เกี่ยวกับโครงสร้างและความสัมพันธ์ระหว่างข้อมูลภายในฐานข้อมูลที่ออกแบบซึ่งอยู่ในรูปของแนวคิดหรือตรรกะ (logical) ที่ยากแก่การเข้าใจ ให้อยู่ในรูปแบบที่เป็นมาตรฐานซึ่งผู้ใช้ในแต่ละระดับที่มีมุมมองต่างกันสามารถเข้าใจได้ง่ายขึ้น

แบบจำลองข้อมูลในระยะแรก ๆ จะมีการนำเสนอเฉพาะรายละเอียดเกี่ยวกับโครงสร้างของฐานข้อมูล เช่น การครบหน่วยของข้อมูล (atomic data values) การควบคุมบูรณภาพของข้อมูล (integrity constraint) เป็นต้น โดยยังขาดการนำเสนอรายละเอียดเกี่ยวกับความหมายของข้อมูลภายในฐานข้อมูล เช่น น้ำหนักและปริมาณของสินค้ามีความสัมพันธ์กันอย่างไร โดเมน (domain) ของ ข้อมูลสามารถมีค่าใดได้บ้าง ข้อมูลใดทำหน้าที่เป็นคีย์คู่แข่ง (candidate key) หรือคีย์นอก (foreign key) เป็นต้น ดังนั้น ในราวปลายปี ค.ศ.1970 ถึงช่วงต้นปี ค.ศ. 1980 จึงมีการนำเสนอแบบจำลองข้อมูลที่เรียกว่า Semantic โมเดล ขึ้น

Semantic โมเดล เป็นแบบจำลองข้อมูลที่กล่าวถึงแนวคิดหรือความหมายของคำต่าง ๆ คือ เอนทิตี (entity) Property ความสัมพันธ์ (relationship) และ Subtype ดังนี้

1. เอนทิตี

เอนทิตี หมายถึง สิ่งที่น่าสนใจสามารถระบุได้ในความเป็นจริง และต้องการการจัดเก็บข้อมูลที่เกี่ยวข้องด้วย โดยสิ่งนี้อาจเป็นรูปธรรมที่สามารถมองเห็นได้ เช่น หนังสือระบบฐานข้อมูล นาย ศ. มหาวิทยาลัยเกษตรศาสตร์ เป็นต้น หรืออาจเป็นนามธรรมที่ไม่สามารถมองเห็นได้ เช่น จำนวนวันลาพักผ่อนของพนักงาน วันหยุดราชการ เป็นต้น ก็ได้

หากนำแต่ละเอนทิตีที่มีคุณสมบัติใด ๆ เหมือนกันมารวมกัน เอนทิตีเหล่านั้นเรียกว่า Entity Type ซึ่งหมายถึง กลุ่มของเอนทิตีหรือกลุ่มของข้อมูลที่มีคุณสมบัติเหมือนกัน เช่น Entity Type นักศึกษา หมายถึง กลุ่มคนที่เป็นนักศึกษาทุกคน Entity Type บุคคลสัญชาติไทย หมายถึง กลุ่มคนที่มีสัญชาติไทยทุกคน เป็นต้น

ทั้งนี้ เอนทิตีใดเอนทิตีหนึ่งที่กำลังสนใจ เรียกว่า เอนทิตีที่ปรากฏ (entity instance) ดังนั้น ในหนึ่ง Entity Type จึงอาจมีเอนทิตีที่ปรากฏได้จำนวนมาก

2. Property

Property หมายถึง คุณสมบัติหรือคุณลักษณะของเอนทิตี ซึ่งแต่ละเอนทิตีใน Entity Type เดียวกัน จะมี Property ต่าง ๆ เหมือนกัน เช่น Property ของ Entity Type นักศึกษาประกอบด้วย รหัสประจำตัว ชื่อ-สกุล เพศ รหัสคณะ เป็นต้น

แม้ว่าทุกเอนทิตีที่เป็นสมาชิกของ Entity Type เดียวกันจะมี Property ต่าง ๆ เหมือนกัน แต่จะมี ข้อมูลใน Property หนึ่งที่เป็นเอกลักษณ์เฉพาะของแต่ละเอนทิตี เช่น Property รหัสประจำตัวของ Entity Type นักศึกษาจะไม่มีค่าในแต่ละเอนทิตีที่ซ้ำกันเลย เป็นต้น ซึ่ง Property ที่สามารถกำหนดเอกลักษณ์เฉพาะให้กับแต่ละเอนทิตีได้นี้เรียกว่า Identity

Identity		Property		
		รหัสประจำตัว	ชื่อ-สกุล	เพศ
เอนทิตี	→	41010703	ศิริมาศ ตันติศิริ	หญิง
	→	42020018	ทิพวรรณ วงศ์อินทร์ตา	หญิง
	→	42020152	แฉกราม พลจันทร์	ชาย
	→	42020665	เชาววิทย์ วิชิตอำพล	ชาย
			รหัสคณะ	
			01	
			02	
			02	
			02	

ภาพที่ 3.1 เอนทิตี Property และ Identity

จากภาพที่ 3.1 Entity Type นักศึกษาประกอบด้วย 4 เอนทิตี แต่ละเอนทิตีประกอบด้วย 4 Property คือ รหัสประจำตัว ชื่อ-สกุล เพศ และรหัสคณะ โดยมี Property รหัสประจำตัวเป็น Identity

อย่างไรก็ตาม หากพิจารณาดูจะพบว่า แต่ละ Property ก็คือแต่ละแอททริบิวต์ (attribute) ส่วนแต่ละเอนทิตีก็คือแต่ละทูเพิล (tuple) ในแบบจำลองข้อมูลเชิงสัมพันธ์ (relation data model) นั่นเอง หากทว่า Semantic โมเดล ถูกนำมาใช้เพื่อการออกแบบฐานข้อมูลในระดับแนวคิดหรือตรรกะที่มีวัตถุประสงค์เพื่อนำเสนอข้อเท็จจริงต่าง ๆ ที่เกี่ยวข้องกับข้อมูล โดยยังไม่มีกระบวนการสร้างของฐานข้อมูลที่ใช้ ดังนั้น จึงใช้ คำว่า Property แทนคำว่า แอททริบิวต์ และคำว่า เอนทิตี แทนคำว่า ทูเพิล

3. ความสัมพันธ์

ความสัมพันธ์ หมายถึง Entity Type ที่แสดงความสัมพันธ์ระหว่างสอง Entity Type ขึ้นไป เป็นความสัมพันธ์ที่สมาชิกของ Entity Type หนึ่งมีความสัมพันธ์กับสมาชิกของอีก Entity Type หนึ่ง โดยแต่ละความสัมพันธ์จะถูกระบุด้วยชื่อที่อธิบายถึงความสัมพันธ์นั้น ๆ

Entity Type นักศึกษา

รหัสประจำตัว	ชื่อ-สกุล	เพศ	รหัสคณะ
41010703	ศิริมาศ สันติศิริ	หญิง	01
42020018	ทิพวรรณ วงศ์อินทร์ตา	หญิง	02
42020152	แรงราม พลจันทร์	ชาย	02
42020665	เชาววิทย์ วิชิตอำพล	ชาย	02

Entity Type คณะ

รหัสคณะ	ชื่อคณะ
01	เกษตร
02	ศึกษาศาสตร์

ตามสัมพันธ์สังกัด

รหัสประจำตัว	ชื่อ-สกุล	เพศ	รหัสคณะ	ชื่อคณะ
41010703	ศิริมาศ สันติศิริ	หญิง	01	เกษตร
42020018	ทิพวรรณ วงศ์อินทร์ตา	หญิง	02	ศึกษาศาสตร์
42020152	แรงราม พลจันทร์	ชาย	02	ศึกษาศาสตร์
42020665	เชาววิทย์ วิชิตอำพล	ชาย	02	ศึกษาศาสตร์

ภาพที่ 3.2 ตัวอย่างของความสัมพันธ์

จากภาพที่ 3.2 ความสัมพันธ์สังกัดเป็น Entity Type ที่แสดงความสัมพันธ์ระหว่าง Entity Type นักศึกษาและ Entity Type คณะ ซึ่งความสัมพันธ์สังกัดประกอบด้วย 5 Property คือ รหัสประจำตัว ชื่อ-สกุล เพศ รหัสคณะ และชื่อคณะ โดยมี Property รหัสประจำตัวเป็น Identity

นอกจากความสัมพันธ์ระหว่าง Entity Type ที่มี Property ร่วมกันจะเป็นตัวกำหนดความสัมพันธ์ขึ้นมาแล้ว ความสัมพันธ์อาจสร้างขึ้นมาจาก Property ที่เป็น Identity ของแต่ละ Entity Type ก็ได้

Entity Type นักศึกษา

รหัสประจำตัว	ชื่อ-สกุล	เพศ
42020018	ทิพวรรณ วงศ์อินทร์ตา	หญิง
42020152	แรงราม พลจันทร์	ชาย
42020665	เชาว์วิทย์ วิชิตอำพล	ชาย
42020674	จริญญา ตีพิมพ์	หญิง
42020699	นงลักษณ์ ตะกานา	หญิง

ตามสเกันซ์ที่ปรึกษา

รหัสประจำตัว	รหัสอาจารย์
42020018	Q1011
42020152	Q1023
42020665	Q1035
42020674	Q1011
42020699	Q1023

Entity Type อาจารย์

รหัสอาจารย์	ชื่อ-สกุล	เพศ	เงินเดือน
Q1011	ศิริภัทรา เหมือนมาลัย	หญิง	8,000
Q1023	สุพิมพ์ ศรีพันธ์วรสกุล	หญิง	9,500
Q1035	ศิริชัย ศรีพรหม	ชาย	12,000

ภาพที่ 3.3 ตัวอย่างของความสัมพันธ์ที่สร้างจาก Identity ของแต่ละ Entity Type

จากภาพที่ 3.3 ความสัมพันธ์ที่ปรึกษาเป็น Entity Type ที่สร้างจาก Property รหัสประจำตัวซึ่งเป็น Identity ของ Entity Type นักศึกษา และ Property รหัสอาจารย์ซึ่งเป็น Identity ของ Entity Type อาจารย์

4. Subtype

หาก Entity Type ใดมี Property ที่สามารถกำหนดเอกลักษณ์เฉพาะให้กับแต่ละเอนทิตีได้ และ ข้อมูลทุกเอนทิตีใน Entity Type นั้นเป็นข้อมูลส่วนหนึ่งใน Entity Type หลัก เรียก Entity Type นั้นว่าเป็น

Entity Type นักศึกษาหญิง

รหัสประจำตัว	ชื่อ-สกุล	เพศ	รหัสคณะ
41010703	ศิริมาศ ตันติศิริ	หญิง	01
42020018	ทิพวรรณ วงศ์อินทร์ตา	หญิง	02

Entity Type นักศึกษาชาย

รหัสประจำตัว	ชื่อ-สกุล	เพศ	รหัสคณะ
42020152	แรงราม พลจันทร์	ชาย	02
42020665	เชาว์วิทย์ วิชิตอำพล	ชาย	02

Entity Type นักศึกษา

รหัสประจำตัว	ชื่อ-สกุล	เพศ	รหัสคณะ
41010703	ศิริมาศ ตันติศิริ	หญิง	01
42020018	ทิพวรรณ วงศ์อินทร์ตา	หญิง	02
42020152	แรงราม พลจันทร์	ชาย	02
42020665	เชาว์วิทย์ วิชิตอำพล	ชาย	02

Subtype ของ Entity Type หลัก

ภาพที่ 3.4 ตัวอย่างของ Subtype

จากภาพที่ 3.4 Entity Type นักศึกษาหญิงและ Entity Type นักศึกษาชายเป็น Subtype ของ Entity Type นักศึกษา โดยแต่ละเอนทิตีต่างก็มี Property รหัสประจำตัวเป็น Identity

เรื่องที่ 3.1.2 อี-อาร์โมเดล

Entity-Relationship Model หรืออี-อาร์โมเดล เป็นแบบจำลองข้อมูลที่ได้รับการประยุกต์มาจากแนวคิดของ Semantic โมเดล และได้รับความนิยมอย่างมากสำหรับนำมาใช้เพื่อการออกแบบฐานข้อมูลในระดับแนวคิด โดยอี-อาร์โมเดลเป็นผลงานการพัฒนาของ Peter Pin Shan Chen จาก Massachusetts Institute of Technology ในปี ค.ศ.1976

อี-อาร์โมเดล เป็นแบบจำลองข้อมูลซึ่งแสดงถึงโครงสร้างของฐานข้อมูลที่เป็นอิสระจากซอฟต์แวร์ที่จะใช้ในการพัฒนาฐานข้อมูล รวมทั้งรายละเอียดและความสัมพันธ์ระหว่างข้อมูลในระบบในลักษณะที่เป็นภาพรวม ทำให้เป็นประโยชน์อย่างมากต่อการรวบรวมและวิเคราะห์รายละเอียด ตลอดจนความสัมพันธ์ของข้อมูลต่าง ๆ โดยอี-อาร์โมเดลมีการใช้สัญลักษณ์ต่างๆ ที่เรียกว่า Entity Relationship Diagram หรือ อี-อาร์ไดอะแกรม แทนรูปแบบของข้อมูลเชิงตรรกะขององค์กร จึงทำให้บุคลากรที่เกี่ยวข้องกับระบบฐานข้อมูลสามารถเข้าใจลักษณะของข้อมูลและความสัมพันธ์ระหว่างข้อมูลได้ง่ายและถูกต้องตรงกัน ระบบที่ได้รับการออกแบบจึงมีความถูกต้องและสอดคล้องกับวัตถุประสงค์ขององค์กร

ทั้งนี้ อี-อาร์โมเดลประกอบด้วย 4 ส่วน คือ เอนทิตี Property ความสัมพันธ์ Subtype และ Supertype

1. เอนทิตี

เอนทิตี หมายถึง สิ่งที่น่าสนใจสามารถระบุได้ในความเป็นจริง และต้องการเก็บรวบรวมข้อมูลที่เกี่ยวข้องด้วยไว้ในฐานข้อมูล โดยตัวอย่างของเอนทิตีประเภทต่าง ๆ เช่น บุคคล สถานที่ สิ่งของ หรือเหตุการณ์ มีดังนี้

- บุคคล ได้แก่ พนักงาน ผู้ป่วย และ นักศึกษา เป็นต้น
- สถานที่ ได้แก่ เขต จังหวัด และ ภาค เป็นต้น
- วัตถุ ได้แก่ รถยนต์ อาคาร และ เครื่องจักร เป็นต้น
- เหตุการณ์ ได้แก่ การลงทะเบียนเรียน ความชำนาญ เป็นต้น

ในอี-อาร์ไดอะแกรม ใช้สัญลักษณ์รูปสี่เหลี่ยมผืนผ้า (rectangle) แทนหนึ่งเอนทิตี โดยมีชื่อของเอนทิตีนั้น ๆ กำกับอยู่ภายใน

นักศึกษา

ภาพที่ 3.5 เอนทิตีนักศึกษา

เอนทิตีสามารถจำแนกได้เป็น 2 ประเภท คือ เอนทิตีปกติ (regular entity) และเอนทิตีอ่อนแอ (weak entity)

1.1 เอนทิตีปกติ

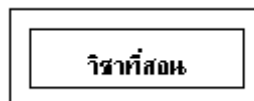
เอนทิตีปกติ หรือ **Strong Entity** หมายถึง เอนทิตีที่สนใจและต้องการจัดเก็บข้อมูลที่เกี่ยวข้องไว้ในระบบฐานข้อมูล ซึ่งการคงอยู่ของเอนทิตีนี้ไม่เกี่ยวข้องกับเอนทิตีอื่น โดยเอนทิตีนี้สามารถมีคุณสมบัติ **Identity** ได้ตัวเอง

ในอี-อาร์ไดอะแกรม สัญลักษณ์ที่ใช้แทนเอนทิตีปกติเป็นเส้นเดียวกับเอนทิตี ดังนั้น เมื่อมีกล่าวถึงเอนทิตีใด ๆ โดยไม่มีการระบุรายละเอียดอื่น จึงมีความหมายถึงเอนทิตีปกตินั่นเอง

1.2 เอนทิตีอ่อนแอ

เอนทิตีอ่อนแอ หมายถึง เอนทิตีที่มีการคงอยู่เกี่ยวข้องกับเอนทิตีอื่นในระบบฐานข้อมูล โดยเอนทิตีอื่นที่มีความสัมพันธ์กับเอนทิตีนี้เรียกว่า **Parent Entity** หรืออาจกล่าวได้ว่า เอนทิตีอ่อนแอจะไม่มี ความหมายหรือไม่สามารถปรากฏในฐานข้อมูลได้ หากปราศจาก **Parent Entity** ที่มีความสัมพันธ์เกี่ยวข้อง กัน ซึ่งสมาชิกของเอนทิตีอ่อนแอจะสามารถมีคุณสมบัติ **Identity** ได้ก็ต่อเมื่ออาศัย **Property** ใด **Property** หนึ่งของเอนทิตีปกติมาประกอบกับ **Property** ของเอนทิตีนั้น ๆ

ในอี-อาร์ไดอะแกรม ใช้สัญลักษณ์รูปสี่เหลี่ยมผืนผ้าสองรูปซ้อนกัน (double rectangle) แทนหนึ่งเอนทิตีอ่อนแอ โดยมีชื่อของเอนทิตีอ่อนแอ นั้น ๆ กำกับอยู่ภายใน



ภาพที่ 3.6 ตัวอย่างของเอนทิตีอ่อนแอ

ทั้งนี้ ตัวอย่างของเอนทิตีปกติและเอนทิตีอ่อนแออาจอธิบายได้ดังภาพที่ 3.7 คือ ในสถานศึกษา แห่งหนึ่ง อาจารย์แต่ละคนจะมีรหัสอาจารย์ที่ไม่ซ้ำกัน ดังนั้น เอนทิตีอาจารย์จะเป็นเอนทิตีปกติ ที่มี **Property** รหัสอาจารย์เป็น **Identity**

ในเอนทิตีตารางสอน อาจารย์แต่ละคนอาจจะสอนชุดวิชาเดียวกันและหมู่เรียนเดียวกัน หรือบางคนอาจจะสอนชุดวิชาเดียวกันแต่คนละหมู่เรียนก็ได้ ฉะนั้นหากไม่มีเอนทิตีอาจารย์ก็จะไม่สามารถทราบได้ว่า อาจารย์คนใด สอนชุดวิชาใด หมู่เรียนใด ในวัน/เวลาใด

นอกจากนี้ เอนทิตีตารางสอนจะสามารถมีคุณสมบัติ **Identity** ได้ก็ต่อเมื่ออาศัย **Property** รหัสอาจารย์ของเอนทิตีอาจารย์ซึ่งเป็นเอนทิตีปกติมาประกอบกับ **Property** วัน/เวลาสอนของ เอนทิตีตารางสอน ฉะนั้น เอนทิตีตารางสอนจะไม่สามารถคงอยู่ได้ หากปราศจากเอนทิตีอาจารย์ ดังนั้น จึงกล่าวได้ว่า เอนทิตีตารางสอนเป็นเอนทิตีอ่อนแอ

เอนทิตีอาจารย์

รหัส	ชื่อ-สกุล	เพศ	เงินเดือน
Q1011	ศิริภัทรา เหมือนมาลัย	หญิง	8,000
Q1023	สุพิมพ์ ศรีพันธ์วรสกุล	หญิง	9,500
Q1035	ศิริชัย ศรีพรหม	ชาย	12,000

เอนทิตีตารางสอน

รหัส	รหัสบุคลากร	หมู่เรียน	วัน/เวลาสอน	ห้องเรียน
Q1011	729113	700	อ.,พ.อ. 08.00-10.00 น.	LH1-201
Q1011	999211	711	จ.,พ. 10.00-12.00 น.	SC4-308
Q1023	999211	712	จ.,พ. 13.00-15.00 น.	SC4-308
Q1023	999211	713	อ.,พ.อ. 08.00-10.00 น.	SC4-308
Q1023	999211	714	อ.,พ.อ. 10.00-12.00 น.	SC4-308
Q1035	172596	700	จ.,พ. 10.00-12.00 น.	LH1-201
Q1035	729113	700	อ.,พ.อ. 08.00-10.00 น.	LH1-201
Q1035	999211	700	ศ. 10.00-12.00 น.	LH1-202
Q1035	999211	715	อ.,พ.อ. 13.00-15.00 น.	SC4-308
Q1035	999211	716	จ.,พ. 08.00-10.00 น.	SC4-308

ภาพที่ 3.7 ตัวอย่างของเอนทิตีปกติและเอนทิตีอ่อนแอ

2. Property

Property หมายถึง ข้อมูลที่แสดงคุณสมบัติหรือคุณลักษณะของเอนทิตีหรือความสัมพันธ์ (มีความหมายเช่นเดียวกับแอททริบิวต์ในแบบจำลองข้อมูลเชิงสัมพันธ์) เช่น **Property** ของเอนทิตีนักศึกษา ประกอบด้วย รหัสประจำตัว ชื่อ-สกุล เพศ รหัสคณะ เป็นต้น

ในอี-อาร์ไดอะแกรมใช้สัญลักษณ์รูปวงรี (ellipse) ที่มีชื่อของ **Property** นั้นกำกับอยู่ภายในแทนหนึ่ง **Property** และเชื่อมต่อกับเอนทิตีที่มี **Property** นั้นด้วยเส้นตรง



ภาพที่ 3.8 Property ของเอนทิตีนักศึกษา

Property สามารถจำแนกได้เป็น 6 ประเภท คือ Simple Property, Composite Property, Key Property, Single Property, Multi-Valued Property และ Derived Property

2.1 Simple Property

Simple Property หมายถึง Property ที่ไม่สามารถแบ่งแยกย่อยลงไปได้อีก เช่น Property ชื่อ นามสกุล และเพศ เป็นต้น

ทั้งนี้ ในอี-อาร์ไดอะแกรม สัญลักษณ์ที่ใช้แทน Simple Property เป็นเช่นเดียวกับ Property

2.2 Composite Property

Composite Property หมายถึง Property ที่มีลักษณะตรงข้ามกับ Simple Property คือ สามารถแบ่งแยกย่อยลงไปได้อีก เช่น Property ชื่อ-สกุล สามารถแบ่งได้ 2 Simple Property คือ Property ชื่อ และนามสกุล หรือ Property ที่อยู่ สามารถแบ่งได้ 5 Simple Property คือ Property บ้านเลขที่ ถนน แขวง เขต และจังหวัด เป็นต้น

ในอี-อาร์ไดอะแกรม ใช้สัญลักษณ์รูปวงรีที่มีชื่อของ Property นั้นกำกับอยู่ภายในแทนหนึ่ง Composite Property และเชื่อมต่อกับ Simple Property ที่จำแนกออกไปด้วยเส้นตรง



ภาพที่ 3.9 ตัวอย่างของ Composite Property

2.3 Key Property

Key Property หมายถึง Property หรือกลุ่มของ Property ที่มีค่าของข้อมูลในแต่ละสมาชิกของ เอนทิตีไม่ซ้ำกัน ทำให้สามารถระบุความแตกต่างของแต่ละสมาชิกในเอนทิตีได้ เช่น เอนทิตีนักศึกษาประกอบด้วย Property รหัสประจำตัว ชื่อ-สกุล และที่อยู่ โดย Property ที่สามารถบอกความแตกต่างของ นักศึกษาแต่ละคนได้ คือ รหัสประจำตัว ดังนั้น Property รหัสประจำตัวจึงเป็น Key Property ของเอนทิตี นักศึกษา เป็นต้น

ในอี-อาร์ไดอะแกรม ใช้สัญลักษณ์รูปวงรีซึ่งภายในกำกับด้วยชื่อของ Property ที่มีการ



ขีดเส้นใต้แทน Key Property และเชื่อมต่อกับเอนทิตีที่มี Property นั้นด้วยเส้นตรง

ภาพที่ 3.10 ตัวอย่างของ Key Property

2.4 Single-Valued Property

Single-Valued Property หมายถึง Property ที่มีค่าของข้อมูลในแต่ละสมาชิกของเอนทิตีได้เพียง ค่าเดียว เช่น บุคคลหนึ่งคนมีเพศเดียว Property เพศจึงเป็น Single-Valued Property หรือพนักงานหนึ่งคนมีเงินเดือนเพียงค่าเดียว Property เงินเดือนจึงเป็น Single-Valued Property เป็นต้น

ทั้งนี้ ในอี-อาร์ไดอะแกรม สัญลักษณ์ที่ใช้แทน Single-Valued Property จะเป็นเช่นเดียวกับ Property

2.5 Multi-Valued Property

Multi-Valued Property หมายถึง Property ที่มีลักษณะตรงข้ามกับ Single-Valued Property โดยเป็น Property ที่สามารถมีค่าของข้อมูลในแต่ละสมาชิกของเอนทิตีได้หลายค่า เช่น บุคคลหนึ่งคนอาจมีวุฒิการศึกษาได้หลายระดับ Property วุฒิการศึกษาจึงเป็น Multi-Valued Property หรือบ้านหลังหนึ่งอาจมีหลายเบอร์โทรศัพท์ Property เบอร์โทรศัพท์จึงเป็น Multi-Valued Property เป็นต้น

ในอี-อาร์ไดอะแกรม ใช้สัญลักษณ์รูปวงรีที่มีชื่อของ Property นั้นกำกับอยู่ภายในแทน Multi-Valued Property และเชื่อมต่อกับเอนทิตีที่มี Property นั้นด้วยเส้นคู่



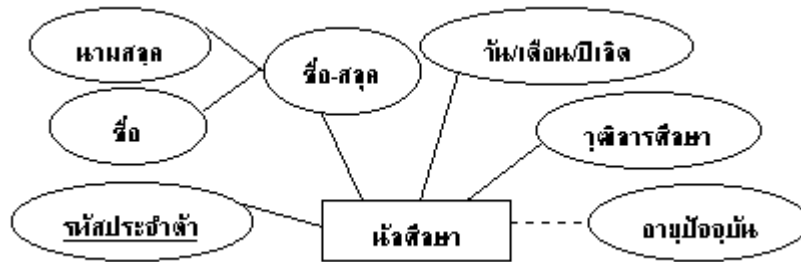
ภาพที่ 3.11 ตัวอย่างของ Multi-Valued Property

2.6 Derived Property

Derived Property หมายถึง Property ที่ค่าของข้อมูลในแต่ละสมาชิกของเอนทิตีได้มาจากการนำ ค่าของข้อมูลใน Property อื่นที่มีอยู่ในแต่ละสมาชิกของเอนทิตีมาทำการคำนวณ ซึ่งโดยทั่วไปไม่จำเป็นต้องจัดเก็บ Property ประเภทนี้ไว้ในระบบฐานข้อมูล เนื่องจาก Property ประเภทนี้มีการเปลี่ยนแปลงค่าของ ข้อมูลในแต่ละสมาชิกของเอนทิตีทุกครั้ง เมื่อมีการเปลี่ยนแปลงค่าของข้อมูลใน Property ที่ถูกนำมา

คำนวณของแต่ละสมาชิกในเอนทิตี เช่น Property อายุปัจจุบันสามารถคำนวณได้จาก Property วัน/เดือน/ปีเกิด เป็นต้น

ในอี-อาร์ไดอะแกรม ใช้สัญลักษณ์รูปวงรี ที่มีชื่อของ Property นั้นกำกับอยู่ภายในแทนหนึ่ง Derived Property และเชื่อมต่อกับเอนทิตีที่มี Property นั้นด้วยเส้นประ



ภาพที่ 3.12 ตัวอย่างของ Derived Property

3. ความสัมพันธ์

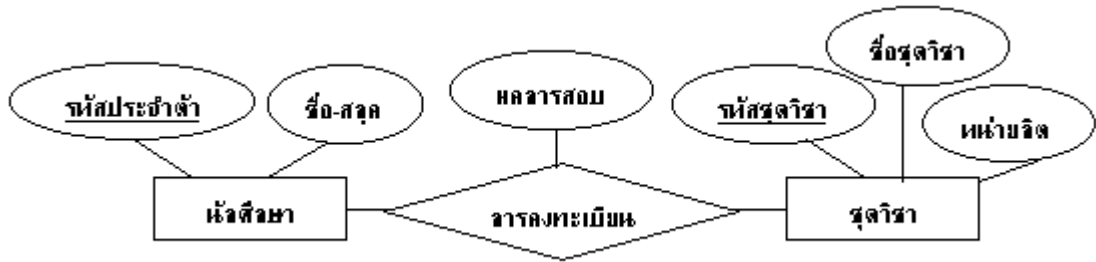
ความสัมพันธ์ หมายถึง เอนทิตีที่แสดงความสัมพันธ์ระหว่างสองเอนทิตีขึ้นไป ซึ่งโดยทั่วไปเป็นความสัมพันธ์ระหว่างเอนทิตีที่มี Property ร่วมกัน โดยแต่ละความสัมพันธ์จะถูกระบุด้วยชื่อที่อธิบายถึงความสัมพันธ์นั้น ๆ เช่น ความสัมพันธ์สังกัดแสดงความสัมพันธ์ระหว่างเอนทิตีนักศึกษาและเอนทิตีคณะ เป็นต้น

ในอี-อาร์ไดอะแกรม ใช้สัญลักษณ์รูปสี่เหลี่ยมข้าวหลามตัด (diamond) ที่มีชื่อของความสัมพันธ์นั้นกำกับอยู่ภายในแทนหนึ่งความสัมพันธ์ และเชื่อมต่อกับเอนทิตีที่เกี่ยวข้องกับความสัมพันธ์นั้นด้วยเส้นตรง



ภาพที่ 3.13 ความสัมพันธ์ระหว่างเอนทิตีนักศึกษาและเอนทิตีคณะ

นอกจากความสัมพันธ์ระหว่างเอนทิตีที่มี Property ร่วมกันจะเป็นตัวกำหนดความสัมพันธ์ขึ้นมาแล้ว ความสัมพันธ์อาจสร้างขึ้นมาจาก Key Property ของเอนทิตีที่มีความสัมพันธ์กัน โดยความสัมพันธ์นั้นอาจมี Property ของตนเองก็ได้ เช่น ความสัมพันธ์การลงทะเบียนแสดงความสัมพันธ์ระหว่างเอนทิตี นักศึกษาและเอนทิตีชุดวิชา รวมทั้งผลการสอบในแต่ละชุดวิชาของนักศึกษาแต่ละคน เป็นต้น



ภาพที่ 3.14 ความสัมพันธ์ที่สร้างจาก Key Property ของเอนทิตีที่มีความสัมพันธ์กัน

ความสัมพันธ์ระหว่างเอนทิตีใด ๆ อาจมีมากกว่าหนึ่งความสัมพันธ์ก็ได้ เช่น ความสัมพันธ์การสอน และความสัมพันธ์การจัดการแสดงความสัมพันธ์ระหว่างเอนทิตีอาจารย์และเอนทิตีชุดวิชา เป็นต้น



ภาพที่ 3.15 ความสัมพันธ์มากกว่าหนึ่งความสัมพันธ์

เอนทิตีที่เกี่ยวข้องกับความสัมพัทธ์เรียกว่า **Participant** ของความสัมพัทธ์ และจำนวนของ Participant ในความสัมพัทธ์เรียกว่า **Degree** ของความสัมพัทธ์ (มีความหมายแตกต่างไปจาก Degree ในแบบจำลองข้อมูลเชิงสัมพันธ์) ทั้งนี้ เอนทิตีซึ่งเป็น Participant ของความสัมพัทธ์อาจมีส่วนร่วมในความสัมพัทธ์ที่สามารถจำแนกได้ 2 ลักษณะ คือ แบบ **Total Participation** และแบบ **Partial Participation**

แบบ Total Participation เป็นความสัมพัทธ์ที่ทุกสมาชิกในเอนทิตีหนึ่งจะมีข้อมูลใน Property หนึ่งที่มีความสัมพันธ์กับข้อมูลในอีกหนึ่งเอนทิตี เช่น อาจารย์ทุกคนต้องสังกัดคณะใดคณะหนึ่งเท่านั้น ดังนั้น แต่ละสมาชิกในเอนทิตีอาจารย์จะมีความสัมพันธ์กับเอนทิตีคณะ เป็นต้น

ในอี-อาร์ไดอะแกรม การระบุความสัมพัทธ์แบบ **Total Participation** ใช้สัญลักษณ์เส้นคู่ เพื่อเชื่อมต่อบetween ความสัมพันธ์กับเอนทิตีที่ทุกสมาชิกมีความสัมพันธ์กับอีกเอนทิตีหนึ่ง



ภาพที่ 3.16 ความสัมพันธ์แบบ Total Participation

แบบ Partial Participation เป็นความสัมพัทธ์ที่บางสมาชิกในเอนทิตีหนึ่งเท่านั้นจะมีข้อมูลใน Property หนึ่งที่มีความสัมพันธ์กับข้อมูลในอีกหนึ่งเอนทิตี เช่น มีนักศึกษาเพียงบางคนเท่านั้นที่เป็นผู้แทนนักศึกษาในแต่ละคณะ ดังนั้น จะมีเพียงบางสมาชิกในเอนทิตีนักศึกษาเท่านั้นที่มีความสัมพันธ์กับเอนทิตีคณะ เป็นต้น

ในอี-อาร์ไดอะแกรม การระบุความสัมพันธ์แบบ Partial Participation ใช้สัญลักษณ์เช่นเดียวกับ การแสดงความสัมพันธ์ระหว่างเอนทิตีใด ๆ ที่เกี่ยวข้อง



ภาพที่ 3.17 ความสัมพันธ์แบบ Partial Participation

ความสัมพันธ์มีหลายประเภทขึ้นอยู่กับปัจจัยที่ใช้ในการจำแนกประเภทของความสัมพันธ์ ซึ่งมี 2 ปัจจัย คือ ความสัมพันธ์ระหว่างเอนทิตี และจำนวนเอนทิตีที่เกี่ยวข้อง

3.1 ความสัมพันธ์ระหว่างเอนทิตี

การจำแนกประเภทของความสัมพันธ์ตามความสัมพันธ์ระหว่างเอนทิตี เป็นการพิจารณาถึงสัดส่วนความสัมพันธ์ระหว่างสมาชิกในเอนทิตีที่เป็น Participant ของความสัมพันธ์ซึ่งอาจเรียกอีกอย่างหนึ่งว่า Cardinality Ratio

วิธีนี้สามารถจำแนกความสัมพันธ์ได้เป็น 3 ประเภท คือ ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (one to one relationship) ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (one to many relationship) และ ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (many to many relationship)

ความสัมพันธ์แบบหนึ่งต่อหนึ่ง หมายถึง ความสัมพันธ์ที่แต่ละสมาชิกในเอนทิตีหนึ่งมีความสัมพันธ์กับสมาชิกในอีกหนึ่งเอนทิตีเพียงสมาชิกเดียว หรือกล่าวได้ว่า ความสัมพันธ์ดังกล่าวเป็นแบบหนึ่งต่อหนึ่ง เช่น เอนทิตีอาจารย์และเอนทิตีคณะ มีความสัมพันธ์กันแบบหนึ่งต่อหนึ่ง กล่าวคือ แต่ละคณะมีคณบดีเพียงหนึ่งคนเท่านั้น และมีอาจารย์เพียงหนึ่งคนเท่านั้นที่เป็นคณบดี เป็นต้น

ในอี-อาร์ไดอะแกรมใช้สัญลักษณ์ 1:1 กำกับเหนือเส้นที่เชื่อมต่อระหว่างความสัมพันธ์และเอนทิตีที่เกี่ยวข้อง



กับความสัมพันธ์นั้น

ภาพที่ 3.18 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง

ความสัมพันธ์แบบหนึ่งต่อกลุ่ม หมายถึง ความสัมพันธ์ที่แต่ละสมาชิกในเอนทิตีหนึ่งมีความสัมพันธ์กับสมาชิกในอีกหนึ่งเอนทิตีมากกว่าหนึ่งสมาชิก หรือกล่าวได้ว่า ความสัมพันธ์ดังกล่าวเป็นแบบหนึ่งต่อกลุ่ม เช่น เอนทิตีคณะและเอนทิตีคณาจารย์มีความสัมพันธ์กันแบบหนึ่งต่อกลุ่ม กล่าวคือ นักศึกษาแต่ละคนมีสังกัดเพียงคณะเดียว และหนึ่งคณะอาจมีนักศึกษานับได้หลายคน

ในอี-อาร์ไดอะแกรมใช้สัญลักษณ์ 1:M กำกับเหนือเส้นที่เชื่อมต่อระหว่างความสัมพันธ์และ เอนทิตีที่เกี่ยวข้องกับความสัมพันธ์ นั้น



ภาพที่ 3.19 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม

จากภาพที่ 3.19 เอนทิตีคณะและเอนทิตีนักศึกษาที่มีความสัมพันธ์กันแบบหนึ่งต่อกลุ่ม และเป็น ความสัมพันธ์แบบ Total Participation

ความสัมพันธ์แบบกลุ่มต่อกลุ่ม หมายถึง ความสัมพันธ์ที่สมาชิกมากกว่าหนึ่งสมาชิกใน เอนทิตีหนึ่งมีความสัมพันธ์กับสมาชิกในอีกหนึ่งเอนทิตีมากกว่าหนึ่งสมาชิก หรือกล่าวได้ว่า ความสัมพันธ์ ดังกล่าวเป็นแบบกลุ่มต่อกลุ่ม เช่น เอนทิตีนักศึกษาและเอนทิตีชุดวิชาที่มีความสัมพันธ์กันแบบกลุ่มต่อกลุ่ม กล่าวคือ นักศึกษาแต่ละคนสามารถลงทะเบียนเรียนได้หลายชุดวิชา และแต่ละชุดวิชาสามารถมีนักศึกษาลงทะเบียนเรียนได้หลายคน

ในอี-อาร์ไดอะแกรม ใช้สัญลักษณ์ M:M กำกับเหนือเส้นที่เชื่อมต่อระหว่างความสัมพันธ์



และเอนทิตีที่เกี่ยวข้องกับความสัมพัธ์ นั้น

ภาพที่ 3.20 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม

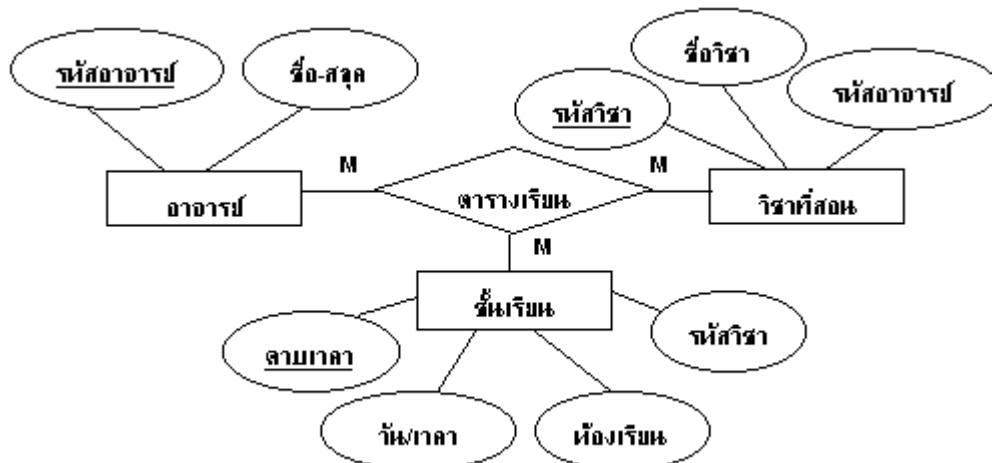
3.2 จำนวนเอนทิตีที่เกี่ยวข้อง

การจำแนกประเภทของความสัมพันธ์ตามจำนวนเอนทิตีที่เกี่ยวข้อง เป็นการพิจารณาถึงจำนวนของ Participant ใน ความสัมพันธ์ หรือ Degree ของ ความสัมพันธ์

วิธีนี้สามารถจำแนกความสัมพันธ์ ได้ 2 ประเภท คือ ความสัมพันธ์แบบ Binary และ ความสัมพันธ์แบบ N-ary

ความสัมพันธ์แบบ Binary เป็นความสัมพันธ์ที่พบได้บ่อยที่สุด โดยเป็นความสัมพันธ์ ที่แสดงความสัมพันธ์ระหว่างสองเอนทิตีใด ๆ เช่น ความสัมพันธ์ที่ปรึกษาแสดงความสัมพันธ์ระหว่างเอนทิตี อาจารย์ที่ปรึกษากับเอนทิตีนักศึกษา

ความสัมพันธ์แบบ N-ary เป็นความสัมพันธ์ที่แสดงความสัมพันธ์ระหว่างสองเอนทิตี ขึ้นไป เช่น ความสัมพันธ์ตารางเรียนแสดงความสัมพันธ์ระหว่างเอนทิตีอาจารย์ เอนทิตีชุดวิชาที่สอน และ เอนทิตี ชั้นเรียน



ภาพที่ 3.21 ความสัมพันธ์แบบ N-ary

นอกจากประเภทของความสัมพันธ์ข้างต้น ยังมีลักษณะอื่น ๆ ที่เกี่ยวข้องกับความสัมพันธ์อีก คือ Existence Dependency ความสัมพันธ์แบบ Recursive และ Composite Entity

Existence Dependency เป็นความสัมพันธ์ที่เกิดขึ้นและคงอยู่ของเอนทิตีหนึ่งขึ้นอยู่กับ การคงอยู่ของอีกเอนทิตีหนึ่งที่มีความสัมพันธ์กัน กล่าวคือ เป็นความสัมพันธ์ระหว่างเอนทิตีปกติและเอนทิตีอ่อนแอ เช่น ความสัมพันธ์แบบหนึ่งต่อกลุ่มของเอนทิตีอาจารย์ซึ่งเป็นเอนทิตีปกติและเอนทิตีตารางสอนซึ่งเป็นเอนทิตีอ่อนแอ ดังภาพที่ 3.7 ซึ่งเอนทิตีตารางสอนจะเกิดขึ้นและคงอยู่ได้ก็ต่อเมื่อมีการกำหนดเอนทิตีอาจารย์ขึ้นมาก่อน

ในอี-อาร์ไดอะแกรม สัญลักษณ์ที่ใช้แทนความสัมพันธ์แบบ Existence Dependency จะเป็นรูปสี่เหลี่ยมข้าวหลามตัดสองรูปซ้อนกัน



ภาพที่ 3.22 ความสัมพันธ์ แบบ Existence Dependency

จากภาพที่ 3.22 เอนทิตีอาจารย์และเอนทิตีตารางสอนมีความสัมพันธ์กันแบบหนึ่งต่อกลุ่ม และเป็นความสัมพันธ์แบบ Total Participation

ความสัมพันธ์แบบ Recursive เป็นความสัมพันธ์ที่เกิดจากเอนทิตีเพียงเอนทิตีเดียว เช่น ในห้องเรียนมีนักศึกษาหลายคน แต่มีนักศึกษาเพียงหนึ่งคนเท่านั้นที่เป็นหัวหน้าห้อง และหัวหน้าห้องเป็นผู้ประสานงานกับนักศึกษาหลายคน



ภาพที่ 3.23 ความสัมพันธ์แบบ Recursive

Composite Entities เป็นเอนทิตีที่ถูกสร้างขึ้นมาเพื่อใช้ในการแปลงความสัมพันธ์ระหว่าง เอนทิตีแบบกลุ่มต่อกลุ่มให้เป็นความสัมพันธ์แบบหนึ่งต่อกลุ่ม เนื่องจากความสัมพันธ์ระหว่างเอนทิตีแบบกลุ่มต่อกลุ่ม มักจะก่อให้เกิดความยุ่งยากเกี่ยวกับความซ้ำซ้อนในการแก้ไขปรับปรุงข้อมูลในการออกแบบฐานข้อมูล ดังนั้น จึงมีการสร้างเอนทิตีใหม่ขึ้นเรียกว่า Composite Entity หรือ Gerund

การสร้าง **Composite Entity** ซึ่งเป็นเอนทิตีที่เชื่อมความสัมพันธ์ระหว่างสองเอนทิตีเดิม ซึ่งเป็น ความสัมพันธ์แบบกลุ่มต่อกลุ่มให้เป็นความสัมพันธ์แบบหนึ่งต่อกลุ่ม ทำได้โดยนำ **Key Property** ของทั้งสองเอนทิตีที่มีความสัมพันธ์กันแบบกลุ่มต่อกลุ่มมารวมกับ **Property** อื่นที่สนใจเป็น **Property** ของ **Composite Entity** และกำหนดให้ **Key Property** ของทั้งสองเอนทิตีที่มีความสัมพันธ์กันแบบกลุ่มต่อกลุ่มเป็น **Key Property** ของ **Composite Entity**

ในอี-อาร์ไดอะแกรม ใช้สัญลักษณ์รูปสี่เหลี่ยมข้าวหลามตัดบรรจุในรูปสี่เหลี่ยมผืนผ้าที่มีชื่อของ **Composite Entity** นั้นกำกับอยู่ภายในแทนหนึ่ง **Composite Entity**

ตัวอย่างเช่น เอนทิตีนักศึกษาและเอนทิตีซุติวิชามีความสัมพันธ์กันแบบกลุ่มต่อกลุ่ม กล่าวคือ นักศึกษาแต่ละคนสามารถลงทะเบียนเรียนได้หลายซุติวิชา และแต่ละซุติวิชาสามารถมีนักศึกษาลงทะเบียนเรียนได้หลายคน ดังภาพที่ 3.24 การแปลงความสัมพันธ์ระหว่างเอนทิตีนักศึกษาและเอนทิตีซุติวิชาจาก ความสัมพันธ์แบบกลุ่มต่อกลุ่มให้เป็นความสัมพันธ์แบบหนึ่งต่อกลุ่มสามารถทำได้โดย การสร้าง **Composite Entity** การเรียน ซึ่งประกอบด้วย **Property** ต่าง ๆ คือ รหัสประจำตัว (key property ของเอนทิตีนักศึกษา) รหัสซุติวิชา (key property ของเอนทิตีซุติวิชา) และ **Property** อื่นที่สนใจ โดย **Composite Entity**



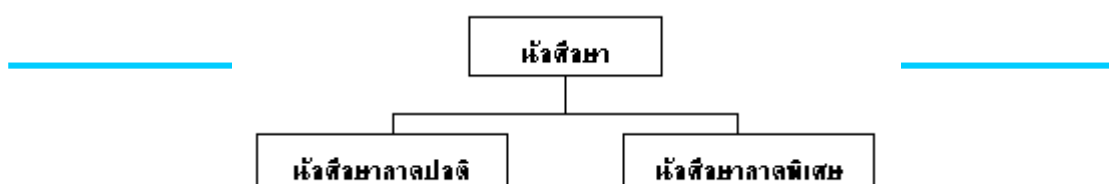
การเรียน จะมี **Property** รหัสประจำตัวและรหัสซุติวิชาเป็น **Key Property**

ภาพที่ 3.24 Composite Property การเรียน

4. Subtype และ Supertype

หากเอนทิตีใดสามารถจำแนกออกเป็นเอนทิตีย่อย ๆ ได้ โดยแต่ละสมาชิกในเอนทิตีย่อยนั้นสามารถมีคุณสมบัติ **Identity** ได้ด้วยตัวเอง เอนทิตีหลักที่ถูกนำมาจำแนกจะเรียกว่า **Supertype** ส่วนเอนทิตีย่อยที่จำแนกออกมาจะเรียกว่า **Subtype** หรืออีกนัยหนึ่งอาจกล่าวได้ว่า เอนทิตีหนึ่งจะเป็น **Subtype** ของอีกเอนทิตีหนึ่งที่เป็น **Supertype** ก็ต่อเมื่อ **Subtype** ประกอบด้วยทุก **Property** ที่มีใน **Supertype** โดย **Subtype** สามารถมี **Property** เพิ่มเติมจาก **Supertype** ได้

ตัวอย่างเช่น เอนทิตีนักศึกษาประกอบด้วย 4 **Property** คือ รหัสประจำตัว ชื่อ-สกุล เพศ และรหัสคณะ โดยมี **Property** รหัสประจำตัวเป็น **Identity** ส่วนเอนทิตีนักศึกษาภาคปกติซึ่งเป็น **Subtype** ของ **Supertype** เอนทิตีนักศึกษาอาจประกอบด้วย 6 **Property** คือ รหัสประจำตัว ชื่อ-สกุล เพศ รหัสคณะ ผู้สนับสนุนการศึกษา และจำนวนเงินที่ได้รับต่อเดือน โดยมี **Property** รหัสประจำตัวเป็น **Identity** และเอนทิตีนักศึกษาภาคพิเศษซึ่งเป็น **Subtype** ของ **Supertype** เอนทิตีนักศึกษาอาจประกอบด้วย 7 **Property** คือ รหัส



ประจำตัว ชื่อ-สกุล เพศ รหัสศัณະ ตำแหน่งงาน สถานที่ทำงาน และเงินเดือน โดยมี Property รหัส ประจำตัวเป็น Identity เป็นต้น

ภาพที่ 3.25 ภาพแสดง Subtype และ Supertype

ตอนที่ 3.2 อี-อาร์โมเดลกับฐานข้อมูลเชิงสัมพันธ์

หัวเรื่อง

- เรื่องที่ 3.2.1 การออกแบบฐานข้อมูลด้วยอี-อาร์โมเดล
- เรื่องที่ 3.2.2 ปัญหาจากการออกแบบฐานข้อมูลด้วยอี-อาร์โมเดล
- เรื่องที่ 3.2.3 การแปลงฐานข้อมูลที่ออกแบบด้วยอี-อาร์โมเดลเป็นฐานข้อมูลเชิงสัมพันธ์

แนวคิด

1. การออกแบบฐานข้อมูลด้วยอี-อาร์โมเดลประกอบด้วย 6 ขั้นตอน คือ การศึกษารายละเอียดและลักษณะหน้าที่งานของระบบ การกำหนดเอนทิตีที่ควรมีในระบบฐานข้อมูล การกำหนดความสัมพันธ์ระหว่างเอนทิตี การกำหนดคุณลักษณะของเอนทิตี การกำหนดคีย์หลักของแต่ละเอนทิตี และการนำสัญลักษณ์ที่ใช้ในอี-อาร์โมเดลมาอธิบายความสัมพันธ์ระหว่างข้อมูล
2. ปัญหาจากการออกแบบฐานข้อมูลด้วยอี-อาร์โมเดลส่วนมากมักเกิดจากการสร้างความสัมพันธ์ระหว่างเอนทิตี ตัวอย่างเช่น ปัญหา Fan Trap เป็นเรื่องเกี่ยวกับลักษณะการจัดความสัมพันธ์ระหว่างเอนทิตีทำให้เกิดความไม่ชัดเจนในการแสดงข้อมูลที่สนใจออกมา ซึ่งเกิดจากการที่เอนทิตีหนึ่งมีความสัมพันธ์แบบหนึ่งต่อกลุ่มกับเอนทิตีอื่นตั้งแต่สองเอนทิตีขึ้นไป และปัญหา Chasm Trap ซึ่งเกิดจากการที่เอนทิตีหนึ่งมีความสัมพันธ์กับเอนทิตีอื่นตั้งแต่สองเอนทิตีขึ้นไป แต่ไม่สามารถเชื่อมโยงความสัมพันธ์ระหว่างข้อมูลได้
3. การแปลงฐานข้อมูลที่ออกแบบด้วยอี-อาร์โมเดลเป็นฐานข้อมูลเชิงสัมพันธ์ ประกอบด้วย 3 ขั้นตอน คือ ขั้นที่หนึ่ง การแปลงเอนทิตีในอี-อาร์โมเดลให้เป็นรีเลชัน และการแปลงความสัมพันธ์ระหว่างเอนทิตีเป็นความสัมพันธ์ระหว่างรีเลชัน ขั้นที่สอง การแปลงรายละเอียดของเอนทิตีให้เป็นแอททริบิวต์ของรีเลชัน รวมทั้งการกำหนดคีย์หลักและคีย์นอกของแต่ละรีเลชัน และขั้นที่สาม การพิจารณาเค้าร่างข้อมูลในแต่ละรีเลชันที่ได้จากสองขั้นตอนแรก

วัตถุประสงค์

หลังจากศึกษาตอนที่ 3.2 แล้ว นักศึกษาสามารถ

1. บอกขั้นตอนการออกแบบฐานข้อมูลด้วยอี-อาร์โมเดลได้
2. บอกตัวอย่างของปัญหาจากการออกแบบฐานข้อมูลด้วยอี-อาร์โมเดลได้
3. บอกขั้นตอนการแปลงฐานข้อมูลที่ออกแบบด้วยอี-อาร์โมเดลเป็นฐานข้อมูลเชิงสัมพันธ์ได้

เรื่องที่ 3.2.1 การออกแบบฐานข้อมูลด้วยอี-อาร์โมเดล

ในการออกแบบฐานข้อมูลจำเป็นต้องทำการศึกษาคูณสมบัติและความสัมพันธ์ระหว่างข้อมูลที่มีอยู่ในระบบเพื่อให้ได้มาซึ่งโครงสร้างพื้นฐานของฐานข้อมูล ซึ่งโดยทั่วไปมักดำเนินการโดยใช้แบบจำลองข้อมูล

อี-อาร์โมเดลเป็นแบบจำลองข้อมูลที่มีความนิยมมากในการใช้เป็นเครื่องมือสำหรับงานออกแบบฐานข้อมูล โดยอี-อาร์โมเดลจะเสนอโครงสร้างของฐานข้อมูลในระดับแนวคิดออกมาในรูปของแผนภาพที่มีโครงสร้างง่ายต่อการทำความเข้าใจ ทำให้เห็นภาพรวมของเอนทิตีทั้งหมดและความสัมพันธ์ระหว่างเอนทิตีในระบบฐานข้อมูล

1. ขั้นตอนการออกแบบฐานข้อมูลด้วยอี-อาร์โมเดล

การออกแบบฐานข้อมูลด้วยอี-อาร์โมเดลประกอบด้วยขั้นตอนต่าง ๆ ดังนี้คือ

1. การศึกษารายละเอียดและลักษณะหน้าทำงานของระบบ

การศึกษารายละเอียดและลักษณะหน้าทำงานของระบบเป็นการศึกษาและรวบรวมรายละเอียด เกี่ยวกับลักษณะหน้าทำงานของระบบ ข้อมูลที่เกี่ยวข้อง ขั้นตอนในการทำงาน ตลอดจนข้อกำหนดและสมมติฐานต่าง ๆ ซึ่งอาจทำได้ด้วยการสัมภาษณ์หรือศึกษาจากแบบฟอร์มต่าง ๆ ที่มีการใช้งานอยู่ในระบบงานขณะนั้น

2. การกำหนดเอนทิตีที่ควรมีในระบบฐานข้อมูล

เนื่องจากฐานข้อมูลหนึ่ง ๆ อาจประกอบด้วยเอนทิตีต่าง ๆ ได้จำนวนมาก ดังนั้น ขั้นตอนนี้จึงเป็นการนำรายละเอียดในข้อ 1 มาทำการกำหนดเอนทิตีที่จำเป็นต้องมีอยู่ในระบบฐานข้อมูล โดยคำนึงถึงการเป็นเอนทิตีประเภทอ้อนแอ ตลอดจน Supertype หรือ Subtype ด้วย

3. การกำหนดความสัมพันธ์ระหว่างเอนทิตี

การกำหนดความสัมพันธ์ระหว่างเอนทิตีเป็นการกำหนดประเภทของความสัมพันธ์ระหว่างเอนทิตี โดยพิจารณาจากข้อกำหนดและสมมติฐานต่าง ๆ ที่ได้ทำการศึกษามาในข้อ 1

3.1 การกำหนดคุณลักษณะของเอนทิตี

การกำหนดคุณลักษณะของเอนทิตีเป็นการกำหนดว่า แต่ละเอนทิตีควรประกอบด้วย Property ไบบ้าง Property ใดที่มีคุณสมบัติเป็น Key Property หรือ Composite Property หรือ Derived Property

การกำหนดคีย์หลักของแต่ละเอนทิตี

การกำหนดคีย์หลักของแต่ละเอนทิตีเป็นการกำหนด Key Property ของแต่ละเอนทิตี เพื่อให้แต่ละสมาชิกในเอนทิตีสามารถมีคุณสมบัติเป็นเอกลักษณ์เฉพาะได้

3.3 การนำสัญลักษณ์ที่ใช้ในอี-อาร์โมเดลมาอธิบายความสัมพันธ์ระหว่างข้อมูล

การนำสัญลักษณ์ที่ใช้ในอี-อาร์โมเดลมาอธิบายความสัมพันธ์ระหว่างข้อมูลเป็นการนำรายละเอียดในขั้นตอนต่าง ๆ มาพิจารณาทบทวนเพื่อเพิ่มหรือลดเอนทิตี Property และความสัมพันธ์ต่าง ๆ จากนั้นจึงนำข้อมูลที่ได้จากขั้นตอนทั้งหมดมาเขียนเป็นแบบจำลองเพื่ออธิบายความสัมพันธ์ระหว่างข้อมูล

ด้วยสัญลักษณ์ต่าง ๆ หรืออี-อาร์ไดอะแกรม ดังนั้น แบบจำลองข้อมูลที่เกิดขึ้นจึงมีความชัดเจน สอดคล้อง ถูกต้องและเหมาะสมกับองค์ประกอบของงานที่กำลังศึกษาทำให้เป็นที่ยอมรับของทุกฝ่ายที่เกี่ยวข้อง

2. ตัวอย่างการออกแบบฐานข้อมูลด้วยอี-อาร์โมเดล

เพื่อแสดงให้เห็นถึงขั้นตอนการออกแบบฐานข้อมูลด้วยอี-อาร์โมเดล ในที่นี้ขอนำตัวอย่างภาระงานสอนของอาจารย์ในสถาบันการศึกษาแห่งหนึ่งมาประกอบดังนี้

การศึกษารายละเอียดและลักษณะหน้าที่งานของระบบ

สถาบันการศึกษาแห่งหนึ่งเปิดทำการสอนโดยแบ่งเป็น 3 คณะ คือ คณะเกษตร คณะวิทยาศาสตร์ และคณะศึกษาศาสตร์ ในแต่ละคณะประกอบด้วยภาควิชาต่าง ๆ และในแต่ละภาควิชาประกอบด้วยอาจารย์ได้หลายคน โดยอาจารย์แต่ละคนจะสังกัดภาควิชาใดภาควิชาหนึ่งเท่านั้น และมีอาจารย์เพียงคนเดียวเท่านั้นที่อยู่ในตำแหน่งหัวหน้าภาควิชาทำหน้าที่บริหารงานภาควิชา โดยในแต่ละภาควิชาจะเปิดทำการสอนได้หลายชุดวิชา ซึ่งอาจารย์แต่ละคนจะต้องสอนไม่น้อยกว่า 2 ชุดวิชา

การกำหนดเอนทิตีที่ควรมีในระบบฐานข้อมูล

เอนทิตีคณะเป็นเอนทิตีที่แสดงรายละเอียดของแต่ละคณะ

เอนทิตีภาควิชาเป็นเอนทิตีที่แสดงรายละเอียดของแต่ละภาควิชา

เอนทิตีอาจารย์เป็นเอนทิตีที่แสดงรายละเอียดของอาจารย์แต่ละคน

เอนทิตีชุดวิชาเป็นเอนทิตีที่แสดงรายละเอียดของแต่ละชุดวิชา

การกำหนดความสัมพันธ์ระหว่างเอนทิตี

ความสัมพันธ์สังกัด เป็นความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีคณะและเอนทิตีภาควิชา เนื่องจากในแต่ละคณะสามารถมีได้หลายภาควิชา และข้อมูลในเอนทิตีภาควิชาจะมีความสัมพันธ์กับเอนทิตีคณะแบบ Total Participation เนื่องจากแต่ละภาควิชาจะต้องสังกัดคณะใดคณะหนึ่ง



ภาพที่ 3.26 ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีคณะและเอนทิตีภาควิชา

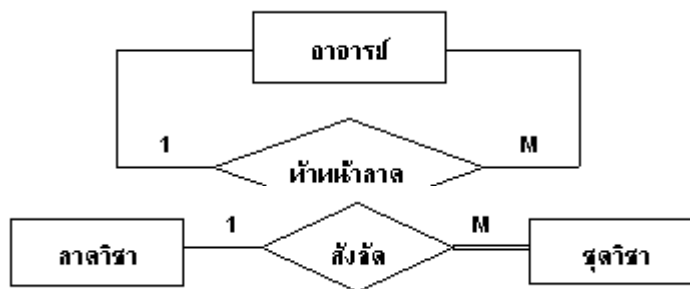
ความสัมพันธ์สังกัด เป็นความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชาและเอนทิตีอาจารย์ เนื่องจากในแต่ละภาควิชาสามารถมีอาจารย์ได้หลายคน และข้อมูลในเอนทิตีอาจารย์จะมี ความสัมพันธ์กับเอนทิตีภาควิชาแบบ Total Participation เนื่องจากอาจารย์แต่ละคนจะต้องสังกัดภาควิชาใดภาควิชาหนึ่ง



ภาพที่ 3.27 ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชาและเอนทิตีอาจารย์

ความสัมพันธ์หัวหน้าภาค เป็นความสัมพันธ์แบบ Recursive เนื่องจากในหนึ่งภาควิชามีอาจารย์ได้หลายคน แต่มีอาจารย์เพียงหนึ่งคนเท่านั้นที่ทำหน้าที่หัวหน้าภาควิชา

ภาพที่ 3.28 ความสัมพันธ์แบบ Recursive ของเอนทิตีอาจารย์



ความสัมพันธ์สังกัด เป็นความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชาและเอนทิตีชุดวิชา เนื่องจากในแต่ละภาควิชาสามารถเปิดสอนได้หลายชุดวิชา และข้อมูลของเอนทิตีชุดวิชามีความสัมพันธ์กับเอนทิตีภาควิชาแบบ Total Participation เนื่องจากแต่ละชุดวิชาจะต้องสังกัดภาควิชาใด ภาควิชาหนึ่ง

ภาพที่ 3.29 ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชาและเอนทิตีชุดวิชา

ความสัมพันธ์การสอน เป็นความสัมพันธ์แบบกลุ่มต่อกลุ่มระหว่างเอนทิตีชุดวิชาและเอนทิตีอาจารย์ โดยข้อมูลของเอนทิตีอาจารย์มีความสัมพันธ์กับเอนทิตีชุดวิชาแบบ Total Participation เนื่องจากในแต่ละชุดวิชาสามารถมีอาจารย์ผู้สอนได้หลายคน และอาจารย์แต่ละคนจะต้องสอนอย่างน้อย 2 ชุดวิชา



ภาพที่ 3.30 ความสัมพันธ์แบบกลุ่มต่อกลุ่มระหว่างเอนทิตีชุดวิชาและเอนทิตีอาจารย์

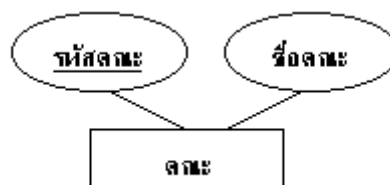
เนื่องจากความสัมพันธ์การสอนความสัมพันธ์การสอนเป็นความสัมพันธ์แบบกลุ่มต่อกลุ่มระหว่างเอนทิตีชุดวิชาและเอนทิตีอาจารย์ ดังนั้น จึงแปลงความสัมพันธ์ดังกล่าวเป็นความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อกลุ่มด้วยการสร้าง Composite Entity การสอน



ภาพที่ 3.31 ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีผู้ศึกษากับ Composite Entity การสอนและเอนทิตีอาจารย์กับ Composite Entity การสอน

4. การกำหนดคุณลักษณะของเอนทิตี

เอนทิตีคณะ ประกอบด้วย Property รหัสคณะ ชื่อคณะ



ภาพที่ 3.32 เอนทิตีคณะ

เอนทิตีภาควิชา ประกอบด้วย Property รหัสภาควิชา ชื่อภาควิชา ที่ทำการ



ภาพที่ 3.33 เอนทิตีภาควิชา

เอนทิตีอาจารย์ ประกอบด้วย Property รหัสอาจารย์ ชื่ออาจารย์ วันที่เริ่มทำงาน เงินเดือน



ภาพที่ 3.34 เอนทิตีอาจารย์

เอนทิตีชุดวิชา ประกอบด้วย Property รหัสชุดวิชา ชื่อชุดวิชา จำนวนหน่วยกิต



ภาพที่ 3.35 เอนทิตีชุดวิชา

ดังนั้น Composite Entity การสอน จะประกอบด้วย Property รหัสชุดวิชา รหัสอาจารย์

5. การกำหนดคีย์หลักของแต่ละเอนทิตี

เอนทิตีคณะ มีรหัสคณะเป็นคีย์หลัก

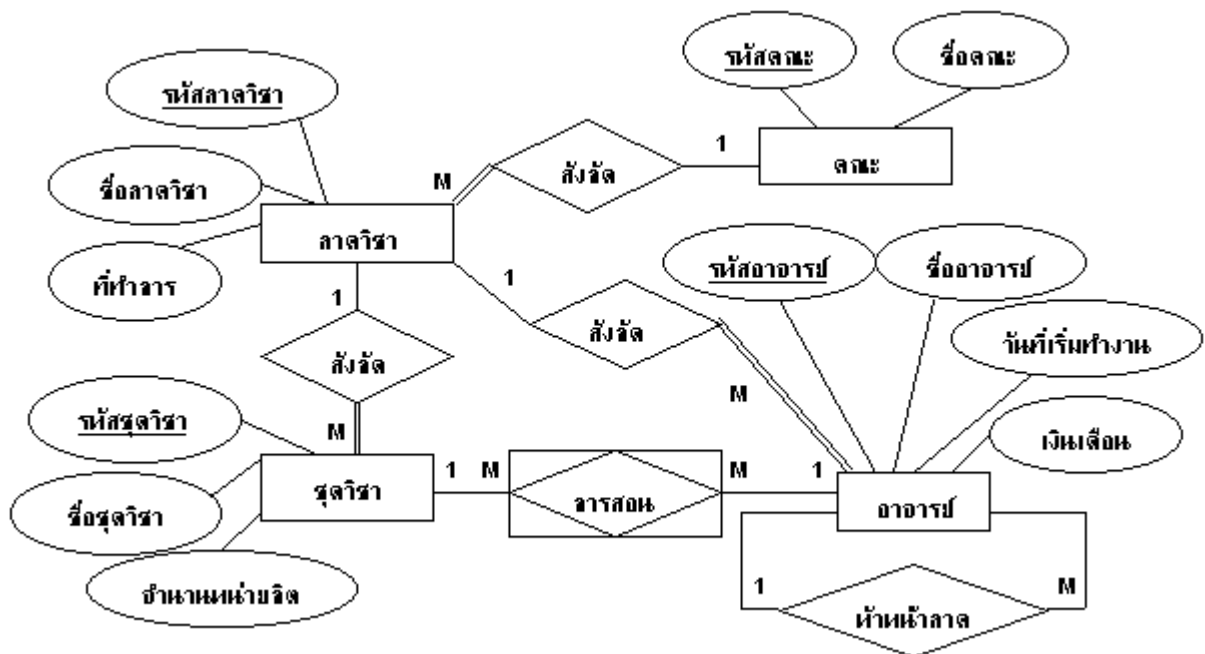
เอนทิตีภาควิชา มีรหัสภาควิชาเป็นคีย์หลัก

เอนทิตีอาจารย์ มีรหัสอาจารย์เป็นคีย์หลัก

เอนทิตีชุดวิชา มีรหัสชุดวิชาเป็นคีย์หลัก

ทั้งนี้ Composite Entity การสอน จะมีรหัสชุดวิชาและรหัสอาจารย์เป็นคีย์หลัก

6. การนำสัญลักษณ์ที่ใช้ในอี-อาร์โมเดล มาอธิบายความสัมพันธ์ระหว่างข้อมูล



ภาพที่ 3.36 อี-อาร์ไดอะแกรมภาระงานสอนของอาจารย์

ปัญหาที่อาจเกิดขึ้นจากการออกแบบฐานข้อมูลด้วยอี-อาร์โมเดลมักเป็นเรื่องเกี่ยวกับการสร้างความสัมพันธ์ระหว่างเอนทิตี ตัวอย่างเช่น ปัญหา Fan Trap และ Chasm Trap

1. Fan Trap

Fan Trap เป็นปัญหาที่เกี่ยวข้องกับลักษณะการจัดความสัมพันธ์ระหว่างเอนทิตี ทำให้เกิดความไม่ชัดเจนในการแสดงข้อมูลที่สนใจออกมา ซึ่งมักพบในกรณีที่เอนทิตีหนึ่งมีความสัมพันธ์แบบหนึ่งต่อกลุ่มกับเอนทิตีอื่นตั้งแต่สองเอนทิตีขึ้นไป

ตัวอย่างเช่น สถาบันการศึกษาแห่งหนึ่งเปิดทำการสอนหลายคณะ โดยในแต่ละคณะประกอบด้วย ภาควิชาต่าง ๆ หลายภาควิชา และอาจารย์หลายคน ซึ่งอาจารย์แต่ละคนจะสังกัดคณะใดคณะหนึ่งเท่านั้น หากกำหนดให้ความสัมพันธ์ระหว่างแต่ละเอนทิตีเป็นดังนี้

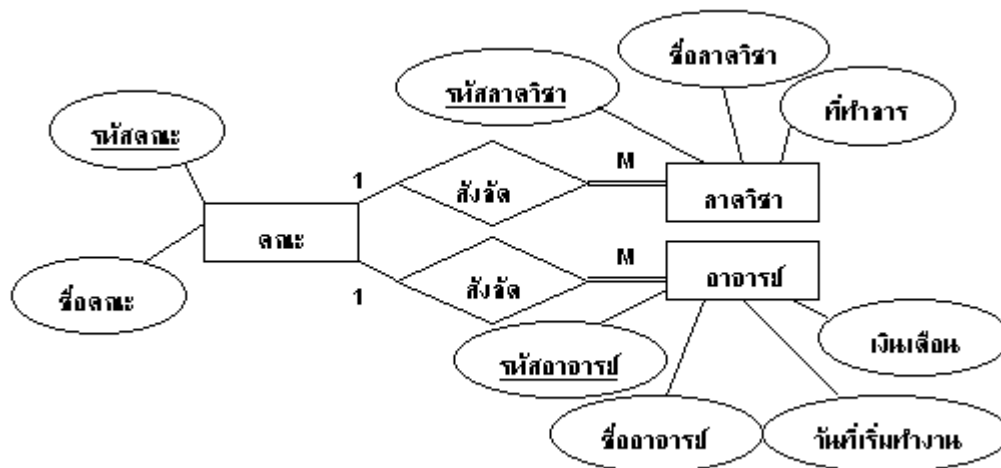


ภาพที่ 3.37 ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีคณะและเอนทิตีภาควิชา



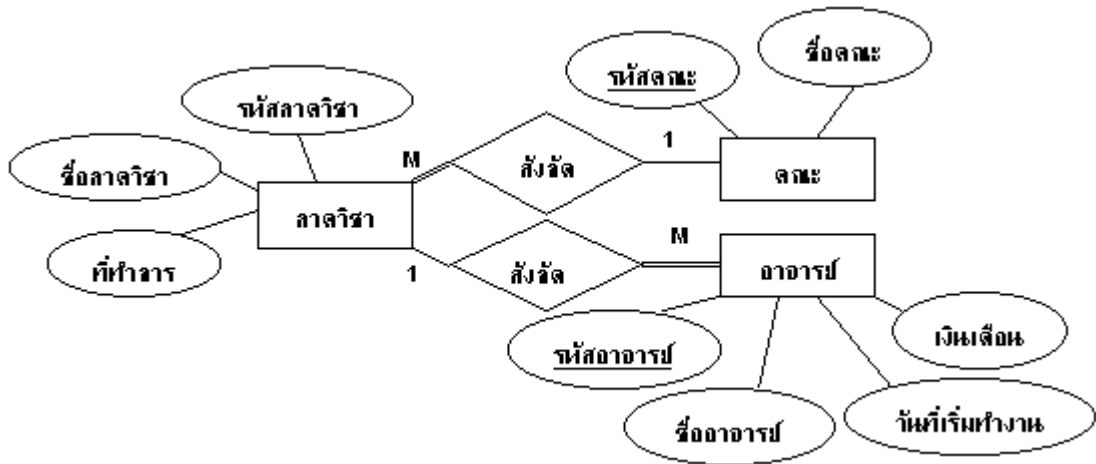
ภาพที่ 3.38 ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีคณะและเอนทิตีอาจารย์

เมื่อทำการกำหนดภาพรวมของเอนทิตีทั้งหมดและความสัมพันธ์ระหว่างเอนทิตีจะพบว่า ไม่สามารถตอบคำถามบางคำถามได้ เช่น อาจารย์คนใดสังกัดภาควิชาใด เนื่องจากไม่มีความสัมพันธ์ระหว่างเอนทิตีอาจารย์และเอนทิตีภาควิชา



ภาพที่ 3.39 ความสัมพันธ์ระหว่างเอนทิตีคณะกับเอนทิตีภาควิชาและเอนทิตีอาจารย์

ทั้งนี้ แนวทางในการแก้ปัญหาอาจทำได้โดยการปรับความสัมพันธ์ระหว่างแต่ละเอนทิตีใหม่เป็นดังนี้คือ ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีคณะกับเอนทิตีภาควิชา และความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชากับเอนทิตีอาจารย์ แล้วจึงทำการกำหนดภาพรวมของเอนทิตีทั้งหมดและ ความสัมพันธ์ระหว่างเอนทิตีใหม่ดังภาพ



ภาพที่ 3.40 ความสัมพันธ์ระหว่างเอนทิตีภาควิชากับเอนทิตีคณะและเอนทิตีอาจารย์

2. Chasm Trap

Chasm Trap เป็นปัญหาที่เกี่ยวข้องกับการแสดงข้อมูลที่เราจะมีไม่ได้ หรือไม่สามารถเชื่อมโยงเพื่อให้ได้ข้อมูลที่ครบถ้วนได้ โดยปัญหานี้มักพบเมื่อเอนทิตีหนึ่งมีความสัมพันธ์กับเอนทิตีอื่นตั้งแต่สองเอนทิตีขึ้นไป แต่ไม่สามารถเชื่อมโยงความสัมพันธ์ระหว่างข้อมูลได้

ตัวอย่างเช่น กำหนดให้แต่ละภาควิชาประกอบด้วยอาจารย์หลายคน ซึ่งอาจารย์แต่ละคนจะสังกัดภาควิชาใดภาควิชาหนึ่งเท่านั้น และอาจารย์แต่ละคนอาจทำการสอนได้มากกว่าหนึ่งชุดวิชา โดยแต่ละชุดวิชาอาจมีอาจารย์ผู้สอนมากกว่าหนึ่งคนก็ได้ หากกำหนดให้ความสัมพันธ์ระหว่างแต่ละเอนทิตีเป็นดังนี้

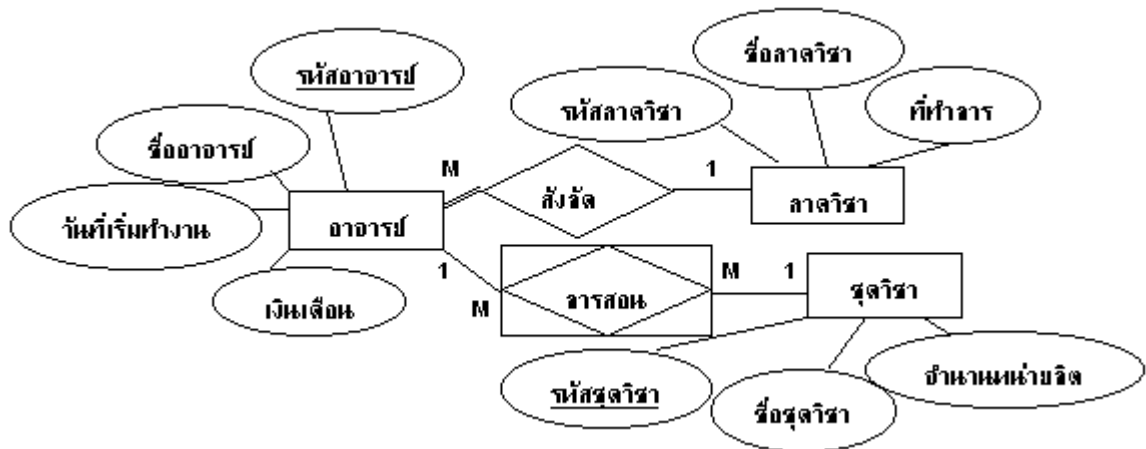


ภาพที่ 3.41 ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชาและเอนทิตีอาจารย์



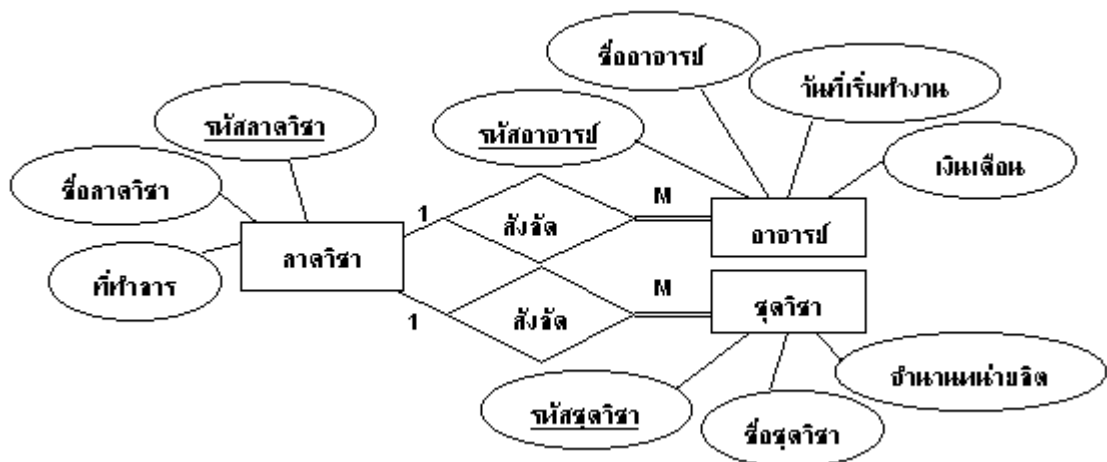
ภาพที่ 3.42 ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีอาจารย์กับ Composite Entity การสอนและเอนทิตีชุดวิชากับ Composite Entity การสอน

เมื่อทำการกำหนดภาพรวมของเอนทิตีทั้งหมดและความสัมพันธ์ระหว่างเอนทิตีจะพบว่า ไม่สามารถแสดงข้อมูลที่เราจะมีได้ หรือไม่สามารถเชื่อมโยงเพื่อให้ได้ข้อมูลที่ครบถ้วนได้ เช่น บางชุดวิชาที่มีอยู่อาจไม่เคยมีการเปิดสอนมาก่อนเลย ก็จะไม่มียาจารย์คนใดเคยสอน ซึ่งจะทำให้ไม่สามารถบอกได้ว่าชุดวิชานั้นอยู่ในสังกัดของภาควิชาใด หรือหากชุดวิชาใดที่มีอาจารย์รับผิดชอบการสอนเพียงคนเดียว หากอาจารย์ท่านนั้นลาออก ก็จะไม่สามารถเชื่อมโยงข้อมูลระหว่างชุดวิชาและภาควิชาได้เลย เนื่องจากไม่มีความสัมพันธ์ระหว่างเอนทิตีภาควิชาและเอนทิตีชุดวิชา



ภาพที่ 3.43 ความสัมพันธ์ระหว่างเอนทิตีอาจารย์กับเอนทิตีภาควิชาและเอนทิตีชุดวิชา

ทั้งนี้ แนวทางในการแก้ปัญหาอาจทำได้โดยการปรับความสัมพันธ์ระหว่างแต่ละเอนทิตีใหม่เป็นดังนี้คือ ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชากับเอนทิตีอาจารย์ และความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชากับเอนทิตีชุดวิชา แล้วจึงกำหนดภาพรวมของเอนทิตีทั้งหมดและความสัมพันธ์ระหว่างเอนทิตีใหม่ดังภาพ



ภาพที่ 3.44 ความสัมพันธ์ระหว่างเอนทิตีภาควิชากับเอนทิตีอาจารย์และเอนทิตีชุดวิชา

เรื่องที่ 3.2.2 การแปลงฐานข้อมูลที่ออกแบบด้วยอี-อาร์โมเดลเป็นฐานข้อมูลเชิงสัมพันธ์

เนื่องจากอี-อาร์โมเดลเป็นการออกแบบฐานข้อมูลในระดับแนวคิดซึ่งจะทำให้ทราบว่า ในฐานข้อมูลนั้นประกอบด้วยเอนทิตีและรายละเอียดอะไรบ้าง ดังนั้น เพื่อให้สามารถใช้งานฐานข้อมูลต่อไปได้ จึงต้องทำการแปลงความสัมพันธ์และอี-อาร์โมเดลให้อยู่ในรูปแบบของแบบจำลองอื่นที่สอดคล้องกับระบบจัดการฐานข้อมูลที่เลือกใช้ ซึ่งในที่นี้จะมุ่งเน้นให้ความสนใจกับฐานข้อมูลเชิงสัมพันธ์เท่านั้น

1. ขั้นตอนการแปลงฐานข้อมูลที่ออกแบบด้วยอี-อาร์โมเดลเป็นฐานข้อมูลเชิงสัมพันธ์

การแปลงฐานข้อมูลที่ออกแบบด้วยอี-อาร์โมเดลเป็นฐานข้อมูลเชิงสัมพันธ์ประกอบด้วย 3 ขั้นตอน ดังนี้คือ

1. การแปลงเอนทิตีในอี-อาร์โมเดลให้เป็นรีเลชัน และการแปลงความสัมพันธ์ระหว่าง เอนทิตีเป็นความสัมพันธ์ระหว่างรีเลชัน

- การแปลงเอนทิตีในอี-อาร์โมเดลให้เป็นรีเลชันในฐานข้อมูลเชิงสัมพันธ์ และการแปลงความสัมพันธ์ระหว่างเอนทิตีเป็นความสัมพันธ์ระหว่างรีเลชันในฐานข้อมูลเชิงสัมพันธ์ ประกอบด้วยขั้นตอนดังนี้คือ
 - การแปลงเอนทิตีในอี-อาร์โมเดลเป็นรีเลชันในฐานข้อมูลเชิงสัมพันธ์ โดยการกำหนดชื่อของ เอนทิตีเป็นชื่อของรีเลชัน
 - การแปลงความสัมพันธ์ระหว่างเอนทิตีเป็นความสัมพันธ์ระหว่างรีเลชันในฐานข้อมูลเชิงสัมพันธ์ หากความสัมพันธ์ระหว่างเอนทิตีในอี-อาร์โมเดลเป็นความสัมพันธ์แบบกลุ่มต่อกลุ่ม จะมีการแปลง ความสัมพันธ์ดังกล่าวเป็นความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อกลุ่มด้วยการสร้าง Composite Entity ขึ้น จากนั้นจึงทำการแปลง Composite Entity ที่สร้างขึ้นให้เป็นรีเลชันในฐานข้อมูลเชิงสัมพันธ์ โดยการกำหนดชื่อของ Composite Entity เป็นชื่อของรีเลชัน

2. การแปลงรายละเอียดของเอนทิตีให้เป็นแอททริบิวต์ของรีเลชัน รวมทั้งการกำหนดคีย์หลักและคีย์นอกของแต่ละรีเลชัน

- จากขั้นตอนที่ 1 เมื่อทราบรีเลชันและความสัมพันธ์ระหว่างรีเลชันแล้ว ในขั้นตอนที่ 2 นี้จะเป็นการแปลง Property ของแต่ละเอนทิตีให้เป็นแอททริบิวต์ของแต่ละรีเลชัน รวมทั้งทำการกำหนดคีย์หลักในแต่ละรีเลชัน ตลอดจนกำหนดคีย์นอกที่อ้างอิงถึงคีย์หลักในอีกรีเลชันหนึ่งหากจำเป็นต้องมี และกำหนดข้อจำกัดต่าง ๆ ของคีย์หลักและคีย์นอกในแต่ละรีเลชัน
- การแปลงรายละเอียดของเอนทิตีให้เป็นแอททริบิวต์ของรีเลชัน รวมทั้งการกำหนดคีย์หลักและคีย์นอกของแต่ละรีเลชัน ประกอบด้วยขั้นตอนดังนี้คือ
 1. การแปลงรายละเอียดของแต่ละเอนทิตีให้เป็นแอททริบิวต์ของแต่ละรีเลชัน ทำได้โดยกำหนดให้ Property ของแต่ละเอนทิตีเป็นแอททริบิวต์ของแต่ละรีเลชัน และ Key Property ของแต่ละเอนทิตีเป็นคีย์หลักของแต่ละรีเลชัน
 2. การกำหนดคีย์นอกของแต่ละรีเลชันสามารถทำได้โดย
 - หากความสัมพันธ์ระหว่างรีเลชันเป็นความสัมพันธ์แบบหนึ่งต่อหนึ่ง ให้เพิ่มคีย์หลักของรีเลชันหนึ่งลงไปเป็นแอททริบิวต์หนึ่งของอีกรีเลชัน

- หากความสัมพันธ์ระหว่างรีเลชันเป็นความสัมพันธ์แบบหนึ่งต่อกลุ่ม ให้เพิ่มคีย์หลักของรีเลชันที่อยู่ด้านความสัมพันธ์เป็นหนึ่งไปเป็นแอททริบิวต์หนึ่งของรีเลชันด้านที่มีความสัมพันธ์เป็นกลุ่ม ทั้งนี้กรณีของรีเลชันที่แปลงมาจาก Composite Entity จะปรากฏแอททริบิวต์ดังกล่าวอยู่แล้ว

- หากรีเลชันมีความสัมพันธ์แบบ Recursive ให้เพิ่มคีย์หลักของรีเลชันที่อยู่ด้านความสัมพันธ์เป็นหนึ่งไปเป็นแอททริบิวต์หนึ่งของรีเลชันด้านที่มีความสัมพันธ์เป็นกลุ่ม โดยเปลี่ยนชื่อของแอททริบิวต์นั้นใหม่

- การพิจารณาเค้าร่างข้อมูลในแต่ละรีเลชันที่ได้จากสองขั้นตอนแรก

ขั้นตอนนี้เป็นขั้นตอนการทบทวนเค้าร่างข้อมูลในแต่ละรีเลชันที่ได้มาจากสองขั้นตอนแรก โดยทำการพิจารณาเรื่องของการซ้ำซ้อนหรือปัญหาที่อาจเกิดขึ้นในการเพิ่มเติม ลบ หรือปรับปรุงข้อมูล

2. ตัวอย่างการแปลงฐานข้อมูลที่ออกแบบด้วยอี-อาร์โมเดลเป็นฐานข้อมูลเชิงสัมพันธ์

เพื่อแสดงให้เห็นถึงขั้นตอนการแปลงฐานข้อมูลที่ออกแบบด้วยอี-อาร์โมเดลเป็นฐานข้อมูลเชิงสัมพันธ์ ในที่นี้ขอเสนอตัวอย่างการออกแบบฐานข้อมูลภาระงานสอนของอาจารย์ด้วยอี-อาร์โมเดลในเรื่องที่ 3.2.1 มาประกอบดังนี้

1. การแปลงเอนทิตีในอี-อาร์โมเดลให้เป็นรีเลชัน และการแปลงความสัมพันธ์ระหว่าง เอนทิตีเป็นความสัมพันธ์ระหว่างรีเลชัน

จากตัวอย่างการออกแบบฐานข้อมูลภาระงานสอนของอาจารย์ด้วยอี-อาร์โมเดลจะพบว่าฐานข้อมูลดังกล่าวประกอบด้วย 4 เอนทิตี คือ

1. เอนทิตีคณะ ประกอบด้วย Property รหัสคณะ ชื่อคณะ
2. เอนทิตีภาควิชา ประกอบด้วย Property รหัสภาควิชา ชื่อภาควิชา ที่ทำการ
3. เอนทิตีอาจารย์ ประกอบด้วย Property รหัสอาจารย์ ชื่ออาจารย์ วันที่เริ่มทำงาน เงินเดือน
4. เอนทิตีชุดวิชา ประกอบด้วย Property รหัสชุดวิชา ชื่อชุดวิชา จำนวนหน่วยกิต

และฐานข้อมูลดังกล่าวมีความสัมพันธ์ระหว่างเอนทิตี 5 ความสัมพันธ์ คือ

1. ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีคณะและเอนทิตีภาควิชา
2. ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชาและเอนทิตีอาจารย์
3. ความสัมพันธ์แบบ Recursive ของเอนทิตีอาจารย์
4. ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชาและเอนทิตีชุดวิชา
5. ความสัมพันธ์แบบกลุ่มต่อกลุ่มระหว่างเอนทิตีชุดวิชาและเอนทิตีอาจารย์

เนื่องจากความสัมพันธ์ระหว่างเอนทิตีชุดวิชาและเอนทิตีอาจารย์เป็นความสัมพันธ์แบบกลุ่มต่อกลุ่ม ดังนั้น จึงมีการแปลงความสัมพันธ์ดังกล่าวเป็นความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อกลุ่ม โดยการสร้าง Composite Entity การสอน ซึ่งประกอบด้วย Property รหัสชุดวิชา รหัสอาจารย์ ขึ้นมา

ดังนั้น ฐานข้อมูลเชิงสัมพันธ์ภาระงานสอนของอาจารย์จะประกอบด้วย 5 รีเลชัน คือ รีเลชันคณะ รีเลชันภาควิชา รีเลชันอาจารย์ รีเลชันชุดวิชา และรีเลชันการสอน โดยมีความสัมพันธ์ระหว่างรีเลชัน 6 ความสัมพันธ์ คือ

1. ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างรีเลชันคณะและรีเลชันภาควิชา
2. ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างรีเลชันภาควิชาและรีเลชันอาจารย์
3. ความสัมพันธ์แบบ Recursive ของรีเลชันอาจารย์
4. ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างรีเลชันภาควิชาและรีเลชันชุดวิชา
5. ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างรีเลชันชุดวิชาและรีเลชันการสอน
6. ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างรีเลชันอาจารย์และรีเลชันการสอน

2. การแปลงรายละเอียดของเอนทิตีให้เป็นแอททริบิวต์ของรีเลชัน รวมทั้งการกำหนดคีย์หลักและคีย์นอกของแต่ละรีเลชัน

ดังนั้น จากตัวอย่างภาระงานสอนของอาจารย์ที่กล่าวมาข้างต้น แต่ละรีเลชันจะประกอบด้วยแอททริบิวต์ คีย์หลัก และคีย์นอก ดังต่อไปนี้คือ

1. รีเลชันคณะ ประกอบด้วยแอททริบิวต์ รหัสคณะ ชื่อคณะ โดยมีแอททริบิวต์รหัสคณะเป็นคีย์หลัก

2. รีเลชันภาควิชา ประกอบด้วยแอททริบิวต์ รหัสภาควิชา ชื่อภาควิชา ที่ทำการ โดยมีแอททริบิวต์รหัสภาควิชาเป็นคีย์หลัก และเนื่องจากรีเลชันคณะมีความสัมพันธ์แบบหนึ่งต่อกลุ่มกับรีเลชันภาควิชา รีเลชันภาควิชาจึงต้องมีแอททริบิวต์รหัสคณะเพิ่มขึ้นด้วย โดยแอททริบิวต์รหัสคณะเป็นคีย์นอกที่ใช้ในการเชื่อมโยงข้อมูลกับรีเลชันคณะ

3. รีเลชันอาจารย์ ประกอบด้วยแอททริบิวต์ รหัสอาจารย์ ชื่ออาจารย์ วันที่เริ่มทำงาน เงินเดือน โดยมีแอททริบิวต์รหัสอาจารย์เป็นคีย์หลัก และเนื่องจากรีเลชันภาควิชามีความสัมพันธ์แบบหนึ่งต่อกลุ่มกับ รีเลชันอาจารย์ รีเลชันอาจารย์จึงต้องมีแอททริบิวต์รหัสภาควิชาเพิ่มขึ้นด้วย โดยแอททริบิวต์รหัสภาควิชาเป็นคีย์นอกที่ใช้ในการเชื่อมโยงข้อมูลกับรีเลชันภาควิชา

เนื่องจากรีเลชันอาจารย์มีความสัมพันธ์แบบ Recursive เป็นความสัมพันธ์หัวหน้าภาคปรากฏอยู่อีกหนึ่งความสัมพันธ์ ฉะนั้น รีเลชันอาจารย์จึงต้องมีแอททริบิวต์ที่แสดงรหัสของอาจารย์ที่เป็นหัวหน้าภาควิชาเพิ่มขึ้นอีกหนึ่งแอททริบิวต์ ซึ่งแอททริบิวต์ดังกล่าวตั้งชื่อว่า รหัสหัวหน้าภาค

4. รีเลชันชุดวิชา ประกอบด้วยแอททริบิวต์ รหัสชุดวิชา ชื่อชุดวิชา จำนวนหน่วยกิต โดยมีแอททริบิวต์รหัสชุดวิชาเป็นคีย์หลัก และเนื่องจากรีเลชันภาควิชามีความสัมพันธ์แบบหนึ่งต่อกลุ่มกับรีเลชันชุดวิชา รีเลชันชุดวิชาจึงต้องมีแอททริบิวต์รหัสภาควิชาเพิ่มขึ้นด้วย โดยแอททริบิวต์รหัสภาควิชาเป็นคีย์นอกที่ใช้ในการเชื่อมโยงข้อมูลกับรีเลชันภาควิชา

5. รีเลชันการสอน ประกอบด้วยแอททริบิวต์ รหัสชุดวิชา รหัสอาจารย์ โดยมีแอททริบิวต์รหัสชุดวิชาและรหัสอาจารย์เป็นคีย์หลัก แอททริบิวต์รหัสชุดวิชาเป็นคีย์นอกที่ใช้ในการเชื่อมโยงข้อมูลกับรีเลชันชุดวิชา และแอททริบิวต์รหัสอาจารย์เป็นคีย์นอกที่ใช้ในการเชื่อมโยงข้อมูลกับรีเลชันอาจารย์

3. การพิจารณาเค้าร่างข้อมูลในแต่ละรีเลชันที่ได้จากสองขั้นตอนแรก

ดังนั้น จากตัวอย่างการออกแบบฐานข้อมูลภาระงานสอนของอาจารย์ด้วยอี-อาร์โมเดลในเรื่องที่

3.2.1 เมื่อทำการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์จะประกอบด้วย รีเลชัน แอททริบิวต์ และคีย์ดังต่อไปนี้คือ

- รีเลชันคณะ ประกอบด้วยแอททริบิวต์ รหัสคณะ ชื่อคณะ โดยมีแอททริบิวต์รหัสคณะเป็นคีย์หลัก

รหัสคณะ	ชื่อคณะ
---------	---------

- รีเลชันภาควิชา ประกอบด้วยแอททริบิวต์ รหัสภาควิชา ชื่อภาควิชา ที่ทำการ รหัสคณะ โดยมีแอททริบิวต์รหัสภาควิชาเป็นคีย์หลัก และแอททริบิวต์รหัสคณะเป็นคีย์นอกที่ใช้ในการเชื่อมโยงข้อมูลกับรีเลชันคณะ

รหัสภาควิชา	ชื่อภาควิชา	ที่ทำการ	รหัสคณะ
-------------	-------------	----------	---------

- รีเลชันอาจารย์ ประกอบด้วยแอททริบิวต์ รหัสอาจารย์ ชื่ออาจารย์ วันที่เริ่มทำงาน เงินเดือน รหัสภาควิชา รหัสหัวหน้าภาค โดยมีแอททริบิวต์รหัสอาจารย์เป็นคีย์หลัก และแอททริบิวต์รหัสภาควิชาเป็นคีย์นอกที่ใช้ในการเชื่อมโยงข้อมูลกับรีเลชันภาควิชา

รหัสอาจารย์	ชื่ออาจารย์	วันที่เริ่มทำงาน	เงินเดือน	รหัสภาควิชา	รหัสหัวหน้าภาค
-------------	-------------	------------------	-----------	-------------	----------------

- รีเลชันชุดวิชา ประกอบด้วยแอททริบิวต์ รหัสชุดวิชา ชื่อชุดวิชา จำนวนหน่วยกิต รหัสภาควิชา โดยมีแอททริบิวต์รหัสชุดวิชาเป็นคีย์หลัก และแอททริบิวต์รหัสภาควิชาเป็นคีย์นอกที่ใช้ในการเชื่อมโยงข้อมูลกับรีเลชันภาควิชา

รหัสชุดวิชา	ชื่อชุดวิชา	จำนวนหน่วยกิต	รหัสภาควิชา
-------------	-------------	---------------	-------------

- รีเลชันการสอน ประกอบด้วยแอททริบิวต์ รหัสชุดวิชา รหัสอาจารย์ โดยมีแอททริบิวต์รหัสชุดวิชาและรหัสอาจารย์เป็นคีย์หลัก โดยมีแอททริบิวต์รหัสชุดวิชาเป็นคีย์นอกที่ใช้ในการเชื่อมโยงข้อมูลกับรีเลชันชุดวิชา และแอททริบิวต์รหัสอาจารย์เป็นคีย์นอกที่ใช้ในการเชื่อมโยงข้อมูลกับรีเลชัน อาจารย์

รหัสชุดวิชา	รหัสอาจารย์
-------------	-------------