

บทที่ 4

แบบจำลองอี อาร์ (Entity-Relationship Model)

จุดประสงค์เชิงพฤติกรรม

1. อธิบายความหมายของแบบจำลองอี อาร์ ได้
2. อธิบายแบบจำลองแบบ Chen Model ได้
3. อธิบายขั้นตอนการแปลงแบบจำลองอี อาร์ เป็นฐานข้อมูลเชิงสัมพันธ์ได้
4. ออกแบบฐานข้อมูลและเขียนแบบจำลองอี อาร์ ได้

บทนำ

แบบจำลองอี อาร์ โมเดล เป็นการทำงานอยู่ในขั้นตอนของการออกแบบเชิงแนวคิด ซึ่งนักออกแบบฐานข้อมูลจะเขียนแผนภาพจำลองฐานข้อมูล และใช้เป็นเครื่องมือสำหรับสื่อสารทำความเข้าใจระหว่างนักออกแบบฐานข้อมูลกับผู้ใช้ ให้เข้าใจตรงกันว่าฐานข้อมูลที่ออกแบบไว้มีโครงสร้างการออกแบบอย่างไร นักออกแบบฐานข้อมูลกับผู้ใช้จะช่วยกันในการพัฒนาและออกแบบฐานข้อมูล ซึ่งกระบวนการในการทำอาจจะมีการทำซ้ำหลาย ๆ รอบเพื่อให้ได้ฐานข้อมูลที่ครบถ้วนตรงตามความต้องการและครอบคลุมการปฏิบัติงานของหน่วยงาน จากที่กล่าวมาเบื้องต้นจะได้อธิบายรายละเอียดในบทที่ 4 ซึ่งมีเนื้อหาเกี่ยวกับ ความหมายของแบบจำลองอี อาร์ การออกแบบฐานข้อมูลด้วยการใช้แบบจำลองอี อาร์ แบบ Chen Model การแปลงอี อาร์ แบบ Chen Model ไปเป็นตาราง

4.1 ความหมายของแบบจำลองอี อาร์ (E-R Model)

แบบจำลองอี อาร์ มีนักวิชาการหลาย ๆ ท่านที่ได้ให้ความหมายของแบบจำลองอี อาร์ ไว้ดังนี้

แบบจำลองอี อาร์ หมายถึง แบบจำลองที่นำมาใช้เป็นตัวแทนของแบบจำลองเชิงแนวคิด เพื่อสื่อถึงเรื่องราวเกี่ยวกับข้อมูลและความสัมพันธ์ระหว่างข้อมูลในระบบงานหนึ่ง ๆ เพื่อสื่อสารความหมายตรงกันระหว่างผู้ออกแบบกับผู้ใช้ได้อย่างเข้าใจ (โอภาส เอี่ยมสิริวงศ์, 2558, หน้า 170)

แบบจำลองอี อาร์ หมายถึง แผนภาพที่แสดงรายละเอียดและแสดงความสัมพันธ์ระหว่างข้อมูลของระบบ โดยนำรูปภาพหรือแผนภาพเข้ามาช่วยในการนำเสนอรายละเอียดของฐานข้อมูล (สุจิตรา อดุลย์เกษม, 2553, หน้า 96)

แบบจำลองอี อาร์ หมายถึง เครื่องมือที่ถูกสร้างขึ้นให้เป็นภาพของข้อมูลที่ต้องจัดเก็บ เหตุการณ์ที่สามารถเกิดขึ้นได้และแสดงถึงความสัมพันธ์ซึ่งกันและกันเพื่อให้เกิดเป็นแนวคิดและกฎกติกาที่ใช้สื่อสารระหว่างผู้ใช้งานกับนักพัฒนาระบบฐานข้อมูล (Neejaj Sharma et al., 2555, Page 62)

แบบจำลองอี อาร์ หมายถึง แบบที่แสดงโครงสร้างข้อมูลในรูปแบบกราฟิก มีส่วนประกอบหลัก 3 ส่วน คือ เอนทิตี เป็นวัตถุหรือสิ่งที่เราสนใจ แอตทริบิวต์ เป็นคุณสมบัติของวัตถุที่สนใจ และ ความสัมพันธ์ของเอนทิตี (Hector Garcia-Molina, Jeffery D. Ullman and Jennifer Widom (2009, Page 126)

สรุปได้ว่า แบบจำลองอี อาร์ หมายถึง แผนภาพที่นักออกแบบฐานข้อมูลเขียนขึ้นมาเพื่อจำลองโครงสร้างและความสัมพันธ์ข้อมูลในของฐานข้อมูล ซึ่งประกอบด้วย เอนทิตี แอตทริบิวต์

ความสัมพันธ์ระหว่างข้อมูลในฐานข้อมูลและเงื่อนไขในการบรรจุข้อมูลในฐานข้อมูล และใช้สื่อสารเพื่อทำความเข้าใจร่วมกันกับผู้ใช้และร่วมกันพัฒนาจนได้แผนภาพที่จำลองฐานข้อมูลที่ตรงกับความต้องการใช้งานของผู้ใช้และลักษณะขององค์กร

แบบจำลองอี อาร์ ถูกแสดงในรูปแบบของแผนภาพแสดงความสัมพันธ์ระหว่างข้อมูล หรือเรียกว่า E-R Diagram เป็นแบบเครื่องมือที่สามารถแทนเอนทิตี แอตทริบิวต์ และความสัมพันธ์ระหว่างเอนทิตี แบบจำลองอี อาร์ จะไม่ยึดติดกับระบบจัดการฐานข้อมูลแต่อย่างใด สามารถนำไปใช้ได้กับทุกระบบจัดการฐานข้อมูล แบบจำลองอี อาร์ มีหลายรูปแบบ และแต่ละรูปแบบจะมีสัญลักษณ์ที่แตกต่างกัน ในบทนี้จะขอกล่าวถึงแบบจำลองแบบ Chen Model

4.2 แบบจำลองแบบ Chen Model

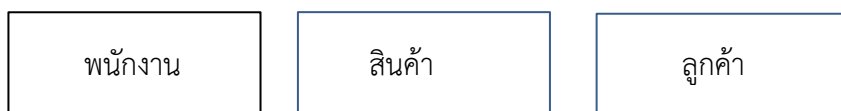
แบบจำลองแบบ Chen Model ถูกพัฒนาขึ้นในปี ค.ศ. 1976 โดยปีเตอร์ เชน อี อาร์ โมเดล นำเสนอด้วยรูปของแผนภาพโดยใช้สัญลักษณ์แทนความหมายของสิ่งที่ต้องมีในฐานข้อมูล (ครรชิต มาลัยวงศ์ สมลักษณ์ ละอองศรี และทัศนีย์วรรณ ศรีประดิษฐ์, 2544, หน้า 245) มีขั้นตอนในการการออกแบบดังนี้

4.2.1 ส่วนประกอบในการเขียนแบบจำลองอี อาร์ แบบ Chen Model มีดังนี้

แบบจำลองแบบ Chen Model มีส่วนประกอบของการเขียนดังนี้

1. เอนทิตี (Entity)

หมายถึง ตารางที่ใช้ในการจัดเก็บข้อมูลที่สนใจ ซึ่งจะรวบรวมเป็นข้อมูลและระบุรายละเอียดของข้อมูลที่จัดเก็บ อาทิ เอนทิตีพนักงาน เอนทิตีสินค้า เอนทิตีลูกค้า เอนทิตีการขายสินค้า ฯลฯ สัญลักษณ์ของเอนทิตีใน แบบจำลองอี อาร์ คือรูปสี่เหลี่ยมผืนผ้า และมีชื่อกำกับภายในเป็นคำนาม เช่น



เอนทิตี ใน Chen Model แบ่งออกเป็น 3 ประเภท คือ

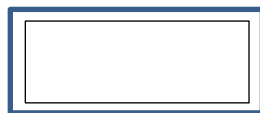
1) เอนทิตีปกติ (Strong Entity) หรือบางครั้งเรียกว่า Regular Entity เป็นเอนทิตีอิสระไม่ต้องพึ่งพิงอาศัย Entity อื่นในการคงอยู่ หรือเอนทิตีที่มีการทำงานในระบบนั่นเอง สัญลักษณ์ที่ใช้ คือ สี่เหลี่ยมผืนผ้า



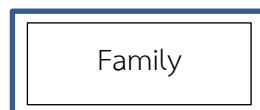
ตัวอย่างเช่น เอนทิตีพนักงาน (Employees) เอนทิตีสินค้า (Product) เป็น Strong Entity เขียนได้ดังนี้



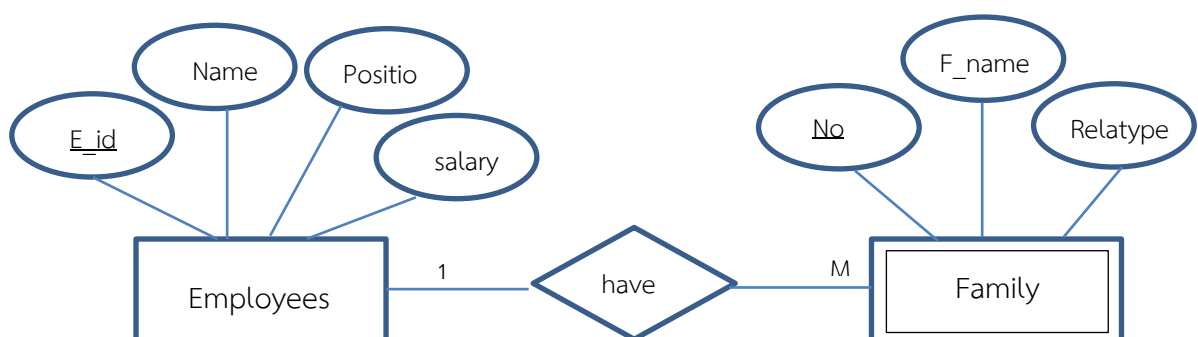
2) เอนทิตีอ่อนแอ (Weak Entity) คือเอนทิตีที่ขึ้นอยู่กับเอนทิตีอื่น จะปรากฏในฐานข้อมูลก็ต่อเมื่อมีเอนทิตีหลักที่สัมพันธ์กันในฐานข้อมูลด้วย สัญลักษณ์ที่ใช้แทนคือ รูปสี่เหลี่ยมผืนผ้าเส้นคู่



ตัวอย่างเช่น ถ้าในบริษัทกำหนดว่า พนักงานทุกคนที่รับเข้ามาทำงานจะมีสวัสดิการให้กับบุตรและครอบครัวของพนักงานด้วย ดังนั้น เอนทิตีพนักงานจะเป็น Strong Entity ส่วน เอนทิตีครอบครัวของพนักงานจะเป็น Weak Entity เขียนได้ดังนี้



เอนทิตี Family ซึ่งเป็น Weak Entity จะเชื่อมความสัมพันธ์เข้ากับ เอนทิตี พนักงานงาน ซึ่งเป็น Strong Entity แบบ 1:M ดังนี้



ภาพที่ 4.1 E-R Model ของความสัมพันธ์ระหว่าง Strong Entity และ Weak Entity

เมื่อทำการแปลงภาพอี อาร์ ตามเงื่อนไขของการแปลงอี อาร์ ไปเป็น รีเลชัน จะได้ เอนทิตีและแอตทริบิวต์ ดังนี้

Employees(E_id, Name, Position, Salary)

Family(E_id, No, F_name, Relatype)

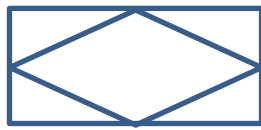
และเมื่อแปลงให้เป็นตารางในการจัดเก็บข้อมูลจะทำให้เห็นความสัมพันธ์ระหว่าง เอนทิตี Employees และ เอนทิตี Family คือ เมื่อทราบข้อมูลพนักงานคือพนักงานรหัส 1001 ลักษณะพร สุริยะ จะทำให้ทราบว่า พนักงานมี สมาชิกในครอบครัว 3 คน คือ ลำดับที่ 1 เด็กหญิงนิชา สุริยะ เกี่ยวข้องกับ บุตร ลำดับที่ 2 คือ เด็กชายนิติ สุริยะ เป็นบุตร และ 3 คือ นายวันชัย สุริยะ เป็นสามี ดังภาพต่อไปนี้

Entity: Employees			
<u>E_id</u>	Name	Position	Salary
1001	ลักษณะพร สุริยะ	บัญชี	27000
1002	สมใจ รักษา	การตลาด	24200
1003	วิชัย ธรรมเรือง	ไอที	29000

Entity: Family			
<u>E_id</u>	<u>No</u>	F_name	Relatype
1001	1	เด็กหญิงนิชา สุริยะ	บุตร
1001	2	เด็กชายนิติ สุริยะ	บุตร
1001	3	นายวันชัย สุริยะ	สามี
1002	1	นายทนง รักษา	สามี
1002	2	นางนภา ไชยดา	มารดา

ภาพที่ 4.2 ตัวอย่างของตารางเก็บข้อมูลที่เกิดจากความสัมพันธ์ระหว่าง Strong Entity และ Weak Entity

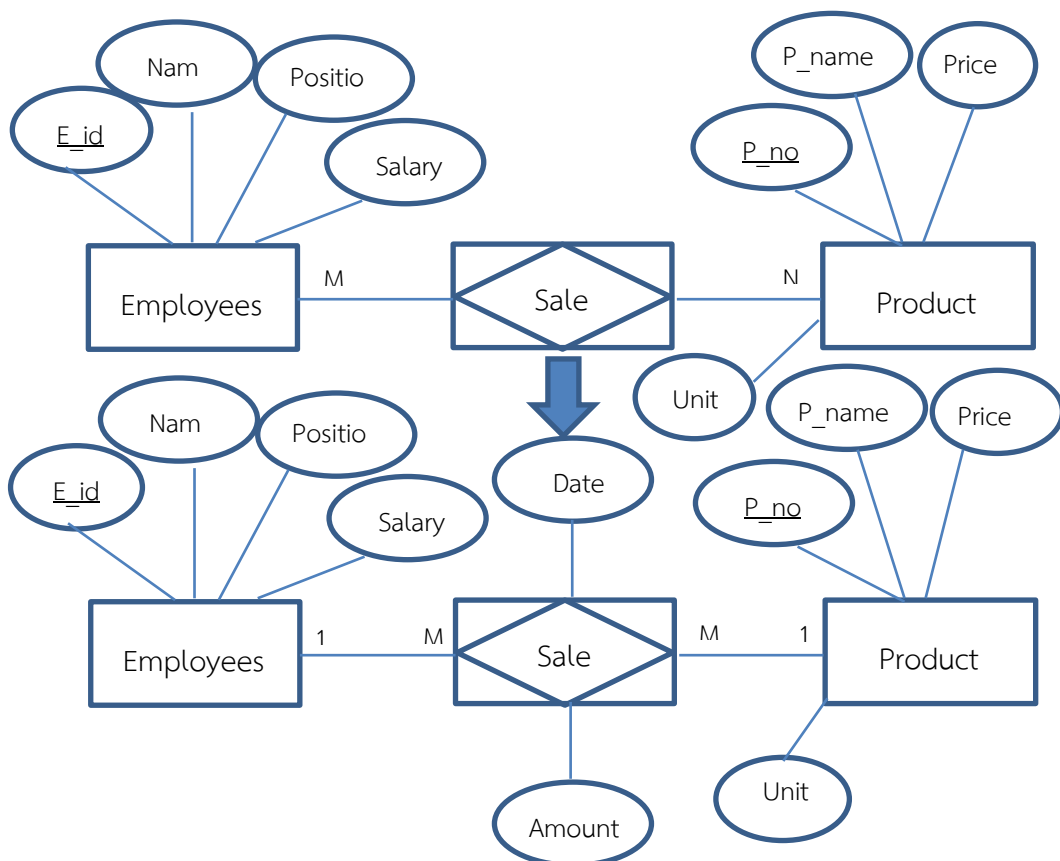
3) เอนทิตีเชิงความสัมพันธ์ (Associative Entity) เป็นเอนทิตีที่เกิดจากความสัมพันธ์ของเอนทิตีแบบ กลุ่มต่อกลุ่ม (M:N) เมื่อแปลงความสัมพันธ์จะเกิดเป็นอีก 1 เอนทิตี เพื่อให้เป็นความสัมพันธ์แบบ หนึ่งต่อกลุ่ม (1:M) และในตัวความสัมพันธ์จะมีแอตทริบิวต์จากความสัมพันธ์ด้วย สัญลักษณ์ที่ใช้คือ สี่เหลี่ยมผืนผ้าซ้อนด้วยสี่เหลี่ยมข้าวหลามตัด



ตัวอย่างเช่น เอนทิตีพนักงานกับสินค้า มีความสัมพันธ์แบบ M:N ชื่อความสัมพันธ์คือขาย (Sale) จะเป็นเอนทิตีแบบ Associative Entity



ตัวอย่าง เอนทิตีพนักงาน (Employees) และเอนทิตีสินค้า (Product) มีความสัมพันธ์กับแบบ M:N แล้วการเปลี่ยนให้เป็น 1:M โดยการทำให้เกิดเอนทิตีแบบ Associative Entity เกิดขึ้นคือเอนทิตีขาย (Sale) ในตัวความสัมพันธ์ Sale มักจะมีแอตทริบิวต์เพื่อระบุรายละเอียดเพิ่มเติม ดังภาพต่อไปนี้



ภาพที่ 4.3 แผนภาพอี อาร์ที่มีการเปลี่ยนแปลงความสัมพันธ์ระหว่าง เอนทิตี

Employees และเอนทิตี Product แบบ M:N ให้เป็น 1:M

จากภาพที่ 4.3 ความสัมพันธ์ระหว่างเอนทิตีพนักงาน (Employees) และ เอนทิตีสินค้า (Product) เป็นแบบ M:N ตัวความสัมพันธ์จึงเกิดเป็นอีก 1 Entity คือ Entity Sale และใน Entity Sale มีแอตทริบิวต์เพิ่มขึ้นคือ Amount และ Date

เมื่อทำการแปลงภาพ E-R Diagram ตามเงื่อนไขของการแปลง E-R Diagram ไปเป็นรีเลชันจะได้เอนทิตีและแอตทริบิวต์ ดังนี้

Employees (E_id, Name, Position, Salary)

Product (P_no, P_name, Price, Unit)

Sale (E_id, P_no, Amount, Date)

2. แอตทริบิวต์ (Attribute) หมายถึง คุณสมบัติหรือรายละเอียดของเอนทิตีที่จะอธิบายเกี่ยวกับรายละเอียดที่เอนทิตีมี ซึ่งแอตทริบิวต์ของเอนทิตีจะบ่งชี้ถึงความแตกต่างของเอนทิตีกับเอนทิตีอื่น ๆ ในหนึ่งเอนทิตีอาจประกอบด้วยแอตทริบิวต์ได้มากกว่าหนึ่งแอตทริบิวต์ ขึ้นกับว่าระบบงานที่กำลังจะพัฒนาต้องการรายละเอียดของเอนทิตีมากน้อยเพียงใด ตัวอย่างเช่น

เอนทิตี พนักงาน ประกอบด้วย แอตทริบิวต์ รหัสพนักงาน คำนำหน้า ชื่อสกุล วดป.เกิด วดป.ที่เริ่มทำงาน ตำแหน่ง เงินเดือน

ซึ่งเอนทิตีพนักงานในแต่ละหน่วยงานอาจจะมีจำนวนของแอตทริบิวต์ไม่เหมือนกันก็ได้เนื่องจากแต่ละที่อาจต้องการรายละเอียดของพนักงานไม่เหมือน ซึ่งบางหน่วยงานอาจมีความต้องการเก็บ รหัสบัตรประชาชน เก็บที่อยู่ เบอร์โทร สถานะสมรส เพิ่มเติมก็ได้

เอนทิตี แผนก ประกอบด้วย รหัสแผนก ชื่อแผนก เบอร์โทรภายใน

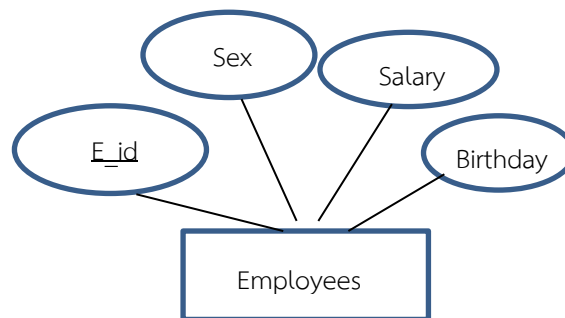
แอตทริบิวต์ใน Chen Model มีอยู่หลายประเภทด้วยกัน และใช้สัญลักษณ์แทนแตกต่างกันออกไป ดังนี้

1) แอตทริบิวต์แบบธรรมดา (Simple Attribute) เป็นแอตทริบิวต์ที่ค่าไม่สามารถแบ่งย่อยได้อีก เช่น รหัสพนักงาน เพศ เงินเดือน วดป.เกิด เป็นต้น

สัญลักษณ์ที่ใช้ คือ รูปวงรี และเขียนชื่อแอตทริบิวต์ในรูปวงรี



ตัวอย่างเช่น แอตทริบิวต์ของเอนทิตีพนักงาน (Employees) ประกอบด้วย รหัสพนักงาน (E_id) เพศ (Sex) เงินเดือน (Salary) วดป.เกิด (Birthday)



ภาพที่ 4.4 แอตทริบิวต์แบบ Simple Attribute ในเอนทิตี Employees

2) แอตทริบิวต์แบบค่าเดียว (Single-Value Attribute) เป็นแอตทริบิวต์ที่มีค่าของข้อมูลเพียงค่าเดียว ไม่สามารถแยกย่อยได้อีก เช่น แอตทริบิวต์ Salary ใช้เก็บเงินเดือนของพนักงาน และพนักงานแต่ละคนจะมีเงินเดือนเพียงค่าเดียว สัญลักษณ์ที่ใช้คือรูปวงรี เหมือนกับ Simple Attribute

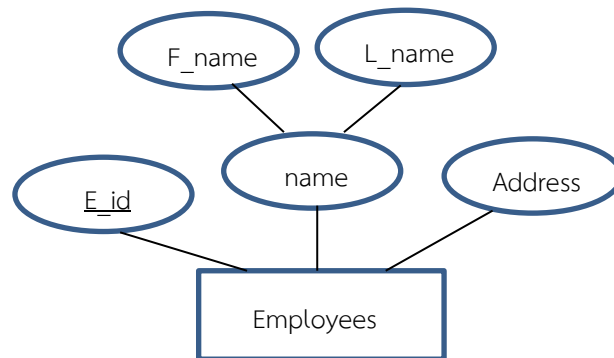


3) แอตทริบิวต์โดเมน (Attribute Domain) เป็นแอตทริบิวต์ที่มีการกำหนดขอบเขตค่าหรือเงื่อนไขของข้อมูลที่บันทึกลงในแอตทริบิวต์นั้น เช่น แอตทริบิวต์เพศ (Sex) จะกำหนดค่าที่สามารถบันทึกลงในแอตทริบิวต์ได้คือ เพศชาย (M) กับ เพศหญิง (F) เท่านั้น หรือ แอตทริบิวต์เงินเดือนของพนักงาน มีเงื่อนไขของการบันทึกข้อมูลลงในแอตทริบิวต์คือ ต้องไม่ต่ำกว่า 0 เป็นต้น สัญลักษณ์ที่ใช้คือรูปวงรี



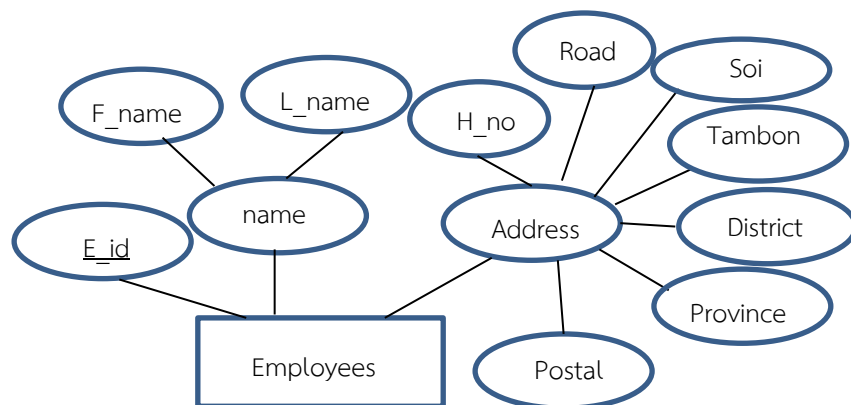
4) แอตทริบิวต์แบบผสม (Composite Attribute) เป็น แอตทริบิวต์ที่ค่าภายในแอตทริบิวต์นั้นสามารถแยกเป็นแอตทริบิวต์ย่อยได้อีก สัญลักษณ์ที่ใช้ จะเป็นวงรี ที่แยกออกจากวงรีของแอตทริบิวต์หลัก

ตัวอย่างเช่น แอตทริบิวต์ ชื่อสกุล (name) สามารถแยกแอตทริบิวต์ ชื่อ (F_name) และ แอตทริบิวต์ สกุล (L_name) สัญลักษณ์ที่ใช้ดังนี้



ภาพที่ 4.5 แอตทริบิวต์แบบ Composite Attribute ในเอนทิตี Employees

แอตทริบิวต์ที่อยู่ (Address) สามารถแยกย่อยได้ ซึ่งประกอบด้วย บ้านเลขที่ (H_No) ถนน (Road) ซอย (Soi) ตำบล (Tambon) อำเภอ (District) จังหวัด (Province) รหัสไปรษณีย์ (Postal)

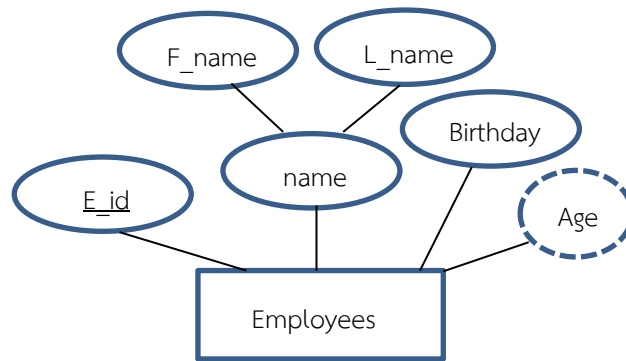


ภาพที่ 4.6 แอตทริบิวต์แบบ Composite Attribute ในเอนทิตี Employees

5) ดีไรฟด์แอตทริบิวต์ (Derived Attribute) เป็นแอตทริบิวต์ที่ได้จากการนำเอาแอตทริบิวต์อื่นมาคำนวณและค่าในแอตทริบิวต์นี้จะเปลี่ยนแปลงทุกครั้งเมื่อมีการเปลี่ยนแปลงค่าของแอตทริบิวต์ที่นำค่ามาคำนวณ โดยปกติแอตทริบิวต์นี้จะไม่นิยมเก็บในฐานข้อมูล เนื่องจากสามารถคำนวณได้จากค่าของแอตทริบิวต์อื่น สัญลักษณ์ที่ใช้ คือ วงรีแบบเส้นประ ดังภาพ



ตัวอย่างเช่น แอตทริบิวต์อายุ (Age) ซึ่งเป็นแอตทริบิวต์ที่ได้จากการคำนวณวันที่ปัจจุบันกับแอตทริบิวต์ วดป.เกิด สามารถใช้สัญลักษณ์แทนได้ดังนี้

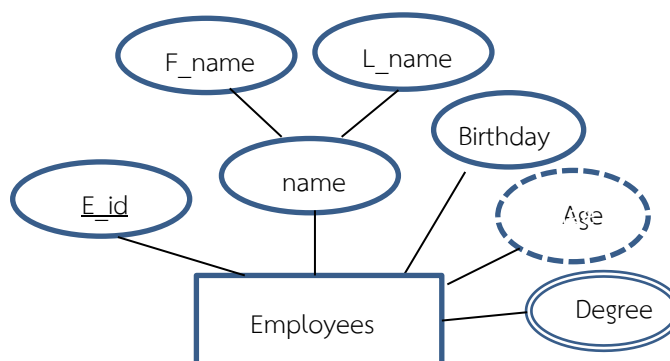


ภาพที่ 4.7 แอตทริบิวต์แบบ Derived Attribute ในเอนทิตี Employees

6) แอตทริบิวต์แบบหลายค่า (Multi-valued Attribute) เป็นแอตทริบิวต์ที่มีการบรรจุข้อมูลได้หลายค่าในแอตทริบิวต์นั้น สัญลักษณ์ที่ใช้คือ รูปวงรีซ้อนกัน ดังภาพ

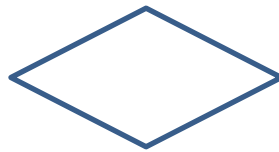


ตัวอย่างเช่น วุฒิการศึกษาของพนักงาน (Degree) ที่ใช้ระบุการศึกษาของพนักงานแต่ละคน ซึ่งพนักงานแต่ละคนจะมีระดับการศึกษาได้หลายระดับ ดังนี้

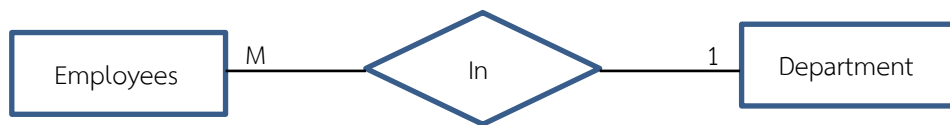


ภาพที่ 4.8 แอตทริบิวต์แบบ Multi-valued Attribute ในเอนทิตี Employees

3. ความสัมพันธ์ (Relationship) หมายถึง ความสัมพันธ์ระหว่างเอนทิตีที่อยู่ในฐานข้อมูลเดียวกัน ซึ่งแต่ละเอนทิตีในฐานข้อมูลต้องมีความสัมพันธ์ระหว่างกัน และจะต้องกำหนดความสัมพันธ์ให้เหมาะสมและสอดคล้องกับความเป็นจริงของการปฏิบัติงานด้วย ซึ่งกระบวนการของการกำหนดความสัมพันธ์นี้จะได้จากขั้นตอนของการศึกษาปัญหาและความต้องการขององค์กร สัญลักษณ์ของการกำหนดความสัมพันธ์จะเป็นรูปสี่เหลี่ยมข้าวหลามตัด และเขียนคำอธิบายถึงความสัมพันธ์ไว้ในสัญลักษณ์นั้น ส่วนใหญ่จะเป็นคำกริยา ดังภาพ



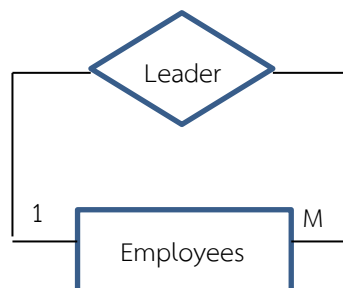
ตัวอย่างเช่น เอนทิตีพนักงาน (Employees) กับเอนทิตีแผนก (Department) โดยพนักงานจะสังกัดแผนกได้เพียง 1 แผนก แต่ใน 1 แผนกจะมีพนักงานสังกัดได้หลายคน ซึ่งเกิดความสัมพันธ์แบบ 1:M ตัวความสัมพันธ์คือ สังกัด เขียนเป็น E-R Diagram ได้ดังนี้



ภาพที่ 4.9 ความสัมพันธ์ระหว่างเอนทิตีแบบ 1:M

ความสัมพันธ์ระหว่างเอนทิตีแบ่งได้เป็น 3 ประเภทดังนี้

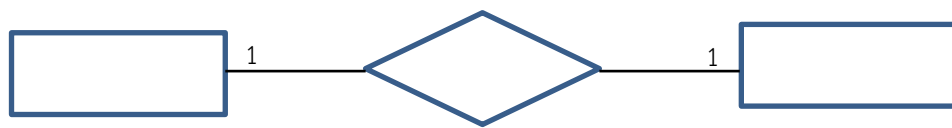
1) ความสัมพันธ์แบบยูนารี (Unary Relationships) ความสัมพันธ์ที่เกิดขึ้นภายในตัวเอง เช่น ในเอนทิตีพนักงาน (Employees) เกิดความสัมพันธ์ คือ มีพนักงาน 1 คนที่ทำหน้าที่เป็นหัวหน้าแผนก (Leader) ซึ่งเป็นความสัมพันธ์จากเอนทิตีพนักงานเอง ลักษณะนี้เรียกว่า เป็นความสัมพันธ์แบบยูนารี เขียนความสัมพันธ์ได้ดังนี้



ภาพที่ 4.10 ความสัมพันธ์ระหว่างเอนทิตีแบบ Unary Relationships

2) ความสัมพันธ์แบบไบนารี (Binary Relationships) เป็นดึกรีของความสัมพันธ์ที่เกิดขึ้นจากเอนทิตี 2 เอนทิตี และเป็นความสัมพันธ์ที่พบเห็นได้บ่อยที่สุด ความสัมพันธ์แบบไบนารีจะแบ่งออกเป็น 3 ชนิดด้วยกันคือ

1. ความสัมพันธ์แบบ หนึ่งต่อหนึ่ง (One-to-One Relationship) ระบุความสัมพันธ์แทนด้วย 1:1 เป็นความสัมพันธ์ระหว่างเอนทิตี 2 เอนทิตีที่มีความสัมพันธ์ที่สมาชิก 1 สมาชิกของเอนทิตีหนึ่งไปสัมพันธ์กับ 1 สมาชิกของเอนทิตีหนึ่ง เขียนภาพความสัมพันธ์ดังนี้



ภาพที่ 4.11 ความสัมพันธ์ระหว่างเอนทิตีแบบ 1:1

ตัวอย่างของความสัมพันธ์ระหว่างเอนทิตีแบบ 1:1 เช่น

- เอนทิตีห้องพักกับเอนทิตีที่จอดรถ โดยห้องพักมีเงื่อนไขของการนำรถเข้ามาจอดในห้องพักโดยจะจัดทำที่จอดรถให้ทุกห้องที่อยู่ในห้องพักแต่มีเงื่อนไขคือใน 1 ห้องพักจะมีที่จอดรถให้เฉพาะเพียงที่เดียวเท่านั้น ตามตารางข้อมูลดังนี้

Entity: ผู้เข้าพัก

ห้อง	ชื่อผู้เข้าพัก	รหัสที่จอด
101	นภัสสร	P001
102	สมชาย	P002
103	กรกนณ	P003

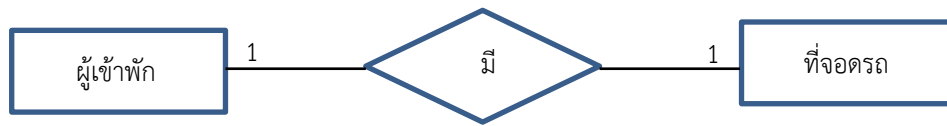
Entity: ที่จอดรถ

รหัสที่จอด	โซน	ลำดับ
P001	A	1
P002	A	2
P003	B	4



ภาพที่ 4.12 ตัวอย่างตารางผู้เข้าพักกับตารางที่จอดรถมีข้อมูลจากความสัมพันธ์แบบ 1:1

จากตารางข้อมูลดังกล่าว แสดงให้เห็นว่า เอนทิตีผู้เข้าพัก กับ เอนทิตีที่จอดรถ มีความสัมพันธ์กับแบบ หนึ่งต่อหนึ่ง (1:1) เขียนสัญลักษณ์แทนความสัมพันธ์ได้ดังนี้



ภาพที่ 4.13 ตัวอย่างเอนทิตีผู้เข้าพักกับเอนทิตีที่จอดรถ มีความสัมพันธ์ระหว่างเอนทิตีแบบ 1:1

- เอนทิตีสาขาวิชากับเอนทิตีหัวหน้าสาขาวิชา โดยใน 1 สาขาจะมีหัวหน้าสาขาได้เพียง 1 คน และ หัวหน้าสาขาจะสังกัดได้เพียง 1 สาขาวิชาเท่านั้น ตามตารางข้อมูลดังนี้

Entity: สาขาวิชา

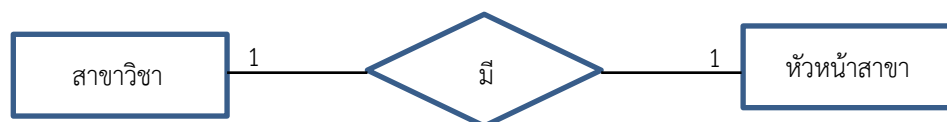
รหัสสาขา	ชื่อสาขา	เบอร์โทร	รหัส
1	ระบบสารสนเทศ	3650	01
2	การจัดการ	3640	02
3	การตลาด	3630	03
4	การบัญชี	3660	04

Entity: หัวหน้าสาขา

รหัส	ตำแหน่ง	ชื่อ-สกุล
01	ผศ.	สุนทวัฒน์ การดี
02	ดร.	ประวิน ชื้อสตัย
03	นาง	ลักขณา ชื้อสตัย
04	รศ.	มิกิ พิมพ์พรรณ

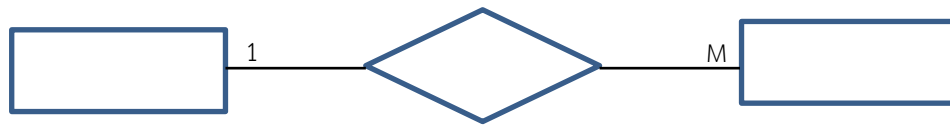
ภาพที่ 4.14 ตัวอย่างตารางสาขาวิชากับตารางหัวหน้าสาขามีข้อมูลจากความสัมพันธ์แบบ 1:1

จากตารางข้อมูลดังกล่าว แสดงให้เห็นว่า เอนทิตีสาขาวิชา กับ เอนทิตีหัวหน้าสาขา มีความสัมพันธ์แบบ หนึ่งต่อหนึ่ง (1:1) เขียนสัญลักษณ์แทนความสัมพันธ์ได้ดังนี้



ภาพที่ 4.15 ตัวอย่างเอนทิตี สาขาวิชากับเอนทิตี หัวหน้าสาขา มีความสัมพันธ์ระหว่างเอนทิตีแบบ 1:1

2. ความสัมพันธ์แบบ หนึ่งต่อกลุ่ม (One-to-Many Relationship)
ระบุความสัมพันธ์แทนด้วย 1:M เป็นความสัมพันธ์ที่สมาชิก 1 ตัวของเอนทิตีหนึ่งไปมีความสัมพันธ์กับสมาชิกเอนทิตีหนึ่งแบบหลายตัว เขียนแผนภาพความสัมพันธ์ดังนี้



ภาพที่ 4.16 ความสัมพันธ์ระหว่างเอนทิตีแบบ 1:M

ตัวอย่างของความสัมพันธ์ระหว่างเอนทิตีแบบ 1:M เช่น

- เอนทิตีแผนก กับ เอนทิตีพนักงาน โดยใน 1 แผนกจะมีพนักงานได้หลายคน แต่พนักงาน 1 คนจะสังกัดแผนกได้เพียงแผนกเดียว ตามตารางข้อมูลต่อไปนี้

Entity: พนักงาน

รหัสพนักงาน	ชื่อ	ตำแหน่ง	เงินเดือน	รหัสแผนก
1001	วงศ์เดือน	PHP Developed	32000	1
1002	วันชัย	IT Manager	42000	1
1003	กลินทิพย์	IT Administrator	28000	1
2001	กานดา	Sale&Marketing	19000	2
2002	น้ำทิพย์	Sale Manager	35000	2
3001	ทองใบ	Accounting Manage	39000	1

Entity: แผนก

รหัสแผนก	ชื่อแผนก
1	IT
2	Marketing
3	Accounting

ภาพที่ 4.17 ตัวอย่างตารางแผนกกับตารางพนักงานมีข้อมูลจากความสัมพันธ์แบบ 1:M

จากตารางข้อมูล แสดงให้เห็นว่า เอนทิตี แผนก กับ เอนทิตีพนักงาน มีความสัมพันธ์กับแบบ หนึ่งต่อหนึ่ง (1:M) เขียนสัญลักษณ์แทนความสัมพันธ์ได้ดังนี้



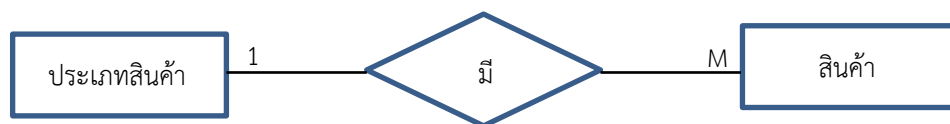
ภาพที่ 4.18 ตัวอย่างเอนทิตี แผนกกับเอนทิตี พนักงาน มีความสัมพันธ์ระหว่างเอนทิตีแบบ 1:M

- เอนทิตี ประเภทสินค้ากับเอนทิตีสินค้า โดยประเภทสินค้า 1 ประเภทจะสามารถมีสินค้าได้หลายชนิดแต่สินค้า 1 ชนิดจะอยู่ได้เพียง 1 ประเภท ตามตารางข้อมูลต่อไปนี้

Entity: สินค้า					Entity: ประเภทสินค้า	
รหัสสินค้า	ชื่อสินค้า	ขนาด	ราคา	รหัสประเภท	รหัสประเภท	ชื่อประเภท
1010	OLAY Rigorist Overnight Cream	40 ml	890	1	1	ครีมบำรุงผิว
1011	Neutrogena ไฮโดร บุสท์เจล	50 g	360	1	2	แผ่นมาร์คหน้า
1020	แผ่นมาร์คหน้าทองคำ	1 แผ่น	90	2	3	อาหารเสริม
1021	แผ่นมาร์คแตงกวา	10 แผ่น	280	2		
1030	Amado P-Collagen	100 g.	520	3		

ภาพที่ 4.19 ตัวอย่างตารางประเภทสินค้ากับตารางสินค้ามีข้อมูลจากความสัมพันธ์แบบ 1:M

จากตารางข้อมูล แสดงให้เห็นว่า เอนทิตีประเภทสินค้า กับ เอนทิตีสินค้า มีความสัมพันธ์แบบ หนึ่งต่อหนึ่ง (1:M) เขียนสัญลักษณ์แทนความสัมพันธ์ได้ดังนี้



ภาพที่ 4.20 ตัวอย่างเอนทิตี ประเภทสินค้ากับเอนทิตีสินค้า มีความสัมพันธ์ระหว่างเอนทิตีแบบ 1:M

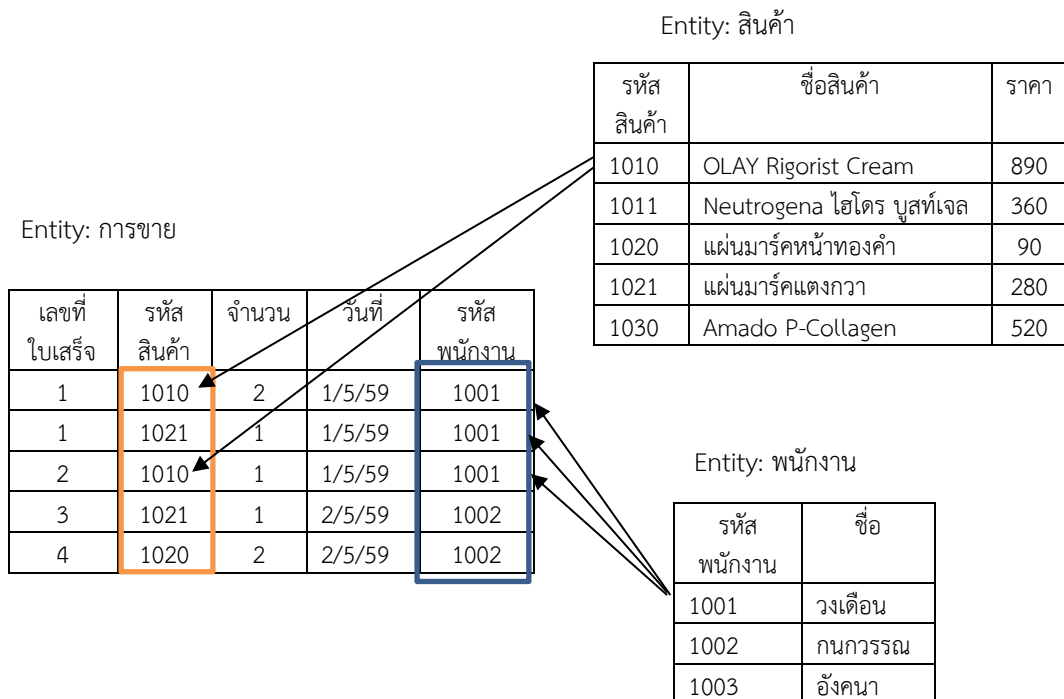
3. ความสัมพันธ์แบบ กลุ่มต่อกลุ่ม (Many-to-Many Relationship) ระบุความสัมพันธ์แทนด้วย M:N เป็นความสัมพันธ์ที่เกิดจากสมาชิกหลายตัวในเอนทิตีหนึ่งไปมีความสัมพันธ์กับสมาชิกอีกหลายตัวในเอนทิตีหนึ่ง เขียนแผนภาพความสัมพันธ์ดังนี้



ภาพที่ 4.21 ความสัมพันธ์ระหว่างเอนทิตีแบบ M:N

ตัวอย่างของความสัมพันธะระหว่างเอนทิตีแบบ M:N เช่น

- เอนทิตีสินค้า กับ เอนทิตีพนักงาน โดยพนักงาน 1 คน สามารถขายสินค้าได้หลายอย่าง และสินค้า 1 อย่างสามารถถูกขายด้วยพนักงานได้หลายคน จึงเกิดความสัมพันธะแบบ M:N ตามตารางข้อมูลต่อไปนี้



ภาพที่ 4.22 ตัวอย่างตารางสินค้ากับพนักงานมีข้อมูลจากความสัมพันธะแบบ M:N

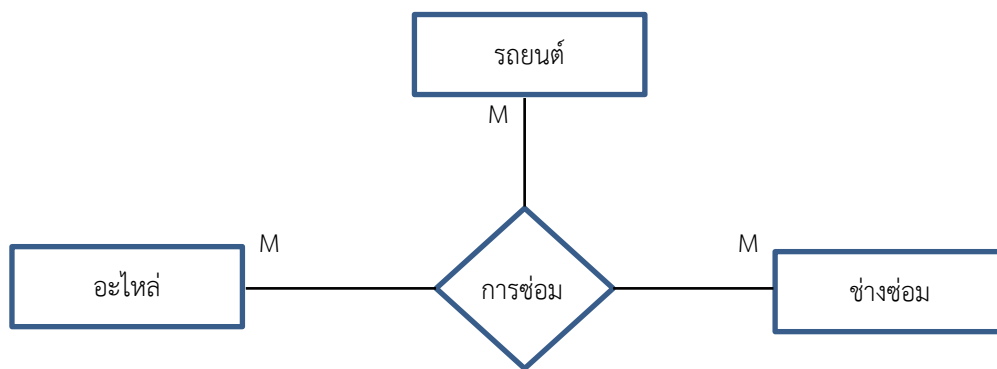
จากตารางข้อมูล แสดงให้เห็นว่า เอนทิตี สินค้า กับ เอนทิตีพนักงาน มีความสัมพันธ์กับแบบ หนึ่งต่อหนึ่ง (M:N) เขียนสัญลักษณ์แทนความสัมพันธ์ได้ดังนี้



ภาพที่ 4.23 ตัวอย่างเอนทิตีสินค้ากับเอนทิตีพนักงานมีความสัมพันธ์ระหว่างเอนทิตีแบบ M:N

3) ความสัมพันธ์แบบเทอร์นารี (Ternary Relationships)

เป็นความสัมพันธ์ที่เกิดจาก 3 เอนทิตีที่มีความสัมพันธ์กัน จึงเกิดเป็นความสัมพันธ์แบบเทอร์นารี ตัวอย่างความสัมพันธ์แบบเทอร์นารี เช่น ในการซ่อมรถยนต์ จะเกิดจากความสัมพันธ์ของ เอนทิตีรถยนต์ เอนทิตีช่างซ่อม และเอนทิตีอะไหล่ ทำให้เกิดเป็นความสัมพันธ์คือ การซ่อม ดังภาพที่ 4.24



ภาพที่ 4.24 ตัวอย่างเอนทิตีที่เกิดความสัมพันธ์แบบ Ternary Relationships

4.2.2 ขั้นตอนการพัฒนาแผนภาพอี อาร์

ขั้นตอนการพัฒนาแผนภาพอี อาร์ เป็นขั้นตอนของการออกแบบฐานข้อมูลในระดับแนวคิด ซึ่งทุกขั้นตอนของการปฏิบัติสามารถที่จะกลับมาทำซ้ำได้ใหม่เพื่อเป็นการตรวจสอบความถูกต้องของฐานข้อมูลที่ได้ออกแบบไว้ร่วมกับผู้ใช้ ว่ามีความถูกต้องหรือไม่ถ้าไม่ถูกต้องสามารถที่จะกลับไปทำในขั้นตอนก่อนหน้าได้ ซึ่งขั้นตอนของการพัฒนาดังนี้

1. ศึกษารายละเอียดและลักษณะหน้าที่งานของระบบ
2. กำหนดเอนทิตีที่ควรมีในระบบฐานข้อมูล
3. กำหนดความสัมพันธ์ระหว่างเอนทิตี
4. กำหนด แอตทริบิวต์ ของแต่ละเอนทิตี
5. กำหนดคีย์หลัก
6. สร้างแผนภาพอี อาร์
7. ตรวจสอบและแก้ไขแผนภาพอี อาร์

1. ศึกษารายละเอียดและลักษณะหน้าที่งานของระบบ

การศึกษารายละเอียดและลักษณะหน้าที่งานของระบบ เพื่อรวบรวมรายละเอียดของลักษณะการปฏิบัติงาน ความต้องการของผู้ใช้ ขั้นตอนการปฏิบัติงานและเงื่อนไขหรือข้อกำหนดต่างๆ ในการปฏิบัติงาน ซึ่งอาจจะศึกษาจากการปฏิบัติงานเดิม ศึกษาจากเอกสารรายงานต่างๆ ที่มีหรือศึกษาจากการสัมภาษณ์ความต้องการของผู้ใช้ ซึ่งอยู่กับผู้ออกแบบฐานข้อมูลจะใช้เทคนิคอะไรในการรวบรวม

2. กำหนดเอนทิตีที่ควรมีในฐานข้อมูล

เมื่อได้ศึกษารายละเอียดและลักษณะหน้าที่ของระบบแล้ว ขั้นตอนต่อไปจะต้องกำหนดเอนทิตี ที่ควรมีในฐานข้อมูล โดยคำนึงถึงข้อมูลทั้งหมดที่จะจัดเก็บลงในฐานข้อมูล การมองข้อมูลให้เป็นเอนทิตีนั้น ข้อมูลนั้นจะต้องมีรายละเอียดที่สามารถชี้ความแตกต่างของข้อมูลได้ ในฐานข้อมูลหนึ่ง ๆ จำนวนเอนทิตีในฐานข้อมูลจะมีมากหรือน้อยขึ้นอยู่กับความเหมาะสมว่าผู้ใช้ต้องการจัดเก็บข้อมูลมากเพียงใด และเมื่อได้เอนทิตีแล้วก็ต้องมีการกำหนดชนิดของเอนทิตีด้วยว่าเป็นเอนทิตีประเภทใด เพื่อที่จะกำหนดสัญลักษณ์ของเอนทิตีให้ถูกต้อง

3. กำหนดความสัมพันธ์ระหว่างเอนทิตี

เมื่อกำหนดเอนทิตี ที่ควรมีในฐานข้อมูลเสร็จแล้ว ขั้นตอนต่อไปก็จะเป็นการกำหนดความสัมพันธ์ระหว่างเอนทิตี ว่ามีเอนทิตีใดที่มีความสัมพันธ์กันบ้าง และเป็นความสัมพันธ์กันอย่างไร เช่น

- ความสัมพันธ์แบบ One-to-One Relationship (1:1)
- ความสัมพันธ์แบบ One-to-Many Relationship (1:M)
- ความสัมพันธ์แบบ Many-to-Many Relationship (M:N)

โดยการกำหนดความสัมพันธ์จะต้องใช้สัญลักษณ์แทนความสัมพันธ์และเขียนชื่อกำกับความสัมพันธ์ด้วย

4. กำหนดแอตทริบิวต์

แอตทริบิวต์จะเป็นการกำหนดรายละเอียดหรือคุณสมบัติของเอนทิตี ซึ่งเอนทิตีหนึ่งๆ อาจประกอบด้วยแอตทริบิวต์ มากกว่าหนึ่งแอตทริบิวต์ขึ้นอยู่กับรายละเอียดของเอนทิตีนั้น ๆ และแอตทริบิวต์ จะเป็นตัวชี้บ่งถึงความแตกต่างของเอนทิตีแต่ละตัว ดังนั้นการกำหนดชื่อของแอตทริบิวต์จะใช้หลักเกณฑ์คือ ชื่อของแอตทริบิวต์จะต้องไม่ซ้ำกันในแต่ละเอนทิตี ถึงแม้จะอยู่กันคนละเอนทิตีในฐานข้อมูลก็ตาม หลังจากกำหนดแอตทริบิวต์แล้วก็ต้องระบุชนิดของแอตทริบิวต์และใช้สัญลักษณ์แทนชนิดของแอตทริบิวต์ให้ถูกต้อง

5. กำหนดคีย์หลัก

เป็นการกำหนดค่าที่เป็นเอกลักษณ์ เพื่อไม่ให้ข้อมูลที่ถูกรวบรวมในเอนทิตีไม่ซ้ำกันในแต่ละเรคคอร์ด และสามารถอ้างอิงได้ โดยแอตทริบิวต์ ที่เป็นคีย์หลักจะต้องขีดเส้นใต้ที่ชื่อของแอตทริบิวต์นั้น และเมื่อพิจารณาแอตทริบิวต์ ที่มีในเอนทิตีแล้วไม่มีแอตทริบิวต์ใดที่มีคุณสมบัติเป็นคีย์หลักได้ สามารถกำหนดแอตทริบิวต์ เพิ่มเพื่อให้มีแอตทริบิวต์นั้นมีคุณสมบัติเป็นคีย์หลักได้

6. สร้างแผนภาพอี อาร์

เป็นขั้นตอนของการสร้างแผนภาพอี อาร์ ด้วยการนำเอนทิตี ที่ได้ ในขั้นตอนที่ 2 มาเชื่อมความสัมพันธ์ตามขั้นตอนที่ 3 การกำหนดแอตทริบิวต์ในขั้นตอนที่ 4 และการกำหนดคีย์หลัก ในขั้นตอนที่ 5 มารวมกันเพื่อเขียนเป็นแผนภาพอี อาร์ ซึ่งมีโปรแกรมสำเร็จรูปหลายโปรแกรมที่มีเครื่องมือช่วยในการเขียนแผนภาพอี อาร์ เช่น โปรแกรม E-R Designer เป็นโปรแกรมที่มีเครื่องมือช่วยในการออกแบบฐานข้อมูลในการวาดรูปของไดอะแกรมในรูปแบบของ Chen's Notation ให้ทำงานง่ายขึ้น

7. ตรวจสอบและแก้ไขแผนภาพอี อาร์

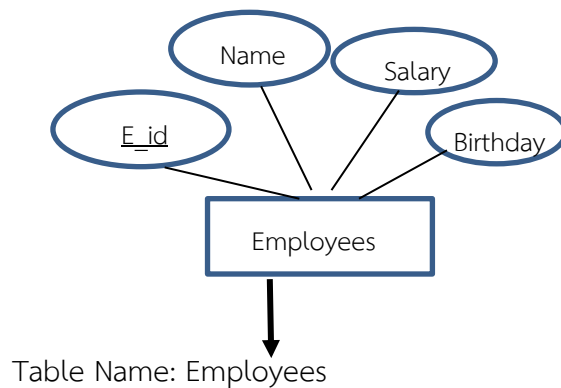
เมื่อได้แผนภาพอี อาร์ แล้วขั้นตอนต่อไปคือ นำแผนภาพไปตรวจสอบกับผู้ใช้งานเพื่อตรวจสอบความถูกต้องตรงตามที่ผู้ใช้งานต้องการและตรงกับลักษณะของการทำงานในองค์กร และทำการปรับปรุงแก้ไขให้ถูกต้อง

4.3 การแปลงแผนภาพอี อาร์ ให้เป็นฐานข้อมูลเชิงสัมพันธ์

แผนภาพอี อาร์ แบบ Chen Model ประกอบด้วยเอนทิตี แอตทริบิวต์และความสัมพันธ์ ดังนั้น จำเป็นต้องมีการแปลงแผนภาพอี อาร์ ให้อยู่ในรูปแบบตารางเก็บข้อมูล หลังจากนั้นจะนำตารางไปตรวจสอบหาความซ้ำซ้อนของข้อมูลด้วยการทำให้อยู่ในรูปของบรรทัดฐาน ถึงจะนำตารางต่าง ๆ ในฐานข้อมูลไปใช้งานได้ ชุดิณพนธ์ บุญมาก และคณะ (2552) ได้แบ่งขั้นตอนของการแปลงอี อาร์ ไปเป็นฐานข้อมูลเชิงสัมพันธ์ออกเป็น 3 ขั้นตอนดังนี้

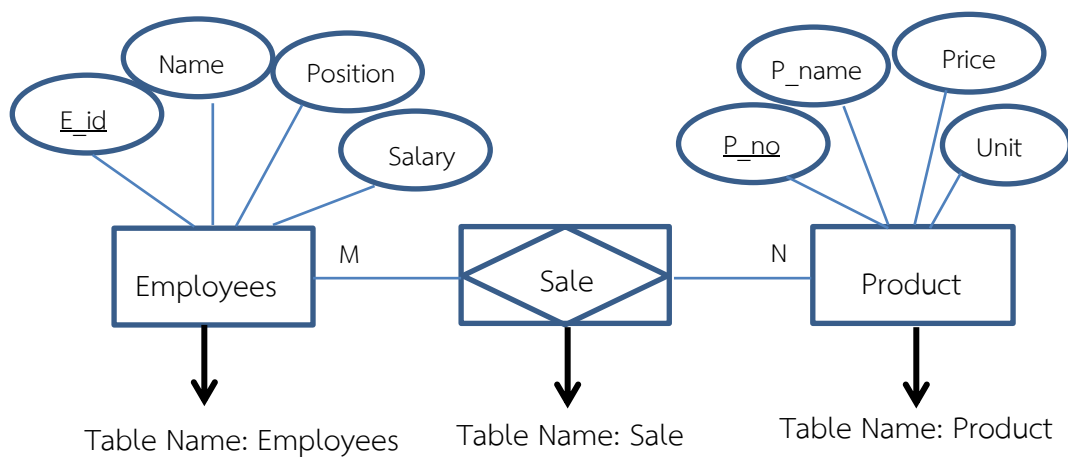
1. การแปลงเอนทิตีใน แผนภาพอี อาร์ ให้เป็นรีเลชัน มีวิธีการแปลงดังนี้

1) แปลงเอนทิตีในแผนภาพอี อาร์ ให้เป็นรีเลชัน โดยกำหนดชื่อของเอนทิตีให้เป็นชื่อของตารางเก็บข้อมูล ตัวอย่างเช่น จาก เอนทิตี Employees ให้แปลงเป็น ตาราง Employees ดังภาพที่ 4.25



ภาพที่ 4.25 การแปลงเอนทิตี Employees ให้เป็นตาราง Employees

2) แปลงความสัมพันธ์ระหว่างเอนทิตีแบบ M:N ซึ่งเกิดเป็น Associative Entity ให้เป็นตารางที่เรียกว่า Composite Entity โดยกำหนดชื่อความสัมพันธ์เป็นชื่อของตารางเก็บข้อมูล ตัวอย่างเช่น เอนทิตี Employees กับเอนทิตี Product มีความสัมพันธ์คือ Sale เป็นความสัมพันธ์แบบ M:N เมื่อแปลงไปเป็นตารางจะได้ 3 ตาราง คือ ตาราง Employees, ตาราง Product และ ตารางแบบ Associative Entity ซึ่งก็คือ ตาราง Sale ดังภาพที่ 4.26

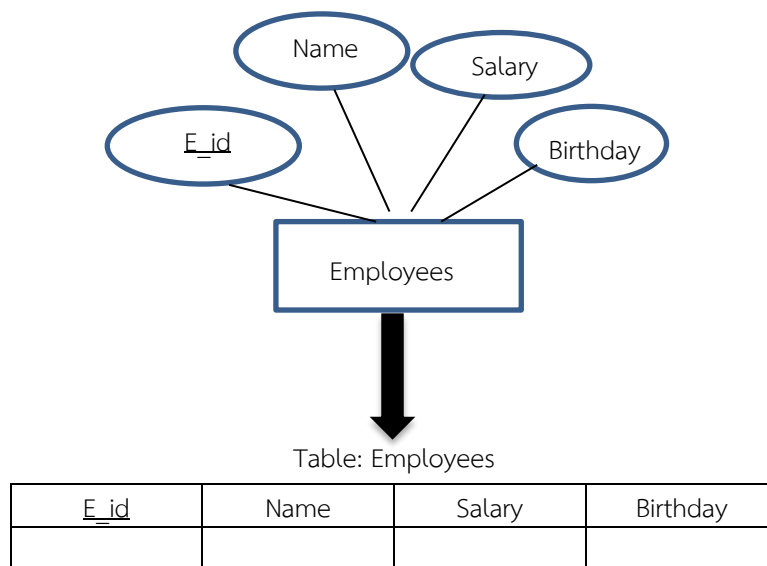


ภาพที่ 4.26 การแปลงเอนทิตี Employees และ Product ที่เกิดความสัมพันธ์แบบ M:N

2. การแปลงรายละเอียดของเอนทิตีให้เป็นแอตทริบิวต์ของตาราง การกำหนดคีย์หลักและคีย์รอง มีขั้นตอนดังนี้

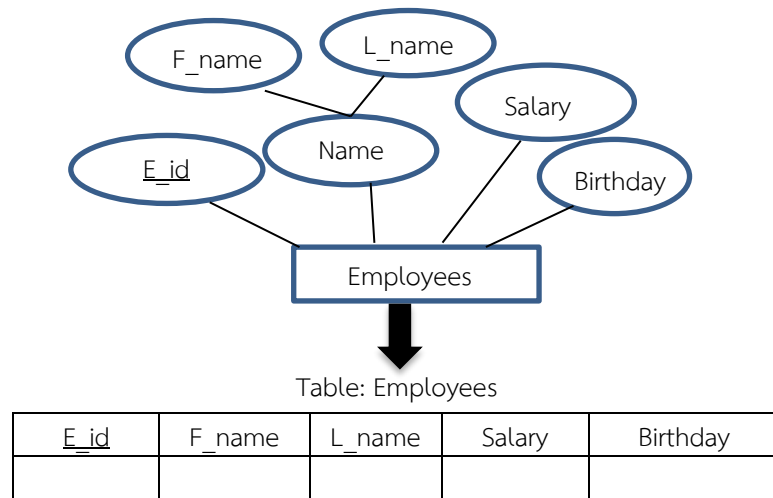
1) การแปลงรายละเอียดของเอนทิตีให้เป็นแอตทริบิวต์ของตาราง และกำหนดคีย์หลัก ตามที่กำหนดไว้ในเอนทิตี มีรายละเอียดดังนี้

- กรณีที่เอนทิตีมีแอตทริบิวต์แบบ Simple Attribute เป็นแอตทริบิวต์ที่มีค่าข้อมูลที่ไม่สามารถแยกย่อยได้อีก ให้กำหนดชื่อ แอตทริบิวต์ตามรายละเอียดของเอนทิตีได้เลย และกำหนดคีย์หลักตามที่ได้ระบุไว้ด้วยการขีดเส้นใต้ ตัวอย่างเช่น เอนทิตี Employees แปลงเป็นตารางได้ดังนี้



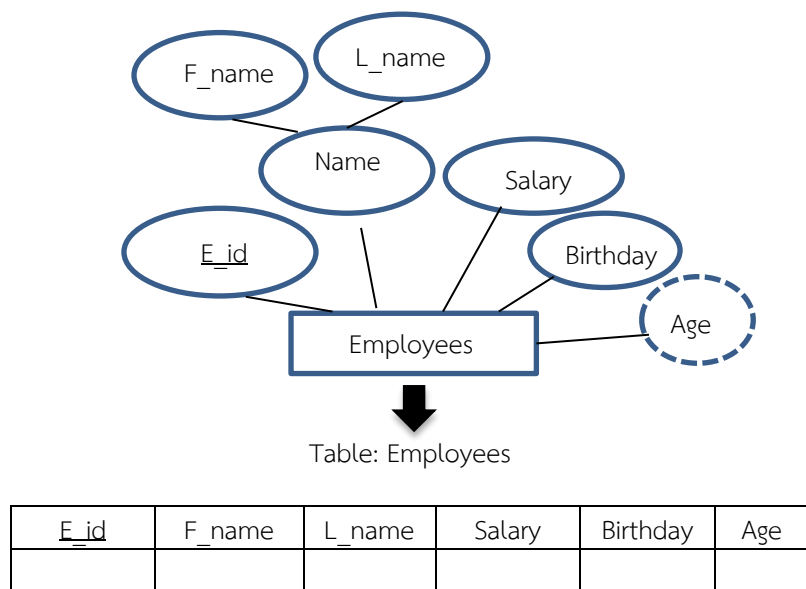
ภาพที่ 4.27 การแปลงแอตทริบิวต์แบบ Simple Attribute ให้เป็นแอตทริบิวต์ในตาราง

- กรณีที่เอนทิตี มีแอตทริบิวต์แบบ Composite Attribute แปลงโดยการระบุแอตทริบิวต์ที่มีการแยกแล้วเท่านั้น ตัวอย่างเช่น ในเอนทิตี Empolyees มีแอตทริบิวต์ Name เป็น Composite Attribute แยกแอตทริบิวต์ย่อยคือ F_name กับ L_name เมื่อแปลงเป็นตารางก็ให้ระบุ แอตทริบิวต์ย่อย เป็นแอตทริบิวต์ในตาราง ดังภาพที่ 4.28



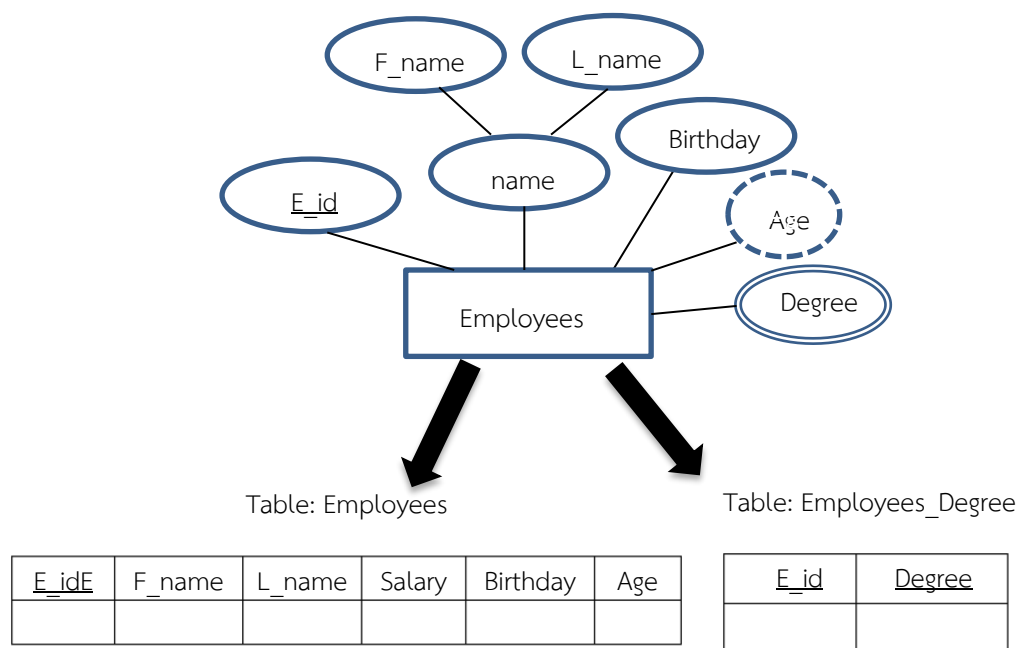
ภาพที่ 4.28 การแปลงแอตทริบิวต์แบบ Composite Attribute

- กรณีที่เอนทิตีมีแอตทริบิวต์แบบ Derived Attribute เมื่อทำการแปลงให้เป็นแอตทริบิวต์ในตารางเก็บข้อมูล ให้กำหนดแอตทริบิวต์ตามชื่อในเอนทิตีได้เลย ปกติจะไม่นิยมเก็บเป็นแอตทริบิวต์เนื่องจากสามารถคำนวณเพื่อหาข้อมูลนี้ได้ แต่ถ้าต้องเก็บข้อมูลนี้ก็สามารถทำได้ ตัวอย่างเช่น เอนทิตี Employees มี แอตทริบิวต์ Age ได้จากการนำเอาแอตทริบิวต์ Birthday ไปคำนวณกับ วันที่ปัจจุบันได้ ดังนั้นสามารถแปลงเป็นแอตทริบิวต์ได้ดังนี้



ภาพที่ 4.29 การแปลงแอตทริบิวต์แบบ Derived Attribute ให้เป็นแอตทริบิวต์ในตาราง

- กรณีที่เอนทิตี มีแอตทริบิวต์แบบ Multi-value Attribute หมายถึง ใน 1 แอตทริบิวต์มีข้อมูลที่บันทึกได้มากกว่า 1 ข้อมูล เมื่อทำการแปลงให้เป็นตารางจะต้องแยกออกให้เป็น 2 ตาราง คือ ให้นำแอตทริบิวต์แบบ Multi-value ไปสร้างเป็นอีกตาราง โดยนำเอาคีย์หลักจากตารางหลักไปรวมกับแอตทริบิวต์ที่เป็น Multi-value รวมกันเป็นคีย์หลักเช่น ตาราง Employee มีแอตทริบิวต์ Degree เป็น Multi-value จึงทำให้เกิดเป็นตารางใหม่ คือตาราง Employee_Degree และใช้ 2 แอตทริบิวต์ คือ E_id กับ Degree รวมกันเป็น Primary Key ดังภาพที่ 4.30



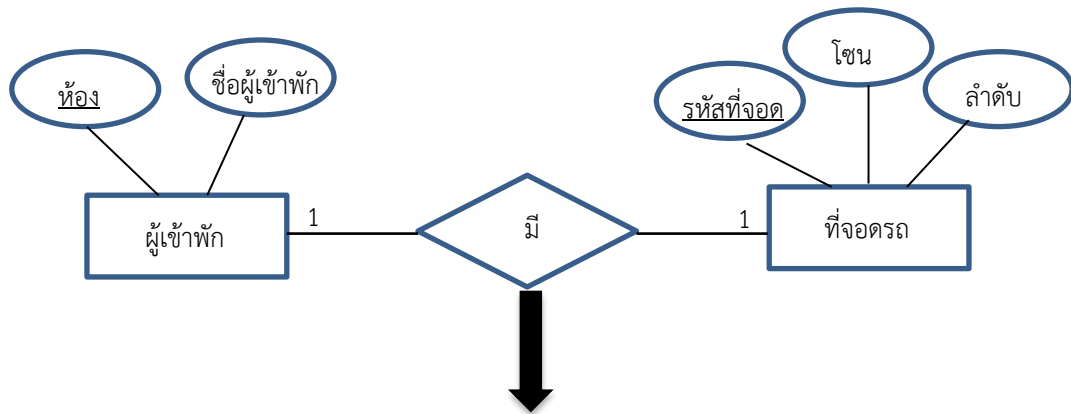
ภาพที่ 4.30 การแปลงแอตทริบิวต์แบบ Multi-value Attribute ให้เป็นแอตทริบิวต์ในตาราง

2) การแปลงแอตทริบิวต์ที่เกิดจากความสัมพันธ์ระหว่างเอนทิตี

ความสัมพันธ์ที่เกิดขึ้นระหว่างเอนทิตี ไม่ว่าจะเป็นความสัมพันธ์แบบ 1:1, 1:M และ M:N เมื่อแปลงไปเป็นตาราง จะทำให้ได้คีย์นอก (Foreign Key) เพื่อใช้เชื่อมความสัมพันธ์ของเอนทิตีที่เกิดขึ้น มีขั้นตอนการทำงานดังนี้

- กรณีที่เอนทิตีมีความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อหนึ่ง (1:1) มีวิธีการแปลงเป็นตารางคือ ให้นำเอาคีย์หลักของเอนทิตีหนึ่งไปเป็นแอตทริบิวต์ของอีกเอนทิตีหนึ่งได้เลย โดยแอตทริบิวต์นั้นมีคุณสมบัติเป็นคีย์นอกตัวอย่างเช่น เอนทิตีผู้เข้าพัก กับเอนทิตีที่จอดรถ มี

ความสัมพันธ์กับแบบ 1:1 สามารถทำการแปลงโดยนำ คีย์หลักของเอนทิตีที่ जोดรองไปเป็นคีย์นอกในเอนทิตีผู้เข้าพัก หรือ นำคีย์หลักในเอนทิตีผู้เข้าพักไปเป็นคีย์นอกในเอนทิตีที่ जोดรองก็ได้任何一种อย่างหนึ่ง ดังภาพ 4.31



วิธีที่ 1 นำคีย์หลักจากเอนทิตีที่ जोดรองมาเป็นคีย์นอกในเอนทิตีผู้เข้าพัก

Table: ผู้เข้าพัก

ห้อง	ชื่อผู้เข้าพัก	รหัสที่ जोด
------	----------------	-------------

Table: ที่ जोดรอง

รหัสที่ जोด	โซน	ลำดับ
-------------	-----	-------



หรือ วิธีที่ 2 นำคีย์หลักจากเอนทิตีผู้เข้าพักมาเป็นคีย์นอกในเอนทิตีที่ जोดรอง

Table: ผู้เข้าพัก

ห้อง	ชื่อผู้เข้าพัก
------	----------------

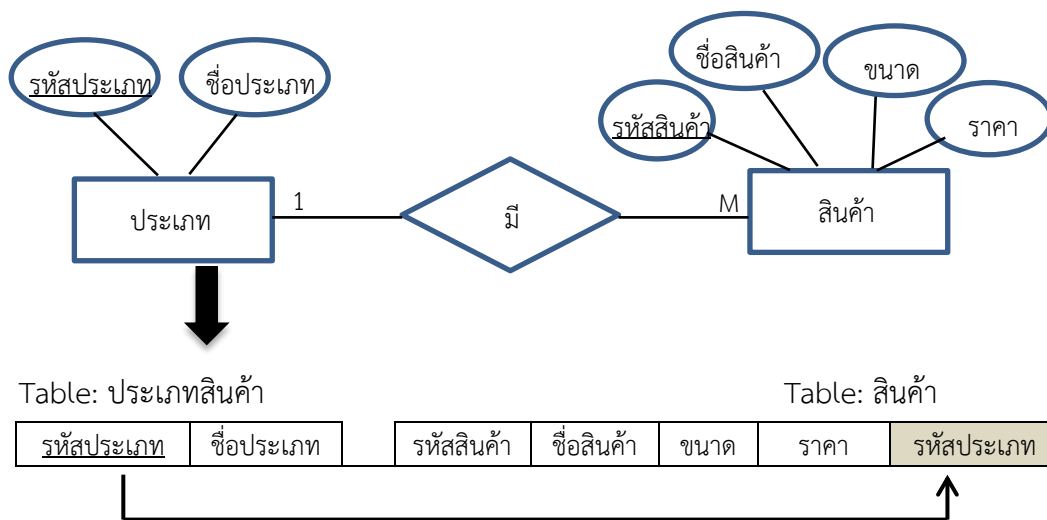
Table: ที่ जोดรอง

รหัสที่ जोด	โซน	ลำดับ	ห้อง
-------------	-----	-------	------



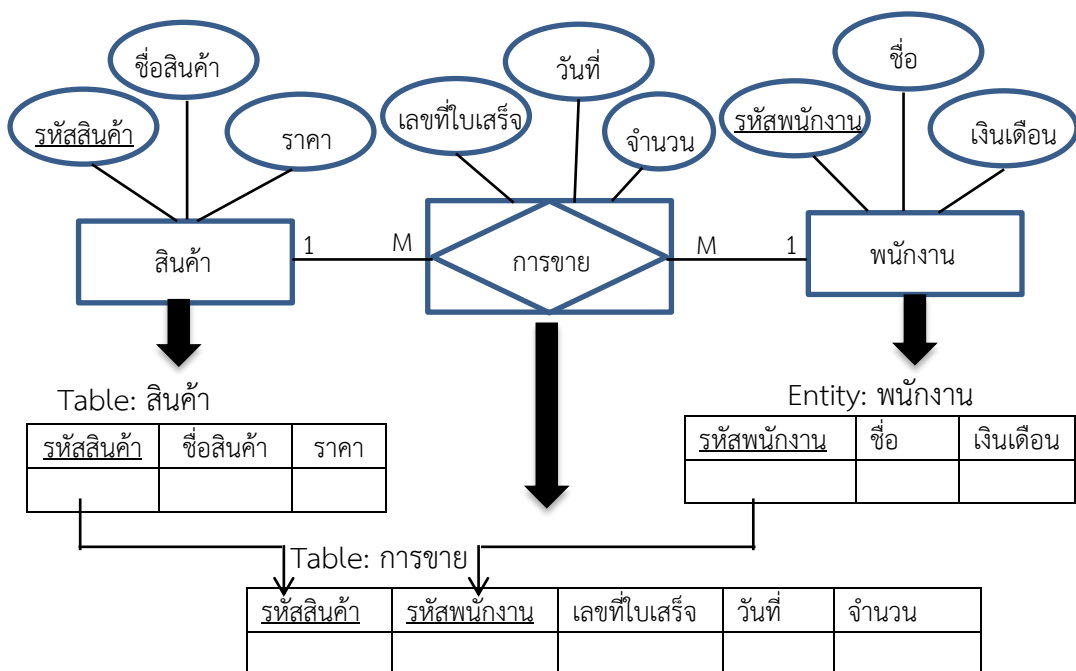
ภาพที่ 4.31 การกำหนดคีย์นอกจากการแปลงความสัมพันธ์ระหว่างเอนทิตีแบบ 1:1

- กรณีที่เอนทิตีมีความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อหนึ่ง (1:M) ให้นำคีย์หลักของเอนทิตีฝั่งที่เป็น 1 ไปเป็นแอตทริบิวต์หนึ่งของเอนทิตีที่ฝั่ง M โดยแอตทริบิวต์นั้นมีคุณสมบัติเป็นคีย์นอก (Foreign Key) ตัวอย่างเช่น เอนทิตีประเภทสินค้ากับเอนทิตีสินค้ามีความสัมพันธ์กันแบบ 1:M เมื่อแปลงไปเป็นตาราง ให้นำเอาคีย์หลักจากเอนทิตีประเภทสินค้าซึ่งเป็น 1 ไปเพิ่มเป็นแอตทริบิวต์ในเอนทิตีสินค้าซึ่งเป็น M เพื่อเป็นคีย์นอก ดังภาพที่ 4.32



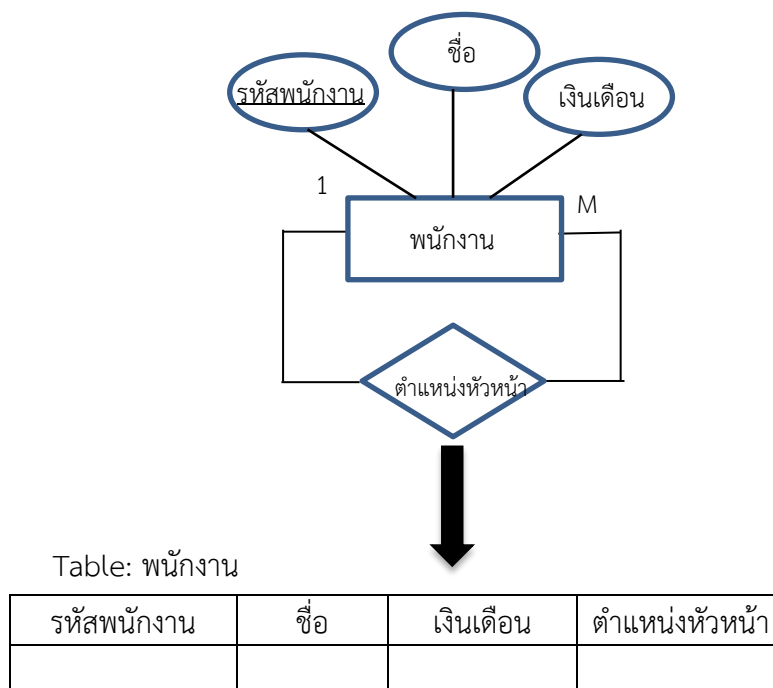
ภาพที่ 4.32 การกำหนดคีย์นอกจากการแปลงความสัมพันธ์ระหว่างเอนทิตีแบบ 1:M

- กรณีที่เป็นความสัมพันธ์ระหว่างเอนทิตีแบบกลุ่มต่อกลุ่ม (M:N) ชื่อความสัมพันธ์จะเป็นเอนทิตีเชิงความสัมพันธ์ (Associate Entity) ซึ่งจะถูกกำหนดให้เป็นอีก 1 เอนทิตี โดยมีเงื่อนไขคือ ให้นำคีย์หลักของเอนทิตีหลักทั้ง 2 มารวมกันเพื่อกำหนดให้เป็นคีย์หลักของเอนทิตีที่เกิดจากความสัมพันธ์ ตัวอย่างเช่น เอนทิตีสินค้ากับเอนทิตีพนักงาน มีความสัมพันธ์แบบ M:N ชื่อความสัมพันธ์คือ การขาย ดังนั้นเมื่อแปลงเป็นตารางจะทำได้ 3 ตารางคือ ตารางสินค้า ตารางพนักงาน และตารางการขาย ดังภาพที่ 4.33



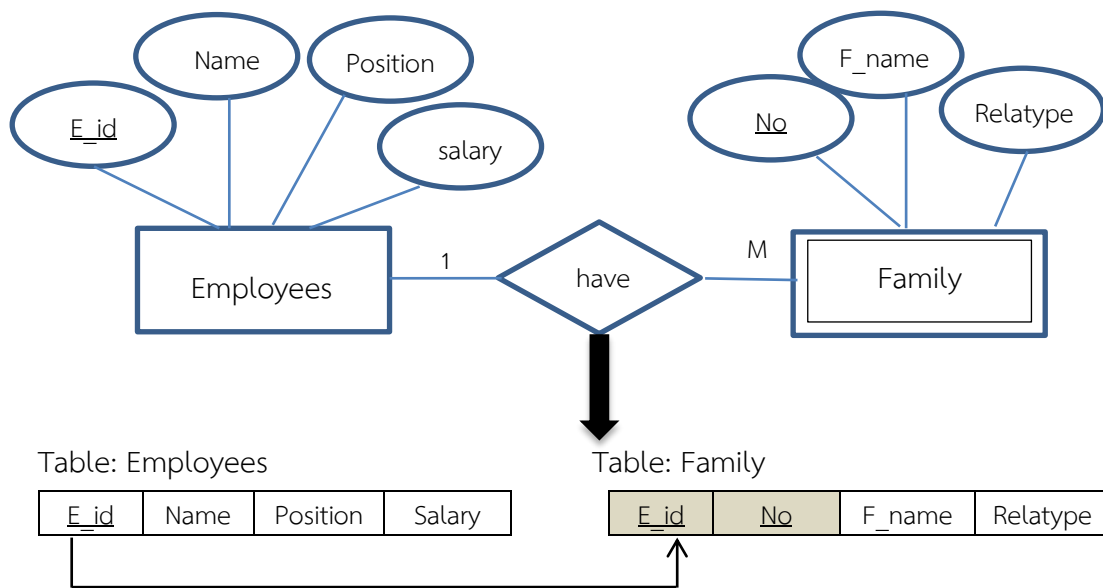
ภาพที่ 4.33 การแปลงความสัมพันธ์ระหว่างเอนทิตีแบบ M:N

- กรณีที่เกิดความสัมพันธ์แบบ ยูนารี (Unary Relationships) ที่เป็นความสัมพันธ์ที่เกิดขึ้นภายในเอนทิตีเอง เช่น ในเอนทิตีพนักงาน เกิดความสัมพันธ์ในคือ มีพนักงาน 1 คนที่ทำหน้าที่เป็นตำแหน่งหัวหน้าแผนก ตัวความสัมพันธ์ที่เกิดขึ้นจะถือเป็นแอตทริบิวต์ในตาราง ดังภาพที่ 4.34



ภาพที่ 4.34 การแปลงความสัมพันธ์ของเอนทิตีแบบ ยูนารี (Unary Relationships)

- กรณีที่เกิดความสัมพันธ์ระหว่างเอนทิตีแบบ Strong Entity กับ เอนทิตีแบบ Weak Entity วิธีการแปลงคือ ให้นำคีย์หลักของเอนทิตีฝั่ง Strong Entity ไปรวมกับคีย์หลักของเอนทิตีฝั่ง Weak Entity เพื่อรวมกันเป็น คีย์หลัก ตัวอย่างเช่น เอนทิตี Employees มีความสัมพันธ์กับเอนทิตี Family ที่เป็น Weak Entity ต้องนำเอาแอตทริบิวต์ E_id ซึ่งเป็นคีย์หลักของ Employees ไปรวมกับแอตทริบิวต์ No รวมกันเป็นคีย์หลัก ของเอนทิตี Family ดังภาพที่ 4.35



ภาพที่ 4.35 การแปลงความสัมพันธ์ระหว่าง Strong Entity และ Weak Entity

3. ตรวจสอบเค้าร่างข้อมูลในแต่ละตาราง

เป็นการนำตารางที่ได้จากการแปลงแผนภาพอี อาร์ มาเป็นตารางทั้งหมดมาตรวจสอบความถูกต้องในภาพรวม โดยจะตรวจสอบในส่วนของ ตาราง แอตทริบิวต์ และคีย์ต่าง ๆ ที่มีในฐานข้อมูลว่ามีความเหมาะสมหรือไม่

4.4 กรณีศึกษาเพื่อพัฒนาแผนภาพอี อาร์

มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน มีหอพักสำหรับให้บริการนักศึกษาจำนวน 2 หลังด้วยกัน แบ่งเป็นหอพักนักศึกษาชายและหญิงแยกกันอยู่คนละหลัง และแต่ละหลังก็จะมีจำนวนห้องพักไม่เท่ากัน แต่ละห้องพักของแต่ละหอ แต่ละห้องพักสามารถจุนักศึกษาได้ห้องละ 4 คน อัตราค่าที่พักจะแตกต่างกันที่เครื่องอำนวยความสะดวกที่มีในห้อง คือ ห้องพัดลม จะมีค่าเช่า 3,000 บาทต่อภาคการศึกษา และห้องที่มีเครื่องปรับอากาศ จะมีค่าเช่า 5,000 บาท ในแต่ละปีมีนักศึกษาที่มีความประสงค์จะเข้าพักในหอพักเป็นจำนวนมาก และมักจะมีปัญหาในการตรวจสอบควบคุม และจัดการเกี่ยวกับการจองห้องพัก และการชำระค่าเช่าห้องพักจึงได้คิดที่จะพัฒนาระบบสารสนเทศเพื่อการจัดการห้องพักขึ้นมา นักศึกษาสามารถเลือกจองห้องพักเองได้ ตามความประสงค์ และตามห้องที่ยังว่างอยู่

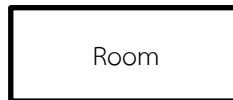
ขั้นตอนในการสร้าง E-R Diagram มีดังนี้

1. กำหนดเอนทิตีที่เกี่ยวข้อง ประกอบด้วย

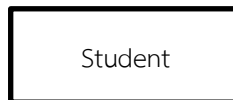
- เอนทิตีห้องพัก Apartment



- เอนทิตีห้องพัก Room



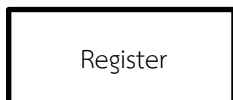
- เอนทิตีนักศึกษา Student



- เอนทิตีการจองห้องพัก Reserve



- เอนทิตี การลงทะเบียนเข้าพัก Register



- เอนทิตี ใบเสร็จ Receipt



2. กำหนดความสัมพันธ์ของเอนทิตี

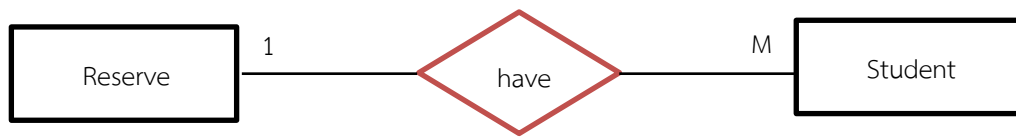
- เอนทิตี Apartment กับ เอนทิตี Room มีความสัมพันธ์แบบ 1:M



- เอนทิตี Room กับ Reserve มีความสัมพันธ์แบบ 1:M



- เอนทิตี Reserve กับ Student มีความสัมพันธ์แบบ 1:M



- เอนทิตี Student กับ Register มีความสัมพันธ์แบบ 1:M



- เอนทิตี Room กับ Register มีความสัมพันธ์แบบ 1:M



- เอนทิตี Reserve กับ Register มีความสัมพันธ์แบบ 1:M

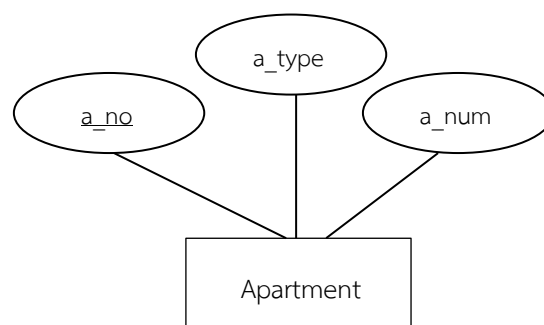


- เอนทิตี Register กับ Receipt มีความสัมพันธ์แบบ 1:M

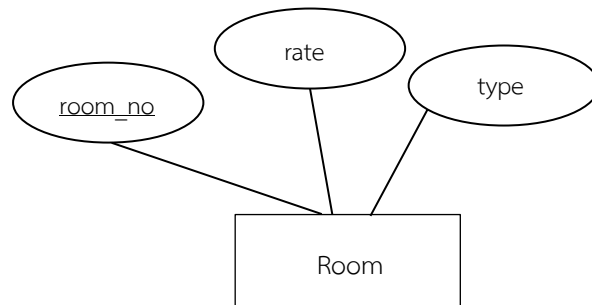


3. กำหนดแอตทริบิวต์ และระบุคีย์หลัก

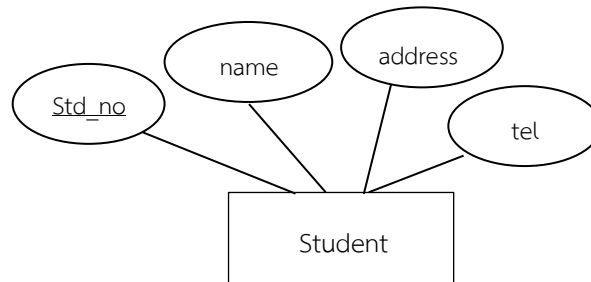
- Apartment ประกอบด้วยแอตทริบิวต์ a_no, a_type, a_num โดยมี a_no เป็นคีย์หลัก



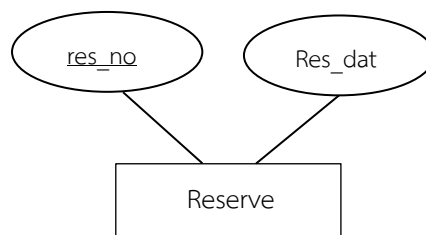
- Room ประกอบด้วยแอตทริบิวต์ room_no, rate, type โดยมี room_no เป็นคีย์หลัก



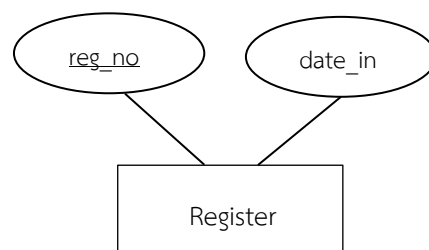
- Student ประกอบด้วย std_no, name, address, tel โดยมี std_no เป็นคีย์หลัก



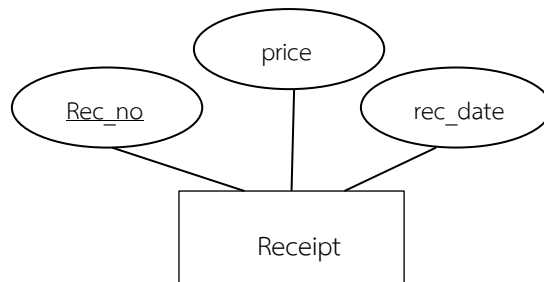
- Reserve ประกอบด้วย res_no, res_date โดยมี res_no เป็นคีย์หลัก



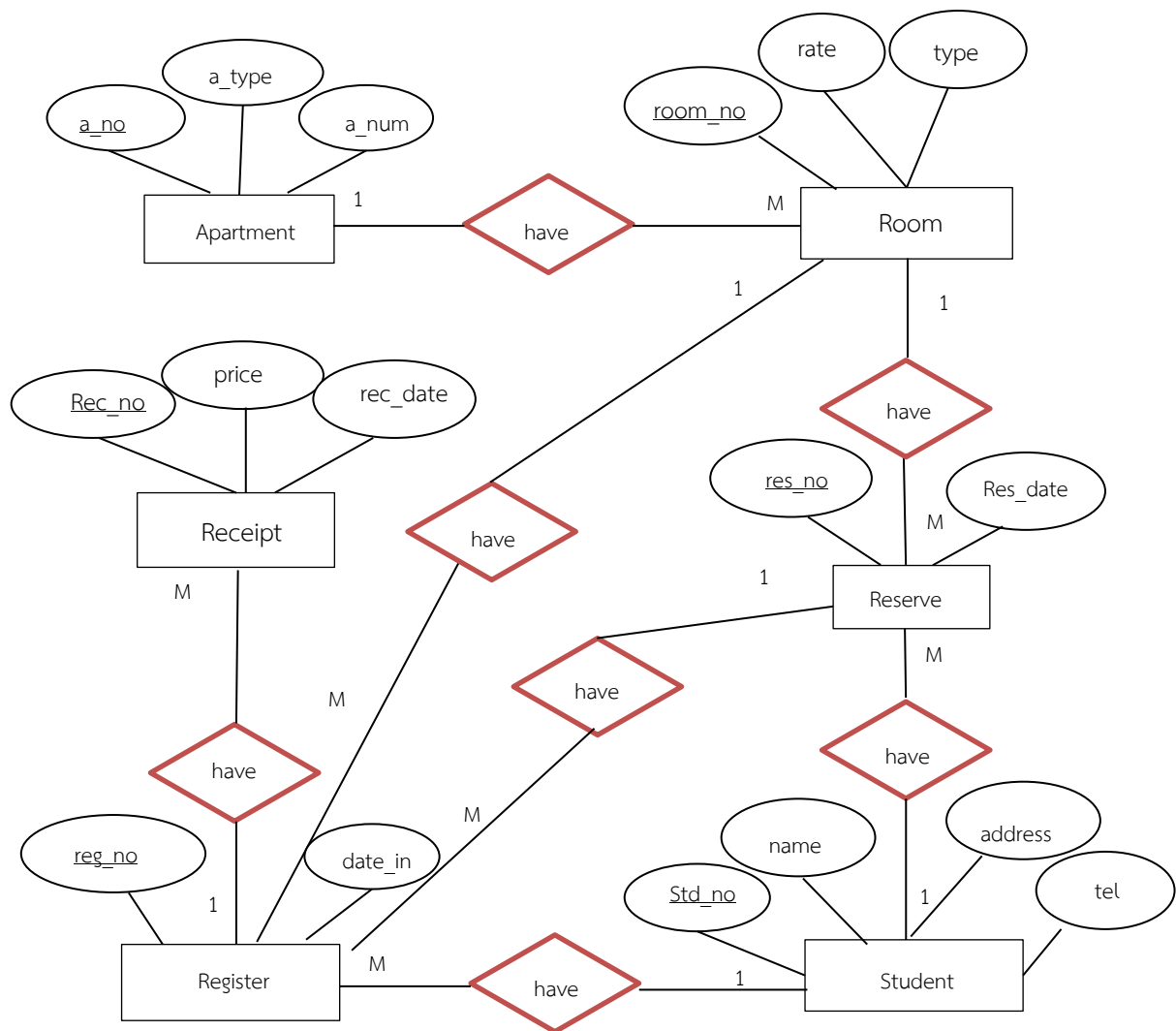
- Register ประกอบด้วย reg_no, date_in โดยมี reg_no เป็น คีย์หลัก



- Receipt ประกอบด้วย rec_no, price, rec_date โดยมี rec_no เป็นคีย์หลัก



4. รวมภาพทั้งหมดเข้าด้วยกันเป็นแผนภาพ E-R Diagram



ภาพที่ 4.36 E-R Diagram ระบบหอพักนักศึกษา

5. การแปลงแผนภาพอี อาร์ให้เป็นตาราง ทำได้ดังนี้

Table : Apartment

<u>a_no</u>	a_type	a_num

Table: Room

<u>room_no</u>	rate	type	a_no

Table: Reserve

<u>res_no</u>	Res_date	room_no	Std_no

Table: Receipt

<u>Rec_no</u>	price	rec_date	reg_no

Table: Register

<u>reg_no</u>	date_in	res_no	Std_no

Table: Student

<u>Std_no</u>	name	address	tel

สรุป

แบบจำลองอี อาร์ เป็นแผนภาพที่พัฒนาขึ้นเพื่ออธิบายภาพรวมของการจัดเก็บข้อมูลในฐานข้อมูลประกอบด้วย เอ็นทิตี แอททริบิวต์ และความสัมพันธ์ โดยจะใช้สัญลักษณ์เพื่อแทนความหมายของส่วนประกอบต่าง ๆ นักออกแบบฐานข้อมูลจะใช้ อี อาร์ โมเดล เพื่อเป็นเครื่องมือในการสื่อสารกับผู้ใช้ให้เข้าใจเกี่ยวกับฐานข้อมูลที่ได้ออกแบบ แบบจำลอง อี อาร์ โมเดล เป็นแบบจำลองที่ไม่ยึดติดกับโปรแกรมจัดการฐานข้อมูล ดังนั้น นักออกแบบฐานข้อมูลจำเป็นต้องทำการแปลงอี อาร์ ไปเป็นตารางเก็บข้อมูล แล้วนำตารางที่ได้ไปตรวจสอบหาความซ้ำซ้อนด้วยการทำให้เป็นบรรทัดฐานต่อไป

แบบฝึกหัดท้ายบท

1. ให้ศึกษารายละเอียดข้อมูลต่อไปนี้

ในคณะหนึ่งประกอบด้วยภาควิชาต่าง ๆ และในแต่ละภาควิชาประกอบด้วยอาจารย์หลายคน ซึ่งอาจารย์แต่ละคนจะสังกัดภาควิชาใดภาควิชาหนึ่งเท่านั้น อาจารย์แต่ละคนจะทำหน้าที่เป็นอาจารย์ที่ปรึกษาให้กับนักศึกษาหลายคนในภาควิชา รวมทั้งทำการสอนชุดวิชาต่าง ๆ ตามหลักสูตรที่เปิดให้นักศึกษาลงทะเบียนเรียนในแต่ละภาคการศึกษา โดยอาจารย์หนึ่งคนอาจสอนหลายชุดวิชา ซึ่งในแต่ละชุดวิชาอาจมีได้มากกว่าหนึ่งหมู่เรียน

นอกจากนี้ในแต่ละภาควิชาประกอบด้วยนักศึกษาหลายคน ซึ่งนักศึกษาแต่ละคนจะสังกัดภาควิชาใดภาควิชาหนึ่งเท่านั้น โดยนักศึกษาแต่ละคนจะมีอาจารย์ที่ปรึกษาหนึ่งคน ในหนึ่งภาคการศึกษา จะมีชุดวิชาในหลักสูตรที่เปิดให้นักศึกษาลงทะเบียนเรียนได้หลายชุดวิชา ซึ่งนักศึกษาทุกคนจะต้องลงทะเบียนเรียนอย่างน้อยหนึ่งชุดวิชา โดยในแต่ละชุดวิชาที่เปิดทำการสอนอาจมีได้มากกว่าหนึ่งหมู่เรียน ซึ่งในแต่ละหมู่เรียนจะมีอาจารย์ผู้สอนเพียงหนึ่งคนเท่านั้น ทั้งนี้ รายงานข้อมูลพื้นฐานทั่วไปที่ต้องการ ได้แก่ รายชื่อนักศึกษาในแต่ละภาควิชา รายชื่อ นักศึกษาในความดูแลของอาจารย์ที่ปรึกษาแต่ละคน เป็นต้น รายงานข้อมูลที่ต้องการในแต่ละภาคการศึกษา ได้แก่ ข้อมูลการลงทะเบียนเรียนของนักศึกษาแต่ละคน รายงานผลการสอบของนักศึกษาแต่ละคน รายชื่อนักศึกษาที่ลงทะเบียนเรียนในแต่ละชุดวิชาจำแนกตามหมู่เรียน รายชื่ออาจารย์ผู้สอนในแต่ละชุดวิชาจำแนกตามหมู่เรียน เป็นต้น

1. ให้นักศึกษาเขียน E-R Model ของระบบงาน

2. แปลง E-R Model เป็นตาราง

2. จาก อี อาร์ ต่อไปนี้ ให้นักศึกษาแปลงเป็นตารางเก็บข้อมูล

