

## บทที่ 3

### แบบจำลองข้อมูล (Data Models)

เป็นที่ทราบกันดีแล้วว่า การใช้ฐานข้อมูลเป็นตัวเลือกอันดับต้น ๆ ของการจัดเก็บข้อมูลของระบบสารสนเทศ และในปัจจุบันได้มีการนำคอมพิวเตอร์มาใช้ในการจัดการกับข้อมูล หรืองานอื่น ๆ ก่อให้เกิดประโยชน์อย่างมหาศาลในทุก ๆ ด้าน ทำให้สามารถจัดเก็บข้อมูลไว้ในฐานข้อมูลได้เป็นจำนวนมาก การเรียกใช้ข้อมูลกระทำได้ง่าย มีความสะดวกมากขึ้น ข้อมูลครอบคลุมตอบสนองความต้องการของผู้ใช้ การนำเสนอรายงานจากฐานข้อมูลมีความหลากหลาย และสวยงาม จะเห็นได้ว่าระบบฐานข้อมูลในปัจจุบันมีการพัฒนาอย่างต่อเนื่องและเจริญก้าวหน้าไปอย่างรวดเร็ว สามารถจัดการข้อมูลได้อย่างมีประสิทธิภาพ

ข้อมูลที่เก็บไว้ในฐานข้อมูลส่วนใหญ่ มักจะมีรายละเอียดของข้อมูลเป็นจำนวนมาก ทำให้ผู้ออกแบบ ผู้เขียนโปรแกรม และผู้ใช้งานระบบฐานข้อมูล มีมุมมองเกี่ยวกับข้อมูลชุดเดียวกันที่อยู่ภายในฐานข้อมูลในมุมมองที่แตกต่างกัน ทำให้การออกแบบฐานข้อมูลเกิดข้อผิดพลาด ไม่สามารถสะท้อนให้เห็นถึงขั้นตอนการดำเนินงานนั้นๆ ได้ เพื่อหลีกเลี่ยงไม่ให้เกิดเหตุการณ์ดังกล่าวเกิดขึ้น ผู้ออกแบบฐานข้อมูล ผู้เขียนโปรแกรม และผู้ใช้งานระบบฐานข้อมูลจะต้องติดต่อสื่อสารกันบ่อย ๆ เพื่อที่จะได้ทราบถึงธรรมชาติของข้อมูล การประยุกต์ใช้งานข้อมูลอย่างชัดเจน และมีความเข้าใจเกี่ยวกับข้อมูลที่ตรงกัน นอกจากนั้นการประยุกต์ใช้แบบจำลองข้อมูลหรือ แบบจำลองฐานข้อมูล (database model) จะทำให้ผู้ออกแบบ ผู้เขียนโปรแกรม และผู้ใช้งานระบบฐานข้อมูลเข้าใจข้อมูลได้ง่าย สามารถลดความซับซ้อนของการออกแบบฐานข้อมูลให้มีความ เข้าใจง่ายมากขึ้นด้วย ดังนั้นการจัดเก็บข้อมูล และการเลือกใช้ข้อมูลให้มีประสิทธิภาพจะต้องอาศัยเทคนิคต่าง ๆ เข้ามาช่วยเพื่อให้ง่ายต่อการใช้งาน

วัตถุประสงค์สำคัญของแบบจำลองข้อมูลก็เพื่อที่จะนำแนวคิด ต่าง ๆ มานำเสนอให้เกิดเป็นรูปแบบจำลองที่ใช้สำหรับการอธิบาย เพื่อให้ทราบว่าข้อมูลขององค์กรเหล่านั้นมีอะไรบ้าง ? โครงสร้างของข้อมูลเป็นอย่างไร ? ความสัมพันธ์ระหว่างข้อมูลภายในฐานข้อมูลนั้นเป็นอย่างไร ? และที่สำคัญคือ จากรูปแบบที่เป็นแนวคิดที่เข้าใจยากให้อยู่ในรูปแบบที่เข้าใจได้ง่าย โดยไม่ขึ้นอยู่กับโครงสร้างของฐานข้อมูลนั้น ๆ และที่สำคัญอีกประการหนึ่งคือเพื่อใช้ในการสื่อสารระหว่าง

ผู้ออกแบบฐานข้อมูล ผู้เขียนโปรแกรม และผู้ใช้งานระบบฐานข้อมูล ให้เกิดความเข้าใจที่ถูกต้องตรงกัน มีผู้ให้นิยาม แบบจำลองข้อมูล หรือแบบจำลองฐานข้อมูล ไว้ ดังนี้

แบบจำลองฐานข้อมูล หมายถึงโครงสร้างข้อมูลระดับตรรกะที่นำเสนอข้อมูล และความสัมพันธ์ระหว่างข้อมูลให้ผู้ใช้เห็น และเข้าใจแบบจำลองข้อมูลที่ใช้โดยพิจารณาจาก ภายในฐานข้อมูลประกอบด้วยข้อมูลอะไรบ้าง? ความสัมพันธ์ของข้อมูลเป็นอย่างไร? โครงสร้างข้อมูลเป็นอย่างไร? และมีกฎควบคุมความถูกต้องบนโครงสร้างข้อมูลอย่างไร? (สมลักษณ์ ละองศรี, 2549 : 95)

แบบจำลองฐานข้อมูล เป็นแหล่งรวมของแนวความคิดที่พรรณนาถึงความเป็นจริงของวัตถุข้อมูล และเหตุการณ์ รวมทั้งความสัมพันธ์ให้มีความถูกต้องตรงกันในกฎเกณฑ์ที่กำหนด (โอภาส เอี่ยมสิริวงศ์, 2549 : 58)

แบบจำลองฐานข้อมูล เป็นแบบจำลองที่ใช้อธิบายถึงโครงสร้าง และความสัมพันธ์ ระหว่างข้อมูลภายในฐานข้อมูล จากแนวความคิดที่เข้าใจยากให้สามารถเข้าใจและจับต้องได้ง่ายขึ้น ดังนั้นแบบจำลองฐานข้อมูลจึงถูกนำไปใช้อธิบายถึงโครงสร้างของฐานข้อมูลประเภทต่าง ๆ ที่อยู่ในรูปของทฤษฎี ที่เข้าใจยาก รวมทั้งนำไปใช้ในการอธิบาย ถึงโครงสร้างของฐานข้อมูลที่นักออกแบบฐานข้อมูลได้ออกแบบไว้ ซึ่งอยู่ในรูปแบบของแนวความคิด และจับต้องได้ยากเช่น เดียวกัน (กิตติ ภัคศิริวัฒนกุล และจำลอง คุรุอุตสาหะ, 2550 : 30)

อาจจะกล่าวอย่างสรุปได้ว่าแบบจำลองฐานข้อมูล คือ เครื่องมือ ที่นำมาใช้ในการจัดการโครงสร้างของข้อมูลในฐานข้อมูล ที่แสดงให้เห็นถึง ความหมายของข้อมูล ส่วนประกอบของข้อมูล ความสัมพันธ์ระหว่างข้อมูลภายในฐานข้อมูล และเงื่อนไขบังคับความสัมพันธ์ของข้อมูล ให้อยู่ในรูปแบบที่ผู้ใช้สามารถเห็น เข้าใจ เข้าถึงข้อมูลได้ง่าย สะดวก และปลอดภัย

ระบบจะสามารถตอบสนองความต้องการของผู้ใช้ได้มากที่สุด จำเป็นอย่างยิ่งที่จะต้องมีการกำหนดโครงสร้างฐานข้อมูลให้ชัดเจนในเบื้องต้น ส่วนที่ต้องออกแบบเพื่อให้เห็นโครงสร้างของข้อมูลที่ชัดเจนคือ สถาปัตยกรรมฐานข้อมูล (Database Architecture) หรือแนวคิดที่ใช้อธิบายโครงสร้างองค์ประกอบหลักของระบบข้อมูลภายในฐานข้อมูล การจัดเก็บข้อมูลภายในฐานข้อมูล รวมถึงการติดต่อกับส่วนอื่น ๆ ที่เกี่ยวข้อง

## 1. องค์ประกอบของแบบจำลองข้อมูล แบบจำลองข้อมูลมีองค์ประกอบทั่ว ๆ ไป ดังนี้

1.1 เอนทิตี (entity) หมายถึง วัตถุ (object) หรือกลุ่มสิ่งของต่างๆที่เราสนใจจะเก็บข้อมูลไว้ในฐานข้อมูล หรือแนวคิดที่สามารถบอกความแตกต่างของแต่ละเอนทิตีได้ ถือเป็นตัวแทนของ

วัตถุในโลกแห่งความเป็นจริง เช่น คน สัตว์ สถานที่ สิ่งของ หรือเหตุการณ์ ซึ่งจะเป็นข้อมูลที่จะถูกจัดเก็บอยู่ในฐานข้อมูล เนื่องจากเอนทิตีหนึ่งจะใช้ แทนข้อมูลชนิดหนึ่ง ๆ ดังนั้นแต่ละเอนทิตีจะต้องมีความแตกต่างกัน และจะต้องมีความเป็นเอกลักษณ์เสมอ เช่น เอนทิตีของนิสิตจะมีข้อมูลนิสิตที่แตกต่างกันหลายคน โดยที่นิสิตแต่ละคน จะมีข้อมูลที่มีความแตกต่างจากนิสิตคนอื่น ๆ เช่น นายมานะ ใจงาม จะมีชื่อและนามสกุลที่แตกต่างจากนายสมชาย ดีพร้อม เป็นต้น

- เอนทิตี ที่เป็นบุคคล เช่น นิสิต อาจารย์ พนักงาน นายแพทย์ พยาบาล วิศวกร แม่บ้าน พนักงานขับรถ ฯลฯ
- เอนทิตี ที่เป็นสถานที่ เช่น หมู่บ้าน ตำบล อำเภอ จังหวัด ประเทศ น้ำตก ภูเขา โรงเรียน ร้านอาหาร ฯลฯ
- เอนทิตี ที่เป็น วัตถุ สิ่งของ เช่น หนังสือ คอมพิวเตอร์ รถ เรือ อาหาร ยา อุปกรณ์ เครื่องมือต่างๆ ฯลฯ
- เอนทิตี ที่เป็น นามธรรม เช่น วิชา ความสามารถ วัน เวลา ฯลฯ

1.2 เอนทิตีเซต (entity set) กลุ่มของเอนทิตีที่มีคุณสมบัติเหมือนกัน สัญลักษณ์จะใช้รูปสี่เหลี่ยม (rectangles)

1.3 แอททริบิวต์ (attribute) หมายถึงคุณลักษณะ หรือ คุณสมบัติ (properties) ของเอนทิตี เช่น

- แอททริบิวต์ ของเอนทิตี “นิสิต” คือข้อมูลเอนทิตี นิสิต ที่อธิบายด้วยแอททริบิวต์ต่าง ๆ เช่น ชื่อ-สกุล, รหัสนิสิต, คณะวิชา, สาขาวิชา, หมายเลขโทรศัพท์, คะแนนสะสม และที่อยู่ เป็นต้น
- แอททริบิวต์ ของเอนทิตี “ลงทะเบียน” คือข้อมูลเอนทิตี ลงทะเบียนที่อธิบายด้วยแอททริบิวต์ต่าง ๆ เช่น รหัสวิชา, รหัสนิสิต, ชื่อวิชา, จำนวนหน่วยกิต, ชื่อผู้สอน , และปีการศึกษา เป็นต้น
- แอททริบิวต์ ของเอนทิตี “ลูกค้า” คือข้อมูลเอนทิตี ลูกค้า ที่อธิบายด้วยแอททริบิวต์ต่าง ๆ เช่น รหัสลูกค้า, ชื่อสกุล, สถานภาพ, อาชีพ, อายุ , หมายเลขโทรศัพท์ และที่อยู่ เป็นต้น

แอททริบิวต์ ในระบบฐานข้อมูลมักจะมีลักษณะเหมือนกับฟิลด์ในแฟ้มข้อมูล สัญลักษณ์จะใช้รูปวงรี (ellipse) แอททริบิวต์ที่ไม่สามารถแยกข้อมูลออกเป็นข้อมูลย่อย ๆ ได้อีกเช่น “รหัสนิสิต” ไม่สามารถที่จะแยกออกเป็นข้อมูลอื่น ได้อีก สัญลักษณ์ที่ใช้คือ วงรีเส้นขอบเส้นเดียว ภายในมีชื่อแอททริบิวต์บรรจุอยู่ แอททริบิวต์ประเภทนี้เรียกว่า Simple Attribute หรือ Atomic Attribute

ส่วนแอททริบิวต์ที่สามารถแบ่งออกเป็นแอททริบิวต์ย่อย ๆ ได้อีก เช่น แอททริบิวต์ “ที่อยู่” สามารถแบ่งออกเป็นแอททริบิวต์ย่อยได้เป็น บ้านเลขที่ ถนน ตำบล อำเภอ จังหวัด ประเทศ เป็นต้น แอททริบิวต์ประเภทนี้เรียกว่า Composite Attribute สัญลักษณ์ที่ใช้ คือวงรีเส้นขอบเส้นเดียว แต่มีวงรีย่อยมาเชื่อมต่อ

1.4 ความสัมพันธ์ระหว่างเอนทิตี (relationship set) มีสัญลักษณ์เป็นรูปขนมเปียกปูน (diamonds) ความสัมพันธ์ระหว่างเอนทิตี มี 3 แบบด้วยกันคือ แบบ one-to-one (1 : 1) one-to-many (1 : M) และแบบ many-to-many (M : M หรือ M : N)

1.5 ข้อจำกัด (Constraints) หรือข้อกำหนด หมายถึงกฎเกณฑ์ หรือข้อตกลงของข้อมูล เพื่อเกิดความมั่นใจในความเป็นอันหนึ่งอันเดียวกัน หรือเกิดความสอดคล้องตรงกันของข้อมูล

ภายในแบบจำลองของข้อมูลจะประกอบด้วยส่วนหลัก ๆ 3 ส่วนด้วยกันคือ

- ส่วนโครงสร้าง (Structural Part) ประกอบด้วยกลุ่มสัญลักษณ์ กฎระเบียบ ที่เห็นพ้องต้องกันเพื่อใช้ในการสร้างฐานข้อมูล เช่น การจัดเก็บข้อมูลในรูปแบบของตาราง (table) ส่วนที่ใช้แทนข้อมูลต่าง ๆ ที่ประกอบกันเป็นฐานข้อมูล เช่น รูปสี่เหลี่ยม ใช้แทนเอนทิตีเซต รูปวงรี ใช้แทนแอททริบิวต์ รูปสี่เหลี่ยมขนมเปียกปูน ใช้แทนรีเลชันชิพ และรูปเส้นตรง ใช้แทนการเชื่อมต่อของแอททริบิวต์กับเอนทิตีเซต และการเชื่อมต่อของเอนทิตีเซตกับรีเลชันชิพ เป็นต้น

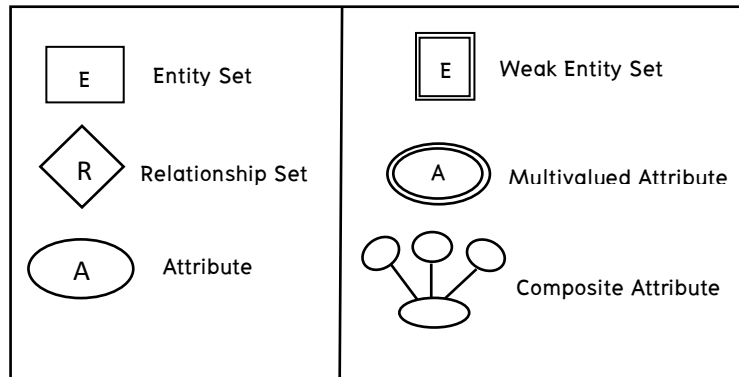
- ส่วนปรับปรุง (Manipulative Part) เป็นส่วนที่กำหนดชนิดของการปฏิบัติการต่าง ๆ ที่เกี่ยวกับข้อมูล ประกอบไปด้วยการปรับปรุง (update) ข้อมูล การเรียกดูข้อมูลจากฐานข้อมูล การเปลี่ยนแปลงโครงสร้างของฐานข้อมูล เป็นต้น

- กฎความคงสภาพ (Integrity Rules) เป็นกฎเกณฑ์ที่ใช้ควบคุมความถูกต้องของข้อมูลภายในฐานข้อมูล เพื่อให้เกิดความมั่นใจในความถูกต้องก่อนการบันทึกข้อมูล

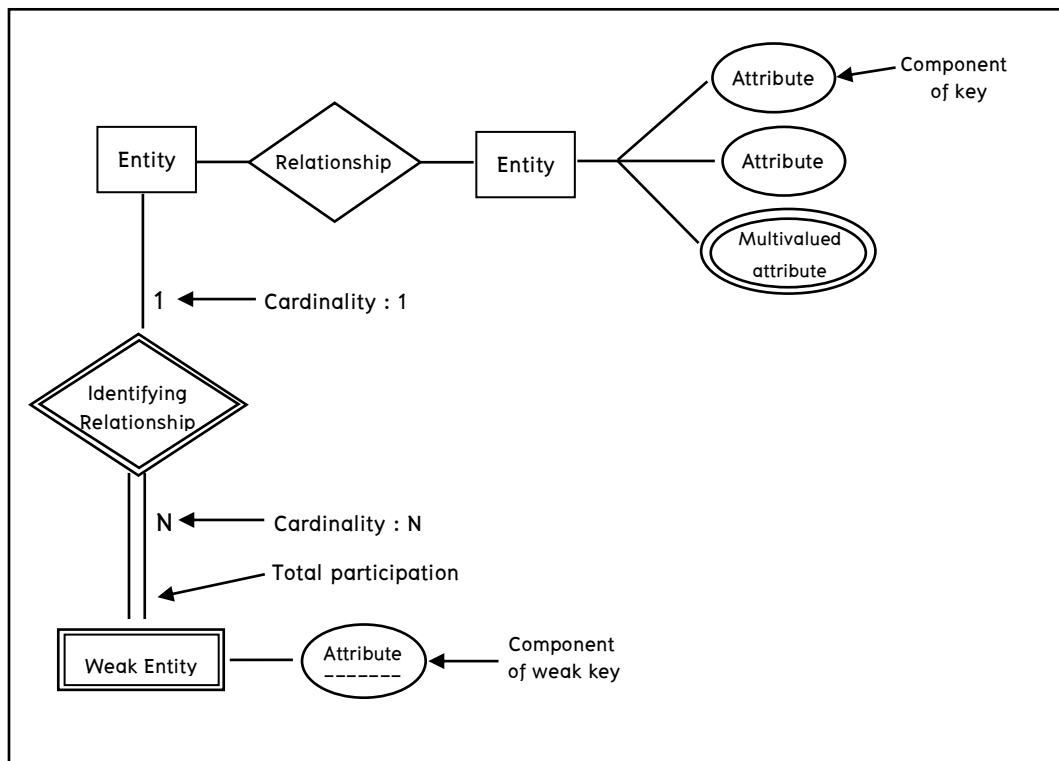
## 2. วิวัฒนาการของแบบจำลองข้อมูล (Evolution of Data Models)

การคิดหาวิธีในการจัดการข้อมูลที่ดี จะนำไปสู่การพัฒนาแบบจำลองข้อมูลที่มีความหลากหลาย และมีประสิทธิภาพในการทำงาน ในยุคแรก ๆ ก่อนที่จะได้นำเอาคอมพิวเตอร์มาช่วยการทำงานในด้านการบันทึกการจัดเก็บข้อมูล การจัดเก็บข้อมูลจะอยู่ในรูปแบบเอกสารต่าง ๆ ซึ่งแต่ละหน่วยงานก็จะมีแฟ้มข้อมูลของตนเอง ต่อมาเมื่อมีการนำคอมพิวเตอร์มาใช้ในการประมวลผลข้อมูล การเก็บรวบรวมข้อมูลเปลี่ยนมาอยู่ในรูปของการจัดเก็บข้อมูลบนสื่อบันทึกข้อมูลอิเล็กทรอนิกส์ต่าง ๆ สื่อบันทึกข้อมูลเหล่านี้สามารถจุข้อมูลได้หลายล้านตัวอักษร จึงเป็นการ

ประหยัดเนื้อที่ในการเก็บรวบรวมข้อมูล และการประมวลผลข้อมูลที่จัดเก็บไว้บนสื่ออิเล็กทรอนิกส์  
เหล่านี้ สามารถกระทำได้อย่างสะดวก รวดเร็ว ทันเวลา ทันต่อการปฏิบัติงาน



ภาพที่ 3.1 แสดงสัญลักษณ์ต่างๆ ของแบบจำลองข้อมูล  
(ที่มา : <https://www.google.co.th/search?q=entity+set+symbol&espv=2&source>)



ภาพที่ 3.2 แสดงองค์ประกอบแบบจำลองข้อมูล  
(ที่มา : <https://www.google.co.th/search?q=entity+set+symbol&espv=2&source>)

การสร้างแบบจำลองข้อมูลและฐานข้อมูลได้มีการพัฒนาร่วมกันและ อาจจะสรุปประวัติความเป็นมา หรือวิวัฒนาการแบบจำลองข้อมูลอาจจะแบ่งออกได้เป็น 5 รุ่น (generations) ด้วยกัน ดังนี้

**รุ่นที่ 1 (First generation) ระบบแฟ้มข้อมูล (File system)** อยู่ในช่วงเวลาระหว่างปี ค.ศ 1960s – 1970s

แบบจำลองข้อมูลในยุคแรก คือระบบแฟ้มข้อมูล ซึ่งเป็นการรวบรวมและการใช้งานไฟล์ต่างๆ ที่เกี่ยวข้องกัน มีการควบคุมโครงสร้างของไฟล์เพื่อให้ใช้งานร่วมกันได้อย่างมีประสิทธิภาพ ระบบนี้มีการประมวลผลข้อมูลที่สามารถทำได้อย่างรวดเร็ว โปรแกรมประยุกต์แต่ละโปรแกรมสามารถควบคุมการใช้ข้อมูลในไฟล์ของตนเองได้ และการใช้งานก็ไม่ยุ่งยาก ไฟล์ต่างๆ มีความปลอดภัยและความคล่องตัวในการประมวลผลสูงถ้ามีผู้ใช้งานเพียงแค่คนเดียว ตัวอย่างเช่น VMS/VSAM

แต่อย่างไรก็ตามระบบแฟ้มข้อมูลของแต่ละหน่วยงานจะถูกเขียนขึ้นด้วยหลาย ๆ โปรแกรม การใช้งานในแต่ละหน่วย งานก็มีความแตกต่างกัน ในการพัฒนาโปรแกรมจะทำให้เสียค่าใช้จ่ายเพิ่ม และการจัดเก็บข้อมูลแบบระบบแฟ้มข้อมูลนี้ อาจจะก่อให้เกิดปัญหาหลายประการเช่น ความซ้ำซ้อนของข้อมูล (data redundancy) ความไม่สอดคล้องกันของข้อมูล (data inconsistency) ความไม่ปลอดภัยของข้อมูล (poor security) ขาดบูรณภาพของข้อมูล (lack of data integrity) ข้อมูลแยกเป็นอิสระต่อกัน (data isolation) และโปรแกรมประยุกต์กับโครงสร้างของแฟ้มข้อมูล (application/data dependence) ไม่มีความเป็นอิสระต่อกัน

**รุ่นที่ 2 (Second generation) แบบจำลองข้อมูลแบบลำดับชั้น (hierarchical model) และ แบบจำลองข้อมูลแบบเครือข่าย (network model)** อยู่ในช่วงเวลา ค.ศ 1970s

แบบจำลองข้อมูลแบบลำดับชั้น ได้เริ่มพัฒนาขึ้นมาในปลายปี ค.ศ 1960 เพื่อแก้ไขปัญหาการเก็บข้อมูลด้วยระบบแฟ้มข้อมูล และจัดการกับปริมาณของข้อมูลที่มีเป็นจำนวนมาก และมีความซับซ้อนขึ้นเรื่อย ๆ โครงสร้างแบบจำลองนี้จะมีลักษณะเป็นลำดับชั้น คล้าย ๆ กับต้นไม้หัวกลับ (upside-down tree) แต่ละชั้นอาจจะประกอบไปด้วยหลาย ๆ ส่วน หรือ segment โดยที่ segment ที่อยู่ด้านบน นิยมเรียกกันว่า parent segment ส่วนที่อยู่ถัดลงมาจะเป็น children segment ซึ่ง children segment จะมีได้หลาย segment แต่ children segment จะมีเพียง parent segment เดียวเท่านั้น

แบบจำลองฐานข้อมูลลำดับชั้นมีข้อดี คือสามารถกำหนดกฎเกณฑ์ที่ใช้ควบคุมความถูกต้องให้กับข้อมูลภายในฐานข้อมูลได้ง่าย เหมาะกับระบบคอมพิวเตอร์ที่มีขนาดใหญ่ แต่ข้อเสียของแบบจำลองฐานข้อมูลแบบลำดับชั้น คือขาดความยืดหยุ่นในการทำงาน ถ้ามีการเปลี่ยนแปลง

โครงสร้างของข้อมูลในแบบจำลองจะส่งผลให้มีการแก้ไขโปรแกรม และแบบจำลองนี้มีโครงสร้างที่เหมาะสมกับข้อมูลที่มีความสัมพันธ์ในแบบหนึ่งต่อกลุ่มหรือ 1 : M (One-to-Many) เท่านั้น

แบบจำลองข้อมูลแบบเครือข่าย ได้เริ่มพัฒนาขึ้นมาในปลายปี ค.ศ 1960 เช่นกัน โดย C.W. Bachman (Bachman 1965) เพื่อเป็นตัวแทนของความสัมพันธ์ของข้อมูลที่มีความซับซ้อน และพัฒนาประสิทธิภาพการทำงานให้ดีกว่าแบบจำลองข้อมูลแบบลำดับชั้น แบบจำลองแบบเครือข่ายนี้ ความสัมพันธ์จะเรียกว่า เซต (set) ในแต่ละเซตจะประกอบด้วยเรคคอร์ดอย่างน้อย 2 ประเภท คือ owner และ member ซึ่งความสัมพันธ์ที่เกิดขึ้นอาจจะเป็นแบบ 1 : N หรือ M : N (หมายถึง member สามารถมีความสัมพันธ์กับ owner ได้หลาย owner) ตัวอย่างเช่น IMS, ADABAS และ IDS-II

ข้อดีของแบบจำลองแบบเครือข่ายคือ สนับสนุนความสัมพันธ์แบบ M : M ความซับซ้อนของข้อมูลเกิดขึ้นน้อยกว่าแบบลำดับชั้น สามารถเชื่อมโยงข้อมูลแบบไปกลับได้ และมีความยืดหยุ่นในการค้นหาข้อมูล แต่ก็ยังมีข้อเสียคือการเปลี่ยนแปลงในโครงสร้างยังมีความยุ่งยากเมื่อมีความต้องการข้อมูลสารสนเทศที่มากขึ้น ความปลอดภัยของข้อมูลมีน้อย เพราะสามารถเข้าถึงข้อมูลได้โดยตรง

**รุ่นที่ 3 (Third generation) แบบจำลองข้อมูลเชิงสัมพันธ์ (relational model)** อยู่ในช่วงกลางปี ค.ศ 1970s – ปัจจุบัน

จากปัญหาที่เกิดขึ้นดังกล่าว จึงเป็นสาเหตุให้มีการพัฒนาแบบจำลองข้อมูลเชิงสัมพันธ์ขึ้น โดย E. F. Codd ในช่วงทศวรรษ 1970–1980 (Codd 1970) แบบจำลองนี้จะช่วยลดปัญหาและอุปสรรคของการสื่อสารระหว่างผู้ออกแบบ และผู้ใช้งานระบบฐานข้อมูล

แนวความคิดพื้นฐานของแบบจำลองข้อมูลเชิงสัมพันธ์ เป็นแนวคิดทางคณิตศาสตร์เรื่อง ความสัมพันธ์ (relation) ซึ่งเป็นเมทริกซ์หรือตาราง (table) ของข้อมูล ที่ประกอบไปด้วยการรวมกันระหว่างแถว (rows) และ สดมภ์ (columns) ทำให้เข้าใจได้ง่าย สามารถแสดงความสัมพันธ์ได้ทั้งแบบ 1 : 1 , 1 : M และ M : M และใช้ระบบคีย์ (keys) ในการเชื่อมความสัมพันธ์ หรือการอ้างอิงกับตารางอื่น

แบบจำลองข้อมูลเชิงสัมพันธ์ สามารถดำเนินการฟังก์ชันพื้นฐานได้เหมือนกับแบบจำลองข้อมูลแบบลำดับชั้น และแบบเครือข่าย แต่มีข้อได้เปรียบจากแบบจำลองทั้งสองแบบหลายประการด้วยกันคือ โครงสร้างของฐานข้อมูลมีความอิสระจากโปรแกรม ความซับซ้อนของข้อมูลมีน้อยมาก สามารถสื่อสารและเข้าใจได้ง่าย มีระบบรักษาความปลอดภัยที่ดี เพราะผู้ใช้งานไม่ต้องทราบถึงโครงสร้างของการเก็บข้อมูลภายในฐานข้อมูล แต่ในการดำเนินการดังกล่าวต้องมีการลงทุนสูง

เนื่องจากต้องใช้ฮาร์ดแวร์และ ซอฟต์แวร์ ที่มีความสามารถสูง ตัวอย่างเช่น DB2, Oracle และ MS SQL-Server

Bachman C.W. 1965. "Integrated Data Store", DPMA Quarterly, (January 1965). Bachman C.W. 1973. "The programmer as Navigator", Communications of the ACM, (November 1973).

ถึงแม้ว่าแบบจำลองข้อมูลเชิงสัมพันธ์จะถูกพัฒนาเพื่อปรับปรุงการทำงานของแบบจำลองแบบลำดับ ชั้นและแบบเครือข่าย แต่แบบจำลองข้อมูลเชิงสัมพันธ์ยังขาดเค้าโครงที่จะทำให้การสร้างเครื่องมือสำหรับออกแบบฐานข้อมูลได้อย่างมีประสิทธิภาพ การแสดงโครงสร้างแบบจำลองด้วยแผนภาพจะทำให้ผู้ออกแบบและผู้ใช้งานฐานข้อมูลเข้าใจแบบจำลองข้อมูลได้ง่ายกว่าการใช้ตัวอักษร ด้วยสาเหตุนี้เอง จึงได้มีการคิดค้น และพัฒนาแบบจำลองความสัมพันธ์เอนทิตี (entity relationship model, ERM) ขึ้นมาในการสร้างแบบจำลองข้อมูลสำหรับการออกแบบฐานข้อมูล

ในปี 1976 โดย Peter Chen ได้พัฒนาแบบจำลองความสัมพันธ์เอนทิตี ขึ้นมาในปี ค.ศ 1976 แบบจำลองนี้ประกอบไปด้วย 2 ส่วน คือ

- *เอนทิตี* – ส่วนที่ใช้จัดเก็บข้อมูล แต่ละเอนทิตีจะสามารถอธิบายได้ด้วยเซตของแอททริบิวท์ที่บ่งบอกถึงคุณลักษณะของเอนทิตินั้น ๆ
- *ความสัมพันธ์* – สิ่งที่ใช้อธิบายเกี่ยวกับความสัมพันธ์หรือการเชื่อมโยงกันของข้อมูล ความสัมพันธ์ส่วนใหญ่จะอธิบายถึงความสัมพันธ์ของเอนทิตีทั้งแบบ 1:1, 1:M และ M:M

**รุ่นที่ 4 (Fourth generation) แบบจำลองข้อมูลเชิงวัตถุ (Object-Oriented database model)** อยู่ในช่วงกลางปี ค.ศ 1980s – ปัจจุบัน

แบบจำลองข้อมูลเชิงวัตถุ เป็นแบบจำลองข้อมูลที่พัฒนาต่อจากแบบจำลองข้อมูลเชิงสัมพันธ์ มีแนวคิดมาจากการเขียนโปรแกรมเชิงวัตถุที่มองของทุกสิ่งเป็นวัตถุ (object) วัตถุนี้จะมีลักษณะคล้ายๆกับเอนทิตี ในแบบจำลองข้อมูลอื่นที่กล่าวมาแล้ว) แต่วัตถุหนึ่ง ๆ จะใช้ในการแสดงถึงการเกิดขึ้นครั้งหนึ่ง ๆ ของข้อมูลในเอนทิตีเท่านั้น โดยที่วัตถุแต่ละวัตถุ จะเป็นแหล่งรวมของข้อมูลและการปฏิบัติงาน มีคลาส (class) หรือกลุ่มของวัตถุที่มีความเหมือนกันในแอททริบิวท์เป็นตัวกำหนดคุณสมบัติของวัตถุ รวมทั้งคุณสมบัติการปกปิดความลับของวัตถุ สามารถจัดเก็บข้อมูลและเรียกดูข้อมูลได้อย่างสะดวกและรวดเร็ว ข้อดีของแบบจำลองนี้คือสามารถจัดการกับข้อมูลที่มีความซับซ้อนเช่น กราฟิก (Graphic) วิดีโอ (VDO) หรือเสียง (Voice) ได้เป็นอย่างดี และที่สำคัญคือสนับสนุนคุณสมบัติของการนำกลับมาใช้ใหม่ (Re useable) แต่แบบจำลองเชิงวัตถุนี้ มีค่าใช้จ่ายสูงเพราะต้องอาศัยบุคลากรที่มีความรู้และประสบการณ์มากในการจัดการกับฐานข้อมูลชนิดนี้ และยังไม่มามีมาตรฐานรองรับ ตัวอย่างเช่น VFS/FastObjects, Objectivity/DB, DB2 UDB และ Oracle 10g



**รุ่นอนาคต (Next generation)** แบบจำลองข้อมูล XML อยู่ในช่วงปัจจุบัน – อนาคต ตัวอย่างเช่น dbXML, Tamino , DB/2 UDB , Oracle 10g และ MS SQL-Server ใช้สำหรับ XML documents (<https://www.slideshare.net/devgocool/dbms-models>)

จะเห็นได้ว่าวิวัฒนาการของระบบจัดการฐานข้อมูลยังคงมีอยู่อย่างต่อเนื่อง เนื่องจากความต้องการที่จะมีวิธีการสร้างแบบจำลองใหม่ ๆ ที่สามารถเพิ่มการรองรับข้อมูลที่มีความสลับซับซ้อน และมีเป็นจำนวนมาก แต่อย่างไรก็ตามแบบจำลองข้อมูลไม่ว่าจะเป็นแบบใดก็ตาม ก็จะต้องมีคุณลักษณะบางอย่างที่เหมือนกันหรือคล้ายกัน อาจจะได้ว่าเป็นคุณลักษณะพื้นฐานสำคัญของแบบจำลองข้อมูล ดังนี้

- แบบจำลองข้อมูลจะต้องสามารถแสดงถึงระดับของความง่ายของกรอบความคิดโดยไม่สูญเสียความครบถ้วนของความหมายของฐานข้อมูล
- แบบจำลองข้อมูลจะต้องสามารถให้กรอบความคิดที่ใกล้เคียงความจริงมากที่สุดเท่าที่จะเป็นไปได้
- การเปลี่ยนแปลง หรือ เปลี่ยนรูปข้อมูลจะต้องเป็นไปตามกฎความสอดคล้องและความสมบูรณ์ของข้อมูล

ปัจจุบันการออกแบบจำลองข้อมูล จะมุ่งเน้นที่การแก้ไขจุดอ่อน ของแบบจำลองก่อนหน้านี้ เช่นแบบจำลองข้อมูลแบบเครือข่าย ได้พัฒนาขึ้นมาเพื่อแทนที่แบบจำลองข้อมูลแบบลำดับชั้น โดยที่แบบจำลองข้อมูลแบบเครือข่าย จะสามารถปรับเปลี่ยนความสัมพันธ์ที่ซับซ้อนให้มีความง่ายมากขึ้น แบบจำลองข้อมูลเชิงสัมพันธ์ จะมีประโยชน์มากกว่าแบบจำลองแบบเครือข่ายในหลาย ๆ ด้าน เช่น การแสดง ข้อมูลด้วยวิธีการที่ง่าย ความไม่สอดคล้องของข้อมูลมีน้อยกว่า และการประยุกต์ใช้ภาษาคิวรีในการเข้าถึงข้อมูล และ แบบจำลองข้อมูลเชิงวัตถุได้พัฒนาต่อจากแบบจำลองข้อมูลเชิงสัมพันธ์ โดยการเพิ่มเติมแนวความคิดเชิงวัตถุ ที่จะช่วยให้สามารถจัดการกับข้อมูลที่มีความซับซ้อนได้

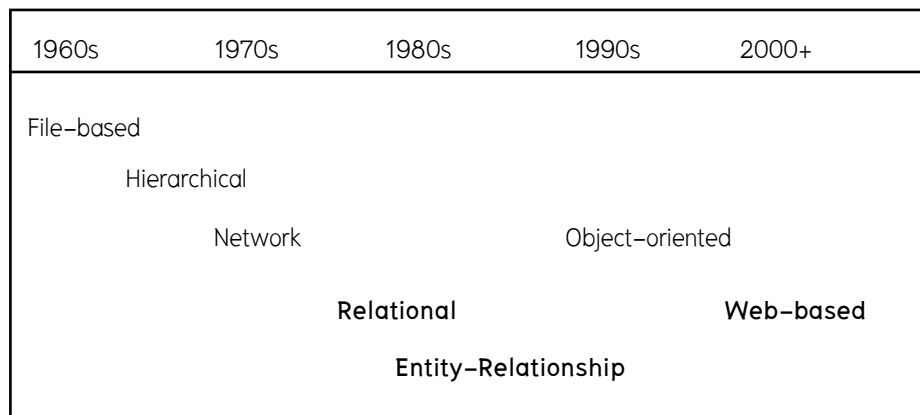
อาจจะกล่าวสรุปได้ว่า แบบจำลองข้อมูลเป็นเครื่องมือที่ใช้อธิบายโครงสร้าง และความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่อยู่ในฐานข้อมูลจากรูปแบบที่เป็นแนวความคิดซึ่งเข้าใจได้ยาก ให้อยู่ในรูปแบบที่เข้าใจได้ง่ายขึ้น และสะดวกในการใช้ เราอาจจะแบ่งแบบจำลองข้อมูลออกได้เป็น 2 ประเภทคือ

- แบบจำลองข้อมูลเชิงแนวคิด (Conceptual Data Models) เป็นแบบจำลองที่ประกอบไปด้วยสัญลักษณ์ที่ใช้อธิบายแทนตัวข้อมูล คุณสมบัติของข้อมูล และความสัมพันธ์ของข้อมูล ความสัมพันธ์ต่าง ๆ ใช้อธิบายภาพหรือลักษณะโดยรวมของข้อมูลทั้งหมดในระบบโดย

แสดงในรูปของแผนภาพ (diagram) ซึ่งประกอบด้วยเอนทิตีต่าง ๆ ความสัมพันธ์ระหว่างเอนทิตีของแบบจำลองเชิงแนวคิดนี้ ต้องการนำเสนอให้เกิดความเข้าใจระหว่างผู้ออกแบบและผู้ใช้งาน ผู้ที่สามารถเข้าไปแก้ไขโครงสร้างข้อมูลนี้ได้ คือผู้บริหารฐานข้อมูล หรือโปรแกรมเมอร์

แบบจำลองข้อมูลเชิงแนวคิดมักถูกนำไปใช้ในการออกแบบฐานข้อมูล เพื่อต้องการอธิบายให้เห็นว่าภายในฐานข้อมูลจะประกอบด้วยข้อมูลอะไรบ้าง ? และแต่ละข้อมูลนั้นมีความสัมพันธ์กันอย่างไร ? ตัวอย่างแบบจำลองข้อมูลประเภทนี้ เช่น แบบจำลองข้อมูล E-R (Entity-Relationship Model) และแบบจำลองข้อมูล เชิงวัตถุ (Object-oriented Model) เป็นต้น

ช่วงเวลา (Timeline)



Y2k						
1990						
1980						
1970						
1960						
1950						
Pre-1950						
	File system	Hierarchical	Network	Relational	Object	Object-Relational

ภาพที่ 3.3 แสดงวิวัฒนาการของแบบจำลองข้อมูล

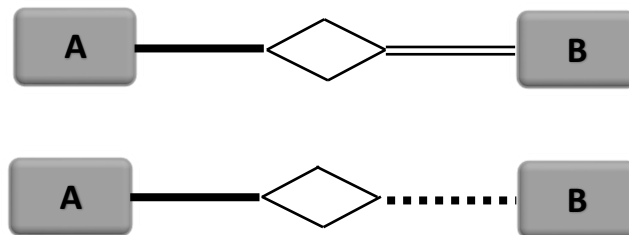
(ที่มา : <https://www.pinterest.com/pin/191825265356325311/>)

- **แบบจำลองข้อมูลแบบนำไปใช้ (Implementation Data Models)** เป็นแบบจำลองข้อมูลที่อธิบายถึงโครงสร้างข้อมูลของฐานข้อมูล ที่แสดงถึงรูปแบบที่อิง กับระบบการจัดการฐานข้อมูลที่ใช้เลือกใช้งาน และเป็นแบบจำลองที่ถูกนำมาใช้ในขั้นตอนการสร้างฐานข้อมูล ตัวอย่าง

แบบจำลองข้อมูลประเภทนี้ เช่น แบบจำลองข้อมูลแบบลำดับชั้น (Hierarchical Model) แบบจำลองข้อมูลแบบเครือข่าย (Network Model) และแบบจำลองข้อมูลเชิงสัมพันธ์ (Relational Model) เป็นต้น

### 3. ความสัมพันธ์ (Relationships)

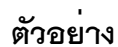
ความสัมพันธ์ในที่นี้จะหมายถึง ความสัมพันธ์ระหว่างเอนทิตีต่าง ๆ ซึ่งจะแสดงถึงความสัมพันธ์ระหว่างเอนทิตีหรือระหว่างข้อมูลที่เราสงสัย ซึ่งพิจารณาได้ 2 แบบ คือความสัมพันธ์แบบจำเป็น หรือแบบบังคับ (mandatory) หมายถึงความสัมพันธ์ที่**จะต้องมี** หรือ **จะต้องเกิดขึ้น** อย่างแน่นอน นิยมแทนด้วยสัญลักษณ์**เส้นตรงทึบ** หรือ**เส้น ตรงสองเส้น** และความสัมพันธ์แบบไม่จำเป็น (optional) เป็นความสัมพันธ์ที่**อาจจะมีหรือไม่มีก็ได้** นิยมแทนด้วยสัญลักษณ์**เส้นตรงประ**หรือ**เส้นตรงหนึ่งเส้น** ดังภาพข้างล่าง



จากภาพอธิบายได้ดังนี้ : ถ้าข้อมูล A มีความสัมพันธ์กับข้อมูล B **อย่างแน่นอน** หรืออย่างน้อยจะต้องมีสมาชิก (element) ของข้อมูล A หนึ่งสมาชิกที่มีความสัมพันธ์กับสมาชิกของข้อมูล B และแทนความสัมพันธ์ด้วย**เส้นตรงทึบ**ออกจากข้อมูล A ไปยังข้อมูล B แต่ในขณะเดียวกันข้อมูล B อาจจะไม่มีความสัมพันธ์กับข้อมูล A หรือไม่มีก็ได้ เส้นตรงที่ออกจากข้อมูล B จึงแทนด้วยเส้นประ

ถ้ากำหนดให้ A และ B เป็นข้อมูลหรือสิ่งที่สนใจ (object of interest) สัญลักษณ์ที่ใช้แทนชนิดความสัมพันธ์ระหว่างข้อมูล A กับข้อมูล B คือ **เส้นตรงเส้นเดียวหรือจะใช้ลูกศร 1 หัว กับ 2 หัว** สามารถที่จะแสดงชนิดความสัมพันธ์ระหว่างข้อมูล A กับข้อมูล B ได้ดังนี้

3.1 **ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One to One Relationships, 1 : 1)** เป็นการแสดงความสัมพันธ์ของข้อมูลของเอนทิตีหนึ่ง ว่ามีความสัมพันธ์กับข้อมูลในอีกเอนทิตีหนึ่ง ในลักษณะ**หนึ่งต่อหนึ่ง** คือข้อมูลหนึ่งข้อมูลของเอนทิตีหนึ่งมีความสัมพันธ์กับข้อมูลในเอนทิตีที่สองเพียงข้อมูลเดียวเท่านั้น เช่นข้อมูล A จะเกี่ยวข้องกับสัมพันธ์กับข้อมูล B ได้เพียง 1 ค่าเท่านั้น และ ข้อมูล B จะเกี่ยวข้องกับข้อมูล A ได้เพียง 1 ค่าเช่นกัน



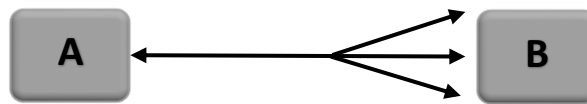
- 
- ```

graph LR
    A[นิสิต] <-->|1 : 1| B[รหัสนิสิต]
    C[คณบดี] <-->|1 : 1| D[คณะวิชา]
  
```

ภาพที่ 3.4 แสดงความสัมพันธ์แบบหนึ่งต่อหนึ่ง หรือ 1 : 1

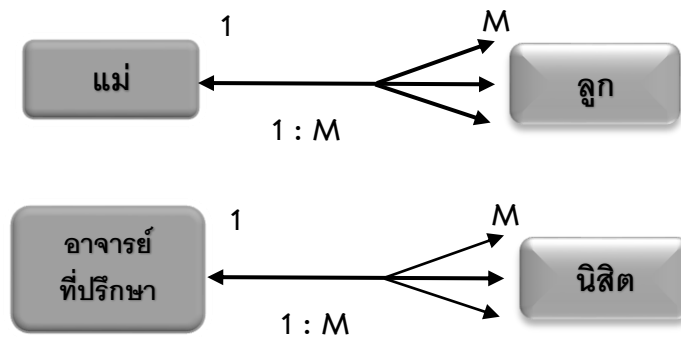
3.2 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One to Many Relationships , 1 : N หรือ 1 : M ) เป็นความสัมพันธ์ระหว่างเอนทิตีหนึ่ง ไปมีความสัมพันธ์กับอีกเอนทิตีหนึ่งได้มากกว่าหนึ่งรายการ ในลักษณะหนึ่งต่อกลุ่ม (ข้อมูลหนึ่งข้อมูลของเอนทิตีที่หนึ่งมีความสัมพันธ์กับข้อมูลในเอนทิตีที่สองหลายข้อมูล) เช่นข้อมูล “A” มีความสัมพันธ์กับข้อมูล “B” ได้มากกว่าหนึ่งรายการ เช่น อาจารย์ที่ปรึกษา 1 คน สามารถเป็นที่ปรึกษาให้กับนิสิตได้หลายคน หรือคณะวิชาหนึ่งคณะวิชามีนิสิตได้หลายคน เป็นต้น

ความสัมพันธ์แบบนี้ข้อมูล A จะเกี่ยวข้องกับข้อมูล B ได้มากกว่า 1 ค่า แต่ข้อมูล B จะเกี่ยวข้องกับข้อมูล A ได้อย่างมากที่สุดเพียง 1 ค่าเท่านั้น



### ตัวอย่าง

- ความสัมพันธ์ระหว่างแม่กับลูก : แม่หนึ่งคนอาจจะมีลูกได้หลายคน แต่ลูกหนึ่งคนจะมีแม่ได้เพียงคนเดียวเท่านั้น
- ความสัมพันธ์ระหว่างอาจารย์ที่ปรึกษากับนิสิต : อาจารย์ที่ปรึกษาหนึ่งคนจะมีนิสิตได้หลายคน แต่นิสิตคนหนึ่ง จะมีอาจารย์ที่ปรึกษาได้เพียง 1 คนเท่านั้น

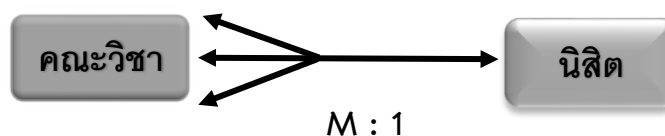


ภาพที่ 3.5 แสดงความสัมพันธ์แบบหนึ่งต่อกลุ่ม หรือ 1 : M

3.3 ความสัมพันธ์แบบกลุ่มต่อหนึ่ง (Many to One Relationships, M : 1) ความสัมพันธ์แบบกลุ่มต่อหนึ่ง เช่น ข้อมูล A จะเกี่ยวข้องกับข้อมูล B ได้เพียง 1 ค่า และข้อมูล B จะเกี่ยวข้องกับข้อมูล A ได้หลายค่า โดยปกติแล้วความสัมพันธ์แบบ M : 1 หรือ 1 : M มักจะรวมไว้เป็นแบบเดียวกัน คือเป็นความสัมพันธ์ระหว่างหนึ่งต่อกลุ่ม

### ตัวอย่าง

- ความสัมพันธ์ระหว่างนิสิตกับคณะวิชาที่สังกัด : สถาบันการศึกษาหนึ่งสามารถมีคณะวิชาได้มากกว่าหนึ่งคณะวิชา แต่นิสิตคนหนึ่ง จะสังกัดคณะวิชาได้เพียงคณะวิชาเดียวเท่านั้น

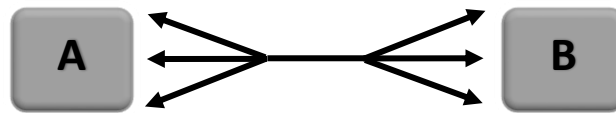


ภาพที่ 3.6 แสดงความสัมพันธ์แบบกลุ่มต่อหนึ่ง หรือ M : 1

### 3.4 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many to Many Relationships , M : N)

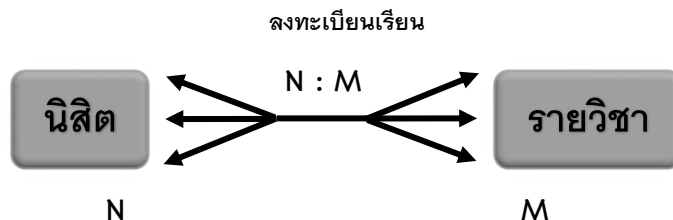
เป็นความสัมพันธ์แบบหลายรายการระหว่างสองเอนทิตี เป็นความสัมพันธ์ของข้อมูลของเอนทิตีหนึ่งที่มีความสัมพันธ์กับข้อมูลในอีกเอนทิตีหนึ่ง ในลักษณะกลุ่มต่อกลุ่ม เช่น ข้อมูล “A” มีความสัมพันธ์กับอีกข้อมูล “B” ได้มากกว่าหนึ่งรายการ ในขณะที่เดียวกันแต่ละรายการของข้อมูล “B” ก็มีความสัมพันธ์กับข้อมูล “A” ได้มากกว่าหนึ่งรายการเช่นเดียวกัน ตัวอย่างเช่น นิสิต 1 คนสามารถลงทะเบียนเรียนได้มากกว่า 1 รายวิชา และรายวิชา 1 รายวิชาจะมีนิสิตลงทะเบียนเรียนได้มากกว่า 1 คน เป็นต้น

ความสัมพันธ์แบบกลุ่มต่อกลุ่ม นี้ ข้อมูล A จะเกี่ยวข้องกับข้อมูล B ได้หลายค่า และในทำนองเดียวกันข้อมูล B เกี่ยวข้องกับข้อมูล A ได้มากกว่า 1 ค่าเช่นกัน



ตัวอย่าง :

- ความสัมพันธ์ระหว่างนิสิตกับรายวิชาที่เปิดสอน : นิสิต 1 คนสามารถลงทะเบียนเรียนได้หลายวิชา และวิชาหนึ่ง ๆ จะมีนิสิตลงทะเบียนเรียนได้มากกว่า 1 คน
- ความสัมพันธ์ระหว่างตารางสินค้าและตารางใบสั่งซื้อ : ใบสั่งซื้อใบเดียวสามารถที่จะมีรายการผลิตภัณฑ์ได้มากกว่าหนึ่งรายการ อย่างไรก็ตามสินค้ารายการเดียวสามารถที่จะปรากฏในใบสั่งซื้อหลายใบได้



ภาพที่ 3.7 แสดงความสัมพันธ์แบบกลุ่มต่อกลุ่ม หรือ M : N

#### 4. สถาปัตยกรรมฐานข้อมูล (Database architecture)

เมื่อเปรียบเทียบระหว่างฐานข้อมูลกับแฟ้มข้อมูล จะพบว่าฐานข้อมูลมีข้อดีกว่าแฟ้มข้อมูลทั่วไป โดยเฉพาะในเรื่องของ*ความเป็นอิสระข้อมูล* เนื่องจากระบบฐานข้อมูล โปรแกรมประยุกต์ไม่ได้ขึ้นอยู่กับโครงสร้างของข้อมูลที่มีการเปลี่ยนแปลงไป เนื่องจากการจัดโครงสร้างของระบบฐานข้อมูล ด้วยเหตุนี้สถาบันมาตรฐานแห่งชาติของสหรัฐอเมริกา หรือ ANSI (American National Standards Institute) ซึ่งเป็นองค์กรสำคัญที่ให้การสนับสนุนการพัฒนามาตรฐานทางเทคโนโลยีของสหรัฐอเมริกา จึงได้มีการกำหนดมาตรฐานขึ้น ในปี 1975 เรียกว่า*สถาปัตยกรรมสามระดับ (Three-level Architecture)* เป็นการอธิบายรูปแบบ และรายละเอียดโครงสร้างที่ใช้กับระบบของฐานข้อมูลโดยรวมในระดับต่าง ๆ ซึ่งจะไม่เกี่ยวข้องกับข้อมูลที่เก็บในฐานข้อมูลแต่ประการใด

สถาปัตยกรรมระบบฐานข้อมูลแบ่งเป็น 3 ระดับ คือ ระดับภายนอก ระดับแนวคิด และระดับภายใน สถาปัตยกรรมฐานข้อมูลในแต่ละระดับจะมี DBMS ทำหน้าที่แปลงข้อมูล (Mapping) จากระดับหนึ่งไปสู่ระดับหนึ่ง ซึ่งจะก่อให้เกิดประโยชน์หลายด้าน เช่น ถ้ามีการเปลี่ยนแปลงวิธีการจัดเก็บข้อมูลในระดับกายภาพ ก็ไม่มีผลกระทบต่อการออกแบบฐานข้อมูลในระดับแนวคิดแต่อย่างใด

##### 4.1 สถาปัตยกรรมระดับภายใน (Internal Level หรือ Physical Level)

สถาปัตยกรรมระดับภายใน เป็นสถาปัตยกรรมระดับล่างสุด เป็นระดับของการจัดเก็บฐานข้อมูลในหน่วยเก็บข้อมูลสำรองจริง มีการจัดเก็บข้อมูลด้วยหน่วยความจำแบบใด? จัดเก็บข้อมูลอย่างไร? มีการดูแลรักษาอย่างไร? มีโครงสร้างข้อมูลเป็นแบบใด? เพราะโครงสร้างข้อมูลแต่ละแบบ จะมีผลต่อประสิทธิภาพของความเร็วในการเข้าถึง (access) ข้อมูล สถาปัตยกรรมระดับภายใน จะทำงานเชื่อมต่อกับระบบการจัดการข้อมูลเกี่ยวกับวิธีการจัดเก็บและวิธีการเข้าถึงข้อมูล โดยมีระบบการจัดการฐานข้อมูลเป็นตัวช่วยในการกำหนดวิธีการเข้าถึงข้อมูล การค้นหา การปรับปรุง การเชื่อมโยงระหว่างตาราง รวมทั้งการบำรุงรักษาฐานข้อมูล ผู้ที่เกี่ยวข้องส่วนใหญ่คือ ผู้ออกแบบ ผู้ดูแลระบบ ผู้เขียนโปรแกรมประยุกต์ และผู้บริหารฐานข้อมูล ผู้ใช้ทั่วไปไม่มีสิทธิ์เข้ามายุ่งในระดับนี้

##### 4.2 สถาปัตยกรรมระดับแนวคิด (Conceptual Level)

สถาปัตยกรรมระดับแนวคิดเป็นมุมมองโครงร่างของฐานข้อมูลในแนวความคิด ซึ่งไม่ใช่โครงร่างจริงที่ถูกสร้างในอุปกรณ์เก็บข้อมูลผ่านการวิเคราะห์จากนักออกแบบระบบ และผู้บริหารฐานข้อมูล สถาปัตยกรรมในระดับแนวคิดจึงเป็นระดับของการออกแบบฐานข้อมูลว่าจะเก็บข้อมูลอะไร ฐานข้อมูลที่สร้างประกอบด้วยเอนทิตีอะไรบ้าง? แต่ละเอนทิตี ประกอบด้วย*เซตข้อมูลหรือ*

แอททริบิวต์ ไต่บ้าง? และความสัมพันธ์แต่ละเอนทิตีเป็นอย่างไร ? ผลลัพธ์ที่ได้ทำให้เกิด สกีมา (Schema) หรือ **โครงสร้างข้อมูล** อาจกล่าวได้ว่า โครงสร้างข้อมูล ระดับเชิงแนวคิดจะเป็นตัวที่ใช้เชื่อมระหว่างโครงสร้างข้อมูล ระดับภายนอกกับโครงสร้างข้อมูลระดับภายใน เช่น ตารางข้อมูล ของนักศึกษา ดังนี้

นิสิต

| รหัส      | ชื่อ-สกุล | สาขา        | คณะวิชา     | ภูมิลำเนา |
|-----------|-----------|-------------|-------------|-----------|
| 256001155 | มานะ อดทน | คอมพิวเตอร์ | วิทยาศาสตร์ | พะเยา     |

โครงสร้างข้อมูลของนิสิตคือ

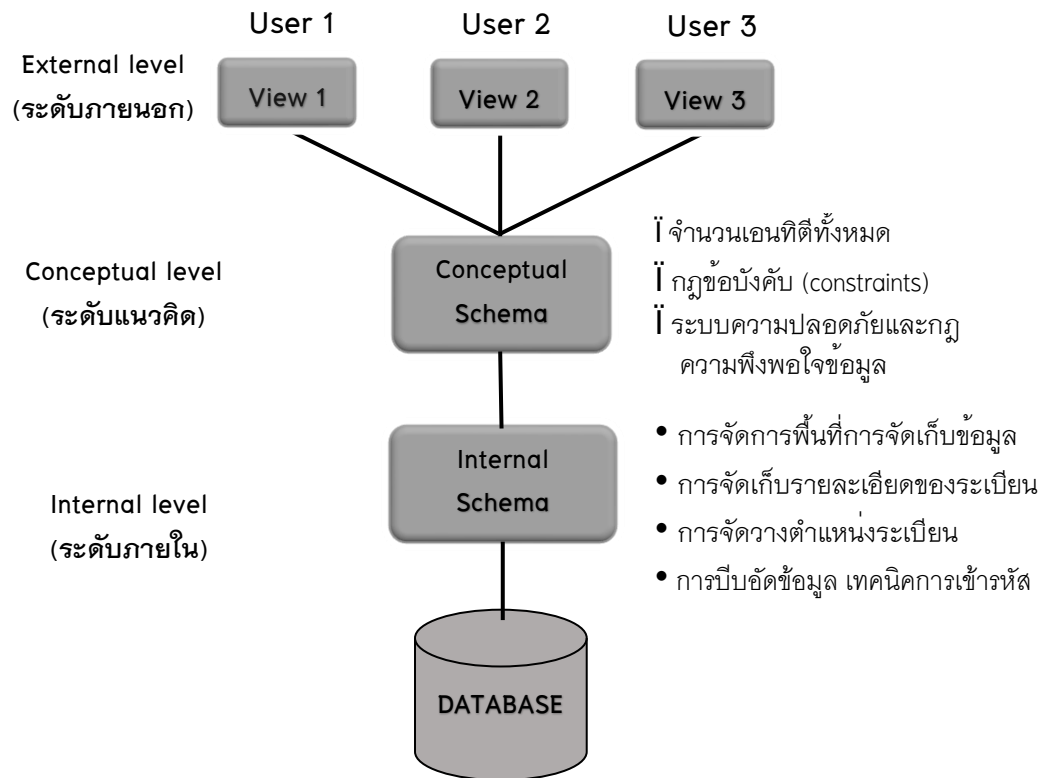
| รหัส | ชื่อ-สกุล | สาขา | คณะวิชา | ภูมิลำเนา |
|------|-----------|------|---------|-----------|
|------|-----------|------|---------|-----------|

#### 4.3 สถาปัตยกรรมระดับภายนอก (External Level หรือ View)

สถาปัตยกรรมระดับภายนอก เป็นระดับบนสุด มีการรวบรวมข้อมูลอย่างง่าย ๆ จากผู้ใช้ให้กับนักวิเคราะห์นำไปศึกษา หรือเป็นมุมมองของผู้ใช้ที่มีต่อฐานข้อมูล ที่เรียกว่า วิว (View) ซึ่งจะเกี่ยวข้องกับผู้ใช้จำนวนมากที่สุด เป็นการมองข้อมูลภายในฐานข้อมูลสำหรับผู้ใช้งานแต่ละคน เป็นระดับที่มีการนำข้อมูลจากฐานข้อมูลไปใช้งาน ผู้ใช้แต่ละคนสามารถเลือกอ่านข้อมูลเฉพาะที่ตนสนใจหรือต้องการใช้เท่านั้น

วัตถุประสงค์หลักของสถาปัตยกรรมทั้งสามระดับ คือ ต้องการให้ข้อมูล มีความเป็นอิสระจาก ซอฟต์แวร์ ทั้งของผู้ใช้ และของระบบ หรืออาจจะกล่าวได้ว่าการเปลี่ยนแปลงใด ๆ ต่อทั้ง ฮาร์ดแวร์ และ ซอฟต์แวร์ จะไม่มีผลต่อข้อมูลนั้น ๆ ดังนั้นสถาปัตยกรรมทั้งสามระดับที่กล่าวมานี้ จะมีความเป็นอิสระของข้อมูล คือ โครงสร้างของข้อมูลระดับภายนอก จะมีรูปแบบที่คงที่ไม่มีการเปลี่ยนแปลงไป แม้จะมีการเปลี่ยนโครงสร้างข้อมูลในระดับแนวคิดก็ตาม เช่น มีการเพิ่มรูปแบบข้อมูลใหม่ หรือมีความสัมพันธ์ใหม่ ๆ เพิ่มขึ้น เรียกว่า **ความเป็นอิสระทางตรรกะ (Logical Data Independence)** และโครงสร้างข้อมูลระดับแนวคิดก็เช่นกัน จะมีรูปแบบที่คงที่แม้จะมีการเปลี่ยนแปลงทางกายภาพ เช่น มีการเปลี่ยนวิธีเข้าถึงข้อมูล หรือลำดับของข้อมูลที่เกิดขึ้นจริงเปลี่ยนไป เรียกว่า **ความเป็นอิสระทางกายภาพ (Physical Data Independence)**





ภาพที่ 3.8 แสดงสถาปัตยกรรมฐานข้อมูล 3 ระดับ

## 5. คุณลักษณะของแบบจำลองข้อมูลที่ดี

คุณลักษณะหรือคุณสมบัติของแบบจำลองข้อมูลที่ดี ควรมีลักษณะดังนี้

5.1 มีความเข้าใจได้ง่าย แบบจำลองข้อมูลที่ดีควรที่จะต้องใช้กฎเกณฑ์ทั่ว ๆ ไป จะทำให้เข้าใจได้ง่าย โดยมีข้อมูลแอมพลิฟิเคชันที่อธิบายในรายละเอียดของแต่ละเอนทิตี

5.2 มีสาระสำคัญและไม่ซ้ำซ้อน แอมพลิฟิเคชันในแต่ละเอนทิตี จะต้องมีความสำคัญ ไม่ควรมีข้อมูลที่ซ้ำซ้อนกัน

5.3 มีความยืดหยุ่นและง่ายต่อการปรับปรุง แบบจำลองข้อมูลที่ดีต้องมีความยืดหยุ่นและง่ายต่อการปรับปรุง หรือการพัฒนาในอนาคต และไม่ควรรู้ขึ้นอยู่กับแอปพลิเคชันโปรแกรม

5.4 ความเป็นอิสระของข้อมูล คือผู้ใช้สามารถเปลี่ยนแปลงข้อมูลในระดับแนวคิด (เชิงตรรกะ) หรือระดับภายใน (เชิงกายภาพ) ได้ โดยที่ไม่กระทบกับโปรแกรม และผู้ใช้งานยังสามารถใช้งานได้ และมองเห็นโครงสร้างข้อมูลในระดับภายนอกได้เหมือนเดิม

## 6. ประเภทของแบบจำลองข้อมูล (Types of Data Models)

### 6.1 แบบจำลองข้อมูลลำดับชั้น (Hierarchical Data Model)

แบบจำลองข้อมูลลำดับชั้น หรือบางครั้งเรียกว่าแบบจำลองข้อมูลโครงสร้างต้นไม้

(Tree-based Data Model) เป็นแบบจำลองข้อมูลที่พัฒนามาจากแนวความคิดในการจัดเก็บข้อมูล เพื่อลดการซ้ำซ้อนของข้อมูล (Data Redundancy) เป็นแบบจำลองต้นแบบสำหรับการแก้ไขปัญหาการเก็บข้อมูลด้วยระบบแฟ้มข้อมูล โดยนำเอาข้อมูลในแต่ละส่วน หรือแต่ละ Part มาจัดเก็บเป็นกลุ่มหรือเป็น Component แล้วรวมแต่ละกลุ่มเป็นกลุ่มใหญ่เรียกว่า Final Component โดยมีโครงสร้างอยู่ในรูปแบบของต้นไม้ที่มีลักษณะคล้ายต้นไม้ที่คว่ำหัวลงเป็นลำดับชั้น (Upside-down Tree) แตกออกเป็นกิ่งก้าน สาขา เป็นการจัดเก็บข้อมูลในลักษณะความสัมพันธ์แบบ พ่อแม่-ลูก (Parent-Child Relationship Type หรือ PCR)

โครงสร้างข้อมูลพื้นฐานของแบบจำลองข้อมูลลำดับชั้น ประกอบด้วยระเบียน (Record) หรือ *กลุ่มค่า* ของข้อมูลของเอนทิตีหนึ่ง ๆ หรือของความสัมพันธ์ระหว่างเอนทิตี ระเบียนข้อมูลชนิดเดียวกันเมื่อนำมารวมกลุ่มกัน เรียกว่าชนิดระเบียน (Record Type) (เอนทิตี หมายถึงวัตถุใด ๆ ทั้งที่เป็น นามธรรมและรูปธรรมที่บรรจุไว้เป็นข้อมูลประกอบด้วยคุณสมบัติ หรือแอททริบิวต์ต่าง ๆ เช่น ฐาน ข้อมูลของคณะวิชาอาจจะประกอบด้วยเอนทิตี อาจารย์ นิสิต เจ้าหน้าที่ งบประมาณ ภาควิชา และ หลักสูตร เป็นต้น เอนทิตีอาจารย์ อาจจะประกอบไปด้วย ชื่อ-สกุล วันเดือนปีเกิด ตำแหน่งวิชาการ คุณวุฒิการศึกษา เลขประจำตัวบัตรประชาชน ที่อยู่ เบอร์โทรศัพท์ เป็นต้น ข้อมูลของหน่วยงานเช่น มหาวิทยาลัย คณะวิชา ภาควิชา จะเหมาะสมกับแบบ จําลองข้อมูลเชิงลำดับชั้นมากที่สุด

แบบจำลองข้อมูลลำดับชั้น มีลักษณะดังนี้

- ข้อมูลทั้งหมดจะอยู่ในฐานข้อมูลร่วมกัน ข้อมูลจะมีความสัมพันธ์แบบ *หนึ่งต่อกลุ่ม*
- มีความสัมพันธ์แบบพ่อแม่-ลูก
- ระเบียนที่อยู่ด้านบน หรือระดับสูงสุดเรียกว่า ระเบียนพ่อแม่ (parent record) ส่วนระเบียนในแถวถัดลงมาเรียกว่า ระเบียนลูก (child record)
- ระเบียนพ่อแม่จะสามารถมีระเบียนลูกได้มากกว่าหนึ่งระเบียน แต่ระเบียนลูกแต่ละระเบียนสามารถมีระเบียนพ่อแม่ได้เพียงหนึ่งระเบียนเท่านั้น
- เป็นการจัดโครงสร้างแบบบนลงล่าง (Top-down) มีลักษณะคล้ายโครงสร้างต้นไม้
- เป็นสถาปัตยกรรมฐานข้อมูลที่เก่าแก่ที่สุด ทำให้ยากต่อการพัฒนาแอปพลิเคชัน (application) เพื่อใช้งาน

- โครงสร้างมีความยืดหยุ่นน้อย
- ไม่สามารถกำหนดความสัมพันธ์แบบ *กลุ่มต่อกลุ่ม* ได้

*ข้อดีแบบจำลองข้อมูลลำดับชั้น*

- เป็นโครงสร้างที่เข้าใจง่าย มีประสิทธิภาพในการค้นหา
- ทำให้การค้นหาทำได้รวดเร็วไม่เสียเวลาในการค้นหาข้อมูลที่ไม่จำเป็นเพราะมีการจัดลำดับชั้นและความสัมพันธ์เอาไว้แล้ว
- รูปแบบโครงสร้างมีความซับซ้อนน้อย ทำให้เข้าใจง่ายเหมาะกับข้อมูลที่มีความสัมพันธ์ แบบ *หนึ่งต่อกลุ่ม*
- มีโครงสร้างที่เหมาะสมกับระบบคอมพิวเตอร์ที่มีขนาดใหญ่
- สามารถกำหนดกฎเกณฑ์ที่ใช้ควบคุมความถูกต้องให้กับข้อมูลในฐานข้อมูลได้ง่าย
- การรักษาความปลอดภัยสามารถควบคุมได้โดยระบบการจัดการฐานข้อมูล มีความปลอดภัยของข้อมูล เพราะต้องอ่านข้อมูลที่เป็นต้นกำเนิดก่อน
- ระบบการจัดการฐานข้อมูลสามารถสร้างสถานะความเป็นอิสระของข้อมูล ทำให้การแก้ไข และการบำรุงรักษา โปรแกรมที่เขียนขึ้นไม่มีผลต่อข้อมูลแต่ประการใด
- เหมาะกับข้อมูลที่มีการเรียงลำดับแบบต่อเนื่อง
- มีค่าใช้จ่ายในการจัดสร้างฐานข้อมูลน้อย

*ข้อจำกัดแบบจำลองข้อมูลลำดับชั้น*

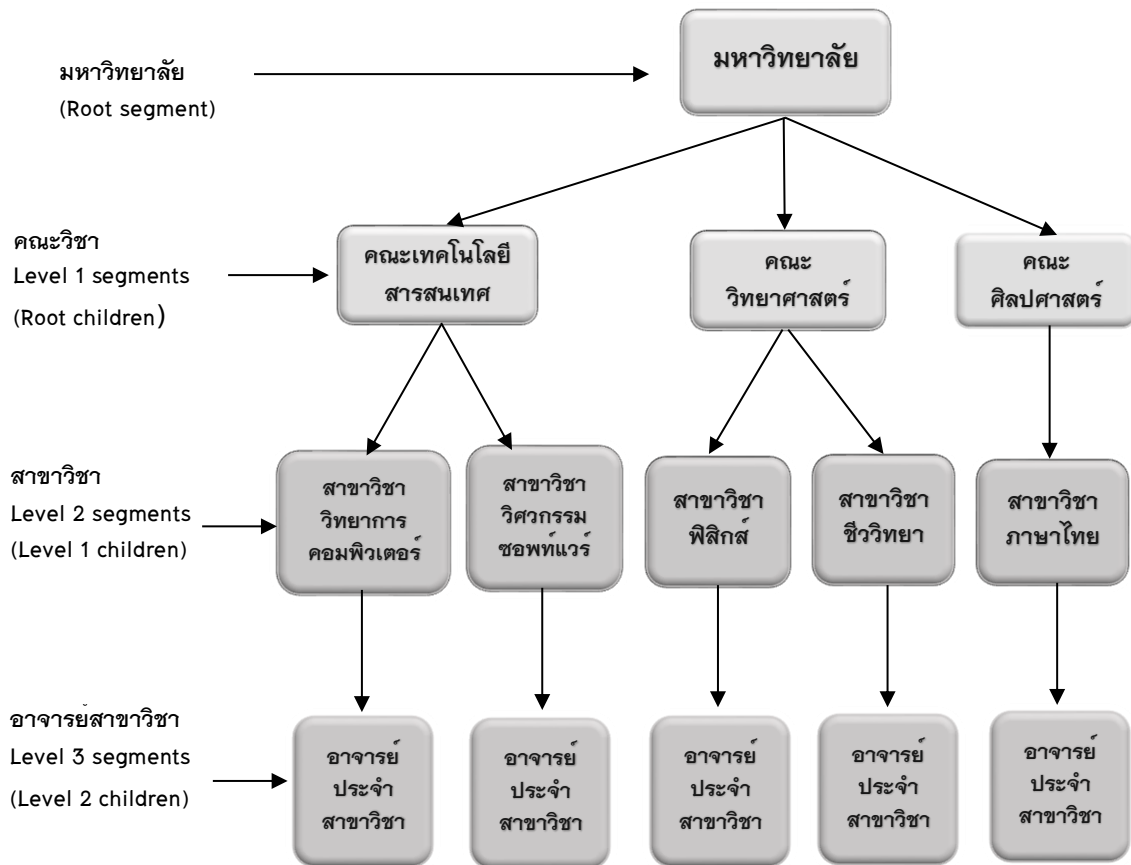
ถึงแม้ว่าระบบการจัดการฐานข้อมูลแบบจำลองฐานข้อมูลแบบลำดับชั้น จะช่วยแบ่งเบาในปัญหาเรื่องความเป็นอิสระของข้อมูล แต่ระบบจัดการฐานข้อมูลยังคงต้องการความรู้ในระดับกายภาพของการจัดเก็บข้อมูล

- ไม่มีข้อมูลจริงของระเบียบใด ๆ ยกเว้นรากต้นไม้ (root) ที่จะดำรงอยู่ได้ โดยไม่ติดอยู่กับระเบียบพอ
- ถ้าระเบียบลูกมีระเบียบพ่อตั้งแต่ 2 ตัวขึ้นไป จากระเบียบชนิดเดียวกัน ระเบียบลูกจำเป็นต้องทำซ้ำภายในแต่ละระเบียบพอ
- การเปลี่ยนแปลงโครงสร้างของข้อมูลส่งผลให้ต้องมีการปรับเปลี่ยนและแก้ไขโปรแกรมประยุกต์ทั้งหมดที่ใช้ฐานข้อมูล
- รองรับเฉพาะกับข้อมูลที่มีความสัมพันธ์แบบ *หนึ่งต่อกลุ่ม* ไม่สามารถรองรับข้อมูลที่มีความสัมพันธ์แบบ *กลุ่มต่อกลุ่ม* ได้

- จะต้องมีการกำหนดลักษณะความสัมพันธ์ของข้อมูลทุกตัวก่อนเพื่อจะนำมาจัดเป็นโครงสร้าง
- การค้นหาข้อมูลมีจำกัด ถ้าข้อมูลมีจำนวนมาก การเข้าถึงข้อมูลจะใช้เวลานานในการค้นหา เนื่องจากจะต้องเข้าถึงต้นกำเนิดของข้อมูล
- การออกแบบฐานข้อมูลเป็นเรื่องที่ซับซ้อนยาก ทำให้ยากต่อการพัฒนา เพราะจะต้องเข้าใจโครงสร้างทางกายภาพของข้อมูล
- การเรียกใช้ข้อมูลจะต้องผ่านจาก Root เสมอ
- ไม่สะดวกในการค้นหาข้อมูลในระดับล่าง เพราะการค้นหาข้อมูลในระดับล่างต้องค้นหาทั้งแฟ้มข้อมูล
- ไม่มีความเป็นอิสระในโครงสร้างและข้อมูล เนื่องจากการจัดการข้อมูลจะขึ้นอยู่กับชนิดของอุปกรณ์ที่ใช้จัดเก็บข้อมูล
- มีการจัดการที่ซับซ้อนและขาดความยืดหยุ่น อาจจะเป็นผลทำให้เกิดความเสียหายที่คาดไม่ถึงได้
- ขาดมาตรฐานการรองรับที่ชัดเจน

## 6.2 แบบจำลองข้อมูลเครือข่าย (Network Data Model)

แบบจำลองข้อมูลเครือข่าย พัฒนามาจากแบบจำลองข้อมูลลำดับชั้น มีวัตถุประสงค์เพื่อกำหนดให้เป็นรูปแบบโครงสร้างข้อมูลที่เป็นมาตรฐาน และสามารถรองรับข้อมูลที่มีความสัมพันธ์ในแบบ *กลุ่มต่อกลุ่ม* ได้ ลักษณะแบบจำลองข้อมูลเครือข่ายจะคล้ายกับลักษณะแบบจำลองข้อมูลลำดับชั้น แต่จะแตกต่างกันตรงที่แบบจำลองข้อมูลเครือข่ายจะมีความยืดหยุ่นที่สูงกว่า เช่นสามารถมีต้นกำเนิดของข้อมูลได้มากกว่าหนึ่ง และยินยอมให้ระดับชั้นที่อยู่เหนือกว่ามีได้หลายแฟ้มข้อมูลถึงแม้ว่าระดับชั้นถัดลงมาจะมีเพียงแฟ้มข้อมูลเดียว ในขณะที่แบบจำลองข้อมูลลำดับชั้นจะมีเพียงพอเดียวหรือแฟ้มข้อมูลเท่านั้น และแบบจำลองเครือข่ายนี้จะใช้อุปกรณ์เก็บข้อมูลสำหรับตัวบ่งชี้ หรือพอยน์เตอร์ (pointer) เป็นตัวโยงความสัมพันธ์ระหว่างระเบียบในไฟล์ต่างๆ รวมทั้งสนับสนุนความสัมพันธ์ทั้งแบบ หนึ่งต่อกลุ่ม และความสัมพันธ์แบบ กลุ่มต่อกลุ่ม ทำให้โครงสร้างของข้อมูลแต่ละแฟ้มข้อมูลมีความสัมพันธ์คล้าย ๆ ร่างแห เช่น พนักงานขายหนึ่งคนสามารถออกใบส่งของได้หลายใบ แต่ละใบจะมีชื่อพนักงานขายเพียงชื่อเดียว หรือลูกค้าคนหนึ่งอาจจะมีการซื้อสินค้าได้หลายครั้ง จึงอาจจะมีใบส่งของได้หลายใบ และแต่ละใบจะมีชื่อผู้ซื้อได้เพียงหนึ่งชื่อเท่านั้น เป็นต้น



ภาพที่ 3.9 แสดงแบบจำลองข้อมูลแบบลำดับขั้น

แบบจำลองข้อมูลเครือข่ายมีลักษณะดังนี้

- สมาชิกของเซตหนึ่งๆ สามารถเป็นสมาชิกของเซตอื่นได้อีกด้วย โดยกลุ่มที่เรียกว่า CODASYL ซึ่งทุกเอนทิตี ที่มีความสัมพันธ์กันจะมีพอยน์เตอร์กำกับไว้
- ข้อมูลในฐานข้อมูลแบบนี้สามารถมีความสัมพันธ์กันแบบใดก็ได้ ไม่ว่าจะเป็นแบบหนึ่งต่อหนึ่ง แบบหนึ่งต่อกลุ่ม หรือ แบบกลุ่มต่อกลุ่ม
- นำเสนอข้อมูลและความสัมพันธ์ระหว่างข้อมูลแบบมัลติลิสต์ (multilist) หรือหลายรายการ ลักษณะการเชื่อมโยงข้อมูลให้เชื่อมเป็นจุด
- โครงสร้างแบบจำลองเป็นเซตของระเบียน (Record Set) ระเบียนแต่ละชุดประกอบด้วย Owner record และ Member record และ Member record สามารถมี Owner record ได้หลายระเบียน

### ข้อดีแบบจำลองข้อมูลเครือข่าย

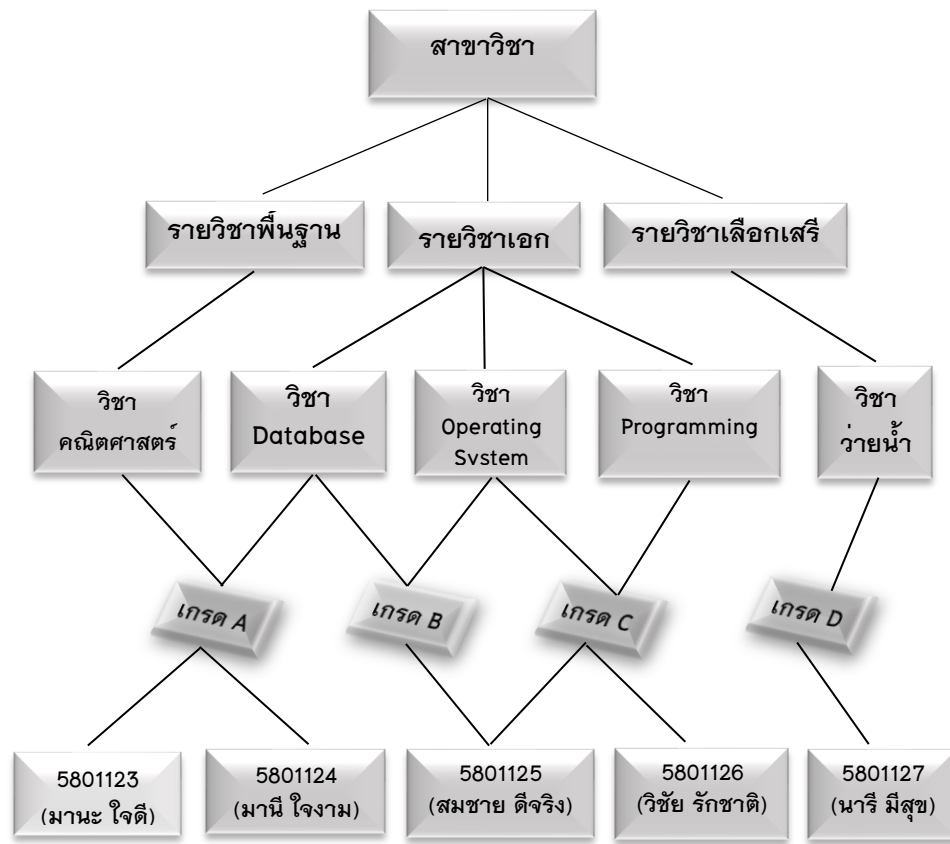
- สนับสนุนความสัมพันธ์แบบ กลุ่มต่อกลุ่ม ได้
- ความซ้ำซ้อนในข้อมูลเกิดขึ้นน้อยกว่าแบบจำลองข้อมูลแบบลำดับชั้น
- สามารถเชื่อมโยงข้อมูลแบบไป-กลับได้
- สะดวกในการค้นหาเพราะไม่ต้องไปเริ่มค้นหาตั้งแต่ข้อมูลต้นกำเนิดโดยทางเดียว
- มีความยืดหยุ่นในการค้นหาข้อมูล เข้าถึงข้อมูลได้ง่าย และเข้าถึงข้อมูลได้ทันที
- สนับสนุนให้โปรแกรมมีความเป็นอิสระจากข้อมูล

### ข้อจำกัดแบบจำลองข้อมูลเครือข่าย

- ด้วยเหตุที่สามารถเข้าถึงระเบียบได้โดยตรง ทำให้การป้องกันความปลอดภัยของข้อมูลมีน้อย
- สิ้นเปลืองค่าใช้จ่ายและเนื้อที่หน่วยความจำในการเก็บพอยน์เตอร์
- การออกแบบและการเปลี่ยนแปลงข้อมูลในแบบจำลองข้อมูลเครือข่ายทำได้ค่อนข้างยากเนื่องจากต้องกำหนดความสัมพันธ์ให้ครอบคลุมทุกข้อมูลในฐานข้อมูล
- ความสัมพันธ์ข้อมูลที่เชื่อมโยงกันไปมา ทำให้ยากต่อการใช้งาน จะเหมาะสำหรับโปรแกรมเมอร์ที่คุ้นเคย แต่จะไม่เหมาะสำหรับผู้ใช้งานทั่วไป
- เนื่องจากแบบจำลองนี้เป็นโครงสร้างที่ง่ายไม่ซับซ้อน ไม่ต้องอ่านแฟ้มข้อมูลที่เป็นต้นกำเนิดก่อน จึงทำให้ป้องกันความลับของข้อมูลได้ยาก

### 6.3 แบบจำลองข้อมูลเชิงสัมพันธ์ (Relational Database Model)

เป็นแบบจำลองที่นิยมแพร่หลายมากที่สุดในปัจจุบัน เพราะใช้งานง่าย ไม่ยุ่งยากซับซ้อน เป็นแบบจำลองข้อมูลที่แสดงโครงสร้างของข้อมูลที่มีอยู่จริงในองค์กร คิดค้นโดย Edgar Frank Codd (จาก บริษัท IBM ในปี ค.ศ.1970) เป็นแบบจำลองข้อมูลที่แสดงข้อมูลในรูปแบบของตาราง ภายในตารางจะประกอบด้วยแถว และสดมภ์ (หรือคอลัมน์) ข้อมูลของแต่ละแถวเรียกว่าทูเพิล (tuple) ส่วนสดมภ์ในตารางจะหมายถึงลักษณะ หรือคุณสมบัติ หรือ แอททริบิวต์ (attribute) ของตาราง แบบจำลองข้อมูลเชิงสัมพันธ์นี้สามารถแสดงความสัมพันธ์ข้อมูลได้ทั้งแบบ *หนึ่งต่อหนึ่ง* *หนึ่งต่อกลุ่ม* และแบบ *กลุ่มต่อกลุ่ม* และใช้ทั้งคีย์หลัก(primary key) และ คีย์รอง (secondary key) ในการอ้างอิงกับตารางอื่น ข้อแตกต่างของแบบจำลองเชิงสัมพันธ์กับแบบจำลองลำดับชั้น และแบบจำลองเครือข่าย คือในส่วนรายละเอียดของการจัดเก็บข้อมูลจะถูกซ่อนไว้จากผู้ใช้งาน การเรียก



ภาพที่ 3.10 ภาพแสดงแบบจำลองข้อมูลเครือข่าย

ใช้ข้อมูลจะผ่านทางภาษาสอบถาม หรือ Query Language ซึ่งเป็นภาษาขั้นสูง มีประสิทธิภาพสูง และใช้ง่าย ภาษาที่ได้รับความนิยมสูงสุดในกลุ่มผู้ใช้ในปัจจุบัน คือภาษา SQL (Structured Query Language) ซึ่งสร้างขึ้นโดยใช้พื้นฐานของแบบจำลองเชิงสัมพันธ์

การเชื่อมโยงข้อมูลระหว่างตาราง จะใช้ แอททริบิวต์ ที่มีอยู่ทั้งสองตารางเป็นตัวเชื่อมโยงข้อมูล ที่สามารถสื่อสัมพันธ์กับผู้ใช้ได้เข้าใจง่ายที่สุด ข้อมูลที่จัดเก็บอยู่ในตารางสามารถจัดเก็บข้อมูลในส่วนของตน และมีความสัมพันธ์กับตารางอื่น ๆ ได้

แบบจำลองข้อมูลเชิงสัมพันธ์ได้พัฒนาใช้งานกับโปรแกรมต่างๆ มากมาย รวมทั้งโปรแกรมระบบการจัดการฐานข้อมูลหรือ DBMS ก็สนับสนุนการทำงานของแบบจำลองดังกล่าวด้วยการใช้ชุดคำสั่ง SQL ในการจัดการฐานข้อมูล ที่ประกอบด้วยตารางต่าง ๆ มากมายด้วยการใช้ดีเอ็นในการกำหนดความสัมพันธ์

แบบจำลองข้อมูลเชิงสัมพันธ์มีลักษณะดังนี้

- ข้อมูลในแต่ละรีเลชันถูกเก็บในรูปของตาราง 2 มิติ คือแถว และสดมภ์
- ข้อมูลในแต่ละแถวต้องมีชื่อกำกับเพื่อใช้ในการอ้างอิง และชื่อจะต้องไม่ซ้ำกัน
- ค่าของข้อมูลในแต่ละสดมภ์ต้องเป็นค่าของข้อมูลชนิดเดียวกัน
- การเรียงลำดับของข้อมูลในแต่ละแถวหรือแอททริบิวต์ไม่เป็นประเด็นสำคัญ
- การเรียงลำดับของแต่ละสดมภ์หรือทูเพิลจะเรียงลำดับก่อนหลังอย่างไรก็ได้
- ค่าของข้อมูลในแต่ละแอททริบิวต์ของทูเพิลหนึ่งๆ จะบรรจุข้อมูลได้เพียงค่าเดียว
- ค่าของข้อมูลในแต่ละแอททริบิวต์ จะต้องเป็นไปตามข้อกำหนดของโดเมนใน

แอททริบิวต์นั้น ๆ

### ชื่อรีเลชัน “นิสิต”

| แอททริบิวต์ 1   แอททริบิวต์ 2   แอททริบิวต์ 3   แอททริบิวต์ 4   แอททริบิวต์ 5 |          |      |                |               |          |
|-------------------------------------------------------------------------------|----------|------|----------------|---------------|----------|
| รหัสนิสิต                                                                     | ชื่อ     | สกุล | คณะวิชา        | สาขาวิชา      |          |
| 580103                                                                        | น.ส.นารี | สดใส | พยาบาล         | การพยาบาล     | ทูเพิล 1 |
| 580314                                                                        | นายสะอาด | ใจดี | วิทยาศาสตร์    | ชีววิทยา      | ทูเพิล 2 |
| 580619                                                                        | นายมานะ  | อดทน | วิศวกรรมศาสตร์ | วิศวกรรมไฟฟ้า | ทูเพิล 3 |
| 580907                                                                        | น.ส.วิไล | โสภา | สหเวชศาสตร์    | กายวิภาค      | ทูเพิล 4 |

คาร์ดินัลลิตี

← ดีกรี →

ภาพที่ 3.11 ตัวอย่างรีเลชันในแบบจำลองข้อมูลเชิงสัมพันธ์

ส่วนประกอบของโครงสร้างฐานข้อมูลเชิงสัมพันธ์ (Relation Database Structure)

- รีเลชัน – ตาราง 2 มิติ ประกอบด้วย แถว และ สดมภ์
- แอททริบิวต์ – คุณสมบัติหรือรายละเอียดของ รีเลชัน
- โดเมน (Domain) – เป็นการกำหนดขอบเขต ค่าข้อมูล และชนิดของข้อมูล

เช่น รหัสนิสิต – มีค่าไม่เกิน 7 หลัก(digits)

- ทัพเพิล(Tuple) – แถวแต่ละแถวในรีเลชัน
- ดีกรี (Degree) – จำนวน แอททริบิวต์ ในรีเลชันนั้นๆ



- คาร์ดินัลลิตี้ (Cardinality) – จำนวนทuples ในรีเลชัน
- คีย์ (key) – แอททริบิวต์ หรือกลุ่มแอททริบิวต์ที่สามารถบอกความแตกต่างของ

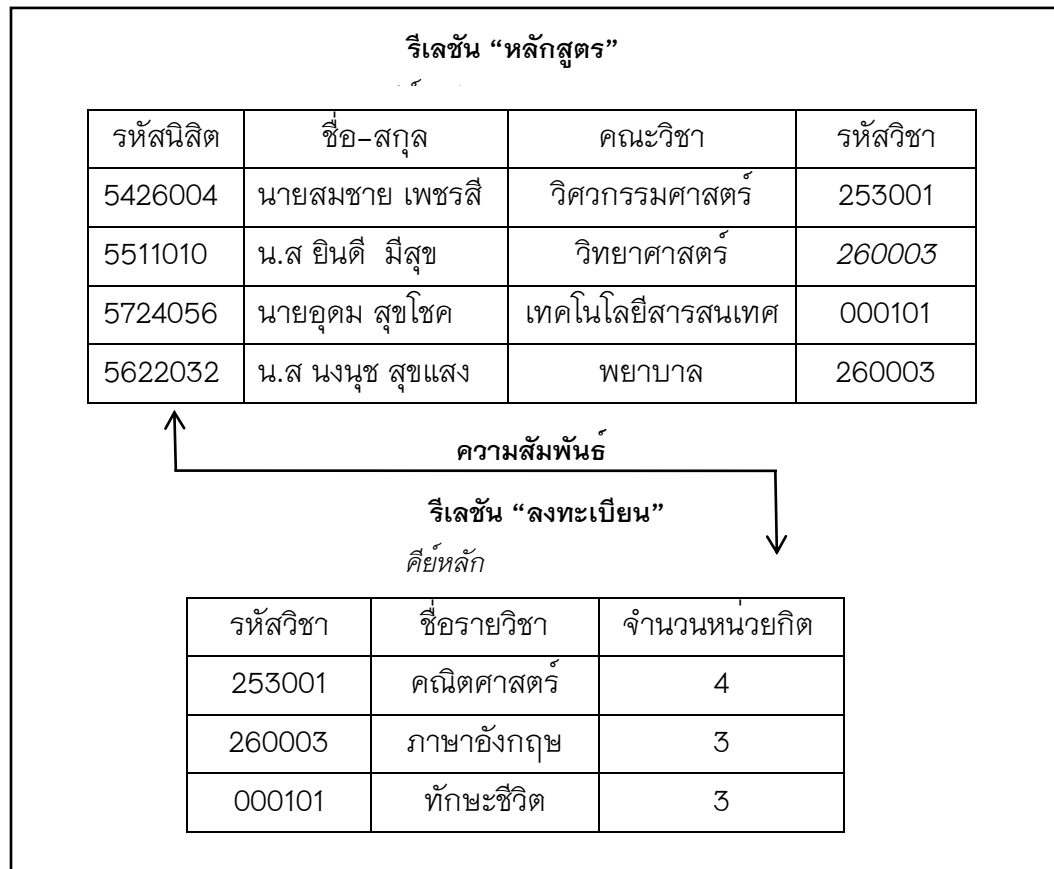
แต่ละทuples ในรีเลชัน

ข้อดีแบบจำลองข้อมูลเชิงสัมพันธ์

- รูปแบบโครงสร้างเข้าใจง่าย สามารถสื่อสารและเข้าใจได้ง่าย
- สนับสนุนความสัมพันธ์แบบกลุ่มต่อกลุ่ม
- มีความยืดหยุ่นในการเข้าถึงข้อมูล ความซับซ้อนของข้อมูลมีน้อยมาก
- มีความคงสภาพที่ดี (Data Integrity) และมีความเป็นมาตรฐาน
- โครงสร้างข้อมูลมีความอิสระจากโปรแกรม
- สามารถเลือกแสดงข้อมูลตามเงื่อนไขได้หลาย Key Field
- มีระบบรักษาความปลอดภัยที่ดี

ข้อจำกัดแบบจำลองข้อมูลเชิงสัมพันธ์

- แก้ไขปรับปรุงแฟ้มข้อมูลทำได้ยาก
- ระบบโดยรวมยังมีความซับซ้อน และมีข้อจำกัดด้านประสิทธิภาพ
- มีการลงทุนสูงเนื่องจากต้องใช้ฮาร์ดแวร์ และซอฟต์แวร์ ที่มีความสามารถสูง
- เมื่อมีการประมวลผลจะต้องทำการสร้างตารางขึ้นมาใหม่



ภาพที่ 3.12 ความสัมพันธ์ของแบบจำลองข้อมูลเชิงสัมพันธ์

#### 6.4 แบบจำลองข้อมูลเชิงวัตถุ (Object-Oriented Database Model)

แบบจำลองข้อมูลเชิงวัตถุ เป็นเทคโนโลยีที่พัฒนาต่อจากแบบจำลองข้อมูลเชิงสัมพันธ์ โดยมีแนวคิดหลักที่ประกอบไปด้วยวัตถุ (Object) การปกปิดความลับของวัตถุ (Encapsulation) คลาส (Class) การสืบทอด (Inheritance) รูปพรรณ (Identity) และ โดเมน (Domain) เป็นต้น

ฐานข้อมูลเชิงวัตถุมีจุดกำเนิดขึ้นมาในช่วงกลางปี ค.ศ. 1980 (Kim, 1990) โดยมีแนวคิดมาจากการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Program หรือ OOP) ที่ได้รับความนิยมในการทำงานเป็นอย่างมาก โดยมองทุกอย่างเป็นวัตถุและแต่ละวัตถุจะประกอบด้วยข้อมูลและโอเปอเรชัน หรือการปฏิบัติงาน ที่มีคลาส (class) หรือต้นแบบข้อมูลเป็นตัวกำหนดคุณสมบัติ (property) ของวัตถุ ซึ่งจะคล้าย ๆ กับแอททริบิวต์ที่ใช้อธิบายเอนทิตีในแบบจำลองข้อมูลเชิงสัมพันธ์นั่นเอง และฐานข้อมูลเชิงสัมพันธ์จะไม่เหมาะกับโครงสร้างฐานข้อมูลที่มีขนาดใหญ่ และมีความซับซ้อน

แบบจำลองข้อมูลเชิงวัตถุเป็นแบบจำลองข้อมูลที่แตกต่างไปจากแบบจำลองข้อมูลแบบเชิงสัมพันธ์ เช่นฐานข้อมูลเชิงวัตถุจะเก็บข้อมูลต่าง ๆ ไว้ในวัตถุแต่แบบจำลองข้อมูลแบบเชิงสัมพันธ์จะเก็บข้อมูลต่าง ๆ ไว้ในตาราง และจากคุณสมบัติที่สำคัญของฐานข้อมูลเชิงวัตถุดังที่กล่าวมา จะช่วยให้การจัดเก็บข้อมูลมีความซับซ้อนลดลง ฐานข้อมูลมีความคงสภาพมากขึ้น ลดความขัดแย้งของข้อมูล และไม่มีการสูญเสียประสิทธิภาพของระบบ

แบบจำลองข้อมูลเชิงวัตถุมีลักษณะสำคัญดังนี้

แบบจำลองข้อมูลเชิงวัตถุเกิดจากแนวคิดการเขียนโปรแกรมเชิงวัตถุ มีลักษณะสำคัญ คือ

- มองทุกสิ่งเป็นวัตถุ (objects) แต่ละวัตถุจะประกอบด้วยข้อมูลและโอเปอเรชัน ทำให้เข้าใจได้ง่าย แต่ละวัตถุจะเป็นแหล่งรวมของข้อมูลและการปฏิบัติ
- มีคลาสเป็นตัวกำหนดคุณสมบัติหรือรายละเอียดของวัตถุ (data and operation)
- โครงสร้างของเชิงวัตถุ ถูกกำหนดโดยใช้พรอปเพอร์ตี้ (property) ของคลาส
- สามารถจัดการกับข้อมูลชนิดต่างๆที่มีความซับซ้อนได้เป็นอย่างดี ไม่ว่าจะเป็นภาพ กราฟฟิก วิดีโอ และเสียง

- การสืบทอดคุณสมบัติ (inheritance) เป็นคุณลักษณะที่มีความสำคัญอย่างยิ่งของฐานข้อมูลเชิงวัตถุ เพราะจะทำให้โครงสร้างเป็นระบบระเบียบ สามารถปรับเปลี่ยนแก้ไขได้ง่าย ทำให้ลดเวลา และค่าใช้จ่ายในการพัฒนาระบบ

- โพลีมอร์ฟิซึม (polymorphism) หมายถึงการออกแบบเดียว แต่สามารถที่จะตอบสนองได้หลายรูปแบบ หรือหมายถึง มีชื่อเมธอดเดียวกันได้ในหลายๆคลาส โดยไม่จำเป็นที่จะต้องมีการมีพฤติกรรม หรือการทำงานที่เหมือนกัน

- การปกปิดความลับของวัตถุ (Encapsulation) หมายถึงการปกปิดความลับของวัตถุ ป้องกันไม่ให้วัตถุอื่นที่อยู่ภายนอกเข้าถึงวัตถุได้อย่างอิสระ

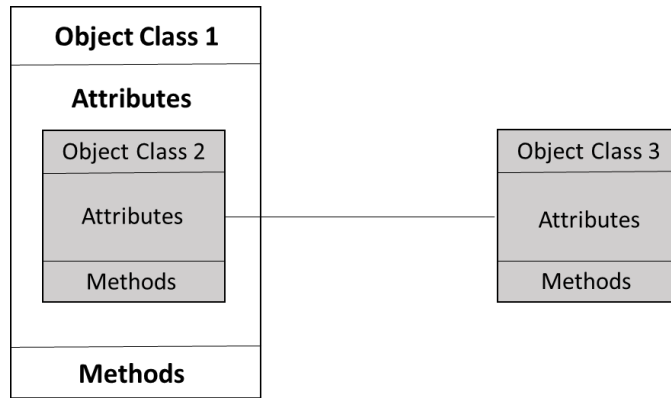
- ความแตกต่างของวัตถุแต่ละตัว (Object Identity , ODI)

- สนับสนุนการนำกลับมาใช้ใหม่

ข้อดีแบบจำลองข้อมูลเชิงวัตถุ

1. คุณสมบัติด้านการสืบทอด (Inheritance) ทำให้ข้อมูลมีความคงสภาพสูง
2. การนำเสนอเป็นแบบ Visual ทำให้อธิบายหัวข้อความหมายได้ดี
2. ในการเข้าถึงข้อมูลในแบบจำลองข้อมูลเชิงวัตถุ ผู้ใช้ไม่จำเป็นต้องรู้เรื่องวิธีการทำงานภายในของแต่ละเมธอดของวัตถุ

2. แบบจำลองข้อมูลเชิงวัตถุ จะสนับสนุนคุณสมบัติของการนำกลับมาใช้ใหม่ (Reusable)
3. แบบจำลองข้อมูลเชิงวัตถุสามารถจัดการกับข้อมูลที่มีความซับซ้อนได้เป็นอย่างดี เช่น ข้อมูลกราฟิก ข้อมูลภาพเคลื่อนไหว เป็นต้น



ภาพที่ 3.13 ความสัมพันธ์ของแบบจำลองข้อมูลเชิงวัตถุ

ข้อจำกัดแบบจำลองข้อมูลเชิงวัตถุ

1. ต้องอาศัยบุคลากรที่มีความรู้ ความเชี่ยวชาญ และประสบการณ์ด้านเทคโนโลยีเชิงวัตถุในการจัดการกับฐานข้อมูลชนิดนี้
2. ค่าใช้จ่ายเกี่ยวกับระบบแบบจำลองข้อมูลเชิงวัตถุค่อนข้างสูง
3. ยังไม่มีมาตรฐานรองรับที่ชัดเจน เมื่อเทียบกับแบบจำลองข้อมูลอื่น

#### 6.5 แบบจำลองข้อมูลมัลติไดเมนชัน (Multidimensional Database Model)

แบบจำลองข้อมูลมัลติไดเมนชัน เป็นแบบจำลองข้อมูลที่พัฒนาจากแบบจำลองข้อมูลเชิงสัมพันธ์ เป็นการแสดงข้อมูลหลายมิติ การจัดเก็บข้อมูลและการแสดงแถว (row) และคอลัมน์ (Column) ข้อมูล จะมีลักษณะเป็นลูกบาศก์ (Data Cube) หรือหลายไดเมนชัน นิยมใช้กับคลังข้อมูล (data warehousing) ทำให้มองเห็นและวิเคราะห์ข้อมูลได้ 2 ทาง ทำให้การวิเคราะห์ข้อมูล กระทำได้ง่ายขึ้น มองเห็นปัญหาและสร้างวิธีการแก้ไขปัญหาได้ดียิ่งขึ้น

แบบจำลองข้อมูลมัลติไดเมนชัน มีลักษณะดังนี้

แบบจำลองข้อมูลมัลติไดเมนชัน เป็นแบบจำลองข้อมูลที่พัฒนามาจากแบบจำลองข้อมูลแบบเชิงสัมพันธ์ จะมีลักษณะดังนี้

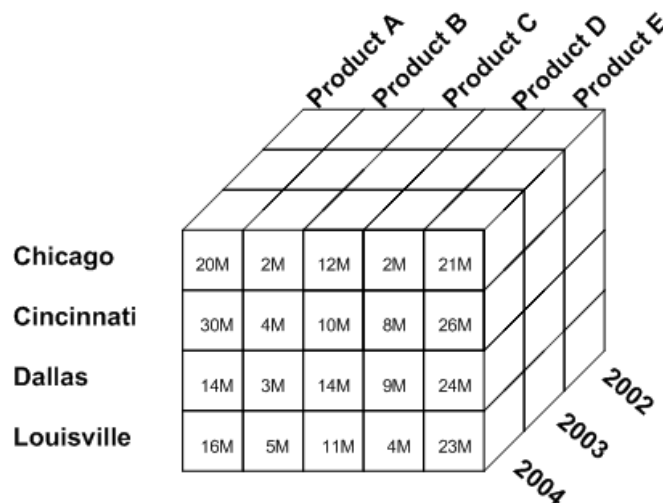
- แกวและคอลัมน์จะมีลักษณะเป็นรูปลูกบาศก์ (Data Cube)
- สามารถมองข้อมูลได้ 2 ทาง เพื่อให้เห็นปัญหาและสร้างวิธีแก้ปัญหาได้ดี
- ใช้แบบจำลอง Star Schema ในการออกแบบ
- มี Fact Table เก็บ fields ทั้งหมดที่จะวัด
- มีตาราง Dimension สำหรับ Join กับ Fact Table

ข้อดีแบบจำลองข้อมูลมัลติไดเมนชัน

1. เป็นแบบจำลองข้อมูลที่น่าเสนอมุมมองข้อมูลได้หลายมิติหรือหลายมุมมอง
2. สามารถนำไปประยุกต์ใช้ในการตัดสินใจ การวางแผนและการแก้ปัญหาทางธุรกิจได้

ข้อจำกัดแบบจำลองข้อมูลมัลติไดเมนชัน

1. การใช้งานแบบจำลองข้อมูลมัลติไดเมนชันต้องอาศัยผู้เชี่ยวชาญเฉพาะด้าน
2. ในการวิเคราะห์ข้อมูล มีค่าใช้จ่ายสูง ทั้งในส่วนของฮาร์ดแวร์และซอฟต์แวร์
3. เป็นแบบจำลองข้อมูลที่เหมาะสมกับธุรกิจขนาดใหญ่ ที่มีปริมาณของข้อมูลจำนวนมาก



ภาพที่ 3.14 ความสัมพันธ์ของแบบจำลองข้อมูลมัลติไดเมนชัน

(ที่มา [https://docs.oracle.com/cd/E12839\\_01/bi.1111/b40105/i\\_olap\\_chapter.htm#BIDPU142](https://docs.oracle.com/cd/E12839_01/bi.1111/b40105/i_olap_chapter.htm#BIDPU142))

### หนังสืออ้างอิง

1. ศุภกฤษฎี นิวัฒนากุล, การออกแบบและพัฒนาฐานข้อมูล, มหาวิทยาลัยเทคโนโลยีสุรนารี : นครราชสีมา, 2545
2. ชนวัฒน์ ศรีสอาน, การออกแบบและพัฒนาฐานข้อมูล, มหาวิทยาลัยเทคโนโลยีสุรนารี : นครราชสีมา, 2542.
3. วรากรณ์ โกวิทวรารังกูร, ระบบฐานข้อมูลและการออกแบบ, ศูนย์หนังสือจุฬาลงกรณ์มหาวิทยาลัย : กรุงเทพฯ, 2543.
4. Bachman C W. 1965. "Integrated Data Store", DPMA Quartely, (January 1965).
5. Bachman C W. 1973. "The programmer as Navigator", Communications of the ACM, (November 1973).
6. (<https://www.slideshare.net/devgocool/dbms-models>)