

ใบความรู้ที่ 1

หน่วยการเรียนรู้ที่ 1 การออกแบบฐานข้อมูล
เรื่อง การออกแบบฐานข้อมูล

วิชา คอมพิวเตอร์ 2
ชั้น มัธยมศึกษาปีที่ 5

ศัพท์ทั่วไปในการประมวลผลแฟ้มข้อมูล

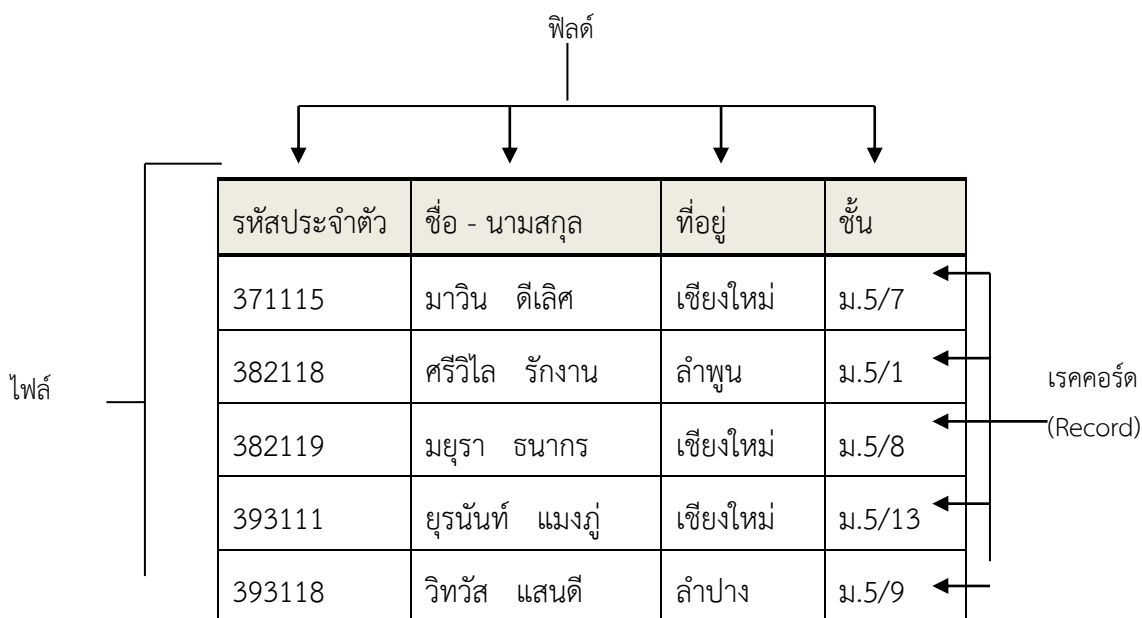
บิต (Bit : Binary Digit) เป็นหน่วยข้อมูลพื้นฐานที่เก็บอยู่ในหน่วยความจำภายในคอมพิวเตอร์ บิตนี้จะแทนด้วยตัวเลข 1 ตัว ได้แก่ 1 หรือ 0 อย่างไม่อย่างหนึ่ง เรียกเลข 1 หรือ 0 นี้ว่าเป็นบิต 1 บิต

ไบต์ (Byte) เป็นอักขระ (Character) แต่ละตัว เช่น A, B, ..., Z, 0, 1, 2, ..., 9 และสัญลักษณ์พิเศษอื่น ๆ เช่น \$, &, +, -, *, / ฯลฯ เมื่อจะนำไปเก็บไว้ในคอมพิวเตอร์ จะต้องถูกแปลงให้อยู่ในรูปของบิตหลายบิตที่มาประกอบกัน โดยอักขระ 1 ตัวจะแทนด้วยบิต 8 บิต ตัวอย่างเช่น อักขระ A เมื่อเก็บอยู่ในคอมพิวเตอร์จะเก็บเป็น 10000001 ตัวอักขระ B จะเก็บเป็น 10000010

เขตข้อมูลหรือฟิลด์ (Field) เป็นการนำเอาอักขระแต่ละตัวมาประกอบกันเป็นกลุ่มคำที่มีความหมาย เช่นกลุ่มตัวอักษรที่ประกอบกันเป็นชื่อหรือนามสกุล กลุ่มตัวเลขที่ประกอบกันเป็นรหัสประจำตัว ซึ่งจะเรียกเป็น ฟิลด์ชื่อ-นามสกุล ฟิลด์รหัสประจำตัว เป็นต้น

ระเบียบหรือเรคคอร์ด (Record) เป็นการนำฟิลด์หลายฟิลด์มารวมกัน เช่น เรคคอร์ดนักเรียนประกอบด้วยฟิลด์รหัสประจำตัว ชื่อ-นามสกุล ที่อยู่ เป็นต้น

แฟ้มข้อมูลหรือไฟล์ (File) เป็นการนำเอาเรคคอร์ด หลาย ๆ เรคคอร์ด ของข้อมูลชนิดเดียวกันมารวมกัน

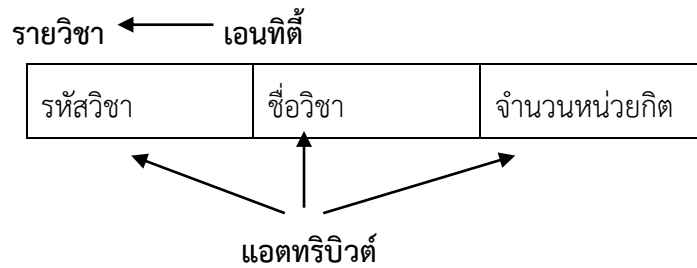


รูปที่ 1 แสดงตัวอย่างของเอนทิตี และแอตทริบิวต์

ศัพท์ที่พบในการจัดการฐานข้อมูล

เอนทิตี (Entity) เป็นสิ่งที่อ้างอิงถึงบุคคล สถานที่ และสิ่งของต่าง ๆ ที่เราสนใจ ในการออกแบบฐานข้อมูล เช่น เอนทิตีการลงทะเบียน การทำงาน การสั่งซื้อ เป็นต้น ตัวอย่างในระบบการลงทะเบียนจะประกอบด้วยเอนทิตีรายวิชานักเรียน การลงทะเบียน เป็นต้น

แอตทริบิวต์ (Attribute) เป็นข้อมูลที่แสดงลักษณะหรืออธิบายคุณสมบัติของเอนทิตี เช่น เอนทิตีรายวิชาจะประกอบด้วย แอตทริบิวต์รหัสวิชา ชื่อวิชา และจำนวนหน่วยกิต เป็นต้น ซึ่งก็คือ เขตข้อมูลหรือฟิลด์ (Field) ในแฟ้มข้อมูลนั่นเอง



รูปที่ 2 แสดงตัวอย่างของเอนทิตี และแอตทริบิวต์

ความสัมพันธ์ (Relationship) หมายถึง ความสัมพันธ์ระหว่างเอนทิตีต่าง ๆ ในระบบ

ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) คือ การรวบรวมเอนทิตีที่อยู่ในระบบที่มีความสัมพันธ์กันไว้ด้วยกัน

การออกแบบฐานข้อมูล

หลังจากที่เราได้กำหนดเป้าหมายของงานได้ชัดเจน และรู้ว่าจะต้องทำรายงานอะไร และใช้อะไรเป็นข้อมูลแล้ว ก็มาถึงจุดที่เราจะต้องกำหนดว่าจะต้องสร้างแฟ้มข้อมูลอะไรบ้างในงานนี้ หรือสำหรับการใช้โปรแกรมเอกเซลก็คือการกำหนดว่าจะใช้ตารางข้อมูลอะไรนั่นเอง

โดยทั่วไปแล้วเรามักจะใช้แฟ้มฐานข้อมูลเก็บข้อมูลต่อไปนี้

- ข้อมูลสำหรับนำไปใช้ประมวลผล
- ข้อมูลสำหรับการคำนวณ
- ข้อมูลที่เป็นค่าสะสม เช่น แฟ้มข้อมูลเงินเดือน ซึ่งอาจเก็บยอดเงินที่ได้รับทั้งหมดตั้งแต่ต้นปี เป็นต้น

ในทางปฏิบัติ เรามักจะไม่เก็บค่าที่สามารถคำนวณได้จากข้อมูล ยกตัวอย่างเช่น อายุซึ่งเปลี่ยนแปลงตลอดเวลา ทุกวัน เรามักไม่เก็บเอาไว้ในแฟ้มข้อมูล แต่เก็บวันเดือนปีเกิดแทน เมื่อต้องการอายุก็เอาวันที่ปัจจุบันตั้งลบด้วยวันเดือนปีเกิด ข้อดีอีกอย่างหนึ่งก็คือ วัน เดือน ปีเกิด ของบุคคลจะมีลักษณะคงที่ไม่มีมีการเปลี่ยนแปลง ฐานข้อมูลที่ใช้ในงานหนึ่ง ๆ นั้น ปกติมักจะประกอบด้วยแฟ้มข้อมูลหลายแฟ้มหรือหลายตาราง แต่ละแฟ้มเก็บข้อมูลแตกต่างกันออกไป ปัญหาสำคัญในการออกแบบฐานข้อมูลก็คือ การกำหนดว่าควรจะใช้แฟ้มข้อมูลกี่แฟ้ม และแต่ละแฟ้มควรเก็บข้อมูลอย่างไร

วิธีการออกแบบฐานข้อมูลวิธีหนึ่งที่ใช้กันมาก คือวิธีวิเคราะห์ความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่ต้องการเก็บ ซึ่งความสัมพันธ์นี้อาจจำแนกได้เป็น 3 ชนิด คือ

1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One to One) ความสัมพันธ์แบบนี้พบบ่อย และเป็นเรื่องที่ยากที่สุดสำหรับการออกแบบฐานข้อมูล ตัวอย่างเช่น สมาชิกชมรมแต่ละคนมีชื่อเพียงชื่อเดียว มีหมายเลขสมาชิกเพียงเลขเดียว เป็นต้น ในกรณีเช่นนี้ให้นำข้อมูลที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่งเหล่านั้นมาบรรจรรวมกันไว้เป็นตารางเดียวกันได้ ดังตัวอย่างแฟ้มข้อมูลสมาชิกชมรมนักร้อง ในรูปที่ 3

เขตข้อมูลลำดับที่	ชื่อเขตข้อมูล	ชนิดของเขตข้อมูล	รายละเอียด
1	Id	Text	รหัสสมาชิก
2	Name	Text	ชื่อสมาชิก
3	Numcopy	Number	จำนวนหนังสือที่มีสิทธิ์ยืม
4	Roomno	Text	ที่อยู่ในโรงเรียน

รูป 3 ตัวอย่างโครงสร้างแฟ้มข้อมูลสมาชิกชมรมนักร้อง

2. ความสัมพันธ์แบบหนึ่งต่อหลาย (One to Many) ความสัมพันธ์แบบนี้เกิดขึ้นระหว่างข้อมูลต่าง ๆ เช่น หนังสือหนึ่งเล่มมีผู้แต่งหลายคน ครูประจำชั้นหนึ่งคนมีนักเรียนที่จะต้องดูแลรับผิดชอบหลายคน และสมาชิกชมรมหนึ่งคนยืมหนังสือได้หลายเล่ม เป็นต้น

ถ้าเราสร้างแฟ้มข้อมูลสำหรับบรรจุข้อมูลเกี่ยวกับหนังสือเพียงแฟ้มเดียว เราจะประสบปัญหาว่าจะต้องสร้างเขตข้อมูลหลายเขตสำหรับเก็บชื่อผู้แต่ง ดังรูปที่ 4

Regno	Title	Author 1	Author 2	Author 3	Avcopy
-------	-------	----------	----------	----------	--------

รูปที่ 4 ระเบียบแฟ้มข้อมูลที่มีเขตข้อมูลชื่อผู้แต่ง 3 เขต

ระเบียบในรูปนี้ไม่เป็นที่ยอมรับ เพราะสิ้นเปลืองเนื้อที่เก็บมาก เนื่องจากส่วนใหญ่ผู้แต่งเพียงคนเดียว ดังนั้นเขตข้อมูล Author2 และ Author3 ส่วนใหญ่จะไม่ได้ใช้ ในขณะที่เดียวกันระเบียบตามรูปนี้ ก็จะใช้ไม่ได้ ถ้าหากหนังสือที่ต้องการเก็บข้อมูลนั้นมีผู้แต่งมากกว่าสามคน

หากเราเปลี่ยนรูปแบบของแฟ้มใหม่ โดยให้แต่ละระเบียบมีเขตข้อมูล Author เพียงเขตเดียว ถ้าหนังสือเล่มใดมีผู้แต่งหลายคนก็ให้บรรจุข้อมูลหนังสือเล่มนั้นซ้ำหลายครั้ง โดยเปลี่ยนเฉพาะชื่อผู้แต่ง ในกรณีนี้เราจะพบปัญหาอีกแบบหนึ่ง ปัญหาในที่นี้ก็คือ ความสิ้นเปลืองเนื้อที่เก็บข้อมูลและความซ้ำซ้อน เพราะจะต้องบันทึกชื่อหนังสือซ้ำกันหลายหน

Regno	Title	Author	Avcopy
B1	ระบบฐานข้อมูล	สมจิตร อาจินทร์	3
B1	ระบบฐานข้อมูล	งามนิจ อาจหาญ	3
B1	ระบบฐานข้อมูล	สวยสด หาญกล้า	3
B2	การใช้ Microsoft	สิทธิชัย ประสาน	2

B2	Access 97	วงศ์	2
B3	การใช้ Microsoft Access 97 จุดประกาย Access 97 : :	สมศักดิ์ ศรีสะอาด ปรารธนา สุขสวัสดิ์	1

รูปที่ 5 การจัดระเบียบข้อมูลแบบซ้ำซ้อน

วิธีการแก้ปัญหาที่เหมาะสม คือ แยกข้อมูลชุดนี้ออกเป็นสองแฟ้ม แฟ้มแรกบันทึกรายการข้อมูลที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง และแฟ้มที่สองบันทึกเฉพาะข้อมูลส่วนที่มีความสัมพันธ์แบบหนึ่งต่อหลาย ดังแสดงในรูปที่ 6 ก และ ข

Regno	Title	Avcopy
B1	ระบบฐานข้อมูล	3
B2	การใช้ Microsoft Access	2
B3	97 จุดประกาย Access 97 : : :	1

รูปที่ 6 ก แฟ้มทะเบียนชื่อหนังสือ

Regno	Author
B1	สมจิตร อัจฉรินทร์
B1	งามนิจ อัจฉาญ
B1	สวยสด หาญกล้า
B2	สิทธิชัย ประสานวงศ์
B2	สมศักดิ์ ศรีสะอาด
B3	ปรารธนา สุขสวัสดิ์

รูปที่ 6 ข แฟ้มชื่อผู้แต่ง

รูปที่ 6 ก และ ข นี้เป็นข้อมูลของหนังสือชุดเดียวกับรูปที่ 5 แต่แบ่งเป็น แฟ้มข้อมูลชื่อหนังสือและจำนวนเล่มแฟ้มหนึ่ง และแฟ้มชื่อผู้แต่งอีกแฟ้มหนึ่ง แฟ้มทั้งสองนี้มีเขตข้อมูลที่ซ้ำกันอยู่เขตหนึ่งคือ เลขทะเบียน หรือ Regno การซ้ำซ้อนในที่นี้ไม่ถือว่าเป็นความซ้ำซ้อนที่ต้องหลีกเลี่ยง แต่เป็นเรื่องซึ่งจำเป็นต้องซ้ำ เลขทะเบียนในที่นี้เป็นเขตข้อมูลสำคัญ ที่เรียกว่า เขตกุญแจหลัก (primary key field)

ประโยชน์ของกุญแจหลัก ก็คือ ใช้เป็นตัวเชื่อมข้อมูลจากแฟ้มทั้งสองเข้าด้วยกัน ทำให้ทราบว่าผู้แต่งหนังสือระบบฐานข้อมูล มีผู้แต่งสามคน คือ สมจิตร อัจฉรินทร์, งามนิจ อัจฉาญ, สวยสด หาญกล้า

จากตัวอย่างข้างต้นสรุปได้ว่า ถ้าหากมีความสัมพันธ์ชนิดหนึ่งต่อหลายที่จะต้องนำมาใช้สร้างแฟ้มข้อมูลแล้ว เราจำเป็นต้องสร้างเขตข้อมูลขึ้นเองให้เป็นเขตกุญแจ และให้ใช้เขตกุญแจจากแฟ้มแรกนี้ มาเชื่อมโยงเป็นเขตข้อมูลหนึ่งในแฟ้มที่สอง ซึ่งแสดงความสัมพันธ์แบบหนึ่งต่อหลาย

ขอให้สังเกตด้วยว่า เขตกุญแจในแฟ้มแรกนั้นต้องใช้ค่าที่ไม่มีทางซ้ำกัน คือ ภายในแฟ้มนี้จะต้องไม่มีทะเบียนใดเลยที่มีค่าเขตกุญแจซ้ำหรือเหมือนกับทะเบียนอื่น

การเลือกเขตกุญแจสำหรับแฟ้มข้อมูลนั้นปกติมักจะใช้เลขทะเบียน รหัสประจำตัว หรือเลขประจำตัวข้อมูลนั้น ๆ ยกตัวอย่างเช่น ปัจจุบันนี้ชาวไทยทุกคนมีเลขประจำตัวประชาชน 13 หลัก ซึ่งไม่มีการซ้ำกันเลย ดังนั้น จึงเหมาะที่จะใช้เป็นเขตกุญแจสำหรับแฟ้มข้อมูลที่เกี่ยวข้องของคนได้ หรือเลขประจำตัวผู้เสียภาษีอากร ก็ไม่มีเลขใดซ้ำ

กันและเหมาะที่จะใช้เป็นเขตกุญแจได้เช่นกัน ถ้าข้อมูลใดไม่มีเลขรหัสประจำตัว และไม่ต้องการที่จะสร้างรหัสขึ้น เราอาจคิดเลขชุดใด ๆ ที่ไม่ต้องมีความหมายมาใช้เป็นเขตกุญแจก็ได้เช่นกัน ยกตัวอย่างเช่น อาจตั้งค่าเขตกุญแจของระเบียบแรกให้เป็น 1001 ระเบียบที่สองเป็น 1002 เรื่อย ๆ ไป

3. ความสัมพันธ์แบบหลายต่อหลาย (Many to Many) ความสัมพันธ์แบบนี้เกิดขึ้นกับข้อมูลสองกลุ่มในแบบที่ข้อมูลในกลุ่มแรกสัมพันธ์กับข้อมูลหลายชุดในกลุ่มที่สอง และข้อมูลแต่ละชุดในกลุ่มที่สองก็สัมพันธ์กับข้อมูลหลายชุดในกลุ่มแรก ยกตัวอย่างเช่น นักเรียนที่เลือกเรียนวิชาต่าง ๆ จะพบความสัมพันธ์ว่า นักเรียนแต่ละคนอาจเลือกเรียนวิชาต่าง ๆ กันได้หลายวิชา และแต่ละวิชาที่อาจมีนักเรียนเลือกเรียนหลายคน

ในกรณีของความสัมพันธ์ชนิดแบบหลายต่อหลายนี้ เราจำเป็นต้องสร้างแฟ้มข้อมูลขึ้น 3 แฟ้ม ดังแสดงในรูปที่ 7 ก, 7 ข และ 7 ค

Id	Name	...
2001	แดง รักเรียน	
2002	ขาว รักสะอาด	
2003	ดำ รักงาน	

รูปที่ 7 ก แฟ้มชื่อนักเรียน

Cno	Title	...
C101	คอมพิวเตอร์และการประมวลค่า	
C102	การจัดการฐานข้อมูล	
C103	การเขียนโปรแกรม 1	

รูปที่ 7 ข แฟ้มชื่อวิชา

Id	Cno
2001	C101
2001	C103
2002	C102
2002	C103
2003	C102

รูป 7 ค แฟ้มการเลือกวิชาเรียน

เมื่อพิจารณาวิธีจัดแฟ้มทั้งสาม ซึ่งมีความสัมพันธ์แบบหลายต่อหลายเทียบกับการจัดแฟ้มสองแฟ้มของความสัมพันธ์แบบหนึ่งต่อหลายแล้ว จะเห็นว่าใช้หลักการเดียวกันนั่นเอง นั่นคือนำเขตกุญแจของแฟ้มชื่อนักเรียนกับเขตกุญแจของแฟ้มชื่อวิชามาไว้รวมกันในแฟ้มที่ 3 ซึ่งบอกว่านักเรียนหมายเลขใดเลือกเรียนวิชาเลขรหัสเท่าใด

แฟ้มที่ 3 ที่สร้างขึ้นในกรณีที่เชื่อมโยงได้สองรูปแบบ รูปแบบแรกก็คือ ถ้าต้องการค้นหา แแดง รักเรียน เลือกเรียนวิชาใด ก็สามารค้นชื่อในแฟ้มชื่อนักเรียนก่อน ซึ่งจะได้ ID ของนักเรียนผู้นั้นคือ 2001 แล้วใช้ Id ค้นในแฟ้มการเลือกวิชาว่า มีรหัสวิชาในแฟ้มที่มีค่า Id เท่ากับ 2001 บ้างก็ได้รับรหัส Cno เท่ากับ C103 จากนั้นก็นำค่า Cno นี้ไปค้นในแฟ้มชื่อวิชาก็จะได้ชื่อวิชาที่ แแดง รักเรียน เรียนตามต้องการ

ในทางกลับกัน ถ้าต้องการทราบว่านักเรียนคนไหนเรียนวิชา การจัดการฐานข้อมูล บ้างก็ให้เริ่มค้นจากแฟ้มชื่อวิชา แล้วจะได้อรหัสวิชาเป็น C102 นำค่านี้ไปตรวจสอบในแฟ้ม การเลือกวิชาเพื่อหา Id ของนักเรียนที่ตรงกับรหัสวิชา C102 จากนั้นก็นำ Id นั้นไปค้นในแฟ้มชื่อนักเรียนก็จะได้นักเรียนออกมา

ตามตัวอย่างเกี่ยวกับชมรมนักอ่านข้างต้น เรามีวิธีพิจารณารวม ๆ ไป เช่นหนังสือ ระบบฐานข้อมูล มี 3 เล่มซ้ำกัน ก็ให้แยกว่าเป็น เล่ม 1 หรือ 2 หรือ 3 ดังนี้ หนังสือแต่ละเรื่องก็อาจมีผู้ยืมได้หลายคน และเราก็อาจถือว่า

ความสัมพันธ์ระหว่างสมาชิกกับหนังสือเป็นแบบหลายต่อหลาย ในที่นี้เราจะยึดแนวความสัมพันธ์แบบหลายต่อหลายเป็นหลัก

การพิจารณาว่าเราควรใช้ความสัมพันธ์ชนิดใดนั้น ขึ้นอยู่กับวัตถุประสงค์ในการสร้างฐานข้อมูล ถ้าเราต้องการติดตามการยืมหนังสือแต่ละเล่มให้เฉพาะเจาะจงเลย ก็ควรเลือกความสัมพันธ์แบบหนึ่งต่อหลาย แต่ถ้าต้องการเพียงให้บริการ และเพื่อจัดทำข้อมูลรวมอย่างกว้าง ๆ เท่านั้น ก็เลือกความสัมพันธ์แบบหลายต่อหลาย

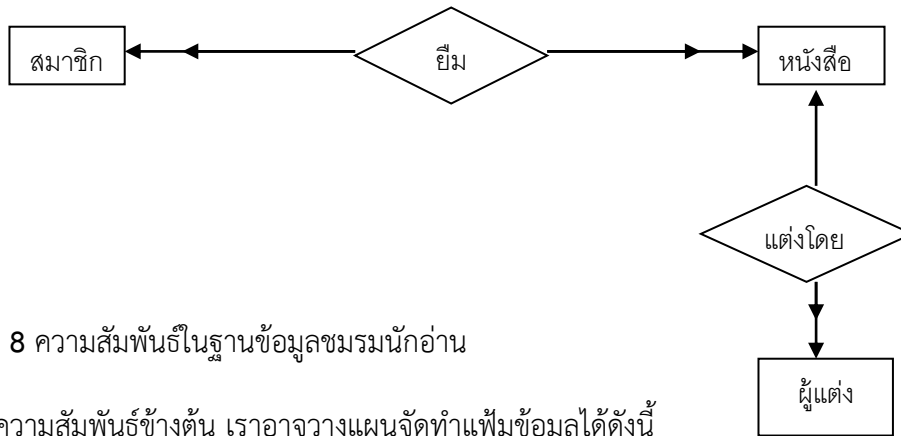
ตัวอย่างของการออกแบบฐานข้อมูล

ความสัมพันธ์ในฐานข้อมูลขรมรณิกอ่านที่กล่าวมาแล้วข้างต้น เมื่อพิจารณาแล้วจะเห็นว่ามีความสัมพันธ์ต่าง ๆ ดังนี้

สมาชิกมีความสัมพันธ์แบบหลายต่อหลายกับหนังสือ

หนังสือมีความสัมพันธ์แบบหนึ่งต่อหลายกับผู้แต่ง

เมื่อนำมาเขียนเป็นแผนภาพรวมกันก็จะได้ดังรูปที่ 8



รูปที่ 8 ความสัมพันธ์ในฐานข้อมูลขรมรณิกอ่าน

จากความสัมพันธ์ข้างต้น เราอาจวางแผนจัดทำแฟ้มข้อมูลได้ดังนี้

แฟ้ม member ประกอบด้วยข้อมูลเกี่ยวกับเลขสมาชิก, ชื่อสมาชิก, จำนวนหนังสือที่จะยืมได้ และหมายเลขห้อง

แฟ้ม bkittle ประกอบด้วยข้อมูลเกี่ยวกับเลขหนังสือ, ชื่อหนังสือ และจำนวนเล่มที่มีในขรม

แฟ้ม bkauthor ประกอบด้วยข้อมูลเกี่ยวกับเลขหนังสือ และชื่อผู้แต่ง

แฟ้ม bkserfce ประกอบด้วยข้อมูลเกี่ยวกับเลขสมาชิก, เลขสมาชิก, วันที่ยืม และวันกำหนดส่งคืน

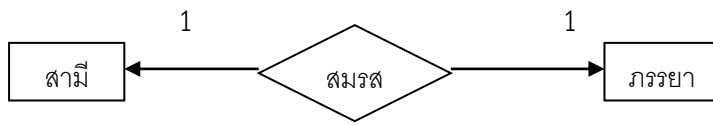
ในการออกแบบโปรแกรมฐานข้อมูลนั้น ผู้ออกแบบควรคำนึงถึงโครงสร้างของแฟ้ม และขนาดของแฟ้มไปด้วยพร้อมกัน ทั้งนี้ก็เพื่อจะได้ทราบว่า ฐานข้อมูลนี้ต้องการหน่วยเก็บขนาดใหญ่เท่าใด และระบบคอมพิวเตอร์ที่มีอยู่จะสามารถบรรจุแฟ้มต่าง ๆ ได้หมดหรือไม่ ถ้าคำนวณแล้วพบว่าฐานข้อมูลมีขนาดใหญ่เกินไป ก็จะต้องแก้ไขปรับปรุงฐานข้อมูลให้เหมาะสมยิ่งขึ้น

ความสัมพันธ์ระหว่างเอนทิตี

ความสัมพันธ์ระหว่างเอนทิตีและแสดงวิธีเขียนความสัมพันธ์ระหว่างเอนทิตีทั้ง 3 แบบดังนี้

1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง เช่น สามีกับภรรยา ครูประจำชั้นกับนักเรียน เช่น

ตัวอย่างที่ 1



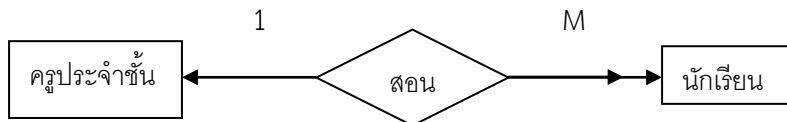
ตัวอย่างที่ 2



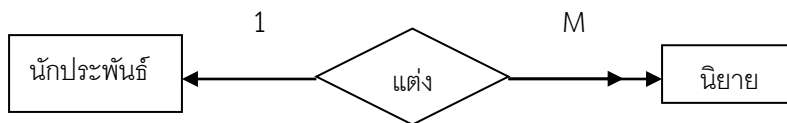
ในแง่ของข้อมูลเอนทิตีทั้งสองมีความเกี่ยวพันกันแนบแน่น เมื่อรู้เอนทิตีหนึ่งจะรู้เอนทิตีอีกอันหนึ่งได้ทันที เช่น เมื่อพูดถึงชื่อครูประจำชั้นคนหนึ่งจะรู้ทันทีว่าประจำอยู่ชั้นใด และถ้าพูดถึงชั้นเรียนหนึ่งก็จะบอกได้ทันทีเช่นกันว่าครูประจำชั้นชื่ออะไร

2. ความสัมพันธ์แบบหนึ่งต่อหลาย เช่น ครูประจำชั้นสอนนักเรียน นักประพันธ์แต่งนิยาย เช่น

ตัวอย่างที่ 1



ตัวอย่างที่ 2



ในแง่ของฐานข้อมูลแล้วความสัมพันธ์ชนิดหนึ่งต่อหลายนี้ ไม่ค่อยแนบแน่นนัก เช่น ถ้าเราพบนักประพันธ์ ที่เราชอบ แล้วเราบอกว่า “ผมชอบนิยายคุณมาก” นักประพันธ์อาจสับสนเพราะไม่ทราบว่าเราชอบนิยายเล่มไหน แต่ถ้าวเอยชื่อนิยายเล่มหนึ่งจะบอกต่อได้ทันทีว่านักประพันธ์ชื่ออะไร (ในกรณีนี้ นิยายแต่ละเรื่องแต่งโดยคน ๆ เดียว)

3. ความสัมพันธ์แบบหลายต่อหลาย เช่น นักเรียนเลือกเรียนวิชาเลือก



ในที่นี้ความสัมพันธ์มีค่อนข้างหลวม การรู้เอนทิตีใด ๆ แล้วจะหารายละเอียดของอีกเอนทิตีหนึ่งจะทำได้ลำบาก บางครั้งอาจถึงกับต้องตรวจหาทั้งฐานข้อมูลเลย เช่น การค้นหาว่ามีนักเรียนคนไหนเรียนวิชาพิมพ์ดีดบ้าง ก็ต้องค้นหาทั้งฐานข้อมูล เป็นต้น

ข้อสำคัญ

1. ถ้าเป็นความสัมพันธ์แบบหนึ่งต่อหนึ่ง และไม่มีควมจำเป็นต้องแยกเก็บเอนทิตีทั้งสองเป็นสองแฟ้มข้อมูลแยกกัน ก็ให้รวมกันไว้เป็นแฟ้มเดียวกันได้ เช่น แฟ้มข้อมูลสมาชิกชมรมนักอ่าน
2. ถ้าเป็นความสัมพันธ์แบบหนึ่งต่อหลาย ให้แยกเก็บแต่ละเอนทิตีเป็นหนึ่งแฟ้ม ให้จัดทำเขตกุญแจของเอนทิตีข้างที่มีความสัมพันธ์เป็นหนึ่ง แล้วนำเขตกุญแจนี้มาเพิ่มเข้าไปในแฟ้มของเอนทิตีข้างที่มีความสัมพันธ์เป็นหลาย เช่น การจัดแฟ้มทะเบียนชื่อหนังสือและแฟ้มชื่อผู้แต่ง
3. ถ้าเป็นความสัมพันธ์แบบหลายต่อหลาย ให้แยกเก็บแต่ละเอนทิตีเป็นหนึ่งแฟ้ม ในแต่ละแฟ้มให้เพิ่มเขตกุญแจของเอนทิตีนั้นเข้าไป หลังจากนั้นให้จัดทำแฟ้มข้อมูลอีกแฟ้มหนึ่งสำหรับเก็บความสัมพันธ์นั้น โดยนำค่าในเขตกุญแจจากแฟ้มเอนทิตีทั้งสองมาบรรจุลงไป เช่น การจัดแฟ้มข้อมูลนักเรียนในการเลือกเรียนวิชา