

จัดการฐานข้อมูลด้วย My SQL

กลุ่มสาระการเรียนรู้ การงานอาชีพและเทคโนโลยี
ตรงตามหลักสูตรแกนกลางการศึกษาขั้นพื้นฐาน พุทธศักราช ๒๕๕๑

สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ ๒๕๕๘

จัดพิมพ์ และ จำหน่ายโดย



บริษัท ซัคเซส มีเดีย จำกัด

1000/19-20 อาคารลิเบอร์ตี้ทาวเวอร์ ชั้น 12A
ซอยสุขุมวิท 55 (ทองหล่อ) แขวงคลองตันเหนือ
เขตวัฒนา กรุงเทพฯ 10110
โทร. 0-2762-9000 โทรสาร 0-2762-9001
E-mail : info@successmedia.com
www.successmedia.com

ข้อมูลบรรณานุกรมของสำนักงานหอสมุดแห่งชาติ

ชาญชัย สุภอรรถกร.

จัดการฐานข้อมูลด้วย My SQL .-- กรุงเทพฯ :
ซัคเซส มีเดีย, 2559.

212 หน้า. -- (แบบเรียนมัธยม).

1. เอสคิวแอลเชอร์ฟเวอร์. I. ชื่อเรื่อง.

005.7585

ISBN 978-616-205-432-7

เรียบเรียงโดย

ผศ. ชาญชัย สุภอรรถกร

การศึกษา :

- ปริญญาตรี คณะวิทยาศาสตร์ ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์
- ปริญญาโท คณะวิทยาศาสตร์ ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยมหิดล

ประวัติการทำงาน :

- ผู้ช่วยนักวิจัย ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC)
- นักโปรแกรมระบบ บริษัทสยามสารสนเทศและประมวลผล ธนาคารไทยพาณิชย์ จำกัด (มหาชน)
- อาจารย์สาขาคอมพิวเตอร์ธุรกิจ คณะบริหารธุรกิจวิทยาลัยโปลีเทคนิคภาคตะวันออกเฉียงเหนือ อุบลราชธานี
- อาจารย์สาขาคอมพิวเตอร์ธุรกิจ คณะการบัญชีและการจัดการ มหาวิทยาลัยมหาสารคาม
- ปัจจุบันเป็นอาจารย์ภาควิชาคณิตศาสตร์ สถิติและคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี

คำอธิบายรายวิชาเพิ่มเติม

จัดการฐานข้อมูลด้วย MySQL

กลุ่มสาระการเรียนรู้การงานอาชีพและเทคโนโลยี

ชั้นมัธยมศึกษาปีที่ ๑-๖

เวลา ชั่วโมง จำนวน หน่วยกิต

ศึกษาและปฏิบัติเกี่ยวกับ ระบบและแบบจำลองสารสนเทศ (Information Models and System) ระบบฐานข้อมูล (Database System) แบบจำลองข้อมูล (Data Modeling) ฐานข้อมูลเชิงสัมพันธ์ (Relational Database System) ภาษาสอบถามฐานข้อมูล (Database Query Language) การออกแบบฐานข้อมูล การทำให้เป็นบรรทัดฐาน (Normalization) การใช้โปรแกรมสำเร็จรูประบบฐานข้อมูล

เพื่อให้ผู้เรียนมีทักษะ และประสบการณ์ในการนำเทคโนโลยีมาประยุกต์ใช้ มีทักษะในการทำงานร่วมกับผู้อื่น สามารถตัดสินใจและแก้ปัญหาในการทำงาน

เพื่อให้ผู้เรียนมีทักษะในการแสวงหาความรู้ มีความคิดริเริ่มสร้างสรรค์ มีคุณธรรม จริยธรรม และค่านิยมที่ดี ต่องาน และเห็นแนวทางในการประกอบอาชีพสุจริต

ผลการเรียนรู้

- ประยุกต์ใช้ระบบจัดการฐานข้อมูลในงานอาชีพ
- วิเคราะห์และออกแบบระบบฐานข้อมูล
- ประยุกต์ใช้โปรแกรมสำเร็จรูปจัดการระบบฐานข้อมูล

รวมทั้งหมด ๓ ผลการเรียนรู้

คำนำ



หนังสือเรียน รายวิชาเพิ่มเติม จัดการฐานข้อมูลด้วยMySQL เล่มนี้ ได้เรียบเรียงขึ้นสำหรับนักเรียนชั้นมัธยมศึกษา โดยยึดกรอบตัวชี้วัดและสาระการเรียนรู้ กลุ่มสาระการเรียนรู้การงานอาชีพและเทคโนโลยี เพื่อนำมาใช้กับตามหลักสูตรแกนกลางการศึกษาขั้นพื้นฐาน พุทธศักราช ๒๕๕๑

ภายในหนังสือเล่มนี้ประกอบเนื้อหาเกี่ยวกับระบบฐานข้อมูลและการใช้โปรแกรม MySQL ได้แก่ รู้จักกับระบบฐานข้อมูลเบื้องต้น ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) การแสดงความสัมพันธ์ของข้อมูลด้วย ER Model ข้อมูลในรูปแบบบรรทัดฐาน (Normalization) Data Definition Language Data Manipulation Language Data Retrieval Command การเรียกดูข้อมูลขั้นสูง การใช้ SQL Functions Data Control Language Transaction Command ฐานข้อมูลแบบกระจาย โปรแกรมฐานข้อมูล MySQL การติดตั้งโปรแกรมฐานข้อมูล MySQL การใช้โปรแกรม PhpMyAdmin และการใช้โปรแกรม MySQL-Front

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่าหนังสือเรียนเล่มนี้ จะเป็นหนังสือประกอบการเรียนที่เป็นประโยชน์ต่อการสอนของคณะอาจารย์ และการเรียนรู้ของนักเรียน เพื่อให้การศึกษาทัดเทียมกับระดับสากล

บรรณาธิการ ชัคเชส มีเดีย
พฤศจิกายน ๒๕๕๙

สารบัญ

บทที่ 1 รู้จักกับระบบฐานข้อมูล	1
1.1 ระบบแฟ้มข้อมูล (File Systems)	2
1.2 Data Independence	7
1.3 ความหมายของระบบฐานข้อมูล	7
1.4 องค์ประกอบของ DBMS	9
กิจกรรมท้ายบทที่ 1	11
บทที่ 2 ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)	13
2.1 โครงสร้างของฐานข้อมูลเชิงสัมพันธ์	14
2.2 การควบคุมความถูกต้องให้กับข้อมูล (Data Integrity)	15
กิจกรรมท้ายบทที่ 2	21
บทที่ 3 การแสดงความสัมพันธ์ของข้อมูลด้วย ER Model	23
3.1 ความหมายของสัญลักษณ์ต่างๆของ ER Model	24
3.2 วิธีการสร้าง ER Diagram	31
3.3 การแปลง ER Diagram เป็นตาราง	35
กิจกรรมท้ายบทที่ 3	42
บทที่ 4 ข้อมูลในรูปแบบบรรทัดฐาน (Normalization)	43
4.1 ปัญหาการจัดการฐานข้อมูล (Anomalies)	44
4.2 Functional Dependency	45
4.3 การจัดการข้อมูลในรูปแบบบรรทัดฐาน (Normalization)	48
กิจกรรมท้ายบทที่ 4	54
บทที่ 5 Data Definition Language	55
5.1 ความหมายของ Data Definition Language	56
5.2 การใช้คำสั่ง CREATE	58
5.3 การใช้คำสั่ง ALTER	66
5.4 การใช้คำสั่ง DROP เป็นคำสั่งสำหรับการจัดการลบฐานข้อมูล และตารางฐานข้อมูล	68
กิจกรรมท้ายบทที่ 5	69

บทที่ 6 Data Manipulation Language	71
6.1 ความหมายของ Data Manipulation Language	72
6.2 การใช้คำสั่ง INSERT	72
6.3 การใช้คำสั่ง UPDATE	74
6.4 การใช้คำสั่ง DELETE	76
กิจกรรมท้ายบทที่ 6	78
บทที่ 7 Data Retrieval Command	79
7.1 การใช้งานคำสั่ง SELECT เบื้องต้น	80
7.2 การใช้งานคำสั่ง SELECT แบบมีเงื่อนไข	83
7.3 การใช้งานตัวดำเนินการร่วมกับคำสั่ง SELECT	85
7.4 การเรียงลำดับข้อมูล	90
กิจกรรมท้ายบทที่ 7	92
บทที่ 8 การเรียกดูข้อมูลขั้นสูง	93
8.1 การเรียกดูข้อมูลจากตารางมากกว่า 1 ตาราง โดยการ Join	94
8.2 การใช้งานคำสั่ง SELECT แบบซ้อนหลายระดับ (Subqueries)	100
กิจกรรมท้ายบทที่ 8	104
บทที่ 9 การใช้ SQL Function	105
9.1 ฟังก์ชันเกี่ยวกับการคำนวณ และสรุปค่าของข้อมูล (Aggregate Function)	106
9.2 การจัดกลุ่มข้อมูลด้วย GROUP BY	111
9.3 การกรองกลุ่มข้อมูลที่มีเงื่อนไข ด้วย HAVING	113
กิจกรรมท้ายบทที่ 9	115
บทที่ 10 Data Control Language	117
10.1 รั้งักสิทธิ (Privileges) ต่างๆ ในการเข้าถึงข้อมูล ของระบบจัดการฐานข้อมูล	118
10.2 การให้สิทธิ (Privileges) ผู้ใช้คนอื่นเข้าถึงข้อมูล ด้วยคำสั่ง GRANT	118
10.3 การยกเลิกสิทธิ (Privileges) ผู้ใช้คนอื่นด้วยคำสั่ง REVOKE	120
กิจกรรมท้ายบทที่ 10	122

บทที่ 11 Transaction Command	123
11.1 คำสั่ง COMMIT และ ROLLBACK	124
11.2 คำสั่ง Autocommit	126
11.3 การทำ Partial Rollback	129
กิจกรรมท้ายบทที่ 11	133
บทที่ 12 แนะนำโปรแกรมฐานข้อมูล MySQL	135
12.1 แนะนำ MySQL	136
12.2 คุณสมบัติของ MySQL	139
กิจกรรมท้ายบทที่ 12	140
บทที่ 13 การติดตั้งโปรแกรมฐานข้อมูล MySQL	141
13.1 การติดตั้งโปรแกรม Appserv	142
13.2 ระบบการจัดเก็บไฟล์ของโปรแกรม AppServ	146
กิจกรรมท้ายบทที่ 13	148
บทที่ 14 การใช้งานโปรแกรมฐานข้อมูล MySQL ด้วย PhpMyAdmin	149
14.1 การใช้งานโปรแกรม Appserv	150
14.2 การสร้างและลบฐานข้อมูล	151
14.3 การสร้างตาราง	152
14.4 การแก้ไขรายละเอียดของตาราง	155
14.5 การเพิ่มเรคคอร์ดใหม่ในตาราง	156
14.6 การดูข้อมูลภายในตาราง	157
14.7 การแก้ไขข้อมูลของเรคคอร์ด	158
14.8 การลบเรคคอร์ด	159
14.9 การลบตาราง	160
14.10 การส่งออกฐานข้อมูล	161
14.11 การนำเข้าฐานข้อมูล	163
14.12 การเขียนคำสั่ง SQL	165
กิจกรรมท้ายบทที่ 14	168

บทที่ 15 การใช้โปรแกรม MySQL-Front

171

15.1 การติดตั้งโปรแกรม MySQL Front

172

15.2 การเข้าสู่โปรแกรม และส่วนประกอบของโปรแกรม

175

15.3 การสร้างฐานข้อมูล/ตาราง

178

15.4 การลบฐานข้อมูล/ตาราง

184

15.5 การเพิ่มเรคคอร์ด

184

15.6 การแก้ไขเรคคอร์ด

187

15.7 การลบเรคคอร์ด

188

15.8 การแสดงฐานข้อมูล/ตาราง

190

15.9 การส่งออก/นำเข้าฐานข้อมูล

193

กิจกรรมท้ายบทที่ 15

199

บรรณานุกรม

202

สาระสำคัญ

ข้อมูลนับว่าเป็นสิ่งที่มีความสำคัญต่อหน่วยงาน หรือองค์กรทุกแห่ง เนื่องจากข้อมูลจะเป็นเสมือนเครื่องมือที่ช่วยประกอบการตัดสินใจ ถ้าหน่วยงาน หรือองค์กรใดขาดซึ่งข้อมูลก็จะไม่สามารถขับเคลื่อนธุรกิจให้บรรลุวัตถุประสงค์ได้เลย เนื้อหาในบทนี้จะเริ่มจากความหมาย ลักษณะของระบบแฟ้มข้อมูล ปัญหาของระบบแฟ้มข้อมูล ความหมาย ลักษณะของระบบฐานข้อมูล ประโยชน์ของระบบฐานข้อมูล

จุดประสงค์ทั่วไป

1. เพื่อให้ทราบความหมาย และลักษณะของระบบแฟ้มข้อมูล
2. เพื่อให้ทราบถึงปัญหาในการจัดเก็บข้อมูลในระบบแฟ้มข้อมูล
3. เพื่อให้ทราบความหมายของความเป็นอิสระของข้อมูล
4. เพื่อให้ทราบความหมาย และองค์ประกอบของระบบฐานข้อมูล
5. เพื่อให้ทราบความหมาย และองค์ประกอบของ DBMS

จุดประสงค์เชิงพฤติกรรม

1. บอกความหมายของของระบบแฟ้มข้อมูลได้
2. ยกตัวอย่างปัญหาในการจัดเก็บข้อมูลในระบบแฟ้มข้อมูลได้
3. บอกระดับความเป็นอิสระของข้อมูลได้
4. สามารถอธิบายองค์ประกอบแต่ละส่วนของระบบฐานข้อมูลได้
5. สามารถอธิบายองค์ประกอบแต่ละส่วนของ DBMS ได้

เนื้อหาสาระ

1. ระบบแฟ้มข้อมูล (File Systems)
2. Data Independence
3. ความหมายของระบบฐานข้อมูล
4. องค์ประกอบของ DBMS

1.1 ระบบแฟ้มข้อมูล (File Systems)

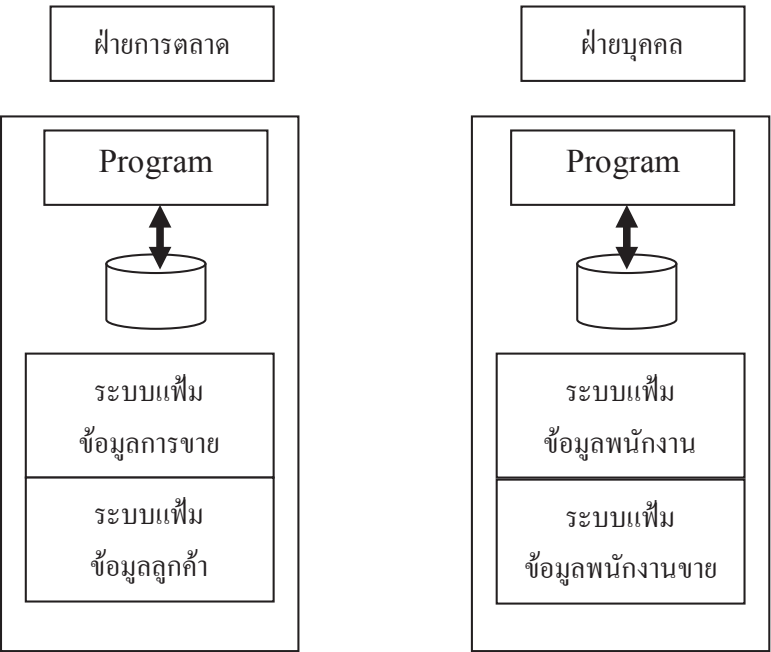
ในอดีตองค์กร หรือบริษัทต่างๆจะใช้วิธีการเก็บข้อมูลต่างๆ เช่น ข้อมูลพนักงาน ข้อมูลลูกค้า ข้อมูลการขาย เป็นต้น ไว้ภายในแฟ้มเอกสาร ตามตัวอย่างในรูปต่อไปนี้



รูปที่ 1.1 แสดงภาพการจัดเก็บเอกสารข้อมูลไว้ภายในแฟ้ม

จากเริ่มแรกองค์กร หรือบริษัทอาจมีข้อมูลไม่มากนัก ดังนั้นการค้นหาข้อมูลจากแฟ้มเอกสารก็จะสามารถทำได้ง่าย แต่ต่อมาเมื่อองค์กร หรือบริษัทมีข้อมูลปริมาณมากขึ้นจำนวนแฟ้มเอกสารก็มีปริมาณมากขึ้นตามตัวอย่างรูปข้างต้น ทำให้การค้นหา และจัดการข้อมูลทำได้อย่างยากลำบาก

ต่อมาเทคโนโลยีสารสนเทศเริ่มมีการพัฒนามากขึ้น คอมพิวเตอร์ได้เข้ามามีบทบาทต่อการจัดเก็บข้อมูลขององค์กร ทำให้การจัดเก็บข้อมูลในแบบเดิมที่จัดเก็บลงในแฟ้มเอกสารก็เปลี่ยนเป็นการจัดเก็บข้อมูลลงในแฟ้มข้อมูลในรูปของไฟล์ในคอมพิวเตอร์แทน และในการเข้าถึงเพื่อจัดการระบบแฟ้มข้อมูลดังกล่าวก็จะใช้วิธีการพัฒนาโปรแกรมเพื่อเรียกใช้ข้อมูลจากระบบแฟ้มข้อมูล โดยแต่ละหน่วยงานซึ่งมีระบบแฟ้มข้อมูลของตนเองก็จะพัฒนาโปรแกรมเพื่อเรียกใช้ระบบแฟ้มข้อมูลนั้นๆเอง เช่น ฝ่ายการตลาด ก็จะพัฒนาโปรแกรมเพื่อเรียกใช้ระบบแฟ้มข้อมูลการขาย เรียกใช้ระบบแฟ้มข้อมูลลูกค้า ฝ่ายบุคคลก็จะมีการพัฒนาโปรแกรมเพื่อเรียกใช้ระบบแฟ้มข้อมูลพนักงาน เป็นต้น



รูปที่ 1.2 แสดงตัวอย่างการจัดเก็บแฟ้มข้อมูล และการเรียกใช้

จากลักษณะการจัดเก็บระบบแฟ้มข้อมูลไว้ภายในคอมพิวเตอร์ตามรูปตัวอย่างข้างต้น จะเห็นว่าข้อมูลมีการกระจายถูกจัดเก็บไว้ตามฝ่ายงาน หรือหน่วยงานต่างๆ ทำให้จะเกิดปัญหาตามมา คือ

Data Redundancy

เป็นปัญหาความซ้ำซ้อนกันของข้อมูล ซึ่งเกิดจากการที่จัดเก็บข้อมูลเรื่องเดียวกันไว้มากกว่า 1 แฟ้มข้อมูล เช่น ฝ่ายบุคคลจัดเก็บระบบแฟ้มข้อมูลพนักงาน และพนักงานขายไว้แยกกัน ดังตัวอย่างต่อไปนี้

แฟ้มข้อมูลพนักงาน (Employee)

Emp_ID	Emp_Name	Emp_Address	Emp_Salary	Emp_Tel
E001	นาย A	กรุงเทพมหานคร	15,000	081-1234567
E002	นาย B	นนทบุรี	12,000	082-9876543
E003	นาย C	สมุทรปราการ	8,500	087-2468021
E004	นาย D	อุบลราชธานี	13,500	089-5302294

แฟ้มข้อมูลพนักงานขาย (Saleman)

Sale_ID	Sale_Name	Sale_Tel
S001	นาย A	081-1234567
S002	นาย C	087-2468021

จากตัวอย่างนี้เกิดข้อมูลซ้ำซ้อนกัน 2 แฟ้มข้อมูล คือข้อมูลของพนักงาน ชื่อ นาย A และ นาย C ซึ่งเป็นพนักงานของบริษัท ดังนั้นจึงมีข้อมูลอยู่ในแฟ้มข้อมูลพนักงาน และเนื่องจาก นาย A และนาย C ทั้งคู่เป็นพนักงานขายด้วย ดังนั้นในแฟ้มข้อมูลพนักงานขายก็ปรากฏข้อมูลของนาย A และ นาย C อีก การซ้ำซ้อนกันของข้อมูลนี้ทำให้สิ้นเปลืองเนื้อที่ในการจัดเก็บข้อมูล

Data Inconsistency

เป็นปัญหาความไม่ถูกต้องของข้อมูล อันเนื่องมาจากปัญหาเรื่องความซ้ำซ้อนที่ผ่านมา มีการจัดเก็บข้อมูลเดียวกันไว้หลายๆแฟ้มข้อมูล ทำให้บางครั้งข้อมูลอาจไม่ตรงกัน สรุปไม่ได้ว่าข้อมูลในแฟ้มข้อมูลไหนเป็นข้อมูลที่ถูกต้องกันแน่ เช่น

แฟ้มข้อมูลพนักงาน (Employee)

Emp_ID	Emp_Name	Emp_Address	Emp_Salary	Emp_Tel
E001	นาย A	กรุงเทพมหานคร	15,000	081-1234567
E002	นาย B	นนทบุรี	12,000	082-9876543
E003	นาย C	สมุทรปราการ	8,500	087-2468021
E004	นาย D	อุบลราชธานี	13,500	089-5302294

แฟ้มข้อมูลพนักงานขาย (Saleman)

Sale_ID	Sale_Name	Sale_Tel
S001	นาย A	081-2034821
S002	นาย C	087-2468021

ข้อมูลเบอร์โทรศัพท์ของนาย A จากแฟ้มข้อมูลพนักงาน และแฟ้มข้อมูลพนักงานขาย ไม่ตรงกัน ไม่ทราบว่าคุณสมบัติโทรศัพท์ของนาย A อันไหนเป็นข้อมูลที่ถูกต้องกันแน่

Data Anomaly

เป็นปัญหาอีกลักษณะหนึ่งของการจัดเก็บข้อมูล อันเนื่องมาจากการจัดเก็บข้อมูลเดียวกันไว้ในหลายๆแฟ้มข้อมูล โดยปัญหา Data Anomaly นี้จะเกิดขึ้นจากการเปลี่ยนแปลงข้อมูล คือ การเพิ่ม ลบ และแก้ไขข้อมูล ดังนั้นปัญหานี้จึงแบ่งออกเป็น 3 ลักษณะ

- **Insertion Anomaly** จะเกิดขึ้นเมื่อมีการเพิ่มข้อมูลใหม่ลงในแฟ้มข้อมูล เช่น มีพนักงานใหม่ชื่อ นาย E มาทำงาน โดยพนักงานคนนี้ทำงานในตำแหน่งพนักงานขายด้วยบริษัทได้ทำการเพิ่มข้อมูลของนาย E ลงไปในแฟ้มข้อมูลพนักงาน แต่ไม่ได้เพิ่มลงในแฟ้มข้อมูลพนักงานขายด้วย หรือในทางกลับกันก็เช่นเดียวกัน

แฟ้มข้อมูลพนักงาน (Employee)

Emp_ID	Emp_Name	Emp_Address	Emp_Salary	Emp_Tel
E001	นาย A	กรุงเทพมหานคร	15,000	081-1234567
E002	นาย B	นนทบุรี	12,000	082-9876543
E003	นาย C	สมุทรปราการ	8,500	087-2468021
E004	นาย D	อุบลราชธานี	13,500	089-5302294
E005	นาย E	เชียงใหม่	17,500	081-8487342

แฟ้มข้อมูลพนักงานขาย (Saleman)

Sale_ID	Sale_Name	Sale_Tel
S001	นาย A	081-1234567
S002	นาย C	087-2468021

- **Deletion Anomaly** จะเกิดขึ้นเมื่อมีการลบข้อมูลในแฟ้มข้อมูล เช่น มีพนักงานชื่อนาย C ได้ลาออกจากการเป็นพนักงานของบริษัท แต่ทางบริษัทได้ทำการลบข้อมูลของนาย C ออกจากแฟ้มข้อมูลพนักงานเท่านั้น แต่ไม่ได้ลบข้อมูลของนาย C ออกจากแฟ้มข้อมูลพนักงานขายด้วย หรือในทางกลับกันก็เช่นเดียวกัน

แฟ้มข้อมูลพนักงาน (Employee)

Emp_ID	Emp_Name	Emp_Address	Emp_Salary	Emp_Tel
E001	นาย A	กรุงเทพมหานคร	15,000	081-1234567
E002	นาย B	นนทบุรี	12,000	082-9876543
E004	นาย D	อุบลราชธานี	13,500	089-5302294

แฟ้มข้อมูลพนักงานขาย (Saleman)

Sale_ID	Sale_Name	Sale_Tel
S001	นาย A	081-1234567
S002	นาย C	087-2468021

- **Update Anomaly** จะเกิดขึ้นเมื่อมีการแก้ไขข้อมูลในแฟ้มข้อมูล เช่น พนักงานชื่อนาย A ได้ทำการเปลี่ยนชื่อเป็น นาย AA บริษัทได้ทำการแก้ไขชื่อเป็น นาย AA ในแฟ้มข้อมูลพนักงาน แต่ไม่ได้แก้ไขในแฟ้มข้อมูลพนักงานขายด้วย หรือในทางกลับกันก็เช่นเดียวกัน

แฟ้มข้อมูลพนักงาน (Employee)

Emp_ID	Emp_Name	Emp_Address	Emp_Salary	Emp_Tel
E001	นาย AA	กรุงเทพมหานคร	15,000	081-1234567
E002	นาย B	นนทบุรี	12,000	082-9876543
E003	นาย C	สมุทรปราการ	8,500	087-2468021
E004	นาย D	อุบลราชธานี	13,500	089-5302294

แฟ้มข้อมูลพนักงานขาย (Saleman)

Sale_ID	Sale_Name	Sale_Tel
S001	นาย A	081-1234567
S002	นาย C	087-2468021

1.2 Data Independence

ปัญหาอย่างหนึ่งในระบบแฟ้มข้อมูล (File Systems) คือ เมื่อมีการเปลี่ยนแปลงโครงสร้างทางกายภาพของแฟ้มข้อมูลจะต้องมีการเปลี่ยนแปลงโปรแกรมต่างๆ ที่เรียกใช้ข้อมูลในแฟ้มข้อมูลนั้นด้วย เช่น เปลี่ยน index file ของแฟ้มข้อมูลพนักงานจากเดิมเรียงตามชื่อพนักงานมาเป็นเรียงตามรหัสพนักงานแทน ทำให้ต้องมีการแก้ไขโปรแกรมตามโครงสร้าง index file ที่เปลี่ยนไป ลักษณะเช่นนี้เราจะเรียกว่าข้อมูลไม่มีความเป็นอิสระกัน (Data Dependence)

ด้วยเหตุนี้ภายในระบบฐานข้อมูลจึงมีการปรับปรุงกำหนดให้ข้อมูลภายในฐานข้อมูลต้องเป็นอิสระจากตัวโปรแกรมที่เรียกใช้ เพื่อที่จะสามารถแก้ไขโครงสร้างทางกายภาพของข้อมูลได้ โดยไม่กระทบต่อโปรแกรมที่เรียกใช้ข้อมูลจากฐานข้อมูลนั้น ลักษณะนี้เรียกว่า ข้อมูลมีความเป็นอิสระต่อกัน (Data Independence) ซึ่งแบ่งได้เป็น 2 ระดับ คือ

• **ความเป็นอิสระของข้อมูลเชิงตรรกะ (Logical Data Independence)**

เป็นความอิสระของข้อมูลในระดับแนวคิด (Conceptual Level) กับระดับภายนอก (External Level) หากมีการเปลี่ยนแปลงโครงสร้างข้อมูลในระดับแนวคิดแล้วจะไม่มีผลต่อโครงสร้างในระดับภายนอก หรือโปรแกรมประยุกต์ เช่นการเพิ่มฟิลด์ใหม่เข้าไปในฐานข้อมูล

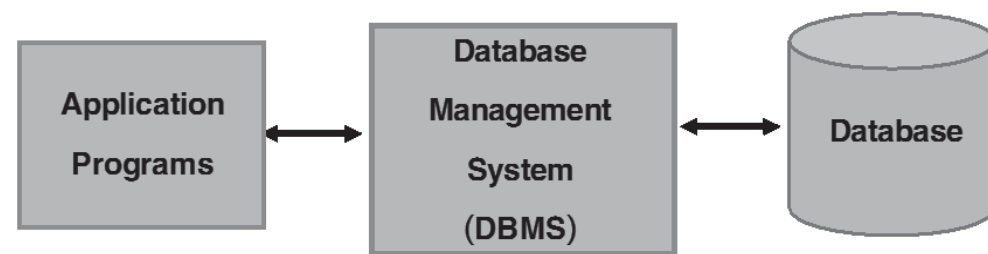
• **ความเป็นอิสระของข้อมูลเชิงกายภาพ (Physical Data Independence)**

เป็นความอิสระของข้อมูลในระดับภายใน (Internal Level) กับระดับแนวคิด (Conceptual Level) หรือระดับภายนอก (External Level) หากมีการเปลี่ยนแปลงในระดับภายใน เช่น การเพิ่มประสิทธิภาพการเรียกดูข้อมูลให้เร็วขึ้น โดยการปรับปรุงโครงสร้างภายใน จะไม่กระทบต่อโครงสร้างระดับแนวคิด หรือโครงสร้างภายนอก

1.3 ความหมายของระบบฐานข้อมูล

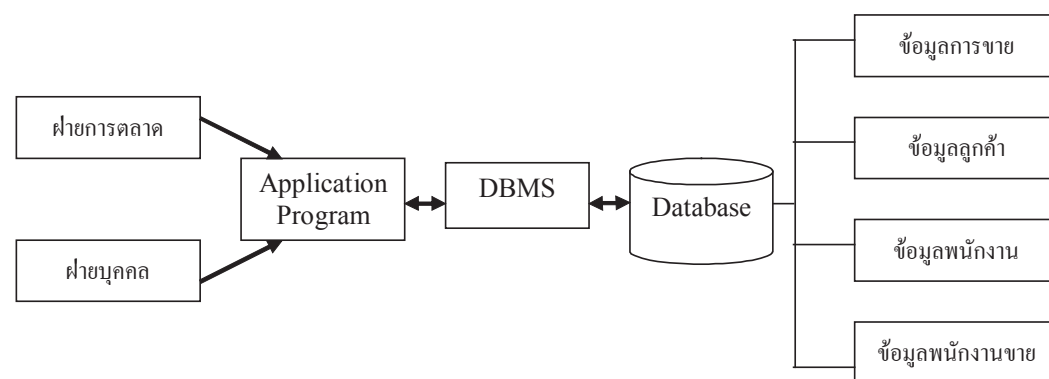
จากปัญหาต่างๆ ของการจัดเก็บข้อมูลในระบบแฟ้มข้อมูล (File Systems) ในหัวข้อที่ผ่านมาทำให้เกิดวิธีการจัดเก็บข้อมูลในรูปแบบใหม่ที่เราเรียกว่า ฐานข้อมูล (database) โดยฐานข้อมูล หมายถึง กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน มีการกำจัดความซ้ำซ้อนของข้อมูลออก และเก็บข้อมูลเหล่านี้ไว้ที่ศูนย์กลาง เพื่อที่จะนำข้อมูลเหล่านี้มาใช้ร่วมกัน โดยทั่วไปองค์กรต่าง ๆ จะสร้างฐานข้อมูลไว้เพื่อเก็บข้อมูลต่าง ๆ ของตัวองค์กร โดยเฉพาะอย่างยิ่งข้อมูลในเชิงธุรกิจ เช่น ข้อมูลของลูกค้า ข้อมูลของสินค้า ข้อมูลของลูกจ้าง และการจ้างงาน เป็นต้น การควบคุมดูแลการใช้ฐานข้อมูลนั้นจะจัดการผ่านตัวที่เรียกว่า

ระบบจัดการฐานข้อมูล (Data Base Management system- DBMS) ระบบจัดการฐานข้อมูล คือ ซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล ซึ่งมีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพ การเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการเรียกดูข้อมูลตามเงื่อนไขต่าง ๆ ออกมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล และสุดท้ายในการที่จะใช้งาน DBMS ได้ก็ต้องกระทำผ่านระบบหรือโปรแกรมที่เรียกใช้ข้อมูล (Application Programs) ดังรูปภาพต่อไปนี้



รูปที่ 1.3 องค์ประกอบของระบบฐานข้อมูล และความสัมพันธ์

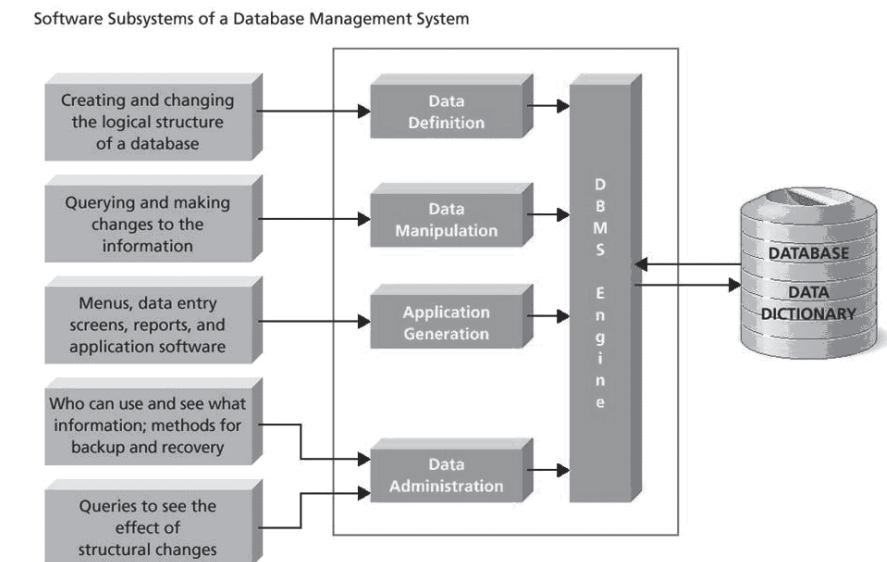
จากวิธีการจัดเก็บข้อมูลแบบแฟ้มข้อมูลของฝ่ายการตลาด และฝ่ายบุคคลในรูปที่ 1.2 ปรับมาเป็นระบบฐานข้อมูลก็จะมีลักษณะ ดังนี้



รูปที่ 1.4 แสดงการจัดการข้อมูลของฝ่ายการตลาด และฝ่ายบุคคลด้วยระบบฐานข้อมูล

1.4 องค์ประกอบของ DBMS

จากในหัวข้อที่ผ่านมาเราได้ทราบความหมายและหน้าที่ของ DBMS ไปแล้วในหัวข้อนี้เราจะมาเจาะลึกในส่วนของ DBMS ให้มากยิ่งขึ้น โดยจะมาทำความเข้าใจหน้าที่ขององค์ประกอบของ DBMS ซึ่งแบ่งองค์ประกอบของ DBMS ได้เป็น 4 ส่วน ดังรูปต่อไปนี้



รูปที่ 1.5 แสดงองค์ประกอบของ DBMS

- Data Definition** เป็นส่วนที่ทำหน้าที่ในการสร้าง และเปลี่ยนแปลงโครงสร้างของฐานข้อมูลโดยจะทำการแปล (Compile) ประโยคคำสั่งของกลุ่มคำสั่ง DDL (Data Definition Language) ให้อยู่ในรูปแบบของ MetaData ที่เก็บไว้ใน Data Dictionary
- Data Manipulation** เป็นส่วนให้ผู้ใช้สามารถทำการ เพิ่ม ลบ แก้ไขข้อมูลในฐานข้อมูลรวมทั้งการคิวรี (Query) ข้อมูลออกมาจากฐานข้อมูล
- Application Generation** เป็นส่วนที่เตรียมให้กับผู้ใช้ในการใช้งานให้ใช้งานได้สะดวก เช่น จัดเตรียมส่วนติดต่อ (Interface) ในลักษณะเป็นเมนู (Menu) หน้าจอกรอกข้อมูล (Data Entry Screen) รายงาน (Report) และ ซอฟต์แวร์ประยุกต์ (Application Software)
- Data Administration** เป็นส่วนที่จัดเตรียมช่วยเหลือผู้ใช้ในภาพรวมทั้งหมด เช่น จัดการเรื่องการสำรองฐานข้อมูล (Backup) การกู้คืน (Recovery) การจัดการระบบรักษาความปลอดภัย (Security Management) การควบคุมการเกิดภาวะพร้อมกัน (Concurrency Control) เป็นต้น

จากรูปที่ 1.5 ในส่วนของตัว Database ยังมีส่วนที่เรียกว่า Data Dictionary อาจเรียกว่า Data Repository โดยเป็นส่วนที่บรรจุข้อมูลของข้อมูล (MetaData) รายละเอียดทั้งหมดของฐานข้อมูลจะถูกเก็บไว้ภายในนี้ ในการทำงานของฐานข้อมูลทุกครั้ง DBMS จะมีการตรวจสอบไปที่ Data Dictionary ภายใน Data Dictionary จะมีข้อมูลที่สำคัญต่อไปนี้

- โครงสร้างทางตรรกะของฐานข้อมูล (Logical structure of database)
- ลักษณะโครงสร้างข้อมูล การแปลงข้อมูล และกฎควบคุมต่างๆ (Schemas, mappings and constraints)
- รายละเอียดของโปรแกรมประยุกต์ (Description about application programs)
- รายละเอียดของประเภทของเรคคอร์ด ประเภทของข้อมูล และการรวมกันของข้อมูลในฐานข้อมูล (Descriptions of record types, data item types, and data aggregates in the database)
- รายละเอียดการออกแบบฐานข้อมูลทางกายภาพ เช่น โครงสร้างการจัดเก็บข้อมูล (Description about physical database design)
- รายละเอียดของในการใช้งานของผู้ใช้ (Descriptions about users of DBMS and their access rights)

สรุป

จากปัญหาของการจัดเก็บข้อมูลในระบบแฟ้มข้อมูล คือ Data Redundancy, Data Inconsistency และ Data Anomaly (Inset Anomaly, Delete Anomaly, Update Anomaly) ทำให้เกิดระบบการจัดการฐานข้อมูล (DataBase Management System - DBMS) ซึ่งเป็นซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และโปรแกรมต่างๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล ซึ่งมีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพ โดย DBMS ประกอบไปด้วย Data Definition, Data Manipulation, Application Generation และ Data Administration

กิจกรรมท้ายบทที่ 1

1. ยกตัวอย่างข้อเสียของการจัดเก็บข้อมูลในระบบแฟ้มข้อมูล (File System)

2. อธิบายความหมาย และบอกความสำคัญของ DataBase Management System - DBMS

3. ความเป็นอิสระของข้อมูล (Data Independence) แบ่งออกเป็นกี่ระดับ อธิบายลักษณะที่สำคัญในแต่ละระดับด้วย

4. จับคู่ระหว่างคำศัพท์ และความหมายต่อไปนี้ให้ถูกต้อง

- | | |
|---------------------------|---|
| 1. Data Redundancy | a. เป็นส่วนให้ผู้ใช้สามารถทำการ เพิ่ม ลบ แก้ไขข้อมูลในฐานข้อมูล |
| 2. Data Inconsistency | b. เป็นส่วนที่จัดเตรียมช่วยเหลือผู้ใช้ในภาพรวมทั้งหมด เช่น จัดการเรื่องการสำรองฐานข้อมูล |
| 3. Inset Anomaly | c. มีพนักงานใหม่ชื่อ นาย E มาทำงาน โดยพนักงานคนนี้ทำงานในตำแหน่งพนักงานขายด้วยบริษัทได้ทำการเพิ่มข้อมูลของนาย E ลงไปในแฟ้มข้อมูลพนักงาน แต่ไม่ได้เพิ่มลงในแฟ้มข้อมูลพนักงานขายด้วย |
| 4. Delete Anomaly | d. Data Repository โดยเป็นส่วนที่บรรจุข้อมูลของข้อมูล (MetaData) รายละเอียดทั้งหมดของฐานข้อมูลจะถูกเก็บไว้ภายในนี้ |
| 5. Update Anomaly | e. เป็นส่วนที่ทำหน้าที่ในการสร้าง และเปลี่ยนแปลงโครงสร้างของฐานข้อมูล โดยจะทำการแปล (Compile) ประโยคคำสั่งของกลุ่มคำสั่ง DDL (Data Definition Language) ให้อยู่ในรูปแบบของ MetaData |
| 6. Data Definition | f. พนักงานชื่อ นาย A ได้ทำการเปลี่ยนชื่อเป็น นาย AA บริษัทได้ทำการแก้ไขชื่อเป็น นาย AA ในแฟ้มข้อมูลพนักงาน แต่ไม่ได้แก้ไขในแฟ้มข้อมูลพนักงานขายด้วย |
| 7. Data Manipulation | g. เป็นส่วนที่เตรียมให้กับผู้ใช้ในการใช้งานให้ใช้งานได้สะดวก เช่น จัดเตรียมส่วนติดต่อ (Interface) |
| 8. Application Generation | h. มีพนักงานชื่อ นาย C ได้ลาออกจากการเป็นพนักงานของบริษัท แต่ทางบริษัทได้ทำการลบข้อมูลของนาย C ออกจากแฟ้มข้อมูลพนักงานเท่านั้น แต่ไม่ได้ลบข้อมูลของนาย C ออกจากแฟ้มข้อมูลพนักงานขายด้วย |
| 9. Data Administration | i. เป็นปัญหาความซ้ำซ้อนกันของข้อมูล ซึ่งเกิดจากการที่จัดเก็บข้อมูลเรื่องเดียวกันไว้มากกว่า 1 แฟ้มข้อมูล |
| 10. Data Dictionary | j. เป็นปัญหาความไม่ถูกต้องของข้อมูล อันเนื่องจากปัญหาเรื่องความซ้ำซ้อน มีการจัดเก็บข้อมูลเดียวกันไว้หลายๆ แฟ้มข้อมูล ทำให้บางครั้งข้อมูลอาจไม่ตรงกัน |

สาระสำคัญ

ฐานข้อมูลเชิงสัมพันธ์ เป็นฐานข้อมูลที่ได้รับความนิยมใช้กันมากที่สุดในปัจจุบัน เนื่องจาก ความง่ายในการทำความเข้าใจ และที่สำคัญคือ เป็นฐานข้อมูลที่สามารถแก้ปัญหาในเรื่องของ Data Redundancy และ Data Inconsistency รวมทั้งมีการแยกโครงสร้างของฐานข้อมูลออกเป็น Logical และ Physical จึงทำให้เกิดความเป็นอิสระของข้อมูล (Data Independence) ทั้งหมดนี้เราได้ศึกษากันในบทที่ 1 แล้ว ดังนั้นในบทนี้เราจะทำการศึกษา และทำความเข้าใจถึงลักษณะฐานข้อมูลเชิงสัมพันธ์ว่ามีลักษณะหรือรูปแบบในการเก็บข้อมูลอย่างไร

จุดประสงค์ทั่วไป

1. เพื่อให้รู้จักโครงสร้างของฐานข้อมูลเชิงสัมพันธ์
2. เพื่อให้ทราบหลักการควบคุมความถูกต้องให้กับข้อมูล
3. เพื่อให้รู้จักคีย์ประเภทต่างๆ
4. เพื่อให้รู้จักกฎความคงสภาพของ Entity
5. เพื่อให้รู้จักกฎความคงสภาพของการอ้างอิง

จุดประสงค์เชิงพฤติกรรม

1. สามารถบอกความหมายของโครงสร้างภายในแต่ละตัวของฐานข้อมูลเชิงสัมพันธ์ได้
2. สามารถบอกลักษณะของคีย์ในแต่ละประเภทได้
3. สามารถกำหนดคีย์ในตารางฐานข้อมูลได้อย่างถูกต้อง
4. สามารถรักษากฎความคงสภาพของ Entity และกฎความคงสภาพของการอ้างอิงเพื่อให้ตารางฐานข้อมูลมีความถูกต้อง
5. สามารถจัดการได้อย่างถูกต้องเมื่อมีการผิดกฎความคงสภาพของการอ้างอิง

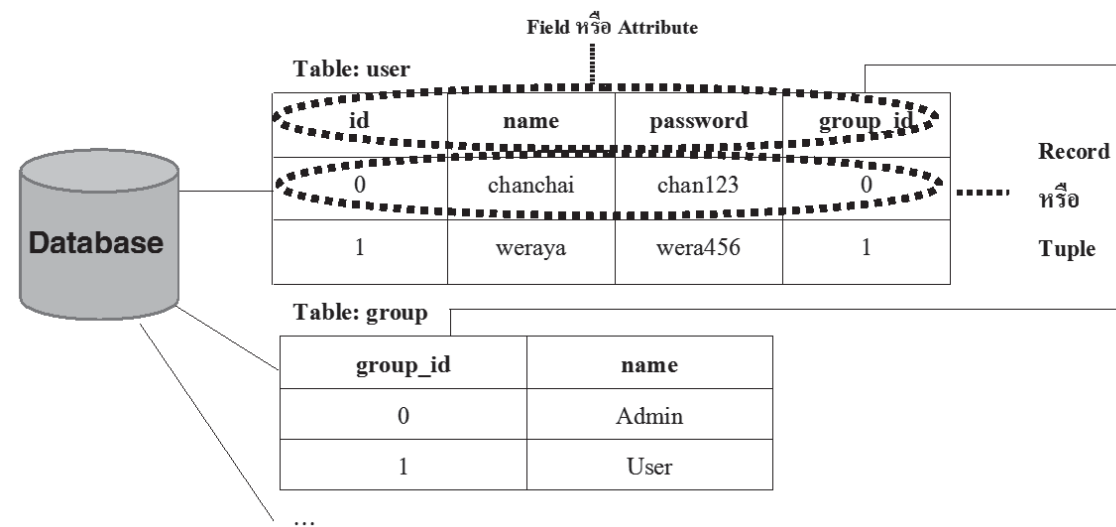
เนื้อหาสาระ

1. โครงสร้างของฐานข้อมูลเชิงสัมพันธ์
2. การควบคุมความถูกต้องให้กับข้อมูล (Data Integrity)

2.1 โครงสร้างของฐานข้อมูลเชิงสัมพันธ์

ฐานข้อมูลเชิงสัมพันธ์ เป็นการจัดเก็บข้อมูลในรูปแบบของตาราง (Table) ที่มีลักษณะเป็นสองมิติ คือ แถว (Row) และคอลัมน์ (Column) ฐานข้อมูล (Database) ลักษณะโครงสร้างการจัดเก็บภายในฐานข้อมูลเชิงสัมพันธ์ที่สำคัญ คือ

- ฐานข้อมูล (Database) สามารถประกอบขึ้นจาก ตาราง (Table) ตั้งแต่ 1 ตารางขึ้นไปฐานข้อมูลเชิงสัมพันธ์จะเรียกตารางนี้ว่า Relation
- ตาราง 1 ตารางประกอบขึ้นจากกลุ่มของข้อมูลเรื่องใดเรื่องหนึ่ง ซึ่งเรียกว่า ฟิลด์ (Field) ในฐานข้อมูลเชิงสัมพันธ์จะเรียกฟิลด์นี้ว่า Attribute
- จำนวนฟิลด์ หรือ Attribute จะเรียกว่า Degree
- และข้อมูลในแต่ละชุด หรือแต่ละแถว จะเรียกว่า เรคคอร์ด (Record) ในฐานข้อมูลเชิงสัมพันธ์จะเรียกเรคคอร์ดนี้ว่า Tuple
- จำนวน Record หรือ Tuple ในตารางหนึ่งที่มีความสัมพันธ์กับ Tuple ในอีกตารางหนึ่ง เรียกว่า Cardinality
- ในการเชื่อมโยงกันระหว่างข้อมูลในตาราง 2 ตาราง หรือมากกว่า จะเชื่อมโยงโดยใช้ฟิลด์ที่มีอยู่ในตารางที่ต้องการเชื่อมโยงข้อมูลกันดังรูปภาพต่อไปนี้



รูปที่ 2.1 ลักษณะโครงสร้างของฐานข้อมูลเชิงสัมพันธ์

จากรูปโครงสร้างฐานข้อมูล สามารถประกอบไปด้วยตารางได้ตั้งแต่ 1 ตาราง ยกตัวอย่างในรูปเราจะเห็นตาราง user ที่ใช้เก็บข้อมูลรายละเอียดของพนักงาน โดยตาราง user นี้ ประกอบขึ้นจาก 4 ฟิลด์ คือ id, name, password และ group_id (กล่าวได้ว่า ตาราง user นี้เป็นตารางที่มี Degree เท่ากับ 4) และในตาราง user นี้ประกอบไปด้วยชุดของข้อมูล 2 เรคคอร์ด คือ

เรคคอร์ดที่ 1 รหัสผู้ใช้ เท่ากับ 0 ชื่อ chanchai รหัสผ่าน chan123 และอยู่กลุ่มผู้ใช้ที่ 0
เรคคอร์ดที่ 2 รหัสผู้ใช้ เท่ากับ 1 ชื่อ weraya รหัสผ่าน wera456 และอยู่กลุ่มผู้ใช้ที่ 1

นอกจากนั้น ในตาราง user ยังเชื่อมโยงข้อมูลไปยังตาราง group ผ่านทางฟิลด์ group_id ดังนั้น เราสามารถสรุปได้ว่า ผู้ใช้ชื่อ chanchai เป็นผู้ใช้ระดับ Admin ส่วนผู้ชื่อ weraya เป็นผู้ใช้ ระดับ User จากความสัมพันธ์ของ 2 ตารางนี้มี Cardinality เท่ากับ (1,1) เนื่องจากผู้ใช้แต่ละคนจะมีสถานะในกลุ่มการใช้งานกลุ่มใดกลุ่มหนึ่งเท่านั้น

2.2 การควบคุมความถูกต้องให้กับข้อมูล (Data Integrity)

เพื่อความถูกต้องของข้อมูลภายในฐานข้อมูลจึงได้มีการตั้งกฎเพื่อควบคุมความถูกต้องของข้อมูลขึ้น เรียกว่า Data Integrity หรือ Integrity Rule โดยฐานข้อมูลแต่ละตัวอาจจะมีกฎที่แตกต่างกันไป เช่น ฐานข้อมูลพนักงานก็จะมี Integrity Rule ดังนี้

- พนักงานแต่ละคนจะต้องสังกัดในแผนกงานใดแผนกหนึ่ง
- การระบุเพศของพนักงานให้ใช้สัญลักษณ์ M= เพศชาย และ F=เพศหญิง
- วันเดือนปีเกิดของพนักงานให้จัดเก็บอยู่ในรูปแบบ วัน/เดือน/ปี
- เงินเดือนของพนักงานต้องเป็นค่าบวก จะติดลบไม่ได้

ประเด็นที่สำคัญของ Data Integrity คือ เรื่องของคีย์ (Key) สามารถสรุปลักษณะที่สำคัญของคีย์ในแต่ละประเภทในฐานข้อมูลเชิงสัมพันธ์ ได้ดังนี้

- **Primary Key** จะเป็นฟิลด์ที่มีค่าไม่ซ้ำกันเลยในแต่ละเรคคอร์ดในตารางนั้น เราสามารถใช้ฟิลด์ที่เป็น Primary Key นี้เป็นตัวแทนของตารางนั้นได้ทันที ฟิลด์ที่จะสามารถเลือกให้เป็น Primary Key ได้จะต้องมีคุณสมบัติความเป็นหนึ่งเดียว (Uniqueness) คือ ค่าของฟิลด์นั้นจะต้องไม่ซ้ำกัน จากตาราง user ในรูปที่ 2.1 เราสามารถเลือกให้ ฟิลด์ id เป็น Primary Key ได้ เนื่องจาก รหัสพนักงาน ไม่มีทางซ้ำกันได้ แต่เราจะไม่เลือกฟิลด์ name เป็น Primary Key เนื่องจาก ชื่อพนักงาน อาจซ้ำกันได้ พนักงานอาจจะมีชื่อเหมือนกันได้

- **Candidate Key** คีย์คู่แข่ง เป็นฟิลด์หนึ่งหรือหลายฟิลด์ที่มีคุณสมบัติที่เป็น Primary Key ได้ แต่ไม่ได้เป็นคีย์หลัก เช่น ชื่อและนามสกุล สามารถรวมกันเป็นคีย์คู่แข่งได้ พิจารณาจากตัวอย่างต่อไปนี้

Table: Employee

EmpID	FName	LName	DepID
001	สมชาย	สุขมาก	01
002	จันจิรา	คียิ่ง	02
003	วิษณุ	รักไทย	01
004	สมชาย	ใจดี	03

รูปที่ 2.2 ตาราง Employee เก็บข้อมูลพนักงาน

จากตัวอย่างนี้ เราสามารถกำหนดให้ฟิลด์ EmpID เป็น Candidate Key ได้ หรือกำหนดให้ฟิลด์ FName + LName เป็น Candidate Key ก็ได้ เนื่องจากทั้ง 2 ฟิลด์นี้เมื่อรวมกันแล้วมีคุณสมบัติเป็น Primary Key ที่สามารถเป็นตัวแทนของตารางได้

ในทางปฏิบัติเราสามารถมี Candidate Key ได้หลาย Key ด้วยกัน ดังนั้นจึงต้องมีการเลือก Candidate Key ที่เหมาะสมมาทำหน้าที่เป็น Primary Key ส่วน Candidate Key ที่เหลือจะทำหน้าที่เป็น Alternate Key แทน ดังตัวอย่างข้างต้น ตาราง Employee ประกอบไปด้วย 2 Candidate Key ได้แก่ Candidate Key “EmpID” และ Candidate Key “FName + LName” หลักในการเลือก คือ เลือกเอา Candidate Key ที่เล็กที่สุด และถูกเรียกใช้โดยการกระทำ (Operation) ต่าง ๆ มากที่สุดมาเป็น Primary Key ซึ่งเมื่อพิจารณาจะพบว่า Candidate Key “EmpID” มีขนาดเล็กกว่า Candidate key “Fname + Lname” รวมทั้งเป็น Candidate Key ที่ถูกเรียกใช้โดยการกระทำต่าง ๆ มากกว่า ดังนั้นจึงเลือก Candidate Key “EmpID” มาทำหน้าที่เป็น Primary Key

- **Composite Key** เป็นฟิลด์ที่ใช้รวมกับฟิลด์อื่นๆ ที่เป็น Composite Key เหมือนกัน มาใช้เป็น Primary Key ของตาราง เนื่องจากบางครั้งเราไม่สามารถกำหนดเพียงฟิลด์เดียวให้เป็น Primary Key ได้ ต้องอาศัยฟิลด์ร่วมกัน เช่น ตาราง Order ซึ่งเก็บรายละเอียดการสั่งซื้อ โดยหมายเลขการสั่งซื้อ (ฟิลด์ order_no) 1 หมายเลข สามารถสั่งซื้อสินค้า (ฟิลด์ product_id) ได้หลายรายการ เราไม่สามารถกำหนดให้ฟิลด์ order_no เป็น Primary Key ได้ เนื่องจากเหตุผลที่รหัสการสั่งซื้อ 1 หมายเลขสามารถซื้อสินค้าได้หลายรายการ แต่ถ้าเรากำหนดให้ฟิลด์ order_no และ product_id ประกอบรวมกันเป็น Composite Key ก็จะสามารถใช้เป็น Primary key ของตารางนี้ได้

Table: Order

order_no	product_id	quantity	price
001	01	4	500
001	02	5	100
002	01	10	250
003	03	15	300

รูปที่ 2.3 ตาราง Order เก็บข้อมูลรายการสั่งซื้อ

- **Foreign Key** เป็นฟิลด์ในตารางหนึ่ง ที่มีความสัมพันธ์กับฟิลด์ที่เป็น Primary Key อีกตารางหนึ่ง เช่น ในรูปที่ 2.1 ความสัมพันธ์ของตาราง user และ ตาราง group โดยตาราง user มีฟิลด์ group_id เป็นสมาชิกของฟิลด์หนึ่งในตาราง โดยฟิลด์ group_id นี้ปรากฏอยู่ในตาราง group ในฐานะ Primary Key ดังนั้น เราจะถือว่าฟิลด์ group_id นี้เป็น Foreign Key
- **Null** ในการจัดเก็บข้อมูลในฐานข้อมูล บ่อยครั้งที่ถูกจัดเก็บไม่ครบถ้วน ซึ่งอาจจะมาจากสาเหตุไม่มีข้อมูลนั้นๆ ทำให้ไม่สามารถระบุข้อมูลลงไปได้ เช่น ข้อมูลวันเดือนปีเกิดของประชากร อาจจะไม่ทราบวันเดือนปี หรืออาจจะทราบเพียงแต่ปีเกิดเท่านั้น เป็นต้น แต่เนื่องจากทุก Attribute ในฐานข้อมูลของ Relational Database จะต้องมีค่าของข้อมูล ดังนั้นเพื่อแก้ปัญหาดังกล่าวจึงได้มีการกำหนดค่าของข้อมูลที่ ไม่ทราบค่าขึ้น เรียกว่า ค่า Null นั่นเอง

พิจารณาตัวอย่างตาราง Emp ซึ่งเก็บข้อมูลพนักงาน ต่อไปนี้

Table: Emp

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7369	SOMPONG	CLERK	7902	17 DEC 1980	800	NULL	20
7499	AEK	SALESMAN	7902	20 FEB 1981	1600	300	30
7521	WILAI	SALESMAN	7902	22 FEB 1981	1250	500	30
7902	KING	PRESIDENT	NULL	17 NOV 1981	5000	NULL	10

รูปที่ 2.4 ตาราง Emp เก็บข้อมูลพนักงาน และแสดงค่า Null ในฟิลด์ Mgr และ Comm

จากตารางข้างต้นมีข้อมูลที่ปรากฏค่า Null อยู่ 2 ฟิลด์ คือ

1. ฟิลด์ Mgr (เก็บหมายเลขพนักงานที่เป็นหัวหน้า) พนักงานชื่อว่า KING เก็บค่าเป็น Null ที่ฟิลด์นี้ เนื่องจากพนักงานชื่อว่า KING เป็นประธานบริษัท ดังนั้นจึงไม่มีใครเป็นหัวหน้าของเขา
2. ฟิลด์ Comm (เก็บค่าคอมมิชชั่นจากการขายสินค้า) พนักงานชื่อว่า SOMPONG และ KING เก็บค่าเป็น Null ที่ฟิลด์นี้ เนื่องจากพนักงานทั้ง 2 คน ไม่ได้ทำงานในตำแหน่ง SALESMAN เช่นเดียวกับพนักงานที่ชื่อว่า AEK และ WILAI ดังนั้นพนักงานชื่อ SOMPONG และ KING จึงไม่มีค่าคอมมิชชั่นจากการขายสินค้า

หมายเหตุ ค่า Null เป็นค่าว่างที่ไม่มีค่า ไม่มีข้อมูล ซึ่งต่างจากค่า 0 (ศูนย์) หรือ ช่องว่าง โดยค่า 0 (ศูนย์) หรือ ช่องว่าง เป็นข้อมูลที่มีค่าอยู่ คือ ค่า 0 (ศูนย์) หรือ ช่องว่าง นั่นเอง

เมื่อเราทราบถึงความหมาย และลักษณะของคีย์ (Key) ในแต่ละประเภทแล้ว ลำดับถัดไป จะได้กล่าวถึง Integrity Rule ซึ่งมีด้วยกัน 2 ข้อ โดยทุกฐานข้อมูลจะมีเหมือนกันแต่ต่างกันที่รายละเอียดเท่านั้น คือ กฎความคงสภาพของ Entity (Entity Integrity Rule) และ กฎความคงสภาพของการอ้างอิง (Referential Integrity Rule)

• กฎความคงสภาพของ Entity (Entity Integrity Rule)

กล่าวว่า จะต้องไม่มี Attribute ใดที่ประกอบขึ้นเป็นคีย์หลักของ Relation มีข้อมูลที่เป็นค่าว่าง (Null Value) หรือพูดง่าย ๆ ก็คือ “ค่าของ Primary Key จะเป็น Null ไม่ได้”

Table: Employee

EmpID	FName	LName	DepID
001	สมชาย	สุขมาก	01
002	จันจิรา	ดียิ่ง	02
	วิษณุ	รักไทย	01
	สมชาย	ใจดี	03

รูปที่ 2.5 ตาราง Employee ที่ผิดกฎ Entity Integrity Rule

จากตารางในรูปที่ 2.5 ผิดกฎ Entity Integrity Rule เนื่องจากเกิดค่าว่างใน Attribute ชื่อว่า EmpID ซึ่งในที่นี้ถือว่าเป็น Primary Key

• กฎความคงสภาพของการอ้างอิง (Referential Integrity Rule)

กล่าวว่า ถ้า Relation ใดมี Attribute ที่เป็น Foreign Key อยู่ ข้อมูลที่เป็น Foreign Key นั้นจะต้องเป็นข้อมูลที่มีอยู่ใน Primary Key ของอีก Relation หรือ ถ้าไม่แล้วข้อมูลที่เป็น Foreign key จะต้องเป็นค่าว่าง พิจารณาตัวอย่างที่ผิดกฎ Referential Integrity Rule ต่อไปนี้

Table: user

id	name	password	group_id
0	chanchai	chan123	0
1	weraya	wera456	1
2	yongyut	Yong789	2

Table: group

group_id	name
0	Admin
1	User

รูปที่ 2.6 ตัวอย่างที่ผิดกฎ Referential Integrity Rule

จากรูปที่ 2.6 ผิดกฎ Referential Integrity Rule เนื่องจากใน Relation ผู้ใช้ (user) กลุ่มผู้ใช้งานหมายเลข 2 ไม่มีใน Relation กลุ่มผู้ใช้ (group)

ในประเด็นของ Referential Integrity Rule นี้ในการแก้ไขหรือลบ Primary Key ใน Relation หนึ่งจะมีผลกระทบถึง Foreign key ในอีก Relation หนึ่ง ดังนั้นเพื่อให้คงความถูกต้องของ Referential Integrity Rule นี้จึงกำหนดวิธีการจัดการไว้ 3 วิธี คือ

1) นร่ทำแบบต่อเนื่อง (Cascade)

คือ ถ้ามีการลบ หรือ เปลี่ยนแปลงข้อมูลที่เป็น Primary key ที่มีการอ้างอิงแล้วจะต้องมีการลบหรือเปลี่ยนแปลงข้อมูลใน Relation ที่มีการอ้างอิงนั้นด้วย เช่นจากตารางในรูปที่ 2.6 ถ้ามีการแก้ไขข้อมูลในฟิวด์ group_id จาก Relation ชื่อว่า group จาก 1 เป็น 3 แล้ว ใน Relation ชื่อว่า user ก็จะต้องมีการแก้ไขข้อมูลของผู้ใช้ที่ชื่อว่า weraya จากฟิวด์ group_id เท่ากับ 1 ให้เป็น 3 ด้วย

2) ระบุว่ามีข้อจำกัด (Restricted)

คือ DBMS จะอนุญาตให้ลบหรือเปลี่ยนแปลงข้อมูลที่เป็น Primary key ก็ต่อเมื่อ Primary key นั้น ไม่ถูกอ้างอิงใน Relation อื่นใด มิฉะนั้นจะปฏิเสธการลบหรือเปลี่ยนแปลง เช่น จากตารางในรูปที่ 2.6 ถ้าจะทำการลบ หรือเปลี่ยนแปลงค่าในฟิลด์ group_id ของ Relation ที่ชื่อว่า group จะถูกปฏิเสธการกระทำดังกล่าว เนื่องจากฟิลด์ group_id นี้เป็น Primary Key และถูกอ้างอิงไปใช้กับ Relation ที่ชื่อว่า user

3) ระบุโดยการใส่ค่าว่าง (Nullifies)

คือ DBMS จะยอมให้มีการลบหรือเปลี่ยนแปลงข้อมูลที่เป็น Primary key ได้ แต่ต้องมีการใส่ค่าว่าง (Null Value) ลงไปยังข้อมูลของ Attribute ที่เป็น Foreign key ที่มีการอ้างอิงถึงข้อมูลที่ถูกลบหรือเปลี่ยนแปลง เช่น จากตารางในรูปที่ 2.6 ถ้าทำการลบ Record ใน relation ที่ชื่อว่า group ที่มีค่าของ group_id เท่ากับ 1 ออกไป ดังนั้นใน Relation ที่ชื่อว่า user จะต้องมีการใส่ค่าว่างในฟิลด์ group_id ของ Record ผู้ใช้ที่ชื่อว่า weraya

สรุป

ฐานข้อมูลเชิงสัมพันธ์ เป็นการจัดเก็บข้อมูลในรูปแบบของตาราง (Table) ที่มีลักษณะเป็นสองมิติ คือ แถว (Row) และคอลัมน์ (Column) 1 ฐานข้อมูล (Database) สามารถประกอบขึ้นจาก ตาราง (Table) ตั้งแต่ 1 ตาราง ขึ้นไป ฐานข้อมูลเชิงสัมพันธ์จะเรียกตารางนี้ว่า Relation ตาราง 1 ตารางประกอบขึ้นจากกลุ่มของข้อมูลเรื่องใดเรื่องหนึ่ง ซึ่งเรียกว่า ฟิลด์ (Field) ในฐานข้อมูลเชิงสัมพันธ์จะเรียกฟิลด์นี้ว่า Attribute จำนวนฟิลด์ หรือ Attribute จะเรียกว่า Degree และข้อมูลในแต่ละชุด หรือแต่ละแถว จะเรียกว่า เรคคอร์ด (Record) ในฐานข้อมูลเชิงสัมพันธ์ จะเรียกเรคคอร์ดนี้ว่า Tuple จำนวน Record หรือ Tuple ในตารางหนึ่งที่มีความสัมพันธ์กับ Tuple ในอีกตารางหนึ่ง เรียกว่า Cardinality ในการเชื่อมโยงกันระหว่างข้อมูลในตาราง 2 ตาราง หรือมากกว่า จะเชื่อมโยงโดยใช้ฟิลด์ที่มีอยู่ในตารางที่ต้องการเชื่อมโยงข้อมูลกัน

เพื่อความถูกต้องของข้อมูลภายในฐานข้อมูล จึงได้มีการตั้งกฎเพื่อควบคุมความถูกต้องของข้อมูลขึ้น เรียกว่า Data Integrity หรือ Integrity Rule โดยฐานข้อมูลแต่ละตัวอาจจะมีกฎที่แตกต่างกันไป ประเด็นที่สำคัญของ Data Integrity คือ เรื่องของคีย์ (Primary Key, Candidate Key, Composite Key, Foreign Key และ Null)

Integrity Rule มีกฎอยู่ 2 ข้อ โดยทุกฐานข้อมูลจะมีเหมือนกันแต่ต่างกันที่รายละเอียดเท่านั้น คือ กฎความคงสภาพของ Entity (Entity Integrity Rule) และ กฎความคงสภาพของการอ้างอิง (Referential Integrity Rule) โดย Entity Integrity Rule กล่าวว่า จะต้องไม่มี Attribute ใดที่ประกอบขึ้นเป็นคีย์หลักของ Relation มีข้อมูลที่เป็นค่าว่าง (Null Value) และ Referential Integrity Rule กล่าวว่า ถ้า Relation ใดมี Attribute ที่เป็น Foreign Key อยู่ ข้อมูลที่เป็น Foreign Key นั้นจะต้องเป็นข้อมูลที่มีอยู่ใน Primary Key ของอีก Relation

เพื่อให้คงความถูกต้องของ Referential Integrity Rule นี้จึงกำหนดวิธีการจัดการไว้ 3 วิธี คือ Cascade, Restricted และ Nullifies

กิจกรรมท้ายบทที่ 2

1. พิจารณาข้อความในแต่ละข้อต่อไปนี้ ข้อใดถูก หรือผิด โดยถ้าข้อใดผิดให้แก้ไขให้ถูกต้องด้วย

1. ใน 1 ตาราง (Table) สามารถประกอบขึ้นจาก ฐานข้อมูล (Database) ตั้งแต่ 1 ฐานข้อมูลขึ้นไป
2. ข้อมูลในแต่ละชุด หรือแต่ละแถว จะเรียกว่า เรคคอร์ด (Record) หรือ Attribute
3. Primary Key จะเป็นฟิลด์ที่มีค่าไม่ซ้ำกันเลยในแต่ละเรคคอร์ดในตารางนั้น เราสามารถใช้ฟิลด์ที่เป็น Primary Key นี้เป็นตัวแทนของตารางนั้นได้ทันที
4. Composite Key คือ คีย์คู่แข่ง เป็นฟิลด์หนึ่งหรือหลายฟิลด์ที่มีคุณสมบัติที่เป็น Primary Key ได้ แต่ไม่ได้เป็นคีย์หลัก
5. ถ้าเราระบุค่าข้อมูลหนึ่งเป็น 0 (ศูนย์) แสดงว่าข้อมูลนั้นมีค่าเป็น Null
6. Cardinality คือ จำนวนฟิลด์ หรือ Attribute
7. Foreign Key เป็นฟิลด์ในตารางหนึ่ง ที่มีความสัมพันธ์กับฟิลด์ที่เป็น Primary Key อีกตารางหนึ่ง
8. ตารางข้อมูลพนักงานเราสามารถกำหนดฟิลด์ชื่อของพนักงานเป็น Primary Key ได้

2. พิจารณาตารางข้อมูลฐานข้อมูลต่อไปนี้

Table: Product

product_id	product_name
01	A
02	B
03	C

Table: Order_Detail

order_no	product_id	quantity	price
001	01	4	500
001	02	5	100
002	01	10	250
003	03	15	300

- 2.1) ให้พิจารณาว่าฟิลด์ใดจากทั้ง 2 ตารางควรจะถูกเลือกให้เป็น Primary Key
- 2.2) ให้พิจารณาว่าฟิลด์ใดจากทั้ง 2 ตารางควรจะถูกเลือกให้เป็น Foreign Key
- 2.3) ให้พิจารณาว่าฟิลด์ใดจากทั้ง 2 ตารางควรจะถูกเลือกให้เป็น Composite Key

3. อธิบายลักษณะของกฎความคงสภาพของ Entity (Entity Integrity Rule) และ กฎความคงสภาพของการอ้างอิง (Referential Integrity Rule) รวมทั้งยกตัวอย่างประกอบด้วย

4. พิจารณาทารางฐานข้อมูลต่อไปนี้

Table: user

id	name	password	group_id
0	chanchai	chan123	0
1	weraya	wera456	1
2	yongyut	Yong789	2

Table: group

group_id	name
0	Admin
1	User

- 4.1) ให้บอกวิธีจัดการแบบ Cascade สำหรับตารางฐานข้อมูลข้างต้นเพื่อให้คงความถูกต้องของ Referential Integrity Rule
- 4.2) ให้บอกวิธีจัดการแบบ Restricted สำหรับตารางฐานข้อมูลข้างต้นเพื่อให้คงความถูกต้องของ Referential Integrity Rule
- 4.3) ให้บอกวิธีจัดการแบบ Nullifies สำหรับตารางฐานข้อมูลข้างต้นเพื่อให้คงความถูกต้องของ Referential Integrity Rule

บทที่

3

การแสดงความสัมพันธ์ของข้อมูลด้วย ER Model

สาระสำคัญ

Entity Relationship Model (ER Model) เป็นเครื่องมือที่ใช้สำหรับการออกแบบฐานข้อมูลในระดับแนวคิด (Conceptual Design) โดยใช้ในการวาดแสดงความสัมพันธ์ของข้อมูลต่างๆภายในฐานข้อมูล ER Model พัฒนาโดย Peter Pin Shan Chen จาก Massachusetts Institute of Technology (MIT) ในปี ค.ศ. 1976 เนื้อหาในบทนี้จะเริ่มจากการให้ความหมายของสัญลักษณ์ต่างๆที่ใช้ใน ER Model ต่อจากนั้นจะแสดงตัวอย่างการสร้าง ER Model และในที่สุดท้ายจะเป็นการเปลี่ยน ER Model ให้เป็นตาราง

จุดประสงค์ทั่วไป

- 1. เพื่อให้ทราบความหมาย และรู้จักสัญลักษณ์ต่างๆ ของ ER Model
- 2. เพื่อให้ทราบแต่ละประเภทของ Entity
- 3. เพื่อให้ทราบแต่ละประเภทของ Attributes
- 4. เพื่อให้ทราบแต่ละประเภทของ Relationship
- 5. เพื่อให้เข้าใจกระบวนการสร้าง ER Diagram
- 6. เพื่อให้ทราบวิธีการแปลง ER Diagram เป็นตาราง

จุดประสงค์เชิงพฤติกรรม

- 1. สามารถวาด ER Diagram โดยใช้สัญลักษณ์ต่างๆ ได้อย่างถูกต้อง
- 2. สามารถแปลง ER Diagram ไปเป็นตารางได้อย่างถูกต้อง

เนื้อหาสาระ:

- 1. ความหมายของสัญลักษณ์ต่างๆของ ER Model
- 2. วิธีการสร้าง ER Diagram
- 3. การแปลง ER Diagram เป็นตาราง

3.1 ความหมายของสัญลักษณ์ต่างๆของ ER Model

Entity คือ สิ่งที่มีอยู่จริง จับต้องได้ ใช้สำหรับเก็บข้อมูลภายในระบบ เช่น บุคคล สินค้า แผนกงาน เป็นต้น จะใช้สัญลักษณ์รูปสี่เหลี่ยม และมีชื่อ Entity อยู่ภายใน



รูปที่ 3.1 แสดงตัวอย่างของ Entity ลูกค้า

Entity แบ่งออกได้เป็น 2 ประเภท คือ

1. **Regular Entity** คือ Entity ทั่วๆ ไปที่ตัว Entity เองมีคุณสมบัติที่สามารถกำหนดให้เป็นเอกลักษณ์เฉพาะกับ Entity ได้ ไม่ต้องอาศัย Entity อื่นในการคงอยู่ ตัวอย่างเช่น Entity Student ต่อไปนี้

SID	Sname	Faculty	GPA
5214768	นายสามารถ รักดี	วิทยาศาสตร์	3.45
5227891	นายพิษณุ ใจกล้า	บริหารธุรกิจ	2.89
5236734	นายสมพงษ์ รักถิ่น	วิศวกรรมศาสตร์	2.50



รูปที่ 3.2 แสดงตัวอย่างของ Entity Student

2. **Weak Entity** คือ Entity อ่อนแอ ที่ไม่สามารถคงอยู่ได้ ต้องอาศัย Entity อื่นในการคงอยู่ ตัวอย่างเช่น Entity TimeStamp ต่อไปนี้

EMP_Id	Date	Time_In	Time_Out
111	13/04/2010	8.00	16.30
222	13/04/2010	8.45	17.00
333	13/04/2010	8.30	16.35

รูปที่ 3.3 ตัวอย่างตารางข้อมูลการลงเวลาของพนักงาน