การออกแบบ
ฐานข้อมูลทางการศึกษาเบื้องต้น

Introduction to Educational Database Design

อาจารย์ ดร.วีระพันธ์ พานิชย์

ภาควิชานวัตกรรมและเทคโนโลยีการศึกษา

คณะศึกษาศาสตร์ มหาวิทยาลัยบูรพา

การออกแบบฐานข้อมูลทางการศึกษาเบื้องต้น
Introduction to Educational Database Design

โดย

ดร.วีระพันธ์ พานิชย์

ภาควิชานวัตกรรมและเทคโนโลยีการศึกษา

คณะศึกษาศาสตร์ มหาวิทยาลัยบูรพา

คำนำ

หนังสือเล่มนี้ได้เขียนขึ้นจากการได้ดูตัวอย่างแนวปฏิบัติที่ดี (Best Practice) และประสบการณ์ในการจัดการเรียนการสอน รวมถึงจากการศึกษาสภาพจริงการนำระบบฐานข้อมูลประยุกต์ใช้ด้านการศึกษา เช่น เป็นเครื่องมือสำหรับจัดเก็บข้อมูลผู้เรียน ผู้สอน การนำระบบฐานข้อมูลเข้ามาสนับสนุนการบริหารจัดการสถานศึกษา ความสำคัญฐานข้อมูลส่งผลให้ระบบฐานข้อมูลถูกพัฒนาขึ้นอย่างต่อเนื่อง โดยข้อมูลที่ถูกจัดเก็บในฐานข้อมูลมีความเชื่อถือได้ และมีความปลอดภัย

หนังสือเล่มนี้แบ่งเนื้อหาออกเป็น 9 บท ประกอบด้วย 1) ความรู้เบื้องต้นเกี่ยวกับฐานข้อมูล 2) การพัฒนาฐานข้อมูล 3) แบบจำลองฐานข้อมูล 4) แบบจำลองฐานข้อมูลเชิงสัมพันธ์ 5) แบบจำลอง อี-อาร์ 6) ภาษา SQL เบื้องต้น 7) ระบบจัดการฐานข้อมูล MySQL และโปรแกรม phpMyAdmin 8) การเขียนโปรแกรมภาษา PHP เชื่อมระบบฐานข้อมูล MySQL) 9) ความปลอดภัยของฐานข้อมูล การจัดทำหนังสือเล่มนี้ ผู้จัดทำได้ศึกษาค้นคว้าข้อมูลจาก หนังสือ ตำรา บทความวิชาการ บทความวิจัย งานวิจัย และข้อมูลจากเวปไซต์ที่เกี่ยวข้อง ผู้เขียนขอขอบพระคุณเจ้าของงานเขียนดังกล่าวทุกท่านมา ณ โอกาสนี้

ผู้เขียนขอขอบคุณต้นแบบแนวปฏิบัติที่ดีทุกคนที่ทำให้ผู้เขียนเกิดการเรียนรู้ และผลักดันจนทำให้หนังสือเล่มนี้เสร็จสมบูรณ์ และหวังว่าหนังสือเล่มนี้จะก่อให้เกิดประโยชน์ต่อนักเทคโนโลยีการศึกษา ครู อาจารย์ นิสิต นักศึกษา ตลอดจนผู้ที่สนใจ ทั้งนี้หากมีข้อผิดพลาดประการใด ผู้เขียนขอน้อมรับและขอผู้อ่านทุกท่านโปรดให้คำแนะนำแก่ผู้เขียนเพื่อจะได้ปรับปรุงแก้ไขให้เกิดความสมบูรณ์ยิ่งขึ้นในโอกาสต่อไป

ดร.วีระพันธ์ พานิชย์

กรกฎาคม 2565

สารบัญ

เรื่อง **หน้า**

คำนำ..... ข
สารบัญ..... ค
สารบัญตาราง..... ฉ
สารบัญภาพ..... ช

บทที่

1 ความรู้เบื้องต้นเกี่ยวกับฐานข้อมูล..... 1
ข้อมูล..... 2
ชนิดของข้อมูล..... 2
การแบ่งประเภทของข้อมูล..... 3
ระบบแฟ้มข้อมูล..... 6
ระบบฐานข้อมูล 11
สรุป..... 17
2 การพัฒนาฐานข้อมูล..... 19
ระบบสารสนเทศ..... 20
วงจรการพัฒนาระบบ..... 22
กระบวนการพัฒนาฐานข้อมูล..... 24
วัฏจักรฐานข้อมูล..... 26
สรุป..... 37
3 แบบจำลองฐานข้อมูล..... 39
องค์ประกอบของแบบจำลองฐานข้อมูล..... 40
รูปแบบความสัมพันธ์สัมพันธ์ระหว่าง Entity..... 42
ประเภทของแบบจำลองฐานข้อมูล..... 45
ชนิดแบบจำลองฐานข้อมูล..... 46
คุณลักษณะพื้นฐานสำคัญของแบบจำลองฐานข้อมูล..... 54
สรุป..... 55
4 แบบจำลองฐานข้อมูลเชิงสัมพันธ์..... 57
ฐานข้อมูลเชิงสัมพันธ์..... 58
โครงสร้างฐานข้อมูลเชิงสัมพันธ์..... 59
คุณสมบัติของ Relation..... 60
ลักษณะของข้อมูลที่จัดเก็บ..... 61
Relation Key..... 62
Properties of Relations 66
กฎ 12 ข้อ ของ E.F Codd..... 68
กฎความคงสภาพข้อมูล..... 69

สารบัญ (ต่อ)

เรื่อง **หน้า**

บทที่

วิว (view)..... 71
สรุป..... 73
5 แบบจำลอง E-R..... 75
องค์ประกอบแบบจำลอง E-R..... 76
Entity 77
ความสัมพันธ์ระหว่าง Entity 80
ชนิดความสัมพันธ์ระหว่างข้อมูล..... 81
การออกแบบฐานข้อมูลโดยใช้ E-R โมเดล..... 88
นอร์มอลฟอร์ม (Normal Forms)..... 89
สรุป..... 93
6 ภาษา SQL เบื้องต้น..... 95
ภาษา SQL 96
องค์ประกอบของภาษา SQL 97
ภาษาสำหรับการนิยามข้อมูล 97
ภาษาสำหรับการจัดการข้อมูล..... 100
ภาษาสำหรับการควบคุมข้อมูล..... 102
หลักการในการแปลงแบบจำลอง E-R มาเป็นตารางโดยใช้ภาษา SQL..... 103
ตัวอย่างการแปลงแบบจำลอง E-R มาเป็นตารางโดยใช้ภาษา SQL..... 103
สรุป..... 108
7 ระบบจัดการฐานข้อมูล MySQL และโปรแกรม phpMyAdmin 109
ระบบจัดการฐานข้อมูล 110
ระบบจัดการฐานข้อมูลมายเอสคิวแอล (MySQL) 116
โปรแกรม phpMyAdmin 117
ตัวอย่าง การใช้โปรแกรม phpMyAdmin สำหรับสร้างฐานข้อมูล..... 122
สรุป..... 126
8 การเขียนโปรแกรมภาษา PHP เชื่อมระบบฐานข้อมูล MySQL 127
ประวัติความเป็นมาของภาษา PHP..... 128
หลักการทำงานของภาษา PHP..... 128
การสร้างเว็บเซิร์ฟเวอร์ (Web Server) บนคอมพิวเตอร์..... 130
ภาษา PHP เบื้องต้น..... 132
การเขียนโปรแกรม PHP เข้าถึงฐานข้อมูล MySQL..... 140
สรุป..... 152

สารบัญ (ต่อ)

เรื่อง **หน้า**

บทที่

9 **ความปลอดภัยของฐานข้อมูล..... 153**
ความหมายความปลอดภัยของฐานข้อมูล..... 154
การสร้างความเสียหาย (Threats)..... 155
ประเภทของภัยระบบฐานข้อมูล..... 156
วัตถุประสงค์การรักษาความปลอดภัย..... 157
การควบคุมความปลอดภัยฐานข้อมูล..... 157
การควบคุมความปลอดภัยของฐานข้อมูลด้วยวิว (VIEW)..... 161
ข้อควรคำนึงในการรักษาความปลอดภัยฐานข้อมูล..... 164
ตัวอย่างนโยบายและแนวปฏิบัติการรักษาความมั่นคงปลอดภัยด้าน
สารสนเทศ มหาวิทยาลัยบูรพา..... 166
สรุป..... 195

บรรณานุกรม..... **197**
Index..... **201**
ประวัติผู้เขียน..... **205**

สารบัญตาราง

ตารางที่ **หน้า**

2.1 แสดงคุณสมบัติที่สำคัญของข้อมูลที่มีคุณภาพ..... 25
6.1 คำสั่งต่าง ๆ ในภาษาสำหรับนิยามข้อมูล..... 97
6.2 ชนิดข้อมูล..... 99
7.1 โครงสร้างตาราง regstudent 124

สารบัญภาพ

ภาพที่ **หน้า**

1.1 ความสัมพันธ์ขององค์ประกอบในโครงสร้างแฟ้มข้อมูล..... 8
1.2 การค้นหาข้อมูลในแฟ้มข้อมูลแบบเรียงลำดับ..... 9
1.3 การค้นหาข้อมูลในแฟ้มข้อมูลแบบสุ่ม..... 9
1.4 การค้นหาข้อมูลในแฟ้มข้อมูลแบบสุ่ม..... 10
1.5 ระบบฐานข้อมูล..... 11
1.6 ตัวอย่างฐานข้อมูลนักศึกษา..... 12
1.7 ระบบจัดการฐานข้อมูล (Database Management System: DBMS)..... 14
1.8 องค์ประกอบระบบจัดการฐานข้อมูล..... 15
2.1 ระบบเทคโนโลยีสารสนเทศเพื่อการตัดสินใจ..... 22
2.2 วงจรการพัฒนาระบบ..... 23
2.3 วัฏจักรฐานข้อมูล..... 27
2.4 ขั้นตอนการออกแบบฐานข้อมูล..... 29
2.5 ตัวอย่างแบบจำลอง E-R ระบบการจัดการของสถาบันการศึกษา..... 31
2.6 การออกแบบฐานข้อมูลเชิงตรรกะ..... 33
2.7 ตัวอย่างระบบ Database virtualization..... 34
3.1 ลักษณะของ Entity และ Attribute..... 40
3.2 Simple Attribute และ Composite Attribute..... 41
3.3 ความสัมพันธ์ระหว่าง Entity (Entity Relationship)..... 42
3.4 แสดงสัญลักษณ์เส้นความสัมพันธ์ระหว่าง Entity..... 42
3.5 แสดงสัญลักษณ์เส้นความสัมพันธ์แบบหนึ่งต่อหนึ่ง..... 43
3.6 แสดงตัวอย่างความสัมพันธ์แบบหนึ่งต่อหนึ่ง..... 43
3.7 แสดงความสัมพันธ์แบบหนึ่งต่อกลุ่ม..... 44
3.8 แสดงตัวอย่างความสัมพันธ์แบบหนึ่งต่อกลุ่ม..... 44
3.9 แสดงตัวอย่างความสัมพันธ์แบบกลุ่มต่อหนึ่ง..... 44
3.10 แสดงความสัมพันธ์แบบกลุ่มต่อกลุ่ม..... 45
3.11 แสดงตัวอย่างความสัมพันธ์แบบกลุ่มต่อกลุ่ม..... 45
3.12 ตัวอย่างแบบจำลองฐานข้อมูลแบบลำดับชั้น..... 46
3.13 ตัวอย่างแบบจำลองฐานข้อมูลแบบเครือข่าย..... 50
3.14 ตัวอย่าง Relation ในแบบจำลองฐานข้อมูลเชิงสัมพันธ์..... 51
3.15 ความสัมพันธ์ของแบบจำลองฐานข้อมูลเชิงสัมพันธ์..... 52
3.16 ความสัมพันธ์ของแบบจำลองฐานข้อมูลเชิงวัตถุ..... 53
3.17 ความสัมพันธ์ของแบบจำลองฐานข้อมูลมัลติไดเมนชั่น..... 54

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

4.1 ส่วนประกอบภายในตารางของฐานข้อมูลเชิงสัมพันธ์..... 58
4.2 ตัวอย่างโครงสร้างฐานข้อมูลเชิงสัมพันธ์..... 59
4.3 แสดงตัวอย่างข้อมูลที่ไม่อิสระจาก Relation อื่น..... 60
4.4 แสดงตัวอย่าง Relation “Student”..... 61
4.5 ข้อมูลซ้ำซ้อนกันใน Relation 61
4.6 Attribute ที่มีค่ามากกว่า 1 ค่า 62
4.7 แสดงตัวอย่าง Super Key..... 63
4.8 แสดงค่าว่าง (NULL)..... 64
4.9 การกำหนด Primary Key..... 64
4.10 การกำหนดคีย์ร่วม 65
4.11 การกำหนดคีย์คู่แข่ง..... 65
4.12 แสดงตัวอย่างคีย์นอก..... 66
4.13 แสดงตัวอย่างคีย์สำรอง..... 66
4.14 แสดงตัวอย่าง Relation Student..... 67
5.1 องค์ประกอบของแบบจำลอง E-R..... 76
5.2 สัญลักษณ์แทนความหมายแบบจำลอง E-R..... 76
5.3 สัญลักษณ์ Entity ปกติ และ Entity อ่อนแอ..... 77
5.4 ตัวอย่าง Entity ปกติ และ Entity อ่อนแอ..... 78
5.5 ตัวอย่าง คอมโพสิต Entity 78
5.6 ตัวอย่าง Simple Attribute 79
5.7 ตัวอย่าง Composite Attribute 79
5.8 แสดง Attribute ชนิดต่าง ๆ 80
5.9 แสดงความสัมพันธ์ระหว่าง Entity หรือระหว่างข้อมูล 81
5.10 แสดงตัวอย่างความสัมพันธ์ของข้อมูล 81
5.11 แสดงความสัมพันธ์หนึ่งต่อหนึ่ง หรือแบบ 1:1..... 82
5.12 ความสัมพันธ์ระหว่างข้อมูลนักศึกษากับรหัสนักศึกษา แบบ 1:1..... 82
5.13 ความสัมพันธ์ระหว่างคณบดีกับคณะวิชา แบบ 1:1 82
5.14 แสดงตัวอย่างความสัมพันธ์แบบหนึ่งต่อหนึ่ง หรือ 1 : 1..... 83
5.15 แสดงความสัมพันธ์แบบหนึ่งต่อกลุ่ม หรือ 1 : M..... 83
5.16 แสดงความสัมพันธ์ระหว่างแม่กับลูกแบบ 1 : M..... 83
5.17 แสดงความสัมพันธ์ระหว่างอาจารย์ที่ปรึกษากับนักศึกษา..... 84
5.18 แสดงความสัมพันธ์แบบกลุ่มต่อกลุ่ม หรือ แบบ M:N 84

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

5.19 ความสัมพันธ์ระหว่างนักศึกษากับรายวิชาที่เปิดสอน..... 85
5.20 แสดงความสัมพันธ์แบบกลุ่มต่อกลุ่ม หรือ M : N..... 85
5.21 ความสัมพันธ์แบบยูนารี 85
5.22 แสดงความสัมพันธ์แบบไบนารี..... 86
5.23 แสดงตัวอย่างความสัมพันธ์แบบไบนารี..... 86
5.24 แสดงความสัมพันธ์แบบสามทาง..... 86
5.25 แสดงตัวอย่างความสัมพันธ์แบบเทอนารี ตารางเรียน..... 87
5.26 แสดงความสัมพันธ์แบบควาเทอร์นารี..... 87
5.27 แสดงตัวอย่างความสัมพันธ์แบบควาเทอร์นารี..... 88
6.1 แสดงองค์ประกอบของภาษา SQL..... 97
6.2 แบบจำลองแบบแบบ One to One และมีความสัมพันธ์แบบบังคับทั้งสองข้าง..... 104
6.3 แบบจำลองแบบ One to One และมีความสัมพันธ์แบบข้างหนึ่งเป็นแบบเลือกได้ และอีกข้างหนึ่งเป็นแบบบังคับ..... 105
6.4 แบบจำลองแบบ One to One และมีความสัมพันธ์แบบบังคับทั้งสองข้าง..... 106
6.5 แบบจำลองแบบ แบบ Many to Many และมีความสัมพันธ์แบบทั้งสองข้างแบบเลือก..... 107
7.1 ระบบการจัดการฐานข้อมูล..... 111
7.2 แสดงการใช้ข้อมูลจากระบบฐานข้อมูลร่วมกัน..... 115
7.3 เว็บไซต์ http://www.appserv.org..... 118
7.4 ขั้นตอนที่ 1 การติดตั้งโปรแกรม AppServ 118
7.5 แสดงรายละเอียดเงื่อนไขการ GNU License..... 119
7.6 เลือกปลายทางการติดตั้งโปรแกรม AppServ..... 119
7.7 เลือก Package Components ที่ต้องการติดตั้ง..... 120
7.8 แสดงการกำหนดค่าคอนฟิกค่า Apache Web Server..... 120
7.9 แสดงการกำหนดค่าคอนฟิกของ MySQL Database..... 121
7.10 แสดงหน้าจอขั้นตอนสิ้นสุดการติดตั้งโปรแกรม AppServ..... 122
7.11 แสดงหน้าจอล๊อคอินเข้าโปรแกรม phpMyAdmin..... 122
7.12 แสดงหน้าจอหลักโปรแกรม phpMyAdmin..... 123
7.13 แสดงขั้นตอนการตั้งชื่อฐานข้อมูล..... 123
7.14 แสดงขั้นตอนการตั้งชื่อตารางและกำหนดจำนวน Columns..... 124
7.15 แสดงตั้งชื่อ Columns ตั้งค่าชนิดของข้อมูล และการกำหนดคีย์หลัก..... 125
7.16 การเพิ่มข้อมูลเข้าฐานข้อมูลด้วยโปรแกรม phpMyAdmin..... 125
7.17 แสดงข้อมูลในตาราง regstudent..... 125

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

8.1 รูปแบบการติดต่อสื่อสารระหว่างเว็บไคลเอ็นท์และเว็บเซิร์ฟเวอร์ผ่านโปรโตคอล HTTP ... 129
8.2 รูปแบบการติดต่อสื่อสารระหว่างเว็บไคลเอ็นท์และเว็บเซิร์ฟเวอร์เพื่อเรียกเว็บเพจธรรมดา ... 129
8.3 กระบวนการประมวลผลภาษา PHP ... 130
8.4 การตรวจสอบ Folder AppServ ... 131
8.5 หน้าเวป localhost ... 131
8.6 หน้าเวปล๊อคอินเข้า phpMyAdmin ... 131
8.7 โปรแกรม phpMyAdmin ... 132
8.8 ผลการเขียน PHP ในไฟล์ HTML ... 133
8.9 ผลการเขียน PHP การกำหนดตัวแปร ... 134
8.10 ผลการเขียน PHP ใช้ตัวดำเนินการ ... 136
8.11 ตัวอย่างผลของการใช้คำสั่ง if/else ... 137
8.12 ตัวอย่างผลของใช้คำสั่ง switch/case ... 138
8.13 ตัวอย่างผลของใช้คำสั่ง while ... 139
8.14 ตัวอย่างผลของใช้คำสั่ง do/while ... 140
8.15 โค้ดภาษา PHP สร้างฐานข้อมูลและตาราง ... 141
8.16 แสดงผลการสร้างฐานข้อมูลและตาราง ... 142
8.17 แสดงงฐานข้อมูลและตาราง ... 142
8.18 โค้ดภาษา HTML สร้างหน้าเว็บหลัก ... 143
8.19 แสดงหน้าเว็บหลัก ... 143
8.20 โค้ดภาษา PHP สร้างโปรแกรมเพิ่มข้อมูลเข้าฐานข้อมูล ... 144
8.21 หน้าเว็บโปรแกรมเพิ่มข้อมูลเข้าฐานข้อมูล ... 145
8.22 หน้าเว็บแสดงผลการเพิ่มข้อมูลเข้าฐานข้อมูล ... 145
8.23 แสดงข้อมูลในโปรแกรม phpMyAdmin ... 145
8.24 โค้ดภาษา PHP สร้างโปรแกรมแสดงข้อมูล ... 146
8.25 หน้าเว็บผลการสร้างโปรแกรมแสดงข้อมูล ... 147
8.26 โค้ดภาษา PHP สร้างโปรแกรมค้นหาข้อมูล ... 147
8.27 หน้าเว็บโปรแกรมค้นหาข้อมูล ... 148
8.28 หน้าเว็บแสดงผลการค้นหาข้อมูล ... 148
8.29 โค้ดภาษา PHP สร้างโปรแกรมแก้ไขข้อมูล และ ลบข้อมูล ... 150
8.30 หน้าเว็บโปรแกรมแก้ไขข้อมูล และ ลบข้อมูล ... 150

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

8.31 หน้าเว็บโปรแกรมแสดงการแก้ไขข้อมูล..... 150
8.32 หน้าเว็บแสดงผลการแก้ไขข้อมูล..... 151
8.33 หน้าเว็บแสดงการยืนยันการลบข้อมูล..... 151
8.34 หน้าเว็บแสดงผลการลบข้อมูล 151

บทที่ 1
ความรู้เบื้องต้นเกี่ยวกับฐานข้อมูล

ฐานข้อมูล (Database) เป็นการจัดโครงสร้างข้อมูลให้เป็นแบบฐานข้อมูลกำลังเป็นที่นิยม เนื่องจากปริมาณข้อมูลมีมากถ้าจัดข้อมูลเป็นแบบแฟ้มข้อมูลจะทำให้มีแฟ้มข้อมูลเป็นจำนวนมาก ซึ่งจะทำให้เกิดข้อมูลที่ซ้ำซ้อนกันได้ ข้อมูลที่ซ้ำซ้อนนี้จะก่อให้เกิดปัญหามากมาย ฐานข้อมูลสามารถรักษาความถูกต้องของข้อมูล ป้องกันและรักษาความปลอดภัยให้กับข้อมูลทำได้อย่างสะดวก สามารถใช้ข้อมูลร่วมกันได้ มีความเป็นอิสระของข้อมูล สามารถขยายงานได้ง่าย ทำให้ข้อมูลบูรณะกลับสู่สภาพปกติได้เร็วและมีมาตรฐาน (ปิยะดนัย วิเคียน, 2564) ปัจจุบัน ระบบฐานข้อมูล (database system) ได้เข้ามามีบทบาทต่อระบบสารสนเทศที่สนับสนุนภารกิจของหน่วยงานทางการศึกษา ดังจะเห็นได้จากฝ่ายงานต่าง ๆ มีการประมวลผลด้วยคอมพิวเตอร์ซึ่งเกี่ยวข้องกับระบบฐานข้อมูลทั้งหมด ประกอบปัจจุบันกับมีเครื่องมือที่สามารถเชื่อมต่อกับอินเตอร์เน็ตได้หลายชนิด เช่น คอมพิวเตอร์ สมาร์ทโฟน IPad หรือแท็บเล็ต ที่ช่วยเพิ่มความสะดวกต่อผู้ใช้งานให้สามารถเข้าถึงและแบ่งปันข้อมูลต่าง ๆ ภายในองค์กรและบนเว็บไซต์ได้ง่ายขึ้น ส่งผลให้มีการพัฒนาฐานข้อมูลเพื่อรองรับภารกิจขององค์กรมากขึ้น ปรากฏการณ์ที่เกิดขึ้นได้ส่งผลกระทบต่อเทคโนโลยีระบบฐานข้อมูลที่มีออกแบบและพัฒนาเพื่อรองรับกับข้อมูลที่มีความหลากหลายในปัจจุบัน ทั้งในรูปแบบฐานข้อมูลภายในองค์กรและฐานข้อมูลบนเว็บไซต์ ดังนั้น ในบทที่ 1 เรื่อง ความรู้เบื้องต้นเกี่ยวกับฐานข้อมูล ได้นำเสนอเนื้อหาสำหรับการศึกษาดังนี้

- ♦ ข้อมูล (Data)
- ♦ ระบบแฟ้มข้อมูล (File system)
- ♦ ระบบฐานข้อมูล (Database system)
  - ◊ ฐานข้อมูล (Database)
  - ◊ ระบบจัดการฐานข้อมูล (Database management system)

ข้อมูล (Data)

ข้อมูล หมายถึง ข้อเท็จจริง หรือสิ่งที่ถือหรือยอมรับว่าเป็นข้อเท็จจริง สำหรับใช้เป็นหลักอนุมานหาความจริง หรือการคำนวณ (พจนานุกรม ฉบับราชบัณฑิตยสถาน, 2554)

ข้อมูล หมายถึง ข้อเท็จจริงต่าง ๆ ที่ถูกจัดเก็บลงในระบบฐานข้อมูล หรือระบบแฟ้มข้อมูล โดยข้อมูลอาจถูกจัดเก็บอยู่ในรูปของตัวเลข ข้อความ หรือสื่อมัลติมีเดียร์ต่าง ๆ ได้ ในโลกปัจจุบันข้อมูลถือได้ว่าเป็นส่วนประกอบที่สำคัญ ดังนั้น ข้อมูลที่ดีจะต้องมีความถูกต้อง (Accuracy) สมบูรณ์ (Integrity) และน่าเชื่อถือ (Reliable) (ณัฐพงษ์ วารีประเสริฐ, 2553)

ข้อมูล หมายถึง ข้อเท็จจริงที่ทราบ สามารถถูกจัดเก็บและมีความหมายต่อผู้ใช้ เช่น ชื่อ รหัสนักศึกษา ที่อยู่ (เทพฤทธิ์ บัณฑิตวัฒนาวงศ์, 2554)

สรุปได้ว่า ข้อมูลหมายถึง ข้อเท็จจริงต่าง ๆ อาจถูกจัดเก็บอยู่ในรูปของตัวเลข ข้อความ หรือสื่อมัลติมีเดียร์ สำหรับใช้เป็นหลักอนุมานหาความจริงหรือการคำนวณ ข้อมูลที่ดีจะต้องมีความถูกต้อง (Accuracy), สมบูรณ์ (Integrity), และน่าเชื่อถือ (Reliable)

ชนิดของข้อมูล (Type of data)

ข้อมูลที่จัดเก็บลงในแฟ้มข้อมูลหรือฐานข้อมูล นอกจากจะเป็นแบบข้อความแล้ว ในปัจจุบันยังมีข้อมูลชนิดอื่น ๆ ที่สามารถนำมาใช้ประกอบร่วมกัน ตัวอย่างเช่น ข้อมูลแบบข้อความที่ใช้อธิบายรายละเอียดเกี่ยวกับผลิตภัณฑ์ชิ้นหนึ่ง ในขณะเดียวกันก็มีรูปภาพของผลิตภัณฑ์นั้น พร้อมเสียงประกอบคำบรรยายตัวผลิตภัณฑ์เป็นต้น โดยชนิดของข้อมูลยังแบ่งออกเป็นรูปแบบต่าง ๆ อันได้แก่ 1) ข้อมูลชนิดข้อความ 2) ข้อมูลที่เป็นรูปแบบ 3) ข้อมูลรูปภาพ และ 4) ข้อมูลชนิดเสียง

1. ข้อมูลชนิดข้อความ (Text)

เป็นข้อมูลที่ประกอบด้วยตัวอักขระ (เช่น ตัวอักษรและตัวเลข) ต่าง ๆ นำมารวมกันเป็นคำหรือประโยค เพื่อใช้สื่อความหมายแทนสิ่งๆ หนึ่ง โดยไม่มีรูปแบบที่ชัดเจนและแน่นอน ข้อมูลชนิดนี้จะมีความหมายในตัวเองโดยไม่จำเป็นต้องนำมาตีความหมายใด ๆ อีก เช่น ชื่อพนักงานและที่อยู่ปัจจุบันของพนักงาน เป็นต้น

2. ข้อมูลที่เป็นรูปแบบ (Formatted data)

ข้อมูลชนิดนี้จะมีความแตกต่างกับแบบแรก กล่าวคือ เป็นข้อมูลที่ประกอบด้วยตัวอักขระต่าง ๆ ที่ถูกกำหนดรูปแบบแน่นอนเอาไว้ ซึ่งปกติมักถูกกำหนดในรูปของรหัส ข้อดีก็คือ ทำให้ไม่สิ้นเปลืองเนื้อที่ในการจัดเก็บ และลดโอกาสการกรอกคำสะกดข้อมูลแบบผิด ๆ รวมถึงความสะดวกต่อการนำไปใช้อ้างอิง แต่ข้อมูลชนิดนี้ จำเป็นต้องนำไปตีความหมายตามรายละเอียดที่กำหนดเอาไว้ ตัวอย่างเช่น รหัสสาขา “CS” ใช้แทนสาขาวิทยาการคอมพิวเตอร์ หรือรหัสสาขา “IT” ใช้แทนสาขาเทคโนโลยีสารสนเทศ หรือรหัสวิชา “CS104” คือวิชาระบบฐานข้อมูล เป็นต้น

3. ข้อมูลรูปภาพ (Images)

ด้วยเทคโนโลยีของอุปกรณ์อย่างกล้องดิจิตอล เครื่องสแกนเนอร์ และโปรแกรมช่วยสร้างภาพต่าง ๆ ดังนั้น การแทนข้อมูลด้วยรูปภาพในยุคปัจจุปันจึงเป็นเรื่องที่ง่ายขึ้นมาก ข้อมูลรูปภาพนิยมนำมาใช้เสริมร่วมกันกับข้อมูลแบบข้อความเพื่อให้มีความสมบูรณ์ยิ่งขึ้น เช่น ข้อมูลประวัตินักศึกษา ที่นอกจากมีข้อมูลเป็นข้อความแล้ว ยังมีรูปถ่ายของนักศึกษา หรือข้อมูลสินค้าที่มีรายละเอียดพร้อมภาพถ่าย เป็นต้น อย่างไรก็ตาม แม้ว่าข้อมูลรูปภาพส่วนใหญ่จำเป็นต้องใช้เนื้อที่ในการจัดเก็บมากกว่าข้อมูลชนิดแบบข้อความก็ตาม แต่ด้วยสื่อจัดเก็บข้อมูลในยุคปัจจุบันก็ได้

พัฒนาขึ้นมาก มีความจุสูงและมีขีดความสามารถในจัดเก็บข้อมูลได้ในปริมาณมาก ดังนั้น ข้อมูลรูปภาพจึงมิใช่ปัญหาใหญ่สำหรับในยุคปัจจุบันแล้ว เพียงแต่ขอให้เลือกใช้ชนิดไฟล์ภาพให้เหมาะสมกับงาน (เช่น BMP, JPG, TIFF, GIF และ PNG) เพราะไฟล์ภาพแต่ละสกุลจะเก็บรายละเอียดภาพที่แตกต่างกัน นั่นหมายถึง มีขนาดความจุที่แตกต่างกันด้วย

4. ข้อมูลชนิดเสียง (Audio/ Sound)

เป็นไฟล์ข้อมูลชนิดหนึ่งที่ใช้จัดเก็บเสียงแบบดิจิตอล เช่น การแปลงเสียงพูดของมนุษย์หรือเสียงดนตรีซึ่งเป็นแบบแอนะล็อก ให้กลายมาเป็นไฟล์ดิจิตอลที่สามารถคลิกเพื่อเปิดฟังผ่านลำโพงที่เชื่อมต่อกับคอมพิวเตอร์ได้ ตัวอย่างเช่น ไฟล์ประเภท MIDI หรือไฟล์ดิจิตอลออดิโอทั่ว ๆ ไป

และจากข้อมูลชนิดต่าง ๆ ข้างตน เรายังสามารถนำข้อมูลเหล่านี้มาใช้งานร่วมกันได้ เช่น งานด้านมัลติมีเดีย หรือสื่อประสมที่นำข้อมูลชนิดต่าง ๆ มาประกอบเข้าด้วยกัน ได้แก่ ข้อความ ภาพนิ่ง คลิปวีดิโอ กราฟิก แอนนิเมชัน ซึ่งเหมาะกับงานสร้างสื่อโฆษณา (โอภาส เอี่ยมศิริวงศ์, 2558)

การแบ่งประเภทของข้อมูล

การแบ่งประเภทข้อมูล มีมีวิธีการแบ่งได้หลายวิธี ดังนี้

1. ประเภทของข้อมูลแบ่งตามลักษณะการเก็บข้อมูล

ประเภทของข้อมูลจำแนกตามลักษณะการเก็บข้อมูล แบ่งได้เป็น 4 ประเภท คือ

1.1 ข้อมูลที่ได้จากการนับ (Counting data) เช่น จำนวนนักศึกษาที่สอบผ่าน จำนวนรถที่ผ่านเข้า-ออกมหาวิทยาลัยในช่วงเวลา 08.00-09.00 น . ซึ่งข้อมูลที่ได้จะเป็นเลขจำนวนเต็ม บางครั้งเรียกว่า เป็นข้อมูลที่ไม่ต่อเนื่อง

1.2 ข้อมูลที่ได้จาการวัด (Measurement data) เช่น น้ำหนักของนักศึกษาแต่ละคน ส่วนสูงของนักศึกษาแต่ละคน ระยะเวลาในการเดินทางจากบ้านมายังที่ทำงานของพนักงานแต่ละคน ปริมาณน้ำฝนที่วัดได้ ข้อมูลที่ได้จะมีลักษณะเป็นเศษส่วน หรือจุดทศนิยม บางครั้งเรียกว่าข้อมูลแบบต่อเนื่อง

1.3 ข้อมูลที่ได้จากการสังเกต (Observation data) เป็นข้อมูลที่ได้จากการติดตามหรือเฝ้าสังเกตพฤติกรรม หรือปรากฏการณ์ต่าง ๆ เป็นต้น

1.4 ข้อมูลที่ได้จากการสัมภาษณ์ (Interview Data) เป็นข้อมูลที่ได้จากการถามตอบโดยตรง ระหว่างผู้สัมภาษณ์ และผู้ถูกสัมภาษณ์

2. ประเภทของข้อมูลแบ่งตามลักษณะข้อมูล แบ่งได้เป็น 2 ประเภท คือ

2.1 ข้อมูลเชิงปริมาณ (Quantitative data) เป็นข้อมูลที่แสดงความแตกต่างในเรื่องปริมาณ หรือขนาด ในลักษณะของตัวเลขโดยตรง เช่น อายุ ส่วนสูง น้ำหนัก แบ่งได้ 2 ประเภท คือ

♦ ข้อมูลแบบไม่ต่อเนื่อง (Discrete data) หมายถึง ข้อมูลที่มีค่าเป็นเลขจำนวนเต็มที่มีความหมาย เช่น จำนวนสิ่งของ จำนวนคน เป็นต้น

♦ ข้อมูลแบบต่อเนื่อง (Continuous data) หมายถึง ข้อมูลที่อยู่ในรูปตัวเลขที่มีค่าได้ทุกค่าในช่วงที่กำหนด และมีความหมายด้วย เช่น รายได้ น้ำหนัก เป็นต้น

2.2 ข้อมูลเชิงคุณภาพ (Qualitative data) เป็นข้อมูลที่แสดงลักษณะที่แตกต่างกัน เช่น เพศชาย เพศหญิง จะเป็นข้อมูลที่ไม่ได้อยู่ในรูปของตัวเลขโดยตรง

3. ประเภทของข้อมูลแบ่งตามการจัดการข้อมูล แบ่งได้เป็น 2 ประเภท คือ

3.1 ข้อมูลดิบ (Raw data) เป็นข้อมูลที่ได้จากการเก็บ ยังไม่ได้จัดรวบรวมเป็นหมู่เป็นกลุ่ม หรือจัดเป็นพวก

3.2 ข้อมูลที่จัดเป็นกลุ่ม (Group data) เป็นข้อมูลที่เกิดจากการนำข้อมูลดิบมารวบรวมเป็นกลุ่มเป็นหมวดหมู่

4. ประเภทของข้อมูลแบ่งตามแหล่งที่มาของข้อมูล แบ่งได้เป็น 2 ชนิด คือ

4.1 ข้อมูลปฐมภูมิ (Primary data) เป็นข้อมูลที่ได้มาจากการที่ผู้ใช้เป็นผู้เก็บข้อมูลโดยตรง ซึ่งอาจจะเก็บด้วยการสัมภาษณ์หรือสังเกตการณ์ เป็นข้อมูลที่มีความน่าเชื่อถือมากที่สุด เนื่องจากยังไม่มีการเปลี่ยนรูป และมีรายละเอียดตามที่ผู้ใช้ต้องการ แต่จะต้องเสียเวลาและค่าใช้จ่ายมาก เช่น ข้อมูลที่ได้จากการนับจำนวนรถที่เข้า-ออก มหาวิทยาลัยในช่วงเวลา 08.00-09.00 น. ข้อมูลจากการสัมภาษณ์นักศึกษา

4.2 ข้อมูลทุติยภูมิ (Secondary data) เป็นข้อมูลที่ได้มาจากแหล่งข้อมูลที่มีผู้เก็บรวบรวมไว้แล้ว เป็นข้อมูลในอดีต และมักจะเป็นข้อมูลที่ได้ผ่านการวิเคราะห์เบื้องต้นมาแล้ว ผู้ใช้นำมาใช้ได้เลย จึงประหยัดทั้งเวลาและค่าใช้จ่าย บางครั้งข้อมูลทุติยภูมิจะไม่ตรงกับความต้องการหรือมีรายละเอียดไม่เพียงพอ นอกจากนั้นผู้ใช้จะไม่ทราบถึงข้อผิดพลาดของข้อมูล ซึ่งอาจจะทำให้ผู้ที่นำมาใช้สรุปผลการวิจัยผิดพลาดไปด้วย เช่น สถิติการเกิดอุบัติเหตุโดยรถจักรยานยนต์ของนักศึกษาในปี พ.ศ. 2560-2564 เป็นข้อมูลที่บางครั้งอาจถูกแปรรูปไปแล้ว แต่เนื่องจากบางครั้งเราไม่สามารถที่จะจัดเก็บข้อมูลปฐมภูมิได้เราจึงต้องศึกษาจากข้อมูลที่มีการเก็บรวบรวมไว้แล้ว

5. ประเภทของข้อมูลแบ่งตามมาตรของการวัด จะแบ่งได้ 4 ชนิด

5.1 มาตรวัดนามบัญญัติ (Nominal scale) เป็นการวัดค่าที่ง่ายที่สุดหรือสะดวกต่อการใช้มากที่สุด เพราะเป็นการแบ่งกลุ่มของข้อมูล เพื่อสะดวกต่อการวิเคราะห์ โดยการแบ่งกลุ่มจะถือว่า แต่ละกลุ่มจะมีความเสมอภาคกันหรือเท่าเทียมกัน ค่าที่กำหนดให้แต่ละกลุ่มจะไม่มีความหมาย และไม่สามารถมาคำนวณได้ เช่น เพศ มี 2 ค่า คือ ชายและหญิง การจำแนกเพศอาจจะกำหนดค่าได้ 2 ค่า คือ ถ้า 0 หมายถึงเพศชาย ถ้า 1 หมายถึงเพศหญิง เป็นต้น

5.2 มาตรวัดอันดับ (Ordinal Scale) เป็นการวัดที่แสดงว่าข้อมูลที่อยู่ในแต่ละกลุ่มจะมีความแตกต่างกัน โดยพิจารณาจากลำดับด้วย นั่นคือสามารถบอกได้ว่า กลุ่มใดดีกว่ากลุ่มอื่น ๆ หรือ กลุ่มใดที่มากกว่าหรือน้อยกว่ากลุ่มอื่น ๆ แต่ไม่สามารถบอกปริมาณความมากกว่าหรือน้อยกว่าเป็นเท่าใด และค่าที่กำหนดให้แต่ละกลุ่มไม่สามารถนำมาคำนวณได้ เช่น คำถามที่ว่า “นักศึกษาอยากทำอะไรเมื่อมีวันหยุดพิเศษ” โดยให้เรียงลำดับตามที่ต้องการจะทำมากที่สุด 5 อันดับ

- ไปเที่ยวห้างสรรพสินค้า ลำดับที่ 4
- ดูทีวีที่บ้าน ลำดับที่ 1
- ไปพักผ่อนที่ต่างจังหวัด ลำดับที่ 2
- ไปเล่นกีฬา ลำดับที่ 5
- ไปดูภาพยนตร์ ลำดับที่ 3

จากข้างต้นจะพบว่า นักศึกษารายนี้ชอบดูทีวีที่บ้านมากกว่าไปพักผ่อนต่างจังหวัด แต่ไม่ทราบว่า ชอบมากกว่าเท่าใด

5.3 มาตรวัดแบบช่วง (Interval Scale) เป็นการวัดที่แบ่งสิ่งที่ศึกษาออกเป็นระดับหรือเป็นช่วง ๆ โดยแต่ละช่วงมีขนาดหรือระยะห่างเท่ากัน ทำให้สามารถบอกระยะห่างของช่วงได้ อีกทั้ง

บอกได้ว่ามากหรือน้อยกว่ากันเท่าไร จึงทำให้มีความแตกต่างกันในเชิงปริมาณ เช่น อุณหภูมิ คะแนนสอบ ซึ่งตัวเลขเหล่านี้ บวก ลบ ได้ แต่ คูณ หาร ไม่ได้ แต่ศูนย์ของข้อมูลชนิดนี้เป็น ศูนย์สมมติ ไม่ใช่ศูนย์แท้ เช่น อุณหภูมิ 0 องศาเซลเซียส ไม่ได้หมายความว่า ณ จุดนั้นไม่มีความร้อนอยู่เลย หรือการที่นักศึกษาได้คะแนน 0 ก็ไม่ได้หมายความว่า นักศึกษาไม่มีความรู้เลย แต่เป็นเพียงตัวเลขที่บอกว่า นักศึกษาทำข้อสอบนั้นไม่ได้

5.4 มาตรวัดอัตราส่วน (Ratio scale) เป็นการวัดที่ละเอียดและสมบูรณ์ที่สุด ที่สามารถบอกความแตกต่างในเชิงปริมาณ โดยแบ่งสิ่งที่ศึกษาออกเป็นช่วง ๆ เหมือนมาตรวัดอันตรภาค ที่แต่ละช่วงมีระยะห่างเท่ากัน และศูนย์ของข้อมูลชนิดนี้เป็นศูนย์แท้ ซึ่งหมายถึงไม่มีอะไรเลย หรือมีจุดที่เริ่มต้นที่แท้จริง และสามารถนำตัวเลขนี้มา บวก ลบ คูณ หารได้ เช่น ความยาว เวลา

6. ประเภทของข้อมูลแบ่งตามเวลาของการเก็บรวบรวมข้อมูล จะแบ่งได้ 2 ชนิด

6.1 ข้อมูลอนุกรมเวลา (Time-series data) เป็นข้อมูลที่ถูกเก็บรวบรวมตามลำดับเวลาที่เกิดขึ้นต่อเนื่องไปเรื่อย ๆ เช่น จำนวนประชากรของประเทศไทยในแต่แต่ละปี จำนวนผู้ป่วยที่เข้ารับการรักษาพยาบาลในโรงพยาบาลต่าง ๆ ในแต่ละปี เป็นต้น ข้อมูลอนุกรมเวลาเป็นประโยชน์ในการวิจัยระยะเวลายาว ทำให้ผู้วิจัยมองเห็นแนวโน้มของเรื่องต่าง ๆ นั้นได้

6.2 ข้อมูลภาคตัดขวาง (Cross-sectional data) เป็นข้อมูลที่เก็บรวบรวม ณ เวลาใดเวลาหนึ่งเท่านั้น เพื่อประโยชน์ในการศึกษาวิจัยอย่างไรก็ตามในการจัดประเภทของข้อมูลนี้ จะขึ้นอยู่กับวัตถุประสงค์ในการนำไปวิเคราะห์และใช้ประโยชน์ด้วย

7. ประเภทของข้อมูลแบ่งตามสภาพของข้อมูลที่เกี่ยวข้องกับกลุ่มตัวอย่าง เช่น

7.1 ข้อมูลส่วนบุคคล (Personal data) หมายถึง ข้อมูลที่เกี่ยวกับข้อเท็จจริงส่วนตัวของกลุ่มตัวอย่าง เช่น ชื่อ-สกุล วันเดือนปีเกิด อายุ เพศ สัญชาติ ศาสนา อาชีพ เป็นต้น

7.2 ข้อมูลสิ่งแวดล้อม (Environmental data) หมายถึง ข้อมูลที่เป็นข้อเท็จจริงเกี่ยวกับสิ่งแวดล้อมของกลุ่มตัวอย่าง เช่น ลักษณะท้องถิ่นที่กลุ่มตัวอย่างอาศัย

7.3 ข้อมูลพฤติกรรม (Behavioral data) หมายถึง ข้อมูลที่เป็นคุณลักษณะของกลุ่มตัวอย่าง เช่น คุณลักษณะด้านความสามารถทางสติปัญญา ตัวอย่าง ผลสัมฤทธิ์ทางการเรียน ความรู้ ความเข้าใจ การวิเคราะห์ ความถนัด ความสนใจ ความวิตกกังวล ความเชื่อ แรงจูงใจ การปฏิบัติ การกระทำสิ่งต่าง ๆ

ในยุคที่คอมพิวเตอร์ยังไม่เป็นที่แพร่หลาย องค์กรต่าง ๆ จัดดำเนินการข้อมูลโดยอาศัยระบบแฟ้มข้อมูลซึ่งเป็นกระดาษ เช่น แฟ้มข้อมูลนักศึกษาที่ถูกจัดดำเนินการโดยฝ่ายงานทะเบียนนักศึกษา ต่อมาคอมพิวเตอร์ถูกนำมาประยุกต์ใช้อย่างแพร่หลายเพื่อเพิ่มประสิทธิภาพในการจัดดำเนินการข้อมูลให้สะดวกรวดเร็วและลดค่าใช้จ่ายในส่วนของกระดาษเพราะข้อมูลจะถูกบันทึกเป็นรูปแบบอิเล็กทรอนิกส์แทน เรียกว่า แฟ้มข้อมูลคอมพิวเตอร์ (Computer-based data file) ในยุคนี้การจัดดำเนินการข้อมูลอาศัยการประมวลผลแฟ้มข้อมูลคอมพิวเตอร์ (Traditional file Processing) ซึ่งเป็นการใช้โปรแกรมประยุกต์ (Application) เข้าถึงแฟ้มข้อมูลคอมพิวเตอร์ในอุปกรณ์หน่วยเก็บผ่านทางระบบปฏิบัติการคอมพิวเตอร์โดยตรง อย่างไรก็ตาม ระบบประมวลผลแฟ้มข้อมูลคอมพิวเตอร์พบขีดจำกัดเมื่อข้อมูลมีขนาดและความหลากหลายมากขึ้น อันเนื่องมาจากความต้องการธุรกรรมที่มีความซับซ้อนมากขึ้น (เทพฤทธิ์ บัณฑิตวัฒนาวงศ์, 2554)

การทำความเข้าใจในข้อดีข้อเสียของระบบประมวลผลแฟ้มข้อมูลคอมพิวเตอร์ซึ่งนำไปสู่การคิดค้นระบบฐานข้อมูล จำเป็นต้องอาศัยความรู้เกี่ยวกับระบบแฟ้มข้อมูลดังมีรายละเอียดในหัวข้อถัดไป

ระบบแฟ้มข้อมูล (File system)

แนวคิดในการจัดการข้อมูลได้เกิดขึ้นมานานแล้ว ซึ่งเป็นไปตามเทคโนโลยีที่พัฒนาขึ้นในแต่ละยุคแต่ละสมัย การจัดการข้อมูลได้เริ่มจากการบันทึกข้อมูลซึ่งอาจเป็นการบันทึกลงในสมุดหรือในกระดาษเพื่อบันทึกข้อมูลช่วยในการจดจำเมื่อต้องการเรียกดูข้อมูลที่เคยบันทึกไว้ก็จะพลิกหน้าสมุดไปยังเลขหน้าที่ต้องการเพื่อดูรายละเอียด ต่อมามีการนำคอมพิวเตอร์มาใช้ในการจัดเก็บข้อมูล สามารถจัดเก็บข้อมูลได้จำนวนมากมายมหาศาล สามารถบันทึกลงในสื่อบันทึกข้อมูล เช่น ดิสก์ ฮาร์ดดิสก์ ซีดีรอม หรือเทป ซึ่งข้อมูลที่บันทึกลงในสื่อบันทึกข้อมูลดังกล่าว สามารถเทียบกับปริมาณของตู้เก็บเอกสารจำนวนมากมายมหาศาล ทั้งยังสามารถค้นหาข้อมูลได้รวดเร็วกว่ามาก

การนำคอมพิวเตอร์มาเป็นเครื่องมือการประมวลผลข้อมูลนั้นสิ่งสำคัญคือข้อมูล ข้อมูลที่จัดเก็บนั้น จะจัดเก็บเป็นแฟ้มข้อมูลเพื่อสามารถอ่านข้อมูลจากแฟ้มนั้นได้ ดังนั้น จึงจำเป็นต้องเข้าใจถึงพื้นฐานของโครงสร้างแฟ้มข้อมูล

1. ประเภทของแฟ้มข้อมูล

ข้อมูลที่บันทึกไว้ สามารถแบ่งประเภทของแฟ้มข้อมูลออกเป็น 2 ประเภท ตามรูปแบบการจัดทำ คือ แฟ้มข้อมูลที่จัดทำด้วยมือและแฟ้มข้อมูลคอมพิวเตอร์

1.1 ประเภทของแฟ้มข้อมูลแบ่งตามรูปแบบการจัดทำ

1.1.1 แฟ้มข้อมูลที่จัดทำด้วยมือ เมื่อมีข้อมูลเพิ่มขึ้น ก็มีการพัฒนารูปแบบการจัดเก็บข้อมูลให้มีระบบมากขึ้นตามไปด้วย มีการบันทึกลงในแฟ้มเอกสารต่าง ๆ ที่จัดไว้เป็นหมวดเป็นหมู่มีการจัดสารบัญ และจำเป็นต้องมีอุปกรณ์หรือเครื่องมือที่ใช้จัดเก็บแฟ้มเอกสารเหล่านั้นเพื่อให้เกิดความปลอดภัยยิ่งขึ้น เช่น มีตู้เก็บเอกสารซึ่งก็มีทั้งขนาดเล็กและขนาดใหญ่ให้เลือกใช้งานตามความเหมาะสมเพื่อเก็บแฟ้มเอกสารเหล่านั้น รวมทั้งอาจจะมีการทำดัชนีเพื่อให้การค้นหาข้อมูลมีความรวดเร็วมากยิ่งขึ้นการจัดเก็บข้อมูลลงในแฟ้มต่าง ๆ นำไปเก็บไว้ในตู้เก็บเอกสารอย่างมิดชิดและปลอดภัย เป็นการจัดเก็บข้อมูลที่ทำกันมานานตั้งแต่สมัยก่อนจนถึงสมัยปัจจุบัน ซึ่งบางหน่วยงานก็ยังคงดำเนินการกันอยู่ การจัดเก็บเอกสารในลักษณะนี้ จะพบว่า จำนวนตู้เอกสารจะเพิ่มมากขึ้นเรื่อย ๆ ทำให้การค้นหาข้อมูลเกิดความล่าช้า

1.1.2 แฟ้มข้อมูลคอมพิวเตอร์ การนำคอมพิวเตอร์มาใช้ในการจัดเก็บข้อมูลจะช่วยได้มาก ในกรณีที่หน่วยงานมีข้อมูลเป็นจำนวนมาก เพราะจะทำให้สามารถจัดเก็บข้อมูลได้เป็นปริมาณมาก เพียงแค่ทำการบันทึกข้อมูลที่ต้องการเก็บลงในสื่อบันทึกข้อมูลต่าง ๆ ซึ่งมีอยู่หลายชนิด เช่น ดิสก์ (Disk) ฮาร์ดดิสก์ (Hard disk) ซีดีรอม (CD ROM) หรือ เทป (Tape) เป็นต้น ข้อมูลที่บันทึกลงในสื่อบันทึกข้อมูลเหล่านี้ เทียบได้กับปริมาณของตู้เก็บเอกสารเป็นจำนวนมาก และที่สำคัญคือทำให้ค้นหาหรือเรียกใช้ข้อมูลได้รวดเร็ว มีข้อมูลที่ถูกต้องทันสมัย สามารถออกแบบแฟ้มข้อมูล และทำการพัฒนาได้ง่าย

1.2 ประเภทของแฟ้มข้อมูลแบ่งตามลักษณะการจัดเก็บและการใช้งาน

1.2.1 แฟ้มข้อมูลหลัก (Master file) เป็นแฟ้มข้อมูลหรือไฟล์ที่จัดเก็บข้อมูลที่ไม่มีรายการเปลี่ยนแปลง หรือมีสภาพที่ค่อนข้างคงที่ เช่น แฟ้มข้อมูลประวัตินักศึกษาซึ่งประกอบด้วย

ข้อมูลต่าง ๆ เช่น รหัสนักศึกษา ชื่อ-สกุล ที่อยู่ คณะวิชา สาขาวิชา การปรับปรุงแก้ไขข้อมูลในแฟ้มข้อมูลหลักเพื่อให้ทันสมัยนั้น สามารทำได้ 3 รูปแบบ ด้วยกัน คือ การเพิ่ม (Insert) การลบออก (Delete) และการแก้ไข (Modify) เช่น การเพิ่มระเบียนของนักศึกษาในกรณีที่เป็นนักศึกษาใหม่ การลบระเบียนของนักศึกษาที่ลาออก การเปลี่ยนที่อยู่ของนักศึกษา เป็นต้น

1.2.2 แฟ้มข้อมูลปรับปรุง (Transaction file) เป็นไฟล์ หรือ แฟ้มที่จัดเก็บข้อมูลการดำเนินธุรกรรมประจำวันที่มักมีความเคลื่อนไหวอยู่เสมอ เช่น แฟ้มข้อมูลการลงทะเบียนเรียนของนักศึกษาที่จะต้องมีการลงทะเบียนเรียนในทุก ๆ ภาคการศึกษา เป็นต้น

1.2.3 แฟ้มข้อมูลเอกสาร (Document file) เป็นแฟ้มข้อมูลเอกสาร หรือไฟล์รายงาน (Report file) ต่าง ๆ ที่เคยผ่านกระบวนการพิมพ์ด้วยโปรแกรมมาก่อน และทำการจัดเก็บในรูปของแฟ้มข้อมูลเอกสารด้วยการสำเนาเก็บไว้ในคอมพิวเตอร์ เมื่อต้องการใช้งานก็สามารถเรียกขึ้นมาใช้งานได้อย่างรวดเร็ว เพราะไม่ต้องผ่านโปรแกรมเพื่อประมวลเป็นรายงานอีก

1.2.4 แฟ้มข้อมูลตาราง (Table file) เป็นแฟ้มข้อมูลตารางที่ใช้สำหรับการอ้างอิง (Reference) เพื่อใช้งานร่วมกัน โดยข้อมูลต่าง ๆ ที่จัดเก็บลงในไฟล์นี้ค่อนข้างคงที่ หรือมักไม่ค่อยเปลี่ยนแปลงใด ๆ เช่น ตารางข้อมูลคณะ ตารางสาขาวิชา เป็นต้น

2. โครงสร้างแฟ้มข้อมูล

โครงสร้างแฟ้มข้อมูลเรียงลำดับจากหน่วยที่เล็กที่สุดไปยังหน่วยที่ใหญ่ขึ้น ดังนี้

2.1 บิท (Binary digit หรือ Bit)

บิท (Bit) เป็นหน่วยข้อมูลที่มีขนาดเล็กที่สุด ที่เก็บอยู่ในหน่วยความจำภายในคอมพิวเตอร์ ซึ่งเป็นข้อมูลที่เครื่องคอมพิวเตอร์สามารถเข้าใจและนำไปใช้งานได้ ประกอบด้วยเลขฐาน 2 (Binary digit) บิทจะมีอยู่เพียง 1 ใน 2 สถานะเท่านั้น คือ เลข 0 หรือเลข 1 อย่างใดอย่างหนึ่ง

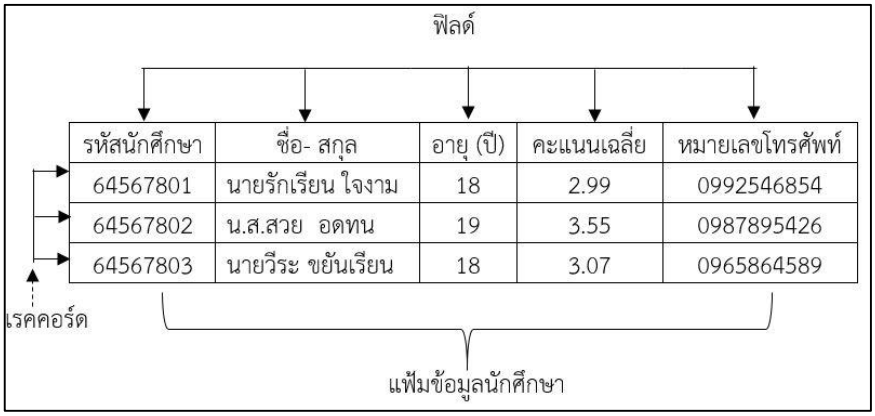
2.2 ไบท์ (Byte) หรือ อักขระ (Character) คือหน่วยของข้อมูลที่นำบิทหลาย ๆ บิทมารวมกัน แทนด้วยตัวเลข หรือ ตัวอักษร หรือ สัญลักษณ์พิเศษ 1 ตัว เช่น ก, ข.....ฮ, 0, 1, 2....9, A, B, C....Z และสัญลักษณ์พิเศษอื่น ๆ เช่น &, @, $, +, % ฯลฯ โดยตัวอักษร 1 ตัวจะแทนด้วยบิท 8 บิท เนื่องจากว่า 1 บิทจะสามารถใช้แทนรหัสได้เพียง 1 ใน 2 สถานะเท่านั้น (คือ 0 กับ 1) ดังนั้นจำเป็นต้องนำบิทหลาย ๆ บิทมารวมกันเป็น ไบท์ เช่น ตัวอักษร A เมื่อเก็บอยู่ในคอมพิวเตอร์จะเก็บเป็น 1000001 ส่วนตัวอักษร B จะเก็บเป็น 1000010 เป็นต้น จึงทำให้สามารถสร้างรหัสแทนข้อมูลขึ้นเพื่อใช้สำหรับแทนอักษรที่แตกต่างกันได้ถึง 256 ตัวด้วยกัน

2.3 ฟิลด์ (Field) หรือ เขตข้อมูล คือการนำตัวอักขระตั้งแต่ตัวหนึ่งขึ้นไปมารวมกันเพื่อให้เกิดความหมาย เช่น อายุเลขประจำตัว ชื่อสกุล ที่อยู่ หมายเลขโทรศัพท์ เป็นต้น

2.4 เรคคอร์ด (Record) หรือระเบียน หมายถึง การนำเอาฟิลด์จำนวนหลายฟิลด์ที่มีความสัมพันธ์กันมารวมกันเป็นกลุ่ม เช่น นักศึกษาแต่ละคนจะมีข้อมูลที่เกี่ยวกับรหัสนักศึกษา ชื่อ สกุล เพศ วันเดือนปีเกิด ที่อยู่ คณะวิชา และสาขาวิชา เป็นต้น ข้อมูลลักษณะนี้ คือ 1 เรคคอร์ด ดังนั้น 1 เรคคอร์ด จะต้องมีอย่างน้อย 1 ฟิลด์ หรือมากกว่านั้น เพื่อใช้ในการอ้างอิงข้อมูลในเรคคอร์ดนั้น ๆ

2.5 แฟ้มข้อมูล หรือ ไฟล์ (File) คือ เรคคอร์ดจำนวนหลาย ๆ เรคคอร์ดที่เกี่ยวข้องในเรื่องเดียวกันมารวมกันเป็นกลุ่มของเรคคอร์ด เช่น แฟ้มข้อมูลนักศึกษาปี 1 จำนวน 50 คน ทุกคนจะ

มีข้อมูล เช่น ชื่อ สกุล เพศ อายุ เกรดเฉลี่ย ฯลฯ ข้อมูลทั้งหมดนี้ของนักศึกษา 50 คนนี้ เรียกว่า แฟ้มข้อมูล



ภาพที่ 1.1 ความสัมพันธ์ขององค์ประกอบในโครงสร้างแฟ้มข้อมูล

3. วัตถุประสงค์การจัดสร้างโครงสร้างแฟ้มข้อมูล

การจัดโครงสร้างแฟ้มข้อมูล เป็นเทคนิคที่ใช้สำหรับการจัดการระเบียนแฟ้มข้อมูล เพื่อการเข้าถึงและจัดเก็บแฟ้มข้อมูลลงบนอุปกรณ์สื่อบันทึกข้อมูลสำรอง (Secondary storage) มีวัตถุประสงค์ในการจัดสร้างโครงสร้างแฟ้มข้อมูลดังนี้

3.1 เพื่อทำให้การค้นหา และการเข้าถึงข้อมูลมีความรวดเร็วขึ้น

3.2 เป็นการใช้อุปกรณ์จัดเก็บข้อมูลได้อย่างมีประสิทธิภาพ

3.3 เป็นการป้องกันข้อมูลที่อาจสูญเสีย

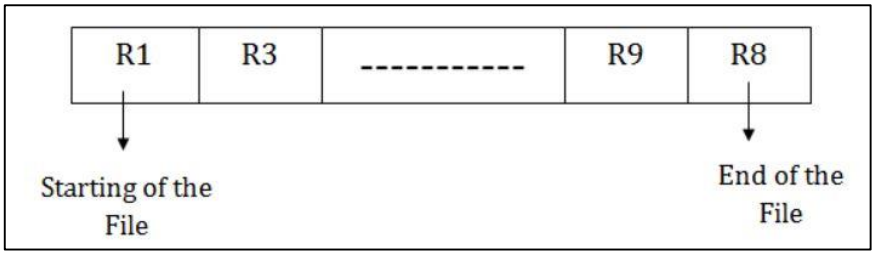
3.4 เพื่อรองรับการเติบโตของข้อมูล

3.5 เพื่อความปลอดภัยจากผู้ใช้ที่ไม่มีสิทธิใช้งาน

4. รูปแบบการจัดโครงสร้างแฟ้มข้อมูล

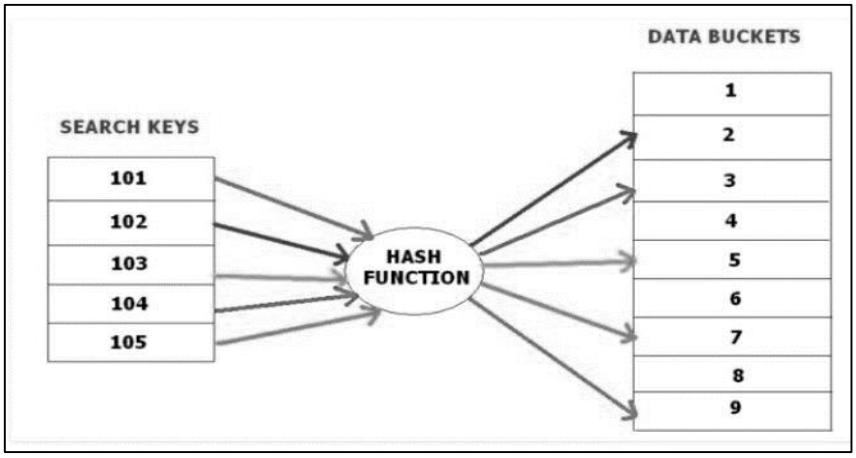
4.1 โครงสร้างแฟ้มข้อมูลแบบเรียงลำดับ (Sequential file organization) เป็นการจัดแฟ้มข้อมูล ซึ่งระเบียนภายในแฟ้มข้อมูลจะถูกบันทึกโดยเรียงหรือไม่เรียงตามลำดับคีย์ฟิลด์ (Key field) ก็ได้ เช่น แฟ้มข้อมูลนักศึกษา อาจจะกำหนดให้รหัสของนักศึกษาเป็นคีย์ ดังนั้น ในการเรียงระเบียนเพื่อเก็บข้อมูลลงในแฟ้มข้อมูลก็จะเรียงลำดับตามรหัสนักศึกษา

ข้อมูลจะถูกบันทึกลงในสื่อบันทึกข้อมูลในตำแหน่งที่อยู่ติด ๆ กัน การนำข้อมูลมาใช้ของโครงสร้างแฟ้มข้อมูลแบบลำดับจะต้องอ่านข้อมูลไปตามลำดับ ไม่สามารถเข้าถึงข้อมูลโดยตรงได้ โครงสร้างแฟ้มข้อมูลแบบนี้ จะบันทึกระเบียนเรียงไปตามลำดับ เช่น จากระเบียนที่ 1 ไปจนถึงระเบียนที่ N การอ่านระเบียนก็จะต้องอ่านแบบต่อเนื่องตามลำดับเช่นกัน เช่น ถ้าต้องการอ่านระเบียนที่ 5 ก็จะต้องเริ่มอ่านตั้งแต่ระเบียนที่ 1, 2, 3, 4 ก่อน เป็นต้น



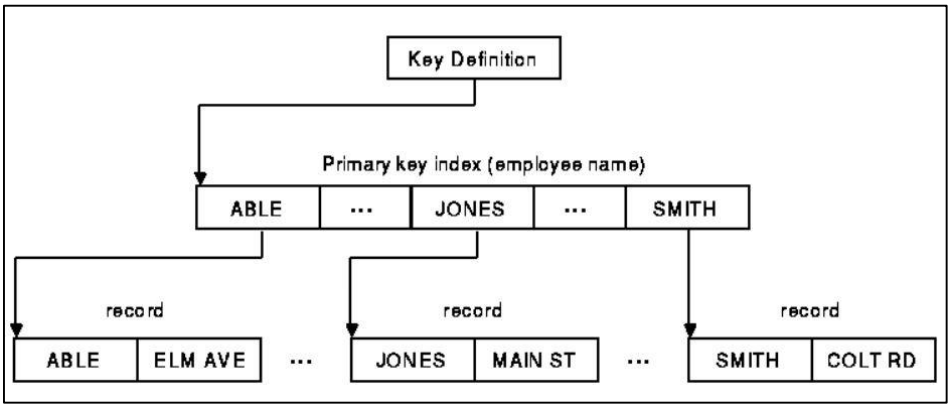
ภาพที่ 1.2 การค้นหาข้อมูลในแฟ้มข้อมูลแบบเรียงลำดับ

4.2 โครงสร้างของแฟ้มข้อมูลแบบสุ่ม (Direct/Random access file) เป็นลักษณะโครงสร้างแฟ้มข้อมูลที่เข้าถึงข้อมูลได้โดยตรง เมื่อต้องการอ่านค่าระเบียนใด ๆ สามารถทำการเลือกหรืออ่านค่าระเบียนนั้นได้ทันที ไม่จำเป็นต้องผ่านระเบียนแรก ๆ เหมือนกับแฟ้มข้อมูลแบบเรียงลำดับ โดยการเข้าถึงข้อมูลจะใช้วิธีการที่เรียกว่า แฮชชิง (Hashing) โดยจะนำค่าคีย์ฟิลด์ไปคำนวณหาตำแหน่งของระเบียนที่เก็บข้อมูล ทำให้การเข้าถึงข้อมูลได้รวดเร็วกว่าปกติ



ภาพที่ 1.3 การค้นหาข้อมูลในแฟ้มข้อมูลแบบสุ่ม

4.3 โครงสร้างแฟ้มข้อมูลแบบลำดับดัชนี (Index sequential file) เป็นวิธีการเก็บข้อมูล โดยแต่ละระเบียนในแฟ้มข้อมูลจะมีค่าของคีย์ฟิลด์ที่ใช้เป็นตัวระบุระเบียนนั้น และจัดทำดัชนีหรือตารางดัชนีไว้สำหรับค้นหาข้อมูลแต่ละระเบียน คล้ายกับการจัดทำสารบัญหนังสือ โดยที่ค่าคีย์ฟิลด์แต่ละระเบียนจะต้องไม่ซ้ำกันกับค่าคีย์ฟิลด์อื่นในแฟ้มข้อมูลเดียวกัน เพราะการจัดโครงสร้างแฟ้มข้อมูลแบบนี้จะใช้คีย์ฟิลด์เป็นตัวเข้าถึงข้อมูล ข้อมูลในแฟ้มจะถูกแบ่งออกเป็นช่วง ๆ หรือเป็นเซกเมนต์ (Segment) โดยมีดัชนีเป็นตัวบอกว่าข้อมูลที่ต้องการอยู่ที่เซกเมนต์ใด การเก็บข้อมูลส่วนมากมักจะใช้จานแม่เหล็กเป็นหน่วยเก็บข้อมูล สามารถเข้าถึงข้อมูลโดยตรง การเก็บข้อมูลโดยวิธีนี้จะทำให้เข้าถึงระเบียนที่ต้องการได้เร็วขึ้น เพราะจะอ่านข้อมูลเพียงแค่เซกเมนต์เดียว ไม่ต้องอ่านทั้งแฟ้มข้อมูล



ภาพที่ 1.4 การค้นหาข้อมูลในแฟ้มข้อมูลแบบสุ่ม (Compaq COBOL, 2022)

5. ปัญหาที่เกิดขึ้นในการใช้แฟ้มข้อมูล

ปัญหาที่เกิดขึ้นในการใช้แฟ้มข้อมูลที่พบกันมี 3 ประเด็นสำคัญคือ ความซ้ำซ้อนของข้อมูล (Data redundancy) รูปแบบที่ไม่ตรงกัน (Incompatible file formats) และความยุ่งยากในการเข้าถึงข้อมูล

5.1 ความซ้ำซ้อนของข้อมูล (Data redundancy)

ความซ้ำซ้อนของข้อมูล หมายถึง การที่มีข้อมูลเดียวกัน ถูกจัดเก็บไว้มากกว่า 1 แห่ง เนื่องจากความยุ่งยากในการรวบรวมข้อมูลมาไว้ที่เดียวกัน อาจจะทำให้เกิดความผิดพลาดของข้อมูล (Data anomalies) กล่าวคือ เมื่อมีความต้องการในการเพิ่ม เปลี่ยนแปลงแก้ไข และลบข้อมูลที่มีความซ้ำซ้อนทำให้ต้องมีการเปลี่ยนแปลงค่าในฟิลด์หลายแห่งแทนที่จะกระทำเพียงที่เดียวเท่านั้น ส่งผลให้ข้อมูลเดียวกันที่จัดเก็บไว้หลายแห่งมีเนื้อหาไม่ตรงกัน ตัวอย่างข้อผิดพลาดที่เกิดจากความซ้ำซ้อนของข้อมูล เช่น

5.1.1 ข้อผิดพลาดจากการเพิ่มข้อมูล (Insertion anomalies) ในกรณีที่มีการเพิ่มข้อมูล เช่น มีนักศึกษาเพิ่มเข้ามา คือ นายมานะ เรียนดี และได้มีการเพิ่มข้อมูลนักศึกษาคนดังกล่าวไว้ในแฟ้มข้อมูลนักศึกษาฝ่ายทะเบียน หากฝ่ายห้องสมุดจะทำการเพิ่มข้อมูลสมาชิกจะต้องกรอกชื่อสมาชิกให้ตรงกันกับชื่อนักศึกษาจากแฟ้มข้อมูลฝ่ายทะเบียน แต่ห้องสมุดมีการกรอกข้อมูลผิดพลาด จากชื่อนายมานะ เรียนดี เป็นชื่อนายมานี เรียนดี ทำให้เมื่อมีการตรวจสอบข้อมูลเกิดการเข้าใจผิดว่านักศึกษาดังกล่าวเป็นคนละคนกัน เป็นต้น

5.1.2 ข้อผิดพลาดจากการลบข้อมูล (Deletion anomalies) ในกรณีที่มีการลบข้อมูล เช่น มีนักศึกษาชื่อ นายมานะ เรียนดี ได้ลาออกจากมหาวิทยาลัย ดังนั้น มหาวิทยาลัยจึงทำการจำหน่ายชื่อนักศึกษาคนนี้ออกจากแฟ้มข้อมูล แต่ห้องสมุดยังไม่ได้มีการลบข้อมูลนักศึกษาคนนี้ออกจากแฟ้มข้อมูลสมาชิกส่งผลให้ นายมานะ เรียนดี ยังมีสถานะภาพเป็นสมาชิกของห้องสมุดอยู่ทั้งที่ไม่มีได้ลาออกจากมหาวิทยาลัยไปแล้ว ซึ่งถือเป็นสิ่งที่ไม่ถูกต้อง

5.1.3 ข้อผิดพลาดจากการแก้ไขข้อมูล (Modification anomalies) ข้อมูลต่าง ๆ ไม่ใช่สิ่งที่ไม่มีการเปลี่ยนแปลงหรืออยู่คงที่ตลอดไป เช่น กรณีที่นักศึกษารหัส 64578009 นายมานะ เรียนดี ได้เปลี่ยนชื่อใหม่เป็น นายนิติพงษ์ เรียนดี ทางฝ่ายทะเบียนมหาวิทยาลัย จึงดำเนินการเปลี่ยนแปลงชื่อดังกล่าว แต่ในขณะเดียวกันทางห้องสมุดไม่ได้มีการเปลี่ยนแปลงชื่อใด ๆ ข้อมูลฝ่ายทะเบียนและข้อมูลห้องสมุดจึงไม่ถูกต้องตรงกัน

5.2 มีรูปแบบที่ไม่ตรงกัน (Incompatible file formats) เมื่อโครงสร้างข้อมูลถูกฝังตัว (Embedded) ด้วยโปรแกรมประยุกต์ โครงสร้างข้อมูลก็จะขึ้นอยู่กับภาษาของโปรแกรมประยุกต์นั้น เช่น โครงสร้างข้อมูลที่สร้างด้วยภาษา COBOL อาจมีความแตกต่างจากโครงสร้างของไฟล์ข้อมูลที่สร้างด้วยภาษา C และเป็นการยากสำหรับการนำไฟล์ทั้ง 2 มาประมวลผลร่วมกัน

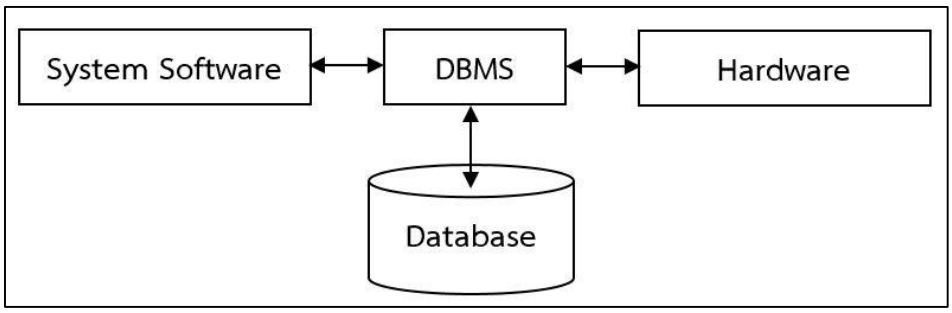
5.3 ความยุ่งยากในการเข้าถึงข้อมูล ถ้าแต่ละหน่วยงานมีการจัดเก็บข้อมูลแยกกัน คือ จัดเก็บข้อมูลของตนเอง จะเป็นการยากสำหรับการเข้าถึงข้อมูลได้โดยตรง โปรแกรมเมอร์จำเป็นต้องเขียนโปรแกรมด้วยการนำข้อมูลที่เก็บแยกกัน แล้วนำมาประมวลผลร่วมกัน ดังนั้น หากผู้ใช้ต้องการรายงานเพิ่มเติม ต้องว่าจ้างโปรแกรมเมอร์ให้เขียนโปรแกรมใหม่

จากปัญหาข้อจำกัดของระบบแฟ้มข้อมูล ทำให้เกิดวิวัฒนาการของระบบการจัดเก็บและการรวบรวมข้อมูลเป็นระบบฐานข้อมูล (Database systems) ที่ได้แก้ปัญหาหรือลดข้อจำกัดต่าง ๆ ของระบบแฟ้มข้อมูลลงได้ (สุจิตรา อดุลย์เกษม, 2553)

ระบบฐานข้อมูล (Database system)

ระบบฐานข้อมูล (Database system) คือ ระบบที่รวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกันเข้าไว้ด้วยกันอย่างมีระบบ มีความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่ชัดเจน ในระบบฐานข้อมูลประกอบด้วยแฟ้มข้อมูลหลายแฟ้มที่มีข้อมูล เกี่ยวข้องสัมพันธ์กันเข้าไว้ด้วยกันอย่างเป็นระบบและเปิดโอกาสให้ผู้ใช้สามารถใช้งานและดูแลรักษาป้องกันข้อมูลเหล่านี้ ได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (Database management system) มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพ การเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการตั้งคำถามเพื่อให้ได้ข้อมูลมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล

ระบบฐานข้อมูลประกอบด้วย ฐานข้อมูล (Database) และระบบจัดการฐานข้อมูล (Database management system: DBMS) ดังภาพที่ 1.5



ภาพที่ 1.5 ระบบฐานข้อมูล

1. ฐานข้อมูล (Database)

ฐานข้อมูล หมายถึง ชุดของข้อมูลที่มีความสัมพันธ์กันที่ถูกนำมาจัดเก็บไว้ด้วยกัน เพื่อให้สามารถใช้ข้อมูลเหล่านั้นร่วมกันได้อย่างรวดเร็วและมีประสิทธิภาพ ตัวอย่างของฐานข้อมูลอย่างง่ายและเกี่ยวข้องกับชีวิตประจำวันของเรา ได้แก่ สมุดโทรศัพท์ ซึ่งเป็นการจัดเก็บรวบรวมรายชื่อ

และเบอร์โทรศัพท์ของผู้ที่เราต้องการติดต่อด้วย หรือการจัดเก็บข้อมูลค่าใช้จ่ายภายในครอบครัว เป็นต้น การจัดเก็บข้อมูลจะมีประสิทธิภาพได้ ก็ต่อเมื่อมีวิธีการจัดการข้อมูลที่ดี กล่าวคือ วิธีการจัดเก็บและค้นคืนข้อมูลต้องเป็นไปอย่างมีประสิทธิภาพและรวดเร็ว เช่น มีการจัดเก็บรายชื่อแบ่งตามลำดับตัวอักษร เป็นต้น โดยทั่วไปเมื่อข้อมูลมีขนาดใหญ่ขึ้น การสร้างฐานข้อมูลมักจะกระทำโดยใช้เครื่องคอมพิวเตอร์เข้ามาช่วย เพื่อให้สามารถจัดเก็บและใช้ข้อมูลเหล่านี้ร่วมกัน ตลอดจนสามารถค้นคืนได้อย่างรวดเร็ว (วิเชียร เปรมชัยสวัสดิ์, 2555)

ตัวอย่างฐานข้อมูลนักศึกษา เป็นที่เก็บรายการข้อมูลของนักศึกษา ซึ่งมีรายละเอียดต่าง ๆ เกี่ยวกับนักศึกษาตามที่ระบบต้องการ เช่น รหัสประจำตัวชื่อนักศึกษา เกรดเฉลี่ยและชื่ออาจารย์ที่ปรึกษา โดยที่รหัสประจำตัวนักศึกษาชื่อนักศึกษาและชื่ออาจารย์ที่ปรึกษา มีชนิดข้อมูลเป็น Non-numeric data ที่กำหนดให้แต่ละฟิลด์ดังกล่าวนั้น ใช้สำหรับบรรจุข้อมูลที่เป็นตัวเลข หรือตัวอักษร เราจึงไม่สามารถนำฟิลด์เหล่านี้ไปใช้ในการคำนวณได้ ส่วนเกรดเฉลี่ยมีชนิดข้อมูลเป็น Numeric data ทำให้เราสามารถนำฟิลด์นี้ไปใช้ในการคำนวณได้ เป็นต้น

ภาพที่ 1.6 ตัวอย่างฐานข้อมูลนักศึกษา

ฐานข้อมูล เป็นการจัดเก็บข้อมูลที่มีความสัมพันธ์กัน นำมาเก็บรวบรวมให้อยู่ในที่เดียวกันอย่างเป็นระบบ ทำให้ผู้ใช้สามารถใช้ข้อมูลที่เกี่ยวข้องร่วมกันได้ เป็นการลดความซ้ำซ้อนของข้อมูล ข้อมูลมีความถูกต้อง น่าเชื่อถือ และเป็นมาตรฐานเดียวกัน ฐานข้อมูลคือระบบการเก็บข้อมูลที่รวบรวมไว้อย่างมีระบบ และมีความสัมพันธ์ในการเรียกฐานข้อมูลมาใช้งานตามจุดประสงค์ต่าง ๆ ในยุคสารสนเทศ ฐานข้อมูลเป็นหัวใจสำคัญของระบบสารสนเทศเพื่อการบริหาร (Management information system หรือ MIS) เพราะว่าสารสนเทศที่ดีและมีคุณภาพจะมาจากข้อมูลที่ดี จะต้องมีความทันสมัย เชื่อถือได้ และถูกจัดเก็บอย่างเป็นระบบ ผู้ใช้สามารถเข้าถึงและใช้ประโยชน์ข้อมูลได้อย่างสะดวกและรวดเร็ว ดังนั้น ฐานข้อมูลจึงเป็นส่วนประกอบสำคัญที่ช่วยให้ระบบสารสนเทศมีความสมบูรณ์ และปฏิบัติงานได้อย่างมีประสิทธิภาพ

1.1 ลักษณะของฐานข้อมูล (Database Characteristics) ฐานข้อมูลที่ดีมีดังนี้

1.1.1 Persistent ข้อมูลที่บันทึกในฐานข้อมูลต้องคงอยู่ตลอดไป ทั้งนี้ เนื่องจากข้อมูลในฐานข้อมูลต่าง ๆ นั้น จะถูกบันทึกไว้ในหน่วยความจำสำรอง เช่น ดิสก์ ทำให้ข้อมูลเหล่านั้น

มีความคงทนถาวร ไม่เกิดการสูญหาย ยกเว้นกรณีที่มีความเสียหายเกิดขึ้นกับหน่วยความจำสำรอง เช่น ดิสก์เสียหาย (Disk failure)

อย่างไรก็ตาม พบว่า การที่ระบบยอมให้ข้อมูลทั้งหมดของระบบคงอยู่ในฐานข้อมูลของระบบตลอดไปนั้น จะทำให้ฐานข้อมูลของระบบมีขนาดใหญ่มาก โดยเฉพาะระบบที่มีข้อมูลเพิ่มมากขึ้นตลอดเวลา (Growth) ดังนั้น ผู้บริหารข้อมูล หรือผู้บริหารฐานข้อมูลระบบ ต้องกำหนดระยะเวลาที่จะให้ข้อมูลคงอยู่ในฐานข้อมูลของระบบ

1.1.2 Shared ข้อมูลในฐานข้อมูลนั้นต้องสามารถใช้งานร่วมกันได้โดยผู้ใช้หลายคน (Multiuser) ทำให้ผู้ใช้แต่ละคนไม่จำเป็นต้องเก็บข้อมูลไว้เป็นของตัวเอง ช่วยลดความซ้ำซ้อนของข้อมูล เช่น มหาวิทยาลัยมีฐานข้อมูลทะเบียนนักศึกษา (Personnel database) ซึ่งเป็นฐานข้อมูลที่เก็บรายละเอียดของนักศึกษา ข้อมูลในฐานข้อมูลนี้จะถูกเรียกใช้โดยผู้ใช้ต่าง ๆ ได้แก่ อาจารย์ที่ปรึกษาเรียกดูผลการเรียน หรือการลงทะเบียนเรียน ฝ่ายทะเบียนวัดผลตรวจสอบผลการเรียน เป็นต้น

1.1.3 Interrelated ข้อมูลต่าง ๆ ในฐานข้อมูลต้องมีความสัมพันธ์ระหว่างกัน เช่น มหาวิทยาลัยมีฐานข้อมูลนักศึกษา ซึ่งเก็บรายละเอียดเกี่ยวกับนักศึกษา และฐานข้อมูลผลการเรียน ซึ่งเก็บรายละเอียดเกี่ยวกับผลการเรียนในรายวิชาต่าง ๆ เมื่ออาจารย์ที่ปรึกษาต้องการแนะนำการลงทะเบียนให้กับนักศึกษาต้องสามารถดูข้อมูลผลการเรียนที่ผ่านมา และรายวิชาที่เปิดเรียน จะเห็นว่าระบบฐานข้อมูลของมหาวิทยาลัยจำเป็นต้องสร้างความสัมพันธ์ระหว่างฐานข้อมูล นักศึกษา และฐานข้อมูลรายวิชาเรียน เพื่อให้อาจารย์ที่ปรึกษาสามารถให้คำแนะนำได้อย่างมีประสิทธิภาพ

1.1.4 ความสามารถในการจัดเก็บข้อมูล (Data storage) การเรียกค้น (Retrieve) การแก้ไขเปลี่ยนแปลงข้อมูล (Update) การลบข้อมูล (Delete) และการสร้างข้อมูลเพิ่มเติม (Add)

1.1.5 การมีข้อมูลซ้ำซ้อนกันน้อยที่สุด (Minimum redundancy) เป็นคุณลักษณะที่สำคัญประการหนึ่งของฐานข้อมูล การนำข้อมูลทั้งหมดมารวมกัน ตัดส่วนที่ซ้ำกันทิ้งไป ให้เหลืออยู่เพียงแห่งเดียว จะทำให้ปัญหาข้อมูลไม่ตรงกันลดน้อยลงหรือหมดไปได

1.1.6 ความถูกต้องของข้อมูล (Data integrity) รักษาความสัมพันธ์ (Relationship) ของข้อมูลภายในฐานข้อมูลให้มีความถูกต้องอยู่เสมอ ฐานข้อมูลมีระบบคอยตรวจสอบกฎเกณฑ์ หรือเงื่อนไขต่าง ๆ (Integrity rules) ทุกครั้งที่มีการแก้ไขข้อมูล หรือเพิ่มเติมข้อมูลเข้าไปในระบบฐานข้อมูลนั้น

1.1.7 ความเป็นอิสระของข้อมูล (Data independence) หมายถึง การที่โปรแกรมเป็นอิสระจากการเปลี่ยนแปลงโครงสร้างข้อมูลทางด้านกายภาพ (Physical data independence) และทางด้านตรรกะภาพ (Logical data independence) ทำให้แก้ปัญหาความขัดแย้งของข้อมูลได้

1.1.8 ความปลอดภัยของข้อมูล (Data security) ฐานข้อมูลจะมีระบบรักษาความปลอดภัยของข้อมูลสูง โดยระบบบริหารจัดการฐานข้อมูลจะกำหนดสิทธิ์ในการใช้งานของผู้ใช้ เช่นตรวจสอบรหัสผ่าน (Password) ของผู้ใช้ก่อนที่จะเข้าสู่ระบบ หลังจากนั้นจะตรวจสอบว่าผู้ใช้มีสิทธิใช้ข้อมูลได้มากน้อยเพียงใด ระบบบริหารจัดการฐานข้อมูลจะไม่ยอมให้โปรแกรมใด ๆ เข้าถึงข้อมูลได้โดยไม่ผ่านระบบ

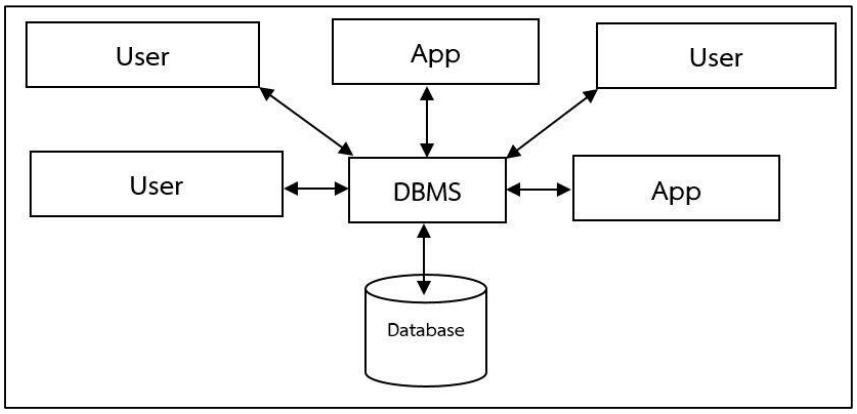
1.1.9 การสำรองข้อมูลและการกู้คืนข้อมูล (Data backup and recovery) การสำรองข้อมูลและการกู้คืนข้อมูล เป็นสิ่งจำเป็นอย่างยิ่งของระบบฐานข้อมูล เพื่อให้ผู้ใช้ระบบ

ฐานข้อมูลมั่นใจว่า ข้อมูลที่จัดเก็บอยู่ในเครื่องคอมพิวเตอร์ไม่ได้เสียหาย ยังมีความสมบูรณ์อยู่ตลอดเวลา ผู้บริหารฐานข้อมูลสามารถใช้คำสั่งสำรองข้อมูลและคำสั่งกู้คืนข้อมูลได้

2. ระบบจัดการฐานข้อมูล (Database management system: DBMS)

ระบบจัดการฐานข้อมูล หรือแทนด้วยอักษรย่อภาษาอังกฤษ DBMS คือ กลุ่มของข้อมูลที่เกี่ยวข้องกัน และชุดของโปรแกรมที่จัดการข้อมูลเหล่านี้ นั่นคือ DBMS = ข้อมูล + โปรแกรมจัดการข้อมูล ตัวอย่างเช่น ORACLE, MS-SQL, DB2, SYBASE โดยทั่ว ๆ ไป กลุ่มข้อมูลนี้ เรียกว่า ฐานข้อมูล (Database) ซึ่งเป็นข้อมูลที่มีประโยชน์ เพื่อใช้ในหน่วยงานนั้น ๆ และจุดประสงค์หลักของ DBMS คือ เพื่ออำนวยความสะดวกในการเก็บ และเรียกใช้ข้อมูลเหล่านี้ โดยเฉพาะอย่างยิ่งข้อมูลจำนวนมาก และมีความสัมพันธ์ที่ซับซ้อน นอกจากนี้ จะต้องทนทานต่อเหตุการณ์ต่าง ๆ ที่อาจจะทำให้ข้อมูลเสียหาย เช่น การลักลอบใช้ข้อมูลโดยไม่ได้รับอนุญาต ความเสียหายที่เกิดจากฮาร์ดแวร์ การเข้าใช้ระบบพร้อม ๆ กันจากผู้ใช้หลายคน เป็นต้น (ชนวัฒน์ ศรีสอ้าน, 2553)

DBMS เป็นซอฟแวร์ที่ช่วยจัดการฐานข้อมูลของระบบ โดยมีเป้าหมายสำคัญ คือ เพื่อให้การเก็บข้อมูลลงฐานข้อมูล (Store) และการดึงข้อมูลออกจากฐานข้อมูล (Retrieve) เป็นไปด้วยความสะดวกสบาย (Convenience) และมีประสิทธิภาพ (Efficiency) โดยให้ความมั่นใจกับผู้ใช้ว่าข้อมูลของระบบมีความคงสภาพ (Integrity) และมีความมั่นคง (Consistency)



ภาพที่ 1.7 ระบบจัดการฐานข้อมูล (Database management system: DBMS)

DBMS ต้องสามารถตอบสนองความต้องการของผู้ใช้ในเรื่องต่าง ๆ เป็นต้นว่า Data persistency, System reliability และ Scalability

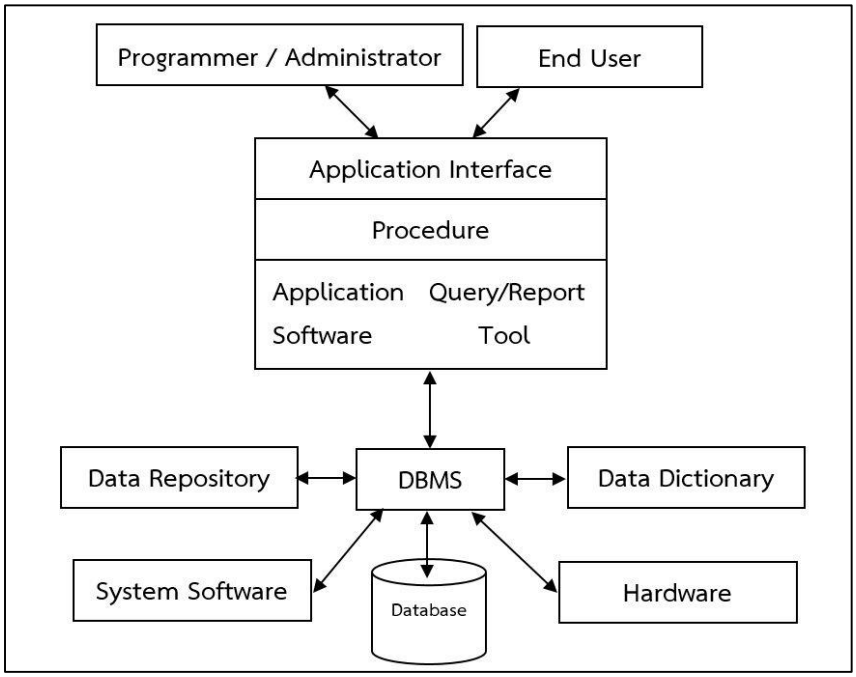
Data persistency หมายถึง การคงอยู่ของข้อมูล กล่าวคือ เมื่อข้อมูลได้ถูกบันทึกไว้ในฐานข้อมูลเรียบร้อยแล้ว ข้อมูลนั้นจะต้องคงอยู่ตลอดไป

System reliability หมายถึง ข้อมูลในฐานข้อมูลนั้นต้องมีความเชื่อถือได้ กล่าวคือ ข้อมูลที่ถูกเก็บในฐานข้อมูลต้องมีความถูกต้อง และเมื่อมีความเสียหายเกิดขึ้น ระบบต้องสามารถกู้คืนข้อมูลกลับคืนมาได้

Scalability หมายถึง ระบบต้องสามารถเก็บข้อมูลปริมาณมาก และสามารถรองรับผู้ใช้งานจำนวนมาก (สุจิตรา อดุลย์เกษม, 2553)

2.1 องค์ประกอบของระบบจัดการฐานข้อมูล

DBMS คือซอฟต์แวร์ที่ทำหน้าที่เป็นตัวกลางให้ผู้ใช้สามารถติดต่อกับฐานข้อมูลเพื่อทำการสร้าง เรียกดูและบำรุงรักษาฐานข้อมูล รวมถึงการจัดการควบคุมการเข้าถึงข้อมูล แต่ในขณะเดียวกัน ก็ยังมีส่วนประกอบทางสภาพแวดล้อมอื่น ๆ ที่เกี่ยวข้องกับระบบจัดการฐานข้อมูลอยู่หลายส่วนด้วยกัน ซึ่งส่วนประกอบเหล่านี้จะทำให้ระบบจัดการฐานข้อมูลมีความสมบูรณ์ยิ่งขึ้น สำหรับส่วนประกอบทางสภาพแวดล้อมของระบบจัดการฐานข้อมูล จะประกอบด้วย 5 ส่วนด้วยกัน คือ 1) ฮาร์ดแวร์ (Hardware) 2) ซอฟต์แวร์ (Software) 3) ข้อมูล (Data) 4) ขั้นตอนการปฏิบัติงาน (Procedure) และ 5) บุคลากร (Personnel)



ภาพที่ 1.8 องค์ประกอบระบบจัดการฐานข้อมูล

2.1.1 ฮาร์ดแวร์ (Hardware) หมายถึง อุปกรณ์ต่าง ๆ ที่ประกอบกันขึ้นเป็นเครื่องคอมพิวเตอร์มีลักษณะเป็นโครงร่างที่สามารถมองเห็นด้วยตาและสัมผัสได้ เช่น จานแม่เหล็ก หน่วยประมวลผล หน่วยความจำหลัก จอภาพ คีย์บอร์ด เครื่องพิมพ์ และ I/O device เป็นต้น ซึ่งสามารถแบ่งออกเป็นส่วนต่าง ๆ ตามลักษณะการทำงาน ได้ 4 หน่วย คือ หน่วยรับข้อมูล (Input Unit) หน่วยประมวลผลกลาง (Central processing unit: CPU) หน่วยแสดงผล (Output unit) และหน่วยเก็บข้อมูลสำรอง (Secondary storage) โดยอุปกรณ์แต่ละหน่วยมีหน้าที่การทำงานแตกต่างกัน ฮาร์ดแวร์ เป็นองค์ประกอบแรกของระบบสารสนเทศ ซึ่งฮาร์ดแวร์นี้ จะหมายถึงอุปกรณ์ต่าง ๆ ทางคอมพิวเตอร์ที่ ใช้ในการเก็บข้อมูลและประมวลผลข้อมูลเพื่อสร้างสารสนเทศขึ้น ได้แก่ เครื่องคอมพิวเตอร์ซึ่งอาจเป็นได้ตั้งแต่เครื่องระดับไมโครคอมพิวเตอร์ มินิคอมพิวเตอร์ไปจนถึงเมนเฟรมคอมพิวเตอร์ นอกจากนี้ สารสนเทศยังสามารถถูกเก็บอยู่ในระบบเครือข่าย (Network) ซึ่งเป็นการเชื่อมโยงไมโครคอมพิวเตอร์หลายตัวเข้าด้วยกัน

2.1.2 ซอฟต์แวร์ (Software) หมายถึงส่วนที่ไม่สามารถสัมผัสได้โดยตรง เป็นโปรแกรมหรือชุดคำสั่งที่ถูกเขียนขึ้นเพื่อสั่งให้เครื่องคอมพิวเตอร์ทำงาน จึงเปรียบเสมือนเป็นตัวเชื่อมระหว่างผู้ใช้ และเครื่องคอมพิวเตอร์ ถ้าไม่มีซอฟท์แวร์ก็จะไม่สามารถใช้คอมพิวเตอร์ทำอะไรได้ โดยทั่ว ๆ ไป ซอฟท์แวร์ จะเป็นโปรแกรมที่ถูกพัฒนาขึ้นมาเพื่อการทำงานในเรื่องใดเรื่องหนึ่งโดยเฉพาะ เช่น โปรแกรมการลงทะเบียนเรียนของนักศึกษา เป็นต้น ซึ่งอาจใช้ภาษาในการพัฒนาโปรแกรม เช่น ภาษาจาวา (JAVA) ภาษาซี (C) และภาษาพีเอชพี (PHP) เป็นต้น

2.1.3 ขั้นตอนปฏิบัติงาน (Procedure) เกี่ยวข้องกับชุดคำสั่งที่โปรแกรมเมอร์หรือผู้ที่ทำหน้าที่ดูแลฐานข้อมูล กำหนดขึ้นไว้เพื่อจัดการกับฐานข้อมูล รวมถึงคำสั่งที่ใช้ในการกำหนดความปลอดภัยให้กับฐานข้อมูล

2.1.4 ข้อมูล (Data) ได้แก่ข้อมูลทั้งหมดของระบบงานที่นำมาเก็บไว้ในฐานข้อมูลตามโครงสร้างที่กำหนดไว้ โดยข้อมูลทั้งหมดจะต้องมีความสัมพันธ์ซึ่งกันและกัน

2.1.5 บุคลากร (Personnel) ระบบสารสนเทศจะไม่สามารถปฏิบัติงานต่าง ๆ เองได้ ถ้าไม่มีคนเป็นผู้จัดการ ซึ่งสามารถแบ่งบุคลากรเกี่ยวกับระบบสารสนเทศออกเป็นประเภทต่าง ๆ ดังต่อไปนี้

♦ ผู้ใช้งาน (Users) จะเป็นผู้ให้ข้อมูลความต้องการในการนำคอมพิวเตอร์มาใช้งานในหน่วยงาน หรือเป็นผู้ใช้ระบบคอมพิวเตอร์ที่ได้พัฒนาขึ้น หรือใช้โปรแกรมประยุกต์อื่น เป็นผู้ที่นำสารสนเทศที่เกิดจากระบบคอมพิวเตอร์ไปใช้

♦ ผู้พัฒนาโปรแกรมประยุกต์ (Application programmer) ได้แก่ บุคลากรที่มีหน้าที่เกี่ยวกับการเขียนและพัฒนาโปรแกรมประยุกต์ต่าง ๆ เพื่อสั่งงานให้คอมพิวเตอร์ทำการประมวลผลและสร้างสารสนเทศในระบบงานใด ๆ เป็นต้น โดยการนำผลที่นักวิเคราะห์ระบบได้ออกแบบไว้ ผู้พัฒนาโปรแกรมประยุกต์จะต้องทำการทดสอบ แก้ไขโปรแกรม ติดตั้งและบำรุงรักษาโปรแกรมที่พัฒนาขึ้น

♦ ผู้ดูแลระบบ (System administrator) จะทำหน้าที่ควบคุมระบบด้าน ฮาร์ดแวร์ เช่น ดูแลระบบฐานข้อมูล ควบคุมเครื่องคอมพิวเตอร์ให้สามารถปฏิบัติงานได้อย่างราบรื่น ไม่มีปัญหาหรือคอยแก้ไขปัญหาที่อาจจะเกิดขึ้นจากการปฏิบัติงานของคอมพิวเตอร์ ตลอดจนบำรุงรักษาและแก้ไขปัญหาที่อาจเกิดขึ้นกับฐานข้อมูลคอมพิวเตอร์ขององค์การ

2.2 ประโยชน์ของระบบฐานข้อมูล

2.2.1 ลดความซ้ำซ้อนของข้อมูล เนื่องจากข้อมูลมีการจัดเก็บเป็นฐานข้อมูลไว้ที่ส่วนกลาง ทำให้ลดความซ้ำซ้อนและช่วยแก้ปัญหาความขัดแย้งกันของข้อมูลได้

2.2.2 การบริหารจัดการฐานข้อมูลทำได้ง่าย เนื่องจากมีการจัดเก็บข้อมูลไว้ที่ส่วนกลาง มีผู้บริหารฐานข้อมูล (Database administrator: DBA) ทำให้การจัดการข้อมูลทำได้ง่าย รวมทั้งสามารถกำหนดมาตรฐานของข้อมูลได้

2.2.3 สามารถใช้ฐานข้อมูลร่วมกันได้ เนื่องจากโครงสร้างการจัดเก็บข้อมูลในฮาร์ดดิสก์ถูกกำหนดด้วย DBMS และผู้ใช้แต่ละคนจะต้องใช้งานผ่าน DBMS

2.2.4 มีความเป็นอิสระระหว่างข้อมูลกับโปรแกรม เนื่องจากการใช้งานต่าง ๆ จะต้องใช้งานไว้เพียงที่เดียว จึงช่วยแก้ปัญหาความขัดแย้งกันของข้อมูลได้

2.2.5 มีความปลอดภัยของข้อมูลสูง เนื่องจากข้อมูลแต่ละข้อมูลจะมีความสำคัญไม่เท่ากัน ดังนั้น จึงต้องมีการกำหนดสิทธิในการใช้งาน โดยผู้บริหารฐานข้อมูลเป็นผู้กำหนดผู้มีสิทธิใช้งานข้อมูล เช่น รหัสผู้ใช้ และรหัสผ่าน ฐานข้อมูลเป็นศูนย์รวมของข้อมูลที่มีความสัมพันธ์กัน มีระบบการจัดหมวดหมู่ของข้อมูลที่มีแบบแผนซึ่งก่อให้เกิดฐานข้อมูลที่เป็นแหล่งรวมของข้อมูลจากแผนกต่าง ๆ และจัดเก็บไว้อย่างเป็นระบบภายในฐานข้อมูลชุดเดียว ผู้ใช้งานต่าง ๆ ในแต่ละแผนกสามารถใช้ข้อมูลส่วนกลางนี้เพื่อนำไปประมวลผลร่วมกันได้ และสนับสนุนการใช้งานฐานข้อมูลร่วมกันทำให้ไม่เกิดความซ้ำซ้อนในข้อมูล ระบบแฟ้มข้อมูลและแนวคิดของฐานข้อมูลสามารถช่วยแก้ไขปัญหาต่าง ๆ อันที่เกิดขึ้นจากการประมวลผลด้วยวิธีแฟ้มข้อมูลได้

สรุป

ข้อมูล (Data) คือ ข้อเท็จจริงเกี่ยวกับเรื่องต่าง ๆ ที่มีลักษณะเป็นตัวเลข ตัวอักษร สัญลักษณ์ ภาพ เสียง หรือมีลักษณะประสมกัน ประเภทของข้อมูลมีวิธีการแบ่งได้หลายวิธีตามเกณฑ์ในการจำแนก ได้แก่ 1) จำแนกตามลักษณะการเก็บข้อมูล ประกอบด้วย ข้อมูลที่ได้จากการนับ (Counting data) และข้อมูลที่ได้จาการวัด (Measurement data) ข้อมูลที่ได้จากการสังเกต (Observation data) ข้อมูลที่ได้จากการสัมภาษณ์ (Interview data) 2) จำแนกตามลักษณะข้อมูล ประกอบด้วย ข้อมูลเชิงปริมาณ (Quantitative data) ข้อมูลเชิงคุณภาพ (Qualitative data) 3) จำแนกตามการจัดการข้อมูล ประกอบด้วย ข้อมูลดิบ (Raw data) ข้อมูลที่จัดเป็นกลุ่ม (Group data) 4) จำแนกตามแหล่งที่มาของข้อมูล ประกอบด้วย ข้อมูลปฐมภูมิ (Primary Data) ข้อมูลทุติยภูมิ (Secondary data) 5) แบ่งตามมาตรของการวัด ประกอบด้วย มาตรวัดนามบัญญัติ (Nominal scale) มาตรวัดอันดับ (Ordinal scale) มาตรวัดแบบช่วง (Interval scale) มาตรวัดอัตราส่วน (Ratio scale) 6) แบ่งตามเวลาของการเก็บรวบรวมข้อมูล ประกอบด้วย ข้อมูลอนุกรมเวลา (Time-series data) ข้อมูลภาคตัดขวาง (Cross-sectional data) 7) แบ่งตามสภาพของข้อมูลที่เกี่ยวข้องกับกลุ่มตัวอย่าง ประกอบด้วย ข้อมูลส่วนบุคคล (Personal data) ข้อมูลสิ่งแวดล้อม (Environmental data) ข้อมูลพฤติกรรม (Behavioral data) 8) จำแนกตามการนำไปใช้กับคอมพิวเตอร์ ประกอบด้วย ข้อมูลตัวเลข (Numeric data) ข้อมูลตัวอักษร (Text data) ข้อมูลเสียง (Audio data) ข้อมูลภาพ (Images data) ข้อมูลภาพเคลื่อนไหว (Video data)

ระบบแฟ้มข้อมูล (File system) ประกอบด้วยโครงสร้างเรียงลำดับจากหน่วยที่เล็กที่สุดไปยังหน่วยที่ใหญ่ขึ้น คือ บิท (Binary digit หรือ Bit) ไบท์ (Byte) หรืออักขระ ฟิลด์ (Field) หรือเขตข้อมูล เรคคอร์ด (Record) หรือระเบียน แฟ้มข้อมูล หรือไฟล์ (File) ปัญหาสำคัญของการรวบรวมข้อมูลต่าง ๆ ไว้ด้วยกันโดยใช้ระบบแฟ้มข้อมูล คือ ข้อมูลมีความซ้ำซ้อน (Data redundancy) และข้อมูลมีความสัมพันธ์กัน (Data dependence)

ระบบฐานข้อมูล (Database system) ช่วยให้มีการจัดการข้อมูลปริมาณมากได้อย่างมีประสิทธิภาพ (Efficiency) และเกิดความสะดวกสบายสำหรับผู้ใช้ (Convenience) ระบบฐานข้อมูลประกอบด้วย ฐานข้อมูล (Database) และระบบจัดการฐานข้อมูล (Database management system: DBMS) ฐานข้อมูล เป็นการจัดเก็บข้อมูลที่มีความสัมพันธ์กัน นำมาเก็บรวบรวมให้อยู่ในที่เดียวกันอย่างเป็นระบบ ทำให้ผู้ใช้สามารถใช้ข้อมูลที่เกี่ยวข้องร่วมกันได้ เป็นการลดความซ้ำซ้อนของข้อมูล ข้อมูลมีความถูกต้องน่าเชื่อถือ เป็นมาตรฐานเดียวกัน และมีความปลอดภัย

ระบบจัดการฐานข้อมูล (Database management system) เป็นซอฟต์แวร์จัดการฐานข้อมูล ที่มีความสัมพันธ์กันตั้งแต่ 2 ฐานข้อมูลขึ้นไป รวมเข้าไว้ด้วยกันอย่างเป็นระบบ หรือหมายถึงระบบที่รวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกันเข้าไว้ด้วยกันเป็นกลุ่มของข้อมูล เพื่อช่วยให้การเข้าถึงข้อมูลในฐานข้อมูลเป็นไปด้วยความสะดวกสบายและมีประสิทธิภาพ โดยให้ความมั่นใจกับผู้ใช้ได้ว่า ข้อมูลของระบบจะมีความคงสภาพ (Data integrity) มีความต้องกัน (Consistency) และมีความน่าเชื่อถือ (Reliability)

บทที่ 2
การพัฒนาฐานข้อมูล

ฐานข้อมูลมีความสำคัญต่อระบบสารสนเทศที่สนับสนุนการทำงานของหน่วยงานต่าง ๆ ขององค์กร ไม่ว่าจะเป็นการเก็บ การจัดการฐานข้อมูลให้ปลอดภัย และความน่าเชื่อถือของข้อมูล ทำให้ผู้ใช้ข้อมูลสามารถนำข้อมูลไปช่วยในการวางแผนการตัดสินใจได้ทันเวลาอย่างมีประสิทธิภาพ การออกแบบฐานข้อมูลขององค์กรขนาดกลางหรือใหญ่ จะซับซ้อนมากกว่าองค์กรขนาดเล็ก เนื่องจากต้องตอบสนองกับระบบประยุกต์ใช้งาน (Application system) ของผู้ใช้จำนวนมาก การออกแบบฐานข้อมูลที่ดีเป็นสิ่งที่จำเป็นมากเพื่อให้ได้ผลตรงตามความต้องการของผู้ใช้ ระบบสารสนเทศที่มีคุณค่านั้น จะต้องตอบสนองต่อผู้ใช้แต่ละระดับ จากการที่ฐานข้อมูลเป็นส่วนหนึ่งของระบบสารสนเทศ (Information system) ดังนั้น การออกแบบระบบฐานข้อมูลจำเป็นต้องพิจารณาถึงวิธีการที่จะได้มาของข้อมูลหรือสารสนเทศที่มีคุณภาพ การจัดระบบที่ดีต้องมีการวิเคราะห์ความต้องการความจำเป็นของระบบโดยยึดถือขอบข่ายงาน การวางแผนและการตัดสินใจการบริหารงานในองค์กรเป็นหลัก ในการผลิตสารสนเทศนั้น ต้องคำนึงถึงความแม่นยำครอบคลุม รวดเร็วทันต่อเหตุการณ์และสามารถเรียกใช้ได้ในเวลาที่ต้องการ การจัดระบบสารสนเทศเพื่อให้สอดคล้องกับวัตถุประสงค์ขององค์กร จะต้องเริ่มต้นจากการจัดทำฐานข้อมูลขององค์กรให้สมบูรณ์ ในบทที่ 1 ได้กล่าวถึงความรู้พื้นฐานของระบบฐานข้อมูล ซึ่งเป็นส่วนหนึ่งของระบบสารสนเทศ ดังนั้น ในบทนี้ จะกล่าวถึงการพัฒนาฐานข้อมูล (Database development) โดยนำเสนอเนื้อหาสำหรับการศึกษาดังต่อไปนี้

- ♦ ระบบสารสนเทศ (Information dystem)
- ♦ วงจรการพัฒนาระบบ (System development life cycle: SDLC)
- ♦ กระบวนการพัฒนาฐานข้อมูล
  - ◊ เป้าหมายการพัฒนาฐานข้อมูล (Goals of database development)
  - ◊ วัฏจักรฐานข้อมูล (Database life cycle: DBLC)

ระบบสารสนเทศ (Information System)

การประยุกต์ใช้ฐานข้อมูล ถูกใช้เป็นส่วนหนึ่งของระบบเทคโนโลยีสารสนเทศที่ซึ่งจะให้บริการในด้านการจัดเก็บและการค้นคืนข้อมูล แต่สำหรับระบบเทคโนโลยีสารสนเทศจะมีวัตถุประสงค์ที่จะอำนวยความสะดวกให้แก่ผู้ใช้งานในการปรับเปลี่ยนข้อมูลให้กลายเป็นสารสนเทศ และจะมีฟังก์ชัน สำหรับจัดการ/จัดเก็บข้อมูลและสารสนเทศ ระบบเทคโนโลยีสารสนเทศจะประกอบไปด้วยบุคคลในหน้าที่ต่าง ๆ ฮาร์ดแวร์ ซอฟต์แวร์ฐานข้อมูลแอพพลิเคชัน/ โปรแกรม และกระบวนการ

สารสนเทศ หมายถึง การแสดงหรือชี้แจงข่าวสารข้อมูลต่าง ๆ (พจนานุกรมฉบับราชบัณฑิตยสถาน, 2554)

ข้อมูลสารสนเทศไม่ได้จำกัดเฉพาะเพียงตัวเลขเพียงอย่างเดียวเท่านั้น แต่อาจจะประกอบด้วยข้อมูลเอกสาร เสียง หรือรูปภาพต่าง ๆ แต่จัดเนื้อเรื่องให้อยู่ในรูปที่มีความหมาย แหล่งที่มาของข้อมูลสารสนเทศ อาจจะเป็นข้อมูลที่เกิดขึ้นภายในองค์กร หรือข้อมูลที่เกิดขึ้นภายนอกองค์กร หรือข้อมูลของหน่วยงานอื่นก็ได้

1. ระบบสารสนเทศ (Information System)

ระบบสารสนเทศ หมายถึง กระบวนการประมวลผลข้อมูลดิบที่มีอยู่ ให้อยู่ในรูปของข้อมูลหรือสารสนเทศที่เป็นประโยชน์สูงสุด เพื่อเป็นข้อสรุปที่ใช้สนับสนุนการตัดสินใจของผู้บริหาร การประมวลผลข้อมูลต่าง ๆ ในสมัยก่อนจะใช้คนเป็นผู้ดำเนินการ มีการเก็บรวบรวมในแฟ้มข้อมูล ระบบสารสนเทศรุ่นแรก ๆ มักจะเป็นการเก็บในรูปแฟ้มเอกสาร ที่แยกเป็นหมวดหมู่โดยมีผู้รับผิดชอบกับข้อมูลนั้น ๆ ต่อมาได้มีการคิดค้นเครื่องพิวเตอร์ขึ้น เพื่อช่วยในการประมวลผลข้อมูลที่รวดเร็วแม่นยำ ทำให้ระบบสารสนเทศสมัยใหม่เริ่มเกิดขึ้นนับแต่นั้นเป็นต้นมา

คอมพิวเตอร์ที่มีความสามารถสูงในการประมวลผลข้อมูล ทำให้การเก็บรวบรวม และการประมวลผลสามารถสร้างระบบสารสนเทศที่มีประสิทธิภาพ กระบวนการที่ทำให้เกิดสารสนเทศเรียกว่า การประมวลผลสารสนเทศ (Information processing) และเรียกวิธีการประมวลผลสารสนเทศด้วยเครื่องมือทางอิเล็กทรอนิกส์ว่า เทคโนโลยีสารสนเทศ ซึ่งหมายถึงเทคโนโลยีทุกรูปแบบที่นำมาประยุกต์ ในการประมวลผลข้อมูล การจัดเก็บ การสื่อสารและการส่งผ่านสารสนเทศ โดยระบบทางกายภาพจะประกอบไปด้วยคอมพิวเตอร์ อุปกรณ์ติดต่อสื่อสาร และระบบเครือข่าย และระบบนามธรรมจะเกี่ยวข้องกับการจัดรูปแบบการปฏิสัมพันธ์ด้านสารสนทศ ทั้งภายในและภายนอกระบบให้ดำเนินร่วมกันได้อย่างประสิทธิภาพ

1.1 คุณลักษณะของสารสนเทศที่ดี

1.1.1 มีความถูกต้องแม่นยำ (Accuracy) ไม่มีความผิดพลาด เชื่อถือได้

1.1.2 มีความเป็นปัจจุบัน (Up to date) ทันต่อการเปลี่ยนแปลงที่ดำเนินไปอย่างรวดเร็ว เพื่อการนำไปใช้ประโยชน์ได้จริง

1.1.3 ทันเวลา (Timely) ทันต่อความต้องการของผู้ใช้

1.1.4 มีความสมบูรณ์ (Complete) เที่ยงตรง (Relevant) เชื่อถือได้ (Reliable)

1.1.5 สอดคล้องกับความต้องการของผู้ใช้ ผู้ใช้มีความพึงพอใจ

1.1.6 มีรูปแบบการนำเสนอ (Presentation) ที่มีประโยชน์ เข้าใจง่าย เหมาะสมกับผู้ใช้ หรือผู้ที่เกี่ยวข้อง

1.1.7 สามารถตรวจสอบหรือพิสูจน์ได้ (Verifiable)

1.1.8 มีความกะทัดรัด (Conciseness) ชัดเจน (Clarity) ไม่คลุมเครือ

1.1.9 สะดวกในการเข้าถึง (Accessible) ผู้ใช้สามารถเข้าถึงข้อมูลได้ง่าย ในรูปแบบและเวลาที่เหมาะสมตามความต้องการของผู้ใช้

1.1.10 มีความยืดหยุ่น (Flexible) สามารถนำไปใช้ได้กับบุคคลหลายกลุ่ม หลายเป้าหมาย หรือหลายวัตถุประสงค์

1.1.11 มีความปลอดภัย (Secure) มีระบบรักษาความปลอดภัย เพื่อป้องกันการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต ในการเข้าถึงของผู้ไม่มีสิทธิใช้สารสนเทศ

1.2 ประโยชน์ของสารสนเทศและระบบสารสนเทศ

1.2.1 ลดความซ้ำซ้อน

1.2.2 ทำให้เกิดความคิด ความรู้และความเข้าใจ

1.2.3 ใช้ในการวางแผนการบริหารงานและประกอบการตัดสินใจ

1.2.4 เข้าถึงสารสนเทศที่ต้องการได้อย่างรวดเร็วทันต่อความต้องการ

1.2.5 ช่วยในการวางแผนและกำหนดเป้าหมายในการดำเนินงาน

1.2.6 ช่วยในการตรวจสอบประเมินผลการดำเนินงาน

1.2.7 ลดค่าใช้จ่าย ลดขั้นตอนในการทำงาน

1.2.8 เพิ่มประสิทธิภาพและศักยภาพในการแข่งขันทางธุรกิจ

2. เทคโนโลยีสารสนเทศ (Information technology)

เทคโนโลยีสารสนเทศ คือ การประยุกต์ใช้คอมพิวเตอร์และอุปกรณ์โทรคมนาคมเพื่อจัดเก็บ ค้นหา ส่งผ่าน และจัดดำเนินการข้อมูล ซึ่งมักเกี่ยวข้องกับธุรกิจหนึ่งหรือองค์การอื่น ๆ โดยปกติก็ใช้แทนความหมายของเครื่องคอมพิวเตอร์ และเครือข่ายคอมพิวเตอร์ และยังรวมไปถึงเทคโนโลยีการกระจายสารสนเทศอย่างอื่น เช่น โทรทัศน์ และโทรศัพท์ อุตสาหกรรมหลายอย่างเกี่ยวข้องกับเทคโนโลยีสารสนเทศ (วิกิพีเดียสารานุกรมเสรี, 2559)

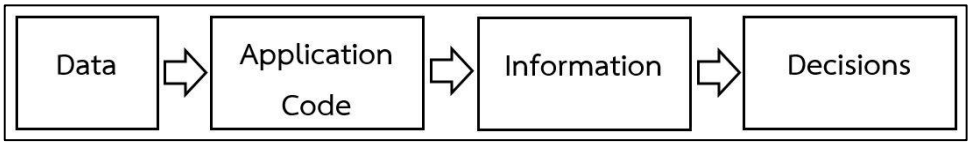
2.1 ความสำคัญของเทคโนโลยีสารสนเทศ

ในปัจจุบันเทคโนโลยีสารสนเทศมีการเปลี่ยนแปลงอย่างมาก เมื่อเข้าสู่ยุคสังคมดิจิทัลที่มีการติดต่อสื่อสารกันด้วยความรวดเร็วและซับซ้อน ตลอดจนมีการแข่งขันที่รุนแรง ทำให้เทคโนโลยีสารสนเทศไม่ได้เป็นเพียงเครื่องมือสนับสนุนในการดำเนินกิจกรรมต่าง ๆ ให้สะดวกและรวดเร็วขึ้นเท่านั้น แต่ได้กลายเป็นส่วนหนึ่งในชีวิตประจำวันของมนุษย์ มีความสำคัญ และเกี่ยวข้องกับคนทุกระดับ ทั้งในระดับประเทศ ระดับองค์กร หรือระดับบุคคล ดังนี้

2.1.1 ระดับประเทศ ตามนโยบายเศรษฐกิจและสังคมดิจิตอลของรัฐบาลให้มีการนำเทคโนโลยีสารสนเทศที่ทันสมัย และหลากหลายมาเปลี่ยนแปลงวิธีการดำเนินชีวิตของประชาชน การดำเนินธุรกิจ การดำเนินงานของภาครัฐ ซึ่งจะส่งผลให้ประชาชนมีความรอบรู้สามารถพัฒนา และใช้ประโยชน์จากเทคโนโลยีสารสนเทศได้อย่างรู้เท่าทัน มีโอกาสในการสร้างรายได้และมีคุณภาพชีวิตที่ดีขึ้น เกิดความมั่งคั่งทางเศรษฐกิจที่แข่งขันได้ในเวทีโลก และความมั่นคงทางสังคมของประเทศ ต่อไประดับองค์กร ในยุคของสังคมดิจิตอลที่ทำให้สามารถทำงานได้ทุกสถานที่ และทุกเวลาตลอด 24 ชั่วโมง ทั้งองค์กรภาครัฐและเอกชน ต้องพบกับข้อมูลจำนวนมหาศาลอย่างหลีกเลี่ยงไม่ได้ ดังนั้นความสามารถในการทำงานผ่านเครือข่ายอินเตอร์เน็ตความเข้าใจเครือข่ายสังคมออนไลน์การซื้อสินค้าและบริการทางอินเตอร์เน็ต และการวิเคราะห์ข้อมูลอย่างชาญฉลาด จึงเป็นสิ่งจำเป็นเพื่อความอยู่รอดขององค์กร การดำเนินงานในองค์กรจึงต้องมีเทคโนโลยีสารสนเทศเพื่อเป็นเครื่องมือ

ช่วยในการปฏิบัติงานที่รวดเร็ว ช่วยเพิ่มผลผลิต ลดต้นทุน ใช้เป็นกลยุทธ์เพื่อความได้เปรียบในการแข่งขัน สร้างความพึงพอใจกับลูกค้าหรือกลุ่มเป้าหมาย และเกิดประสิทธิภาพในการดำเนินงาน

2.1.2 ระดับบุคคล สำหรับคนทั่วไปเทคโนโลยีสารสนเทศช่วยให้เข้าถึงข้อมูลข่าวสาร สามารถพัฒนาตนเองให้เป็นคนที่ฉลาด รู้เท่าทันสื่อ เท่าทันโลก และยังเป็นเครื่องมือในการสร้างศักยภาพของบุคคล ช่วยยกระดับคนไปสู่สังคมแห่งการเรียนรู้และสามารถนำไปใช้ในการประกอบอาชีพได้อย่างมีคุณภาพ นอกจากนี้ เทคโนโลยีสารสนเทศยังเป็นส่วนหนึ่งในดำเนินชีวิตประจำวันของคนปัจจุบัน เช่น ดูรายการโทรทัศน์ผ่านเครือข่ายสังคมออนไลน์ ซื้อสินค้าผ่านทางอินเตอร์เน็ต จองตั๋วเดินทางแบบออนไลน์ การลงทะเบียนหรือดูผลการเรียนทางเว็บไซต์เป็นต้น



ภาพที่ 2.1 ระบบเทคโนโลยีสารสนเทศเพื่อการตัดสินใจ

ภายใต้กรอบการทำงานระบบเทคโนโลยีสารสนเทศ จะเป็นการปรับเปลี่ยนข้อมูลให้กลายเป็นสารสนเทศที่พร้อมสำหรับสนับสนุนการตัดสินใจ โดยสารสนเทศที่ได้จะอยู่ในรูปแบบของรายงาน ตารางรูปภาพ ชาร์ตและกราฟต่าง ๆ ดังแสดงในภาพที่ 2.1 ที่แสดงถึงแอพพลิเคชันที่ประกอบด้วย 2 ส่วนหลัก ๆ คือ ข้อมูลและโค้ด หรือคำสั่งของโปรแกรมที่ใช้ในการปรับเปลี่ยนข้อมูลให้กลายเป็นสารสนเทศ เพื่อนำไปใช้ในกระบวนการและการดำเนินการต่าง ๆ

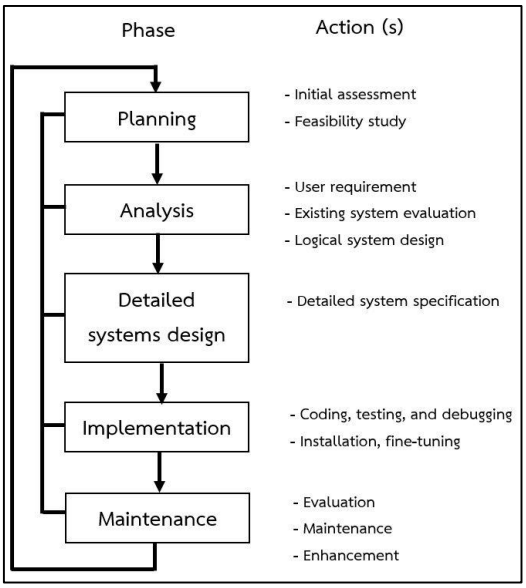
การพัฒนาฐานข้อมูล จำเป็นต้องอาศัยวัฏจักรฐานข้อมูลเป็นแนวทางการดำเนินงานที่ประกอบไปด้วยขั้นตอนต่าง ๆ ตามช่วงวัฏจักรฐานข้อมูล (Database life cycle: DBLC) โดยเฉพาะขั้นตอนสำคัญ คือ “การออกแบบฐานข้อมูล” โดยทั่วไปฐานข้อมูลมักถูกพัฒนาขึ้นพร้อม ๆ กับการพัฒนาระบบงานสารสนเทศ ซึ่งส่วนใหญ่จะนิยมใช้ “วงรอบชีวิตการพัฒนาระบบ (System development life cycle: SDLC)” เป็นแนวทางการดำเนินงานเช่นกัน การพัฒนาฐานข้อมูลเป็นขั้นตอนหนึ่งในการพัฒนาระบบ ดังนั้น จึงกล่าวได้ว่า DBDLC และ SDLC มีความสอดคล้องสัมพันธ์กัน (พนิดา พานิชกุล และณัฐพงษ์ วารีประเสริฐ, 2552)

วงจรการพัฒนาระบบ (System development life cycle: SDLC)

วงจรการพัฒนาระบบ (Systems development life cycle) เป็นวงจรที่แสดงถึงกิจกรรมต่าง ๆ ในแต่ละขั้นตอน ตั้งแต่ต้นจนเสร็จเป็นระบบงานที่ใช้ได้ ซึ่งนักวิเคราะห์ระบบต้องทำความเข้าใจว่า ในแต่ละขั้นตอนทำอะไรและทำอย่างไร โดยมีอยู่ 5 ขั้นตอน ได้แก่ 1) ขั้นตอนการวางแผน (Planning) 2) ขั้นตอนการวิเคราะห์ (Analysis) 3) ขั้นตอนการออกแบบและพิจารณารายละเอียด (Detailed systems design) 4) ขั้นตอนการสร้างระบบ (Implementation) และ 5) ขั้นตอนการดูแลรักษาระบบ (Maintenance)

วงจร SDLC จะเป็นการทำงานแบบทำซ้ำ ตัวอย่างเช่น ในการทำงานรอบแรก เราอาจทำการศึกษาขั้นตอนการทำงานองค์กร จากนั้นทำการลงรายละเอียดการออกแบบเบื้องต้น จากนั้น

ทำการประเมินว่ามีส่วนใดที่ยังไม่สมบูรณ์ ทำการพิจารณาใหม่หลาย ๆ รอบ จนกระทั่งได้ระบบที่มีการดำเนินงานครบถ้วน ที่สามารถตอบสนองต่อเป้าหมายได้



ภาพที่ 2.2 วงจรการพัฒนาระบบ (Coronel, Morris & Rob, 2012)

1. การวางแผนระบบ (Systems planning)

ขั้นตอนแรกของการพัฒนาระบบ จำเป็นต้องพิจารณาและกำหนดปัญหาของระบบ และมีการวางแผนสร้างระบบสารสนเทศ หรือแก้ไขระบบสารสนเทศที่มีอยู่เดิม โดยพิจารณาความต้องการทางด้านฮาร์ดแวร์ ซอฟต์แวร์ บุคลากร ตลอดจนค่าใช้จ่ายและเวลาที่ต้องใช้ในการพัฒนาระบบ รวมถึงความเป็นไปได้ที่ระบบงานนั้นจะสามารถพัฒนาให้สำเร็จได้ภายใต้เงื่อนไขต่าง ๆ

2. การวิเคราะห์ระบบ (Systems analysis)

เป็นการวิเคราะห์ระบบงานปัจจุบัน รวบรวมความต้องการในด้านต่าง ๆ เพื่อนำไปใช้ออกแบบระบบในขั้นตอนต่อไป ในการรวบรวมความต้องการของผู้ใช้นั้นอาจใช้วิธีการที่แตกต่างกันไป เช่น ศึกษาเอกสารที่มีอยู่ (Documentation) สังเกตการณ์การทำงานในปัจจุบัน (Observation) สัมภาษณ์ผู้ใช้งานระบบในระดับต่าง ๆ (Interview) ฯลฯ

3. การออกแบบระบบ (Systems design)

ในขั้นตอนการวิเคราะห์ระบบนั้น เป็นการหาคำตอบว่า “จะต้องทำอะไร (What)” แต่ในขั้นตอนการออกแบบระบบจะเป็นการหาคำตอบว่า “จะต้องทำอย่างไร (How)” ดังนั้น ขั้นตอนนี้จะเป็นการออกแบบระบบใหม่ เพื่อให้สอดคล้องกับความต้องการของผู้ใช้ โดยที่นักวิเคราะห์ระบบต้องกำหนดว่า จะต้องมีโปรแกรมอะไรบ้าง รวมถึงโครงสร้างของโปรแกรม และการเชื่อมโยงระหว่างโปรแกรม ฯลฯ

4. การอิมพลีเมนต์ระบบ (Systems implementation)

ขั้นตอนนี้โปรแกรมเมอร์จะเขียนโปรแกรมตามข้อมูลที่ได้จากขั้นตอนการออกแบบระบบ และทำการทดสอบโปรแกรมว่าทำงานถูกต้องหรือไม่ โดยมีการทดสอบกับข้อมูลจริง ถ้าทุกอย่าง

เรียบร้อย เราจะได้โปรแกรมที่พร้อมที่จะนำไปใช้งานจริงต่อไป หลังจากนั้นต้องเตรียมคู่มือการใช้และการฝึกอบรมผู้ใช้งานจริงของระบบ

5. การบำรุงรักษา (Maintenance)

หลังจากที่โปรแกรมถูกนำไปใช้งานได้ระยะหนึ่ง อาจจะต้องมีการแก้ไขโปรแกรม อันเนื่องมาจากพบปัญหาในการทำงานของโปรแกรม (Bug) และการดำเนินงานในองค์กร มีการเปลี่ยนแปลง

กระบวนการพัฒนาฐานข้อมูล

การออกแบบฐานข้อมูล จำเป็นต้องดำเนินการอย่างเป็นระบบ เริ่มตั้งแต่การวิเคราะห์ความต้องการ การออกแบบระบบและการนำระบบไปใช้งานซึ่งจะแสดงให้เห็นประสิทธิภาพการทำงานของระบบ กล่าวคือ การออกแบบฐานข้อมูล มีผลต่อการนำข้อมูลไปใช้ในระบบงานต่าง ๆ ขององค์กร ลักษณะการออกแบบฐานข้อมูลและปัจจัยต่าง ๆ ที่เกี่ยวข้อง อาทิเช่น ระบบจัดการฐานข้อมูล (Database management system: DBMS) หรือระบบปฏิบัติการ (Operating systems) ที่เลือกใช้ ล้วนแต่มีความสำคัญต่อประสิทธิภาพการทำงาน หากมีการออกแบบฐานข้อมูลที่ดี การที่จะต้องปรับประสิทธิภาพของระบบต่าง ๆ ในภายหลัง ก็อาจจะมีน้อยลง (Performance tuning) อาทิเช่น การเพิ่มประสิทธิภาพระบบงาน (Application tuning) หรือการเพิ่มประสิทธิภาพระบบปฏิบัติการ (Operating systems tuning) ดังนั้น การออกแบบฐานข้อมูลจึงมีบทบาทสำคัญอย่างยิ่งโดยเฉพาะการนำข้อมูลจากส่วนต่าง ๆ ขององค์กรมารวบรวมออกแบบอย่างเป็นระบบ เพื่อสามารถใช้ในการดำเนินงานได้อย่างมีประสิทธิภาพ (ศิริลักษณ์ โรจนกิจอำนวย, 2552)

1. เป้าหมายการพัฒนาฐานข้อมูล (Goals of database development)

การพัฒนาฐานข้อมูลของระบบใด ๆ นั้นมีเป้าหมาย (Goal) ที่สำคัญคือ ต้องการสร้างฐานข้อมูลที่รวบรวมข้อมูลสำหรับผู้ใช้จำนวนมากในองค์กร (Organization) โดยที่ข้อมูลในฐานข้อมูลต้องเป็นข้อมูลที่มีคุณภาพ (High quality) ที่สอดคล้องกับนโยบาย (Policy) ขององค์กร และมีการควบคุมการเข้าถึงข้อมูลอย่างมีประสิทธิภาพ ดังนั้น ในการพัฒนาฐานข้อมูล เราจึงต้องมีการวางเป้าหมายการทำงานเพื่อให้ได้สิ่งต่าง ๆ ต่อไปนี้

1.1 พัฒนากลุ่มคำสามัญ (Development a common vocabulary)

การทำงานของแต่ละองค์กรประกอบด้วยหน่วยงานย่อย เช่น แผนก ฝ่าย ฯลฯ แต่ละหน่วยงานย่อยมีความจำเป็นและมีความต้องการเก็บรายละเอียดข้อมูลแตกต่างกัน และข้อมูลเหล่านั้นอาจมีรูปแบบ (Format) ของข้อมูลที่เหมือนกันหรือแตกต่างกันไป เช่น แผนกขายต้องการเก็บข้อมูลชื่อลูกค้าและหมายเลขโทรศัพท์ลูกค้า แผนกบัญชีต้องการเก็บข้อมูลชื่อลูกค้า ที่อยู่ลูกค้าและหมายเลขโทรศัพท์ลูกค้า ดังนั้น การพัฒนาฐานข้อมูลของระบบจำเป็นต้องกำหนดรายละเอียดต่าง ๆ ของข้อมูลในองค์กรให้เป็นกลุ่มคำสามัญ (Common vocabulary) ที่มีรูปแบบเดียวกันทั้งหมด โดยที่ผู้ออกแบบฐานข้อมูล (Database designer) ต้องรวบรวมความต้องการข้อมูลของหน่วยงานต่าง ๆ ในองค์กร และพิจารณาสร้างรูปแบบสามัญข้อมูลที่องค์กรต้องการ โดยคำนึงถึงนโยบายขององค์กรเป็นประการสำคัญ และลดความแตกต่างตลอดจนความขัดแย้งที่เกิดขึ้น

ในการสร้างกลุ่มคำสามัญเหล่านั้น เป็นไปได้ที่ผู้ใช้ในองค์กรมีความต้องการแตกต่างกัน มีความคิดเห็นที่ขัดแย้งกัน ผู้ออกแบบฐานข้อมูลจำเป็นต้องประนีประนอม (Compromise) ผู้ใช้ใน

แต่ละหน่วยงานขององค์กร ซึ่งเป็นสิ่งที่ทำได้ยาก ต้องใช้เวลา แต่ถ้าสามารถทำได้ จะช่วยเพิ่มประสิทธิภาพการทำงานของระบบหรือองค์กรนั้น ๆ

1.2 กำหนดความหมายของข้อมูล (Define the meaning of data) ในการกำหนดความหมายของข้อมูลในองค์กรนั้น ผู้ออกแบบฐานข้อมูลต้องกำหนดถึงเงื่อนไขของระบบ หรือองค์กรที่สอดคล้องกับนโยบายขององค์กรนั้น ๆ ซึ่งเป็นการกำหนด Business rules ของระบบนั่นเอง

ตัวอย่างเช่น ระบบห้องสมุด (Library System) ของมหาวิทยาลัย ABC กำหนดให้แบ่งสมาชิก (Member) ออกเป็น 4 ประเภท ดังนี้

- ทั่วไป (General)
- นักศึกษา (Student)
- อาจารย์ (Lecturer)
- เจ้าหน้าที่ (Staff)

เมื่อระเบียบห้องสมุดกำหนดเรื่องการยืม-คืนของสมาชิกแต่ละประเภท ดังนี้

- ทั่วไป สามารถยืมหนังสือได้ไม่เกิน 5 เล่ม ภายใน 5 วัน
- นักศึกษา สามารถยืมหนังสือได้ไม่เกิน 7 เล่ม ภายใน 7 วัน
- อาจารย์ สามารถยืมหนังสือได้ไม่เกิน 30 เล่ม ภายใน 1 ภาคการศึกษา
- เจ้าหน้าที่ สามารถยืมหนังสือได้ไม่เกิน 10 เล่ม ภายใน 10 วัน

1.3 คุณภาพข้อมูล (Ensure data quality)

ในการออกแบบฐานข้อมูลนั้น เราต้องการให้มีการรวบรวมข้อมูลที่มีคุณภาพไว้ในฐานข้อมูลขององค์กร

ตารางที่ 2.1 แสดงคุณสมบัติที่สำคัญของข้อมูลที่มีคุณภาพ

คุณสมบัติ	ความหมาย
สมบูรณ์ (Completeness)	ฐานข้อมูลต้องบรรจุข้อมูลที่จำเป็นต่อการใช้งานของระบบ
ชัดเจนไม่กำกวม (Lack of ambiguity)	ข้อมูลที่บรรจุในฐานข้อมูลของระบบนั้น ต้องมีความหมายเดียว ไม่มีความกำกวม ซึ่งจะทำให้เกิดความสับสนในการตีความ
ถูกต้อง (Correctness)	ข้อมูลที่บรรจุในฐานข้อมูลต้องมีความถูกต้อง
ทันเวลา (Timeliness)	ข้อมูลของระบบที่มีการเปลี่ยนแปลง จะถูกบันทึกในฐานข้อมูลอย่างทันเวลา ไม่เกิดการล่าช้า
เชื่อถือได้ (Reliability)	เมื่อมีข้อขัดข้องใด ๆ เกิดขึ้นกับระบบ จะไม่ส่งผลกระทบต่อข้อมูลที่บรรจุในฐานข้อมูล
ถูกต้องตรงกัน (Consistency)	ข้อมูลที่บรรจุในฐานข้อมูลต้องถูกต้องตรงกัน ไม่ขัดแย้งกัน

องค์กรที่มีข้อมูลที่มีคุณภาพบรรจุในฐานข้อมูล จะช่วยให้ได้การทำงานที่มีประสิทธิภาพ มีการตัดสินใจที่ดีและเหมาะสม ส่งผลต่อคุณภาพและประสิทธิภาพการทำงานขององค์กร

ในการสร้างฐานข้อมูลที่บรรจุข้อมูลที่มีคุณภาพนั้น ผู้ออกแบบฐานข้อมูลจำเป็นต้องคำนึงถึงเรื่อง “Cost-benefit Trade-off” ด้วย เช่น ระบบงานทะเบียนนักศึกษาที่มีการแสดงรายละเอียดเกี่ยวกับรายวิชาและจำนวนนักศึกษาที่ลงทะเบียนเรียนในรายวิชานั้น ๆ ของแต่ละภาค

การศึกษา กรณีที่ระบบต้องการแสดงข้อมูลที่มีคุณภาพ ผู้ออกแบบฐานข้อมูลจำเป็นต้องพิจารณาและตัดสินใจว่าจะให้ระบบปรับเปลี่ยน (Update) ข้อมูลรายวิชาบ่อยครั้งเท่าไร ซึ่งอาจจะเป็นการปรับเปลี่ยน 1 ครั้งต่อวัน หรือ 2 ครั้งต่อวัน หรือ 1 ครั้งต่อสัปดาห์ ฯลฯ ดังนั้น ผู้ออกแบบฐานข้อมูลต้องตัดสินใจเลือกระหว่างค่าใช้จ่ายที่เกิดขึ้นในการปรับเปลี่ยนข้อมูล และการแสดงข้อมูลกับความพึงพอใจของผู้ใช้ระบบที่ได้เห็นข้อมูลที่เป็นปัจจุบันมากที่สุด

1.4 การอิมพลีเมนต์ที่มีประสิทธิภาพ (Find an efficient implementation)

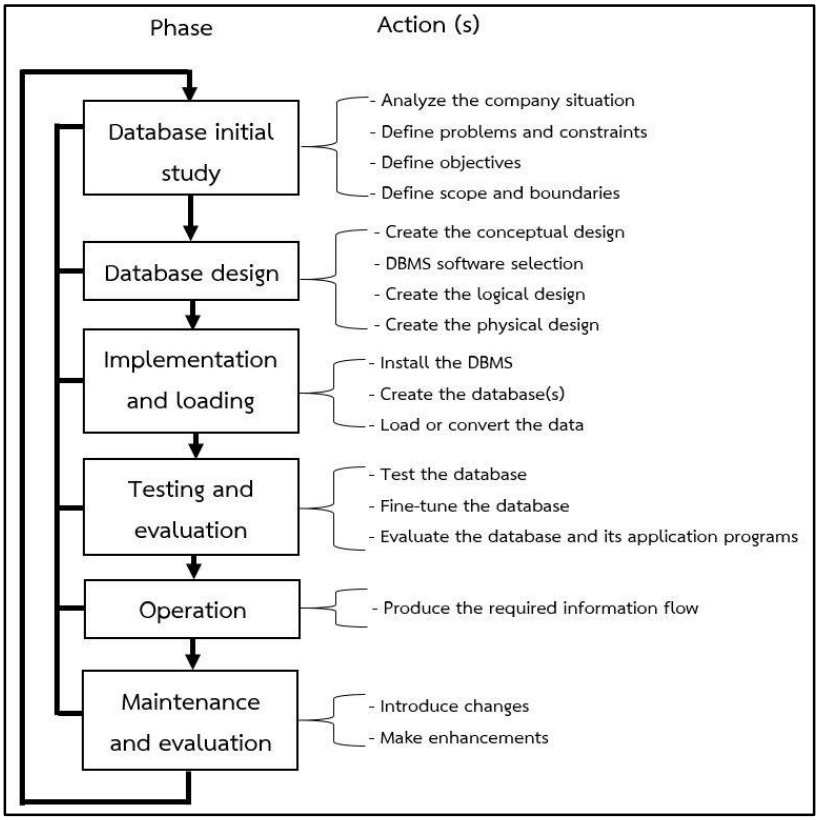
เป้าหมายที่สำคัญที่สุดของการพัฒนาฐานข้อมูล คือ การหาวิธีการอิมพลีเมนต์ที่มีประสิทธิภาพ โดยที่วิธีดังกล่าวนี้ ต้องสอดคล้องกับเป้าหมายของระบบให้มากที่สุด แต่จำเป็นต้องพิจารณาข้อจำกัดต่าง ๆ ของระบบประกอบกันไปด้วยในกรณีที่ต้องการรับประกันประสิทธิภาพการทำงานของระบบนั้น เราจำเป็นต้องมีการติดตามประสิทธิภาพของระบบอย่างสม่ำเสมอ เพื่อจะได้ปรับเปลี่ยนการออกแบบระบบเพื่อให้ได้ประสิทธิภาพการทำงานตามความต้องการ (สุจิตรา อดุลย์เกษม, 2552)

2. วัฏจักรฐานข้อมูล (The database life cycle: DBLC)

การพัฒนาฐานข้อมูล จำเป็นต้องมีแนวทางในการดำเนินงาน “วัฏจักรฐานข้อมูล” ก็เป็นแนวทางหนึ่งที่นิยมนำมาใช้กำหนดขั้นตอนการพัฒนาฐานข้อมูลตามช่วงชีวิตของฐานข้อมูล

วัฏจักรฐานข้อมูล (The database life cycle: DBLC) เป็นขั้นตอนในการพัฒนา หรือจัดทำระบบฐานข้อมูล ซึ่งประกอบด้วย 6 ขั้นตอน ดังภาพตัวอย่าง ประกอบไปด้วยขั้นตอน (Step) หรือระยะ (Phase) การดำเนินงานต่าง ๆ ที่ถูกกำหนดไว้อย่างเป็นระบบเพื่อการออกแบบและพัฒนาระบบฐานข้อมูลให้แล้วเสร็จ ในแต่ละขั้นตอนของ DBLC จะประกอบไปด้วยกิจกรรมหลักหลายกิจกรรม ในกิจกรรมหลักประกอบด้วยกิจกรรมย่อย โดยที่แต่ละขั้นตอน ไม่จำป็นต้องรอให้ขั้นตอนก่อนหน้าดำเนินการเสร็จสิ้นก่อนก็ได้ กล่าวคือ ในระหว่างที่ขั้นตอนหนึ่งกำลังถูกดำเนินการ ทีมงานสามารถดำเนินขั้นตอนอื่นขนานกันไปได้ นอกจากนี้ บางขั้นตอนยังสามารถวนรอบกลับมาแก้ไขปรับปรุงหรือทำเพิ่มได้ (Iterative) จึงทำให้ปัจจุบันวงรอบการพัฒนามีหลายรูปแบบแตกต่างกันออกไป

สำหรับวัฏจักรฐานข้อมูลโดยทั่วไป ประกอบไปด้วย 6 ขั้นตอน ประกอบด้วย 1) การศึกษาฐานข้อมูลขององค์กรเบื้องต้น (Database initial study) 2) การออกแบบฐานข้อมูล (Database design) 3) การสร้างฐานข้อมูลและการโหลดข้อมูลเข้าสู่ฐานข้อมูล (Implementation and loading) 4) การทดสอบและประเมินการทำงานของฐานข้อมูล (Testing and evaluation) 5) การประยุกต์ใช้ฐานข้อมูล (Operation) และ 6) การดูแลรักษาและพัฒนาให้ดียิ่งขึ้น (Maintenance and evolution) ตามลำดับ ดังภาพที่ 2.3



ภาพที่ 2.3 วัฏจักรฐานข้อมูล (The database life cycle: DBLC) (Coronel, Morris & Rob, 2012)

2.1 การศึกษาฐานข้อมูลขององค์กร (Database initial study)

ในการศึกษาเกี่ยวกับฐานข้อมูลขององค์กรเพื่อทำการปรับปรุง หรือสร้างระบบเทคโนโลยีสารสนเทศขึ้นใหม่ ผู้ออกแบบฐานข้อมูลต้องเข้าใจถึงสาเหตุและเหตุผลที่ระบบที่ใช้อยู่ในปัจจุบันเกิดความล้มเหลว ซึ่งจะสามารถทำความเข้าใจได้การพูดคุย สัมภาษณ์ และ/ หรือสอบถามจากผู้ใช้งานระบบ การศึกษาดังกล่าวจะมีวัตถุประสงค์หลัก 3 ประเด็น ดังนี้

- การวิเคราะห์สถานการณ์ขององค์กร (Analysis of the company situation)
- การระบุถึงปัญหาและเงื่อนไขต่าง ๆ (Define problems and constraints)
- การระบุถึงวัตถุประสงค์และขอบเขต (Define objectives, scopes and boundaries)

จากการศึกษาทั้ง 3 ประเด็น ทำให้เราทราบถึงวัตถุประสงค์ของการพัฒนาระบบฐานข้อมูล โดยแต่ประเด็นมีรายละเอียดดังนี้

2.1.1 การวิเคราะห์สถานการณ์ขององค์กร

สถานการณ์ขององค์กรจะบ่งบอกถึงการดำเนินการขององค์กร โครงสร้างขององค์กร และเป้าหมายขององค์กร ด้วยเหตุนี้ จึงเป็นเหตุให้ผู้ออกแบบฐานข้อมูลจะต้องพิจารณาถึงการดำเนินการพื้นฐานและเป้าหมายที่ต้องดำเนินการ และยังรวมถึงการพิจารณาถึงโครงสร้างหลักขององค์กร การวิเคราะห์สถานการณ์ขององค์กรทำให้ทราบถึงลำดับขั้นของพนักงาน และความเกี่ยวเนื่องกันของพนักงานจากสายงานต่าง ๆ ซึ่งการทราบถึงโครงสร้างขององค์กรจะมี

ประโยชน์ในการที่จะระบุถึงความต้องการของตำแหน่งงาน ความต้องการรายงานและคิวรี (Query) ที่เกิดจากหน้าที่ในการทำงาน

2.1.2 การระบุถึงปัญหาและเงื่อนไข

ถ้าองค์กรที่ต้องการออกแบบฐานข้อมูลนั้นได้ดำเนินการทำการมาเป็นเวลานานจะทำให้เราทราบถึงฟังก์ชันการทำงานต่าง ๆ ของระบบดั้งเดิม ประกอบด้วยอินพุตที่ถูกป้อนเข้าไปในระบบ เอกสารหรือรายงานที่ระบบทำการสร้าง รวมถึงวิธีการใช้เอาท์พุตและบุคคลที่ใช้เอาท์พุต อย่างไรก็ตาม ในการระบุถึงปัญหานั้นจะทำได้ค่อยข้างยาก เนื่องจากผู้ใช้งานอาจไม่สามารถระบุปัญหาที่เกิดขึ้นได้ครอบคุลม หรืออาจไม่สามารถระบุถึงปัญหาที่แท้จริงในระหว่างการทำงาน ด้วยเหตุนี้จึงทำให้ผู้ออกแบบฐานข้อมูลจะจต้องทำการจัดเก็บคำอธิบายถึงปัญหาอย่างกว้าง ๆ หลังจากทราบถึงปัญหาและคำอธิบายถึงปัญหาในเบื้องต้นแล้ว ผู้ออกแบบฐานข้อมูลจะต้องไตร่ตรองอย่างละเอียดถี่ถ้วนจึงจะได้ข้อมูลสำหรับระบุถึงปัญหา เมื่อทราบถึงปัญหาที่แน่ชัดแล้ว จะต้องทำการค้นหาคำตอบหรือวิธีการแก้ปัญหาเหล่านั้นที่จะต้องสอดคล้องกับแนวทาง หรือวิธีการในการดำเนินงานขององค์กร

2.1.3 วัตถุประสงค์และระบุขอบเขต

หลังจากการระบุถึงปัญหาและเงื่อนไขแล้ว ผู้ออกแบบฐานข้อมูลจะต้องทำให้แน่ใจได้ว่าวัตถุประสงค์ของการสร้างระบบฐานข้อมูล หรือระบบเทคโนโลยีสารสนเทศมีความสอดคล้องกับความต้องการหรือ ปัญหาของผู้ใช้งานหรือไม่ ในการที่จะมีความแน่ใจเกี่ยวกับคำถามข้างต้น ผู้ออกแบบฐานข้อมูลควรที่จะต้องพิจาณาและหาคำตอบสำหรับคำถามดังต่อไปนี้

- วัตถุประสงค์เบื้องต้นของระบบที่จะทำการสร้างขึ้นใหม่คืออะไร
- ระบบที่จะทำการสร้างขึ้นใหม่สามารถเชื่อมต่อกับระบบเดิมที่มีอยู่ หรือสามารถเชื่อมต่อกับระบบที่จะมีการสร้างขึ้นในอนาคตได้หรือไม่
- ระบบที่จะทำการสร้างขึ้นใหม่มีความสามารถในการแบ่งปันข้อมูลให้กับระบบอื่นหรือผู้ใช้งานอื่น ๆ ได้หรือไม่

จากคำถามทั้ง 3 ข้างต้นทำให้ผู้ออกแบบฐานข้อมูลเข้าใจถึงวัตถุประสงค์ที่แท้จริง และลักษณะ/ คุณลักษณะของระบบที่จะสร้างขึ้นใหม่

ขอบเขตของระบบจะเป็นตัวกำหนดขอบเขตของการออกแบบฐานข้อมูลสำหรับการดำเนินการ ด้วยเหตุนี้ จำเป็นที่จะต้องทราบอย่างแน่ชัดว่าในการออกแบบแต่ละครั้ง เป็นการออกแบบสำหรับทั้งองค์กรหรืออาจเป็นการออกแบบสำหรับแผนกงานเดียวหรือหลายแผนก หรือเป็นการออกแบบสำหรับฟังก์ชันการทำงานภายใต้แผนกงานใดงานหนึ่งเท่านั้น การทราบถึงขอบเขตจะช่วยในการกำหนดถึงโครงสร้างข้อมูลที่ต้องการใช้ชนิดและจำนวนเอ็นทิตี้ที่ต้องการใช้

ในส่วนของขอบเขตทางด้านเวลาในการพัฒนาระบบ ขอบเขต ทางด้านงบประมาณ และขอบเขตของฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการสร้างระบบ จำเป็นที่จะต้องทราบถึงขอบเขตของระบบและเงื่อนไข ทั้งในด้านเวลาและงบประมาณที่จะช่วยให้ตัดสินใจได้ว่าจะเลือกใช้ฮาร์ดแวร์ซอฟต์แวร์และส่วนประกอบอื่น ๆ อย่างไร

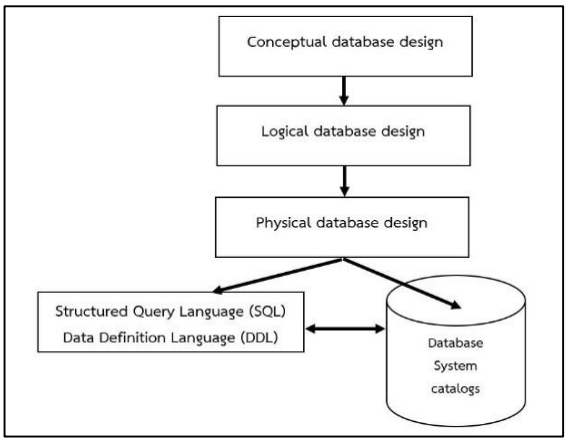
2.2 การออกแบบฐานข้อมูล (Database design)

ขั้นตอนการออกแบบฐานข้อมูลเป็นขั้นตอนสำคัญที่ใช้สำหรับออกแบบฐานข้อมูลเพื่อสนับสนุนการดำเนินการและวัตถุประสงค์ขององค์กร ภายใต้การบวนการออกแบบฐานข้อมูลผู้ออกแบบต้องให้ความใส่ใจกับลักษณะของข้อมูลที่จะทำการจัดเก็บฐานข้อมูล ในการพิจารณาข้อมูลจะมี 2 มุมมองที่เกี่ยวข้อง คือ มุมมองทางภารกิจขององค์กรที่มองข้อมูลเป็นแหล่งกำเนิดของ

สารสนเทศ และมุมมองของการออกแบบที่จะพิจารณาโครงสร้างของข้อมูล การเข้าถึง/ จัดการกับข้อมูล และกระบวนการในการปรับเปลี่ยนข้อมูลให้กลายเป็นสารสนเทศ

ขั้นตอนการออกแบบฐานข้อมูลจะประกอบไปด้วย 4 ขั้นตอนหลัก คือ 1) การออกแบบกรอบความคิด (Conceptual design) 2) การออกแบบเชิงตรรกะ (Logical design) 3) การออกแบบเชิงกายภาพ (Physical design) และ 4) การเลือกใช้ระบบจัดการฐานข้อมูล (DBMS selection)

การออกแบบฐานข้อมูล เริ่มจากการออกแบบกรอบความคิดแล้วทำการปรับเปลี่ยนไปเป็นเชิงตรรกะและการออกแบบทางกายภาพ โดยในแต่ละขั้นตอนจะเป็นการเพิ่มรายละเอียดเกี่ยวกับคุณลักษณะของข้อมูลโครงสร้างการจัดเก็บข้อมูล ความสัมพันธ์ และยังรวมถึงการจัดทำเอกสารเพื่อแสดงถึงรายละเอียดของฐานข้อมูล การออกแบบกรอบความคิดจะเกี่ยวข้องกับข้อมูลทั้งหมดในมุมมองของผู้ใช้งาน แต่ในส่วนของการออกแบบเชิงตรรกะเกี่ยวข้องกับข้อมูลภายใต้การประยุกต์ใช้ DBMS และท้ายสุด การออกแบบเชิงกายภาพจะเกี่ยวข้องกับข้อมูลที่ถูกดำเนินการโดยระบบปฏิบัติการและอุปกรณ์ฮาร์ดแวร์สำหรับการจัดเก็บข้อมูล



ภาพที่ 2.4 ขั้นตอนการออกแบบฐานข้อมูล

2.2.1 การออกแบบกรอบความคิด (Conceptual design)

การออกแบบกรอบความคิดจะเป็นขั้นตอนแรกของการออกแบบฐานข้อมูลที่มีเป้าหมายที่จะออกแบบฐานข้อมูลที่มีความอิสระจากซอฟต์แวร์ฐานข้อมูล และอุปกรณ์ต่าง ๆ ที่ใช้สำหรับจัดเก็บข้อมูล โดยผลลัพธ์ที่ได้จากการออกแบบกรอบความคิดจะเป็นแบบจำลองกรอบความคิด (Conceptual data model) ที่สามารถบ่งบอกได้ถึงเอ็นทิตี้ (Entity) แอทริบิว (Attribute) ความสัมพันธ์ (Relationship) และเงื่อนไขต่าง ๆ การสร้างแบบจำลองกรอบความคิดมักถูกใช้ในการกำหนดโครงสร้างของการจัดเก็บข้อมูลที่ซึ่งจะทำให้มีความเข้าใจในภารกิจและการดำเนินงานขององค์กร ในระหว่างการดำเนินการสร้างแบบจำลองกรอบความคิด การออกแบบจะระบุถึงข้อมูลทั้งหมดที่จะถูกใช้ในภารกิจขององกรค์ แต่ไม่ได้ระบุถึงชนิดของฮาร์ดแวร์และซอฟต์แวร์ที่จะใช้สำหรับจัดเก็บและจัดการกับข้อมูล (หมายเหตุ ด้วยเหตุนี้จึงทำให้การออกแบบแบบจำลองกรอบความคิดจะไม่ขึ้นกับฮาร์ดแวร์และซอฟต์แวร์ใด ๆ)

1) การวิเคราะห์ข้อมูลและความต้องการของผู้ใช้งาน

การออกแบบกรอบความคิดคือการระบุถึงลักษณะของข้อมูล โดยข้อมูลเหล่านี้จะถูกปรับเปลี่ยนไปเป็นสารสนเทศ ดังนั้น ผู้ออกแบบควรที่จะมุ่งเน้นที่

♦ Information needs ผู้ออกแบบจะต้องพิจารณาว่าข้อมูลสารสนเทศใดเป็นที่ต้องการของผู้ใช้และเอาท์พุตของระบบ คืออะไร

♦ Information users ผู้ออกแบบจะต้องพิจารณาว่าใครจะเป็นผู้ใช้ข้อมูลสารสนเทศ ข้อมูลสารสนเทศจะถูกใช้อย่างไร มุมมองข้อมูลของผู้ใช้มีอะไรบ้าง

♦ Information sources ผู้ออกแบบจะต้องพิจารณาว่าข้อมูลสารสนเทศจะสามารถค้นหาได้จากที่ใด เราจะสามารถสกัดข้อมูลสารสนเทศได้อย่างไร

♦ Information Constitution ผู้ออกแบบจะต้องพิจารณาว่าข้อมูลส่วนใดที่จะถูกประยุกต์ใช้เพื่อสร้างเป็นข้อมูลสารสนเทศ แอทริบิวของข้อมูลมีอะไรบ้าง ความสัมพันธ์ระหว่างข้อมูลมีอะไรบ้าง ปริมาณข้อมูลที่จะถูกจัดเก็บเป็นเช่นไร และข้อมูลจะถูกเข้าถึง/ ประยุกต์ใช้บ่อยแค่ไหน

คำถามข้างต้นจะเป็นคำถามที่สำคัญและจำเป็นที่จะต้องค้นหาคำตอบ โดยเราสามารถหาคำตอบได้โดยการดำเนินการดังนี้

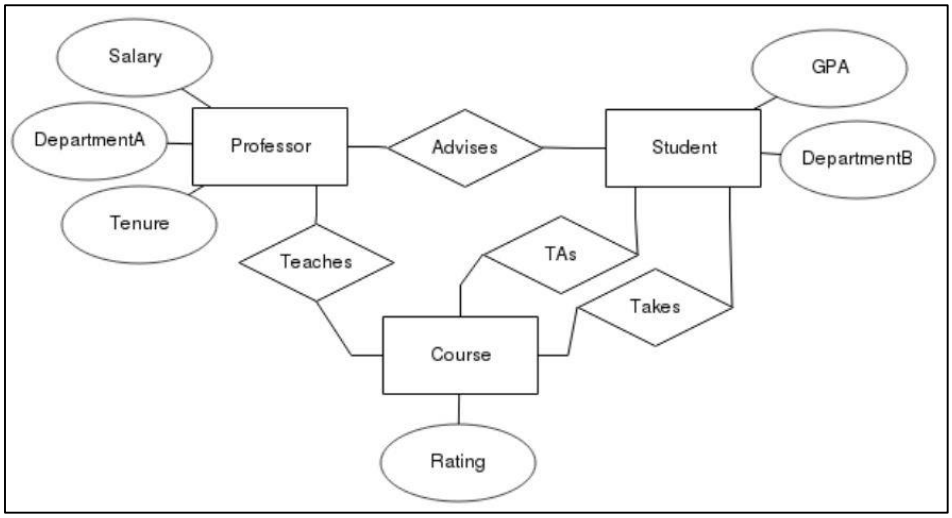
♦ การพัฒนาและรวบรวมมุมมองข้อมูลของผู้ใช้ จะช่วยให้เราสามารถระบุได้ถึงข้อมูลต่าง ๆ ที่จะถูกจัดเก็บในฐานข้อมูล

♦ การสังเกตุการทำงานของระบบปัจจุบัน: เอาท์พุตที่มีอยู่และเอาท์พุตที่ต้องการ จะทำให้เราทราบถึงรายงานต่าง ๆ ที่ผู้ใช้ต้องการและยังสามารถทราบถึงข้อมูลที่สนับสนุนการสร้างรายงานเหล่านั้น

♦ การพูดคุยกับทีมผู้พัฒนาระบบ อย่างที่เราทราบดีว่าการออกแบบฐานข้อมูลจะเป็นส่วนหนึ่งของ SDLC ที่ซึ่งผู้ออกแบบระบบก็จะทำหน้าที่ออกแบบกรอบความคิดด้วยเช่นกัน

2) การออกแบบเชิงแนวคิด

ในขั้นการออกแบบเชิงแนวคิด โดยการพัฒนาแบบจำลองอี-อาร์ (E-R Model) ที่ใช้อธิบายถึงความสัมพันธ์ระหว่างสิ่งที่เราสนใจจะจัดเก็บ ที่เรียกว่า เอนทิตี้ (Entity) และรายละเอียดหรือแอททริบิวต์ (Attribute) ของสิ่งที่จะจัดเก็บ แล้วทำการแปลงแบบจำลองอี-อาร์ โมเดล (E-R model) เป็นโครงสร้างตารางฐานข้อมูล จากนั้นทำการนอร์มัลไลเซชัน (Normalization) เพื่อให้ได้โครงสร้างของตารางที่ดี สามารถควบคุมความซ้ำซ้อนของข้อมูลหลีกเลี่ยงความผิดปกติของข้อมูลในการสร้างแบบจำลอง E-R นั้น ผู้ออกแบบระบบจำเป็นต้องสร้างแบบจำลอง E-R ที่ประกอบด้วยเอนทิตี้และความสัมพันธ์ทั้งหมดที่จำเป็นต้องมีในระบบ (All that is needed is there, and all that is there is needed.)



ภาพที่ 2.5 ตัวอย่างแบบจำลอง E-R ระบบการจัดการของสถาบันการศึกษา

2.2.2 การออกแบบเชิงตรรกะ (Logical design)

การออกแบบเชิงตรรกะเป็นขั้นตอนที่ 2 ของขั้นตอนการออกแบบฐานข้อมูลที่มีเป้าหมายที่จะทำการออกแบบฐานข้อมูลด้วยการประยุกต์จากแบบจำลองข้อมูล การออกแบบเชิงตรรกะจะเป็นขั้นตอนที่ต่อยอดมาจากขั้นตอนการออกแบบกรอบความคิด ด้วยการนำข้อมูลภายใต้กรอบความคิดมากำหนดโครงสร้างการจัดเก็บข้อมูล ตัวอย่างเช่น ถ้าต้องการออกแบบเชิงตรรกะด้วยการประยุกต์ใช้แบบจำลองข้อมูลเชิงสัมพันธ์ การออกแบบต้องระบุถึงตารางข้อมูล แอทริบิว ความสัมพันธ์ และเงื่อนไขต่าง ๆ เป็นต้น การออกแบบเชิงตรรกะจะประกอบไปด้วย 4 ขั้นตอนหลัก ดังนี้

- ♦ การปรับเปลี่ยนแบบจำลองกรอบความคิดให้กลายเป็นแบบจำลองเชิงตรรกะ
- ♦ การตรวจสอบแบบจำลองเชิงตรรกะด้วยการประยุกต์ใช้นอร์มัลไลเซชัน
- ♦ การตรวจสอบแบบจำลองเชิงตรรกะด้วยกฎความสมบูรณ์ต่าง ๆ (Integrity Constraints)
- ♦ การตรวจสอบแบบจำลองเชิงตรรกะกับความต้องการของผู้ใช้

ภายใต้ขั้นตอนการออกแบบทั้ง 4 ขั้นตอน ไม่จำเป็นต้องทำการออกแบบแบบเรียงลำดับ และสามารถดำเนินการซ้ำหลาย ๆ รอบเพื่อให้แน่ใจได้ว่าเราสามารถออกแบบแบบจำลองเชิงตรรกะที่มีความสมบูรณ์และถูกต้องตรงความต้องการของผู้ใช้ โดยรายละเอียดของแต่ละขั้นตอนจะสามารถแสดงได้ดังนี้

2.2.2.1 การปรับเปลี่ยนแบบจำลองกรอบความคิดให้กลายเป็นแบบจำลองเชิงตรรกะ

การออกแบบจำลองข้อมูลเชิงตรรกะจะเป็นการปรับเปลี่ยนแบบจำลองกรอบความคิดให้กลายเป็นแบบจำลองข้อมูลที่ผู้ออกแบบได้ทำการเลือกไว้ การปรับเปลี่ยนแบบจำลองกรอบความคิดให้กลายเป็นแบบจำลองข้อมูลเชิงสัมพันธ์เอ็นทิตี้ เป็นการนำข้อมูลจากแบบจำลองกรอบความคิดให้กลายเป็นตารางข้อมูล แอทริบิว ความสัมพันธ์ และเงื่อนไขต่าง ๆ โดยขั้นตอน

การปรับเปลี่ยนแบบจำลองกรอบความคิดให้กลายเป็นเซตของตารางข้อมูลจะประกอบไปด้วยขั้นตอนต่าง ๆ ดังนี้

- ♦ การพิจารณา Strong entity
- ♦ การพิจารณาความสัมพันธ์ระหว่างเอ็นทิตี้ Super type และ Subtype
- ♦ การพิจารณา Weak entity
- ♦ การพิจารณา Binary relationship
- ♦ การพิจารณา Higher degree relationship

2.2.2.2 การตรวจสอบแบบจำลองเชิงตรรกะด้วยการประยุกต์ใช้นอร์มัลไลเซชัน

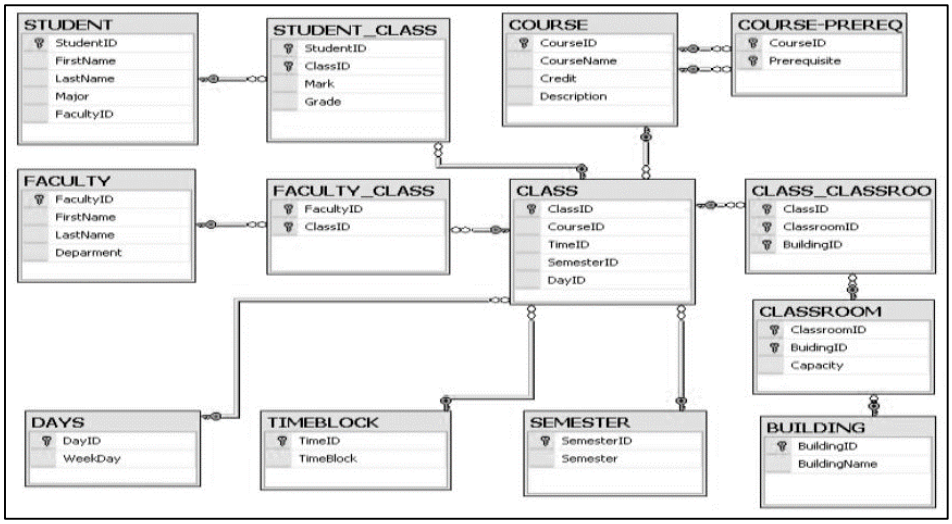
หลังจากทำการปรับเปลี่ยนแบบจำลองกรอบความคิดให้อยู่ในรูปแบบของแบบจำลองข้อมูลเชิงสัมพันธ์เอ็นทิตี้แล้ว เราจะได้แผนภาพเชิงสัมพันธ์เอ็นทิตี้ที่บ่งบอกถึงเอ็นทิตี้ แอทริบิว ความสัมพันธ์และ เงื่อนไขต่าง ๆ แต่อย่างไรก็ตามในเอ็นทิตี้หนึ่งอาจมีแอทริบิวที่มีลักษณะเป็น Composite และ/หรือ Multivalued attribute ที่ซึ่งจะต้องการการดำเนินการปรับเปลี่ยนให้เป็น Simple attribute และ/หรือ Single-valued attribute ตามลำดับ จากนั้นพิจารณาถึงความสัมพันธ์ของแอทริบิวในแต่ละเอ็นทิตี้ ด้วยการประยุกต์ใช้ขั้นตอนนอร์มัลไลเซชัน

2.2.2.3 การตรวจสอบแบบจำลองเชิงตรรกะด้วยกฎความสมบูรณ์

หลังจากการปรับเปลี่ยนแบบจำลองกรอบความคิดให้กลายเป็นแบบจำลองข้อมูลเชิงสัมพันธ์เอ็นทิตี้ แล้วเราจะต้องทำการพิจารณาถึงเงื่อนไขต่าง ๆ รวมถึงการกำหนดขอบเขตค่าของข้อมูลที่ปรากฏในแอทริบิว เพื่อที่จะทำให้ข้อมูลในเอ็นทิตี้มีความสอดคล้องกับเป้าหมาย นอกจากนั้นทำการพิจารณาเกี่ยวกับ Entity integrity และ Referential integrity เพื่อที่จะลดความผิดปกติของข้อมูล สุดท้ายทำการตรวจสอบสิทธิ์ในการเข้าถึง การใช้งานข้อมูลของผู้ใช้งานที่จะช่วยให้ข้อมูลถูกใช้งานอย่างเหมาะสมและมีความปลอดภัย

2.2.2.4 การตรวจสอบแบบจำลองเชิงตรรกะกับความต้องการของผู้ใช้

ขั้นตอนสุดท้ายของการออกแบบแบบจำลองข้อมูลเชิงตรรกะ จะเกี่ยวข้องกับการตรวจสอบ แบบจำลองที่ได้ทำการออกแบบไว้กับความต้องการของผู้ใช้งานที่ซึ่งจะต้องทำการร้องขอให้ผู้ใช้งานทำการตรวจสอบข้อมูล และมาตราการความปลอดภัยด้วยเช่นกัน โดยหลังจากผู้ใช้ทำการตรวจสอบอย่างละเอียดถี่ถ้วนแล้ว ทำให้ได้แบบจำลองข้อมูลเชิงตรรกะที่แสดงถึงโครงสร้างการจัดเก็บข้อมูลที่จะไม่ขึ้นกับฮาร์ดแวร์ และแบบจำลองข้อมูลนี้จะเป็นอินพุตของการออกแบบเชิงกายภาพต่อไป



ภาพที่ 2.6 การออกแบบฐานข้อมูลเชิงตรรกะ

2.2.3 การออกแบบเชิงกายภาพ (Physical design)

การออกแบบเชิงกายภาพ จะเป็นขั้นตอนการออกแบบองค์ประกอบการจัดเก็บข้อมูล และลักษณะการเข้าถึง การใช้งานข้อมูลเพื่อทำให้แน่ใจได้ว่าฐานข้อมูลที่สร้างขึ้น จะมีความสมบูรณ์ มีความปลอดภัยและมีประสิทธิภาพในการทำงาน การออกแบบเชิงกายภาพจะเป็นขั้นตอน ที่เกี่ยวข้องกับฮาร์ดแวร์ และจะต้องคำนึงถึงประสิทธิภาพการทำงานของระบบ ขั้นตอนการออกแบบเชิงกายภาพ ประกอบด้วย 3 ขั้นตอนหลัก ประกอบด้วย 1) การระบุถึงองค์ประกอบการจัดเก็บข้อมูล 2) การประเมินความสมบูรณ์และความปลอดภัย และ 3) การตรวจสอบประสิทธิภาพการทำงาน

2.2.3.1 การระบุองค์ประกอบการจัดเก็บข้อมูล

ก่อนการระบุถึงองค์ประกอบของการจัดเก็บข้อมูล ต้องพิจาณาถึงปริมาณข้อมูลที่ต้องจัดการและรูปแบบการใช้ข้อมูลจากฐานข้อมูล มีประโยชน์ ดังนี้

- การทราบถึงปริมาณข้อมูล ทำให้สามารถคำนวณพื้นที่ในการจัดเก็บข้อมูลในฐานข้อมูล
- การทราบถึงความถี่ในการเพิ่ม อัพเดท ค้นหาข้อมูลทำให้ทราบถึงรูปแบบการใช้งานข้อมูล

เมื่อเราทราบถึงปริมาณข้อมูลและรูปแบบข้อมูล สามารถดำเนินการ ดังนี้

- ทำการตรวจสอบพื้นที่สำหรับจัดเก็บข้อมูลในแต่ละตารางข้อมูลและการจัดองค์ประกอบของตารางข้อมูล
- ระบุถึงดัชนีที่จะใช้ในแต่ละตารางข้อมูล
- ระบุถึงมุมมองและประเภทของมุมมองที่จะใช้ในแต่ละตารางข้อมูล

2.2.3.2 การประเมินความสมบูรณ์และความปลอดภัย

หลังจากทำการระบุถึงองค์ประกอบของการจัดเก็บข้อมูลให้กับตารางข้อมูล การสร้างดัชนี และการสร้างมุมมองสำหรับตารางข้อมูลแล้ว ฐานข้อมูลจะมีความพร้อมให้ผู้ใช้สามารถ

ใช้งานได้ แต่ก่อนที่จะใช้งานผู้ใช้ควรที่จะต้องระบุยืนยันตัวตนเพื่อเข้าใช้งาน ด้วยเหตุนี้ต้องทำการพิจารณาถึง

- การกำหนดให้ผู้ใช้เข้าใช้งานและบทบาทของผู้ใช้งาน
- กำหนดความสามารถในการดำเนินการให้กับผู้ใช้ตามลำดับ

2.2.3.3 การตรวจสอบประสิทธิภาพการทำงาน

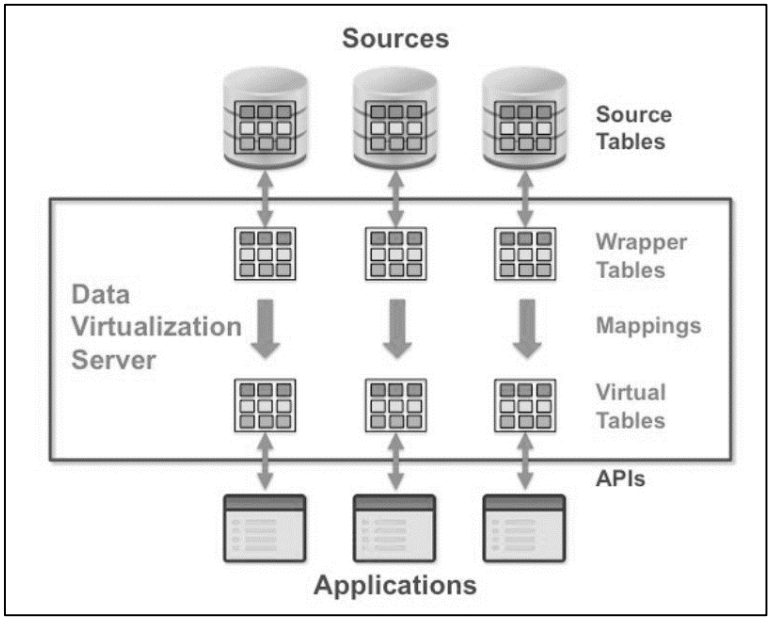
ก่อนเริ่มใช้งานต้องทำการตรวจสอบประสิทธิภาพการดำเนินการของฐานข้อมูล ซึ่งจะกี่ยวกับกระบวนการในการปรับฐานข้อมูลและคิวรีต่าง ๆ เพื่อเพิ่มประสิทธิภาพการดำเนินงานและช่วยให้ประสิทธิภาพการทำงานที่สามารถตอบสนองความต้องการของผู้ใช้งานได้

2.3 การสร้างฐานข้อมูลและการโหลดข้อมูลเข้าสู่ฐานข้อมูล (Implementation and loading)

หลังจากขั้นตอนการออกแบบฐานข้อมูล ทำให้ทราบถึงตารางข้อมูล (เอนทิตี้) แอทริบิว ความสัมพันธ์ ขอบเขต มุมมอง ดัชนี และรายละเอียดต่าง ๆ ที่เกี่ยวข้องกับการจัดเก็บข้อมูล จากนั้นเราจะทำการดำเนินการสร้างฐานข้อมูลตามที่ออกแบบไว้ โดยขั้นตอนการดำเนินการสร้างฐานข้อมูลจะประกอบไปด้วยขั้นตอนย่อยดังนี้

2.3.1 การติดตั้งระบบจัดการฐานข้อมูล

ในหลายครั้งของการสร้างระบบเทคโนโลยีสารสนเทศ หรือระบบฐานข้อมูล ทีมผู้สร้างอาจจำเป็นต้องทำการติดตั้งระบบจัดฐานข้อมูลในเซิร์ฟเวอร์ใหม่หรือเซิร์ฟเวอร์เดิมที่มีอยู่ แต่ ณ ปัจจุบันที่มผู้สร้างมักจะประยุกต์ใช้เทคนิค “Virtualization” ที่จะทำการสร้างเซิร์ฟเวอร์เสมือน พื้นที่ในการจัดเก็บข้อมูลเสมือน และ เครือข่ายเสมือน เพื่อที่จะทำให้การแสดงผลในเชิงตรรกะ จะไม่ขึ้นกับฮาร์ดแวร์ต่าง ๆ หลังจากการสร้างเซิร์ฟเวอร์เสมือนจะทำให้เราสามารถติดตั้งระบบจัดการฐานข้อมูลในเซิร์ฟเวอร์เสมือนได้ (Database virtualization)



ภาพที่ 2.7 ตัวอย่างระบบ Database virtualization

2.3.2 การสร้างฐานข้อมูล

หลังจากทำการติดตั้งระบบจัดการฐานข้อมูลจะเป็นหน้าที่ของผู้ดูแลระบบ (System Administration: SYSADM) และผู้ดูแล (Database administration: DBA) ที่จะทำการสร้างฐานข้อมูล และตารางข้อมูล ต่าง ๆ ในขั้นตอนแรก SYSADM สามารถทำการสร้างฐานข้อมูล และจากนั้น DBA จะทำการกำหนดสิทธิ์ในการใช้งานฐานข้อมูลให้กับผู้ใช้งาน

2.3.3 การนำข้อมูลเข้าสู่ฐานข้อมูล

หลังจากทำการสร้างฐานข้อมูลและตารางข้อมูลแล้ว ขั้นตอนต่อมาดำเนินการนำข้อมูลเข้าสู่ฐานข้อมูล (Upload) การ Upload ข้อมูลอาจทำการเคลื่อนย้ายข้อมูลจากระบบเดิมไปยังฐานข้อมูลที่สร้างขึ้นใหม่ ซึ่งจะอยู่ในรูปแบบฐานข้อมูลเชิงสัมพันธ์เหมือนกัน แต่ก็อาจจะมีบางกรณีที่ต้อง Upload ข้อมูลจากระบบต่าง ๆ หลายระบบที่ข้อมูลไม่ได้อยู่ในรูปแบบของฐานข้อมูลเชิงสัมพันธ์ เช่น ข้อมูลที่อยู่ในรูปแบบของแฟ้มข้อมูลต่าง ๆ สเปรดชีท หรือแม้แต่ข้อมูลที่ถูกจัดเก็บอยู่ในกระดาษ ด้วยเหตุนี้จึงจำเป็นที่จะต้องทำการสร้างโปรแกรมเพื่อทำการปรับเปลี่ยนข้อมูลที่อยู่ในรูปแบบต่าง ๆ ให้กลายเป็นข้อมูลที่อยู่ในรูปแบบของฐานข้อมูลเชิงสัมพันธ์ ซึ่งจะทำให้เราสามารถ Upload ข้อมูลเหล่านั้นเข้าสู่ฐานข้อมูลที่สร้างขึ้นได้ แต่อาจมีบางกรณีที่เราต้องทำการกรอกข้อมูลแบบ Manual ซึ่งจะต้องทำการตรวจสอบอย่างละเอียดเพื่อหลีกเลี่ยงข้อผิดพลาดในการ Upload ข้อมูล

2.4 การทดสอบและประเมินการทำงานของฐานข้อมูล (Testing and evaluation)

ในขั้นตอนการออกแบบฐานข้อมูล เต้องทำการตรวจสอบเกี่ยวกับ Integrity (Entity integrity และ Referential integrity) ความปลอดภัย ประสิทธิภาพและความสามารถในการกู้คืนข้อมูลของฐานข้อมูลที่ถูกออกแบบ โดยขั้นตอนการตรวจสอบจะประกอบไปด้วยขั้นตอนดังต่อไปนี้

2.4.1 การตรวจสอบฐานข้อมูล

การตรวจสอบฐานข้อมูลจะทำมั่นใจได้ว่า 1) ตารางข้อมูลมีการกำหนด Primary key และ Foreign key อย่างเหมาะสม 2) ตารางข้อมูลต่างมีการกำหนดขอบเขตของแอทริบิวขอบเขตของเงื่อนไข และ Database trigger ได้อย่างเหมาะสม และ 3) มีการกำหนดมาตรการความปลอดภัยที่เกี่ยวข้องกับการกำหนดสิทธิ์ในการเข้าถึง การใช้งานข้อมูลในตารางข้อมูลตามลำดับ

ในการที่จะทดสอบความปลอดภัยของฐานข้อมูล ควรพิจารณาประเด็นสำคัญ ดังนี้

♦ Physical security จะเป็นการยินยอมให้ผู้ใช้ที่มีการยืนยันตัวตนสามารถเข้าใช้งานข้อมูล

♦ Password security จะบังคับให้ผู้ใช้งานทำการ Log in เพื่อใช้งานข้อมูลตามที่ถูกกำหนดสิทธิ์

♦ Access rights จะเป็นการกำหนดสิทธิ์ในการใช้งานข้อมูลในตารางข้อมูล เช่น CREATE, UPDATE, DELETE, INSERT, SELECT และอื่น ๆ

♦ Audit trails จะเป็นการตรวจสอบการล่วงละเมิดสิทธิ์ในการดำเนินการกับข้อมูลที่ซึ่งจะดำเนินการโดย DBMS

2.4.2 การประเมินการทำงานของฐานข้อมูลและโปรแกรมที่เกี่ยวข้อง

การประเมินประสิทธิภาพการทำงานของฐานข้อมูลจะทำได้ค่อนข้างยาก เนื่องจากการไม่มีมาตรฐานในการประเมิน แต่อย่างไรก็ตามการประเมินประสิทธิภาพของฐานข้อมูลเป็นสิ่งจำเป็นในการตรวจสอบว่าฐานข้อมูลมีการทำงานอย่างไร ระบบฐานข้อมูลที่แตกต่างกันอาจ

ต้องการประสิทธิภาพของการทำงานในแง่มุมที่แตกต่างกัน ตัวอย่างเช่น บางระบบอาจต้องการให้ฐานข้อมูลสามารถเพิ่ม อัพเดท และลบข้อมูลปริมาณมาก ๆ ได้อย่างรวดเร็ว หรือบางระบบอาจต้องการให้ระบบที่สร้างขึ้นสามารถทำการค้นคืนข้อมูลที่มีความซับซ้อนได้อย่างรวดเร็ว เป็นต้น

จากความจำเป็นของการประเมินการทำงานของฐานข้อมูลและโปรแกรมที่เกี่ยวข้องกับฐานข้อมูล จะต้องทำการทดสอบประเมินส่วนประกอบย่อยของระบบเพื่อทำให้แน่ใจว่าส่วนต่าง ๆ ของระบบสามารถทำงานได้อย่างเหมาะสม และสามารถตอบสนองต่อความต้องการของผู้ใช้งานได้ นอกจากนั้นต้องทำการตรวจสอบเกี่ยวกับกระบวนการสำรองและกู้คืนข้อมูลที่ ภายใต้การดำเนินการได้หลายระดับดังนี้

- Full backup จะเป็นการสำรองข้อมูลทั้งหมดในฐานข้อมูล
- Differential backup จะทำการคัดลอกข้อมูลที่มีการเปลี่ยนแปลงล่าสุดจากการสำรองข้อมูลครั้งก่อนหน้า
- Transaction Log backup จะเป็นการสำรองข้อมูลที่มีการดำเนินการที่มีการจดบันทึกใน log file เท่านั้น

จากการสำรองข้อมูลทั้ง 3 ระดับทำให้สามารถป้องกันอันตรายที่อาจเกิดกับฐานข้อมูล เช่น ไฟไหม้ การลักขโมย น้ำท่วมและอุบัติเหตุหรืออุบัติภัยต่าง ๆ ได้ และการสำรองข้อมูลจะช่วยให้องค์กรมั่นใจเกี่ยวกับการกู้คืนข้อมูลเมื่อประสบปัญหาทางด้านซอฟต์แวร์และฮาร์ดแวร์

2.5 การประยุกต์ใช้ฐานข้อมูล (Operation)

หลังจากทำการประเมินและตรวจสอบการทำงานของฐานข้อมูลอย่างแน่ชัดแล้ว ต่อมาเป็นขั้นตอนการประยุกต์ใช้ฐานข้อมูลเพื่อการดำเนินการขององค์กร โดยจะเริ่มจากการกำหนดให้ผู้ใช้งานที่เกี่ยวข้องเข้าใช้งานฐานข้อมูล เมื่อผู้ใช้งานเริ่มใช้งานอาจทำให้พบปัญหาในการใช้งานเกิดขึ้น และเมื่อทราบปัญหาแล้วต้องทำการระบุถึงวิธีการแก้ไขปัญหาในหลาย ๆ วิธีการ ซึ่งจะช่วยให้แก้ไขปัญหาที่เกิดขึ้นได้อย่างสมบูรณ์

2.6 การดูแลรักษาและพัฒนาให้ดียิ่งขึ้น (Maintenance and evolution)

หลังจากทำการประยุกต์ใช้ฐานข้อมูลแล้ว ผู้ดูแลฐานข้อมูลควรที่จะต้องดำเนินการกิจกรรมต่าง ๆ เพื่อทำให้ฐานข้อมูลสามารถทำงานได้อย่างราบรื่นดังนี้

- การสำรองข้อมูล
- การกู้คืนข้อมูล
- การเพิ่มประสิทธิภาพการทำงานของฐานข้อมูล
- การเพิ่มเอ็นทิตี้ แอทริบิว และอื่น ๆ
- การกำหนดสิทธิ์ในการใช้งานข้อมูลให้กับผู้ใช้
- การจัดเก็บสถิติเกี่ยวกับการเข้าถึงข้อมูลเพื่อพิจาณาถึงการเพิ่มประสิทธิภาพการทำงานของระบบและเพื่อติดตามเฝ้าสังเกตการทำงานของระบบ
- การตรวจสอบมาตรการความปลอดภัยจากสถิติที่ได้รับ

นอกเหนือจากกิจกรรมทั้งหมดข้างต้น อาจมีความต้องการจากผู้ใช้งานที่จะต้องการข้อมูลสารสนเทศที่เพิ่มขึ้นจากเดิม หรือผู้ใช้อาจร้องขอรายงานเพิ่มเติม จำเป็นต้องทำการกำหนดคิวรีใหม่พิ่มเติมเช่นกัน ถ้าการออกแบบฐานข้อมูลมีความยืดหยุ่นจะทำให้ผู้ดูแลระบบสามารถดำเนินการปรับเปลี่ยนการทำงานของฐานข้อมูลตามความต้องการของผู้ใช้งานได้โดยง่าย

หลังจากระบบได้เริ่มดำเนินการ ผู้ดูแลฐานข้อมูลจะต้องเตรียมการบำรุงรักษาฐานข้อมูล โดยการสำรองข้อมูลไว้ เพื่อสะดวกในการกู้คืนข้อมูล เมื่อระบบมีปัญหา และหากมีการใช้งานไปนาน ๆ อาจต้องทำการปรับปรุงแก้ไขโปรแกรมให้เหมาะสมกับเหตุการณ์ และความต้องการของผู้ใช้ที่เปลี่ยนแปลงไป

สรุป

เทคโนโลยีสารสนเทศ คือการประยุกต์ใช้คอมพิวเตอร์และอุปกรณ์โทรคมนาคม เพื่อจัดเก็บ ค้นหา ส่งผ่านและจัดดำเนินการข้อมูล ซึ่งมักเกี่ยวข้องกับธุรกิจหนึ่งหรือองค์การอื่น ๆ โดยปกติก็ใช้แทนความหมายของเครื่องคอมพิวเตอร์ และเครือข่ายคอมพิวเตอร์ และยังรวมไปถึงเทคโนโลยีการกระจายสารสนเทศอย่างอื่น เช่น โทรทัศน์ และโทรศัพท์ อุตสาหกรรมหลายอย่างเกี่ยวข้องกับเทคโนโลยีสารสนเทศ การพัฒนาระบบเทคโนโลยีสารสนเทศสามารถใช้ขั้นตอนวงจรการพัฒนาระบบ (Systems development life Cycle) เป็นแนวทางการพัฒนา โดยมี 5 ขั้นตอนหลัก ได้แก่ 1) ขั้นตอนการวางแผน (Planning) 2) ขั้นตอนการวิเคราะห์ (Analysis) 3) ขั้นตอนการออกแบบและพิจารณารายละเอียด (Detailed systems design) 4) ขั้นตอนการสร้างระบบ (Implementation) และ 5) ขั้นตอนการดูแลรักษาระบบ (Maintenance)

วงจรฐานข้อมูล (Database life cycle: DBLC) ประกอบด้วย 6 ขั้นตอน ได้แก่ 1) การศึกษาฐานข้อมูลขององค์กรเบื้องต้น (Database initial study) 2) การออกแบบฐานข้อมูล (Database design) 3) การสร้างฐานข้อมูลและการโหลดข้อมูลเข้าสู่ฐานข้อมูล (Implementation and loading) 4) การทดสอบและประเมินการทำงานของฐานข้อมูล (Testing and evaluation) 5) การประยุกต์ใช้ฐานข้อมูล (Operation) และ 6) การดูแลรักษาและพัฒนาให้ดียิ่งขึ้น (Maintenance and evolution)

การออกแบบฐานข้อมูล แบ่งเป็น 3 ขั้นตอน ได้แก่ 1) การออกแบบระดับแนวคิด (Conceptual schema design) การออกแบบฐานข้อมูลในระดับแนวคิด เป็นขั้นตอนการรวบรวมและวิเคราะห์ความต้องการในการใช้ข้อมูล เป็นการออกแบบโครงร่างของฐานข้อมูลในระดับแนวคิด (Conceptual schema design) เพื่อกำหนดโครงสร้างพื้นฐานของฐานข้อมูลและรายละเอียดทั้งหมดของฐานข้อมูล 2) การออกแบบฐานข้อมูลในระดับตรรกะ (Logical design) เป็นการนำผลจากการออกแบบฐานข้อมูลในระดับแนวคิดมาทำการปรับ เพื่อให้เหมาะสมกับรูปแบบฐานข้อมูลที่เลือกใช้ ผลที่ได้จะเป็นเค้าร่างของฐานข้อมูลที่มีรายละเอียดสมบูรณ์ ที่สามารถนำไปกำหนดภาษาสำหรับนิยามข้อมูลในขั้นตอนการออกแบบในระดับกายภาพได้ 3) การออกแบบฐานข้อมูลในระดับกายภาพ (Physical design) เป็นการนำโครงสร้างตารางข้อมูลที่มีการกำหนดคุณสมบัติหลักไว้แล้วมากำหนดรายละเอียดคุณสมบัติของโครงสร้างที่ใช้ในการจัดเก็บข้อมูลให้ครบถ้วน กำหนดตำแหน่งของฐานข้อมูลที่จะบันทึกลงบนสื่ออิเล็กทรอนิกส์ กำหนดวิธีในการเข้าถึงข้อมูลในฐานข้อมูล

บทที่ 3

แบบจำลองฐานข้อมูล (Database models)

เทคโนโลยีการจัดเก็บข้อมูลได้ถูกพัฒนามาอย่างต่อเนื่องส่งผลให้มีการ การเปลี่ยนแปลงแก้ไขแบบจำลองข้อมูลหรือโมเดลการเก็บข้อมูลเพื่อให้สอดคล้องกับประสิทธิภาพที่ต้องการมากขึ้น การสร้างแบบจำลองข้อมูลหรือแบบจำลองฐานข้อมูล จะเป็นขั้นตอนแรกของการออกแบบฐานข้อมูลที่มุ่งเน้นที่การกำหนดโครงสร้างของฐานข้อมูล ที่ซึ่งจะใช้ในการจัดเก็บและจัดการข้อมูลของผู้ใช้งานฐานข้อมูล แบบจำลองฐานข้อมูลมีลักษณะเป็นแผนภาพที่ใช้แสดงโครงสร้างที่ซับซ้อนของฐานข้อมูล มีหน้าที่ในการช่วยให้ผู้ออกแบบฐานข้อมูล สามารถเข้าใจความซับซ้อนของข้อมูลที่ถูกใช้ในองค์กร นอกจากนั้น แบบจำลองฐานข้อมูลมักจะแสดงถึงโครงสร้างของข้อมูลในฐานข้อมูลและคุณลักษณะของข้อมูลเหล่าต่าง ๆ จากรูปแบบที่เป็นแนวคิดเข้าใจยาก ให้อยู่ในรูปแบบที่เข้าใจได้ง่าย วัตถุประสงค์ของแบบจำลองฐานข้อมูล ก็เพื่อนำแนวคิดต่าง ๆ มาเสนอให้เกิดเป็นแบบจำลองเพื่อนำเสนอข้อมูล และความสัมพันธ์ระหว่างข้อมูลในรูปแบบที่เข้าใจง่าย ดังนั้น ในบทนี้จะกล่าวถึงแบบจำลองฐานข้อมูล (Database models) โดยนำเสนอเนื้อหาสำหรับการศึกษาดังต่อไปนี้

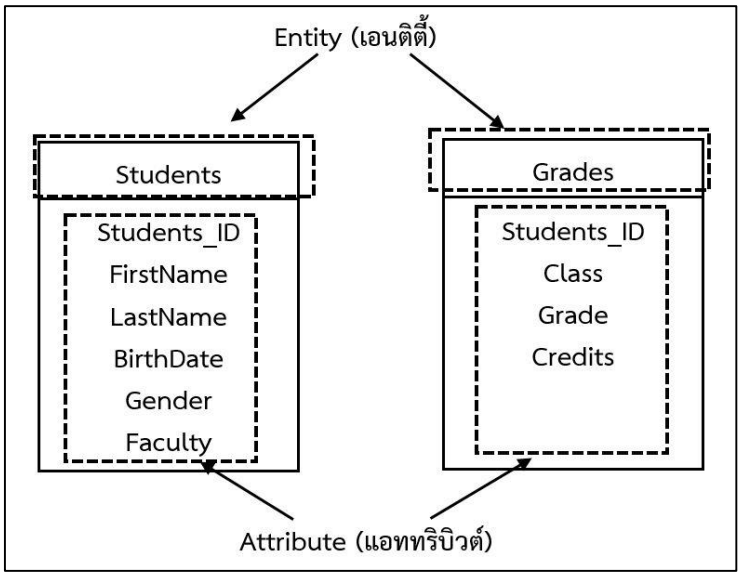
- องค์ประกอบของแบบจำลองฐานข้อมูล
- แบบจำลองฐานข้อมูลลำดับชั้น (Hierarchical database model)
- แบบจำลองฐานข้อมูลเครือข่าย (Network database model)
- แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational database model)
- แบบจำลองฐานข้อมูลเชิงวัตถุ (Object-oriented database model)
- แบบจำลองฐานข้อมูลมัลติไดเมนชัน (Multidimensional database model)

องค์ประกอบของแบบจำลองฐานข้อมูล

แบบจำลองฐานข้อมูล หมายถึง แนวคิดที่ใช้อธิบายโครงสร้างความสัมพันธ์ และความหมายของข้อมูล รวมไปถึงเงื่อนไขบังคับต่าง ๆ เพื่อให้มีความความสอดคล้องของข้อมูลภายในฐานข้อมูล เพื่อให้ผู้ใช้งานเข้าใจได้ง่ายขึ้น

แบบจำลองฐานข้อมูล คือ สถาปัตยกรรมที่มีความสำคัญต่อระบบจัดการฐานข้อมูล (DBMS) ในการนำมาใช้จัดเก็บข้อมูล/ วัตถุลงในฐานข้อมูลและกำหนดความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่เกี่ยวข้อง ซึ่งเทคโนโลยีฐานข้อมูลแต่ละชนิดต่างก็มีวิธีเชื่อมโยงความสัมพันธ์เพื่อเข้าถึงข้อมูลในรูปแบบที่แตกต่างกัน ดังนั้น กระบวนการคัดสรร DBMS เพื่อใช้งาน (ในขั้นตอนการออกแบบฐานข้อมูล) จึงจำเป็นต้องสอดคล้องกับแบบจำลองข้อมูลที่สร้างขึ้นจากระยะการวิเคราะห์ด้วย (โอภาส เอี่ยมสิริวงศ์, 2558)

แบบจำลองฐานข้อมูลส่วนใหญ่ ประกอบด้วย 1) Entity 2) Attribute 3) ความสัมพันธ์ระหว่าง Entity (Entity relationship) และข้อจำกัดต่าง ๆ (Constraints) ดังนี้



ภาพที่ 3.1 ลักษณะของ Entity และ Attribute

1. Entity

Entity หมายถึง วัตถุ (Object) หรือกลุ่มสิ่งของต่าง ๆ ที่เราสนใจจะเก็บข้อมูลไว้ในฐานข้อมูล หรือแนวคิดที่สามารถบอกความแตกต่างของแต่ละ Entity ได้ ถือเป็นตัวแทนของวัตถุในโลกแห่ง ความเป็นจริง เช่น คน สัตว์ สถานที่ สิ่งของ หรือเหตุการณ์ ซึ่งจะเป็นข้อมูลที่จะถูกจัดเก็บอยู่ในฐานข้อมูล เนื่องจาก Entity หนึ่งจะใช้แทนข้อมูลชนิดหนึ่ง ๆ ดังนั้น แต่ละ Entity จะต้องมีความแตกต่างกัน และจะต้องมีความเป็นเอกลักษณ์เสมอ เช่น Entity ของนักศึกษาจะมีข้อมูลนักศึกษาที่แตกต่างกันหลายคน โดยที่นักศึกษาแต่ละคนจะมีข้อมูลที่มีความแตกต่างจากนักศึกษาคนอื่น ๆ เช่น นายมานะ ใจงาม จะมีชื่อและนามสกุลที่แตกต่างจาก นายสมชาย ดีพร้อม เป็นต้น

1.1 Entity ที่เป็นบุคคล เช่น นักศึกษา อาจารย์ พนักงาน แม่บ้าน พนักงานขับรถ ฯลฯ

1.2 Entity ที่เป็นสถานที่ เช่น หมู่บ้าน ตำบล อำเภอ จังหวัด ประเทศ น้ำตก ภูเขา โรงเรียน ร้านอาหาร ฯลฯ

1.3 Entity ที่เป็น วัตถุ สิ่งของ เช่น หนังสือ คอมพิวเตอร์ โทรศัพท์มือถือ อุปกรณ์เครื่องมือต่าง ๆ ฯลฯ

1.4 Entity ที่เป็น นามธรรม เช่น วิชา ความสามารถ วัน เวลา ฯลฯ

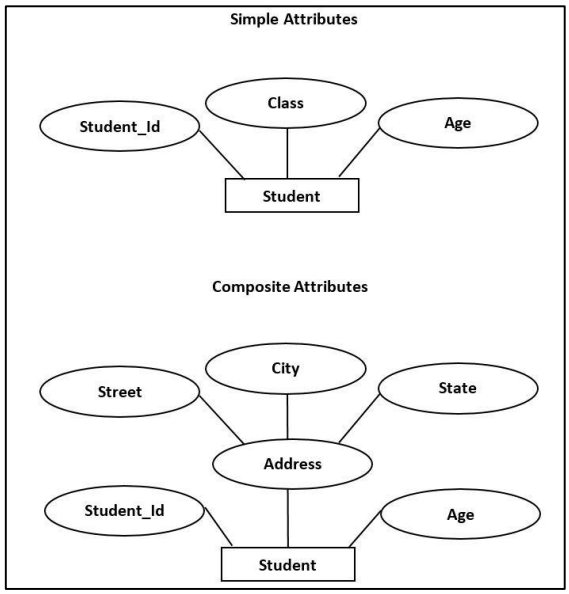
1.5 Entity เซต (Entity Set) คือ กลุ่มของ Entity หรือวัตถุที่เป็น “ชนิดเดียวกัน” คือมีคุณสมบัติร่วมกันบางประการ เช่น นักศึกษาแต่ละคนถือเป็น Entity, กลุ่มของนักศึกษาในคณะศึกษาศาสตร์ ถือเป็นหนึ่ง Entity Set ของนักศึกษา ซึ่งต่างจาก Entity Set ของ “วิชา” ซึ่งเก็บรวบรวมรายวิชาต่าง ๆ เป็นต้น สัญลักษณ์จะใช้รูปสี่เหลี่ยม (Rectangles)

2. Attribute

Attribute หมายถึง คุณลักษณะ หรือ คุณสมบัติ (Properties) ของ Entity เช่น Attribute ของ Entity “นักศึกษา” คือข้อมูล Entity นักศึกษา ที่อธิบายด้วย Attribute ต่าง ๆ เช่น ชื่อ-สกุล รหัสนักศึกษา คณะวิชา สาขาวิชา หมายเลขโทรศัพท์ คะแนนสะสมและที่อยู่

หรือ Attribute ของ Entity “ลงทะเบียน” คือข้อมูล Entity ลงทะเบียนที่อธิบายด้วย Attribute ต่าง ๆ เช่น รหัสวิชา รหัสนักศึกษา ชื่อวิชา จำนวนหน่วยกิต ชื่อผู้สอน และปีการศึกษา เป็นต้น

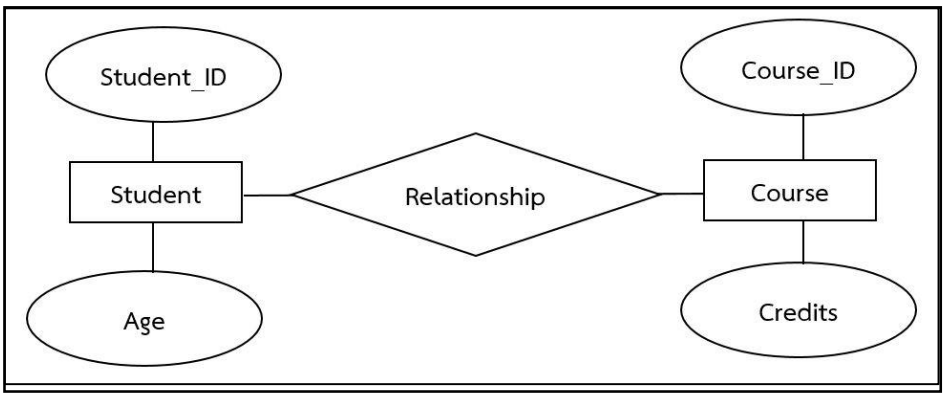
Attribute ในระบบฐานข้อมูลมักจะมีลักษณะเหมือนกับฟิลด์ในแฟ้มข้อมูล สัญลักษณ์จะใช้รูปวงรี (Ellipse) Attribute ที่ไม่สามารถแยกข้อมูลออกเป็นข้อมูลย่อย ๆ ได้อีก เช่น “รหัสนักศึกษา” ไม่สามารถที่จะแยกออกเป็นข้อมูลอื่นได้อีก สัญลักษณ์ที่ใช้คือวงรีเส้นขอบเส้นเดี่ยว ภายในมีชื่อ Attribute บรรจุอยู่ Attribute ประเภทนี้เรียกว่า Simple attribute หรือ Atomic attribute ส่วน Attribute ที่สามารถแบ่งออกเป็น Attribute ย่อย ๆ ได้อีก เช่น Attribute “ที่อยู่” สามารถแบ่งออกเป็น Attribute ย่อยได้เป็น บ้านเลขที่ ถนน ตำบล อำเภอ จังหวัด ประเทศ เป็นต้น Attribute ประเภทนี้ เรียกว่า Composite attribute สัญลักษณ์ที่ใช้ คือ วงรีเส้นขอบเส้นเดี่ยว แต่มีวงรีย่อยมาเชื่อมต่อ



ภาพที่ 3.2 Simple attribute และ Composite attribute

3. ความสัมพันธ์ระหว่าง Entity (Entity relationship)

ความสัมพันธ์ระหว่าง Entity มีสัญลักษณ์เป็นรูปขนมเปียกปูน (Diamonds) ความสัมพันธ์ระหว่าง Entity มี 3 แบบด้วยกันคือ 1) แบบ One-to-one (1: 1) 2) One-to-Many (1: M) และ 3) แบบ Many-to-many (M: M หรือ M: N)



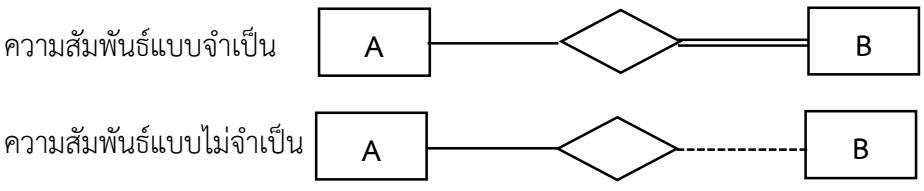
ภาพที่ 3.3 ความสัมพันธ์ระหว่าง Entity (Entity relationship)

4. ข้อจำกัดหรือข้อกำหนด (Constraints)

หมายถึงกฎเกณฑ์หรือข้อตกลงของข้อมูลเพื่อเกิดความมั่นใจในความเป็นอันหนึ่งอันเดียวกัน หรือเกิดความสอดคล้องตรงกันของข้อมูล ข้อจำกัดส่วนใหญ่จะอยู่ในรูปแบบของกฎเกณฑ์ต่าง ๆ เช่น คะแนนเฉลี่ยของนักศึกษาจะอยู่ระหว่าง 0.00-4.00

รูปแบบความสัมพันธ์สัมพันธ์ระหว่าง Entity (Entity relationships)

ความสัมพันธ์ในที่นี้จะหมายถึง ความสัมพันธ์ระหว่าง Entity หรือระหว่างข้อมูลที่เราสนใจ ซึ่งพิจารณาได้ 2 แบบ คือความสัมพันธ์แบบจำเป็น หรือแบบบังคับ (Mandatory) หมายถึง ความสัมพันธ์ที่จะต้องมีหรือจะต้องเกิดขึ้นอย่างแน่นอน แทนด้วยสัญลักษณ์เส้นตรงทึบ หรือเส้นตรง 2 เส้น และความสัมพันธ์แบบไม่จำเป็น (Optional) เป็นความสัมพันธ์ที่อาจจะมีหรือไม่มีก็ได้ แทนด้วยสัญลักษณ์เส้นตรงประ หรือเส้นตรง 1 เส้น ดังภาพที่ 3.4



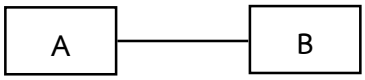
ภาพที่ 3.4 แสดงสัญลักษณ์เส้นความสัมพันธ์ระหว่าง Entity

จากภาพ 3.4 อธิบายได้ ดังนี้: ถ้าข้อมูล A มีความสัมพันธ์กับข้อมูล B อย่างแน่นอน หรืออย่างน้อยจะต้องมีสมาชิก (Element) ของข้อมูล A 1 สมาชิกที่มีความสัมพันธ์กับสมาชิกของข้อมูล B และแทนความสัมพันธ์ด้วยเส้นตรงทึบออกจากข้อมูล A ไปยังข้อมูล B แต่ในขณะเดียวกันข้อมูล B อาจจะมีความสัมพันธ์กับข้อมูล 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

1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One to one relationships, 1: 1) เป็นการแสดงความสัมพันธ์ของข้อมูลของ Entity หนึ่งว่า มีความสัมพันธ์กับข้อมูลในอีก Entity หนึ่งในลักษณะหนึ่งต่อหนึ่ง คือ ข้อมูลหนึ่งข้อมูลของ Entity ที่หนึ่งมีความสัมพันธ์กับข้อมูลใน Entity ที่ 2 เพียงข้อมูลเดียวเท่านั้น เช่น ข้อมูล A จะเกี่ยวข้องสัมพันธ์กับข้อมูล B ได้เพียง 1 ค่า เท่านั้น และข้อมูล B จะเกี่ยวข้องกับข้อมูล A ได้เพียง 1 ค่า เช่นกัน

ตัวอย่างเช่น มี Entity 2 Entity คือ “อาจารย์” และ “คณะวิชา” มีความสัมพันธ์กันชื่อ “บริหาร” แบบ 1: 1 หมายถึง อาจารย์ 1 คน จะสามารถบริหารหรือเป็นคณบดีได้พียง 1 คณะวิชาเท่านั้น และในขณะเดียวกัน แต่ละคณะวิชาก็จะมีคณบดีได้เพียง 1 คน เช่นกัน หรือความสัมพันธ์ระหว่างนักศึกษากับรหัสนักศึกษา จะเป็นแบบ 1: 1 เพราะนักศึกษา 1 คน จะมีรหัสนักศึกษาได้เพียง 1 รหัสเท่านั้น และในขณะเดียวกันรหัสนักศึกษา 1 รหัส จะใช้กับนักศึกษาได้เพียง 1 คน เช่นกัน

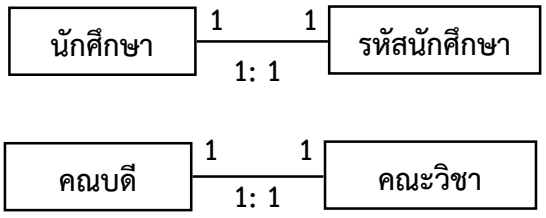


ภาพที่ 3.5 แสดงสัญลักษณ์เส้นความสัมพันธ์แบบหนึ่งต่อหนึ่ง

ตัวอย่าง

ความสัมพันธ์ระหว่างข้อมูลนักศึกษากับรหัสนักศึกษา: นักศึกษา 1 คน จะมีรหัสนักศึกษาได้เพียง 1 รหัสเท่านั้น ในขณะเดียวกันรหัสนักศึกษา 1 รหัส จะใช้กับนักศึกษาได้เพียง 1 คนเช่นกัน

ความสัมพันธ์ระหว่างคณบดีกับคณะวิชา: คณบดี 1 คน จะบริหารคณะวิชาได้เพียง 1 คณะวิชาเท่านั้น ในขณะเดียวกันคณะวิชา 1 คณะ จะมีคณบดีบริหารได้เพียง 1 คน เช่นกัน

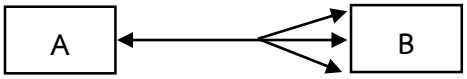


ภาพที่ 3.6 แสดงตัวอย่างความสัมพันธ์แบบหนึ่งต่อหนึ่ง

2. ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One to many relationships, 1: N หรือ 1: M)

เป็นความสัมพันธ์ระหว่าง Entity หนึ่งไปมีความสัมพันธ์กับอีก Entity หนึ่งได้มากกว่า 1 รายการ ในลักษณะหนึ่งต่อกลุ่ม (ข้อมูลหนึ่งข้อมูลของ Entity ที่ 1 มีความสัมพันธ์กับข้อมูลใน Entityที่ 2 หลายข้อมูล) เช่น ข้อมูล “A” มีความสัมพันธ์กับข้อมูล “B” ได้ มากกว่า 1 รายการ

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

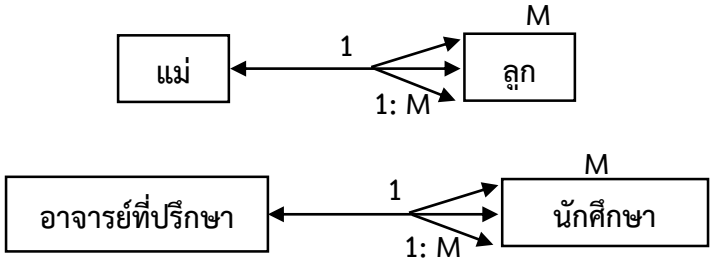


ภาพที่ 3.7 แสดงความสัมพันธ์แบบหนึ่งต่อกลุ่ม

ตัวอย่าง

ความสัมพันธ์ระหว่างแม่กับลูก: แม่ 1 คนอาจจะมีลูกได้หลายคน แต่ลูก 1 คนจะมีแม่ได้เพียงคนเดียวเท่านั้น

ความสัมพันธ์ระหว่างอาจารย์ที่ปรึกษากับนักศึกษา: อาจารย์ที่ปรึกษา 1 คนจะมีนักศึกษาได้หลายคน แต่นักศึกษาคนหนึ่งจะมีอาจารย์ที่ปรึกษาได้เพียง 1 คนเท่านั้น



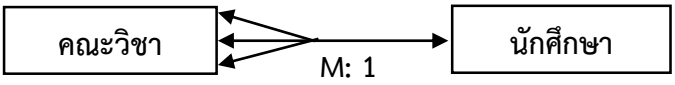
ภาพที่ 3.8 แสดงตัวอย่างความสัมพันธ์แบบหนึ่งต่อกลุ่ม

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 คือ เป็นความสัมพันธ์ระหว่างหนึ่งต่อกลุ่ม

ตัวอย่าง

ความสัมพันธ์ระหว่างนักศึกษากับคณะวิชาที่สังกัดสถาบันการศึกษาหนึ่งสามารถมีคณะวิชาได้มากกว่า 1 คณะวิชา แต่นักศึกษาหนึ่งคนจะสังกัดคณะวิชาได้เพียงคณะวิชาเดียวเท่านั้น



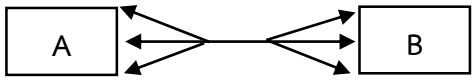
ภาพที่ 3.9 แสดงตัวอย่างความสัมพันธ์แบบกลุ่มต่อหนึ่ง

4. ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many to many relationships, M: N)

เป็นความสัมพันธ์แบบหลายรายการระหว่าง 2 Entity เป็นความสัมพันธ์ของข้อมูลของ Entity หนึ่งมีความสัมพันธ์กับข้อมูลในอีก Entity หนึ่ง ในลักษณะกลุ่มต่อกลุ่ม เช่น ข้อมูล “A” มีความสัมพันธ์กับอีกข้อมูล “B” ได้มากกว่า 1 รายการ ในขณะเดียวกันแต่ละรายการของข้อมูล “B” ก็มีความสัมพันธ์กับข้อมูล “A” ได้มากกว่า 1 รายการ เช่นเดียวกัน ตัวอย่างเช่น นักศึกษา 1 คน

สามารถลงทะเบียนเรียนได้มากกว่า 1 รายวิชา และ รายวิชา 1 รายวิชาจะมีนักศึกษาลงทะเบียนเรียนได้มากกว่า 1 คน เป็นต้น

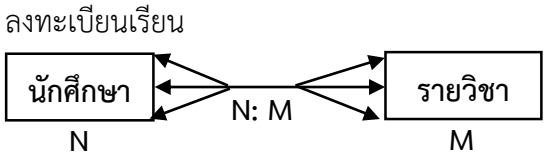
ความสัมพันธ์แบบกลุ่มต่อกลุ่มนี้ ข้อมูล A จะเกี่ยวข้องกับข้อมูล B ได้หลายค่า และในทำนองเดียวกันข้อมูล B เกี่ยวข้องกับข้อมูล A ได้มากกว่า 1 ค่าเช่นกัน



ภาพที่ 3.10 แสดงความสัมพันธ์แบบกลุ่มต่อกลุ่ม

ตัวอย่าง

ความสัมพันธ์ระหว่างนักศึกษากับรายวิชาที่เปิดสอน: นักศึกษา 1 คนสามารถลงทะเบียนเรียนได้หลายวิชา และวิชาหนึ่ง ๆ จะมีนักศึกษาลงทะเบียนเรียนได้มากกว่า 1 คน



ภาพที่ 3.11 แสดงตัวอย่างความสัมพันธ์แบบกลุ่มต่อกลุ่ม

ประเภทของแบบจำลองฐานข้อมูล

แบบจำลองฐานข้อมูลเป็นเครื่องมือที่ใช้อธิบายโครงสร้าง และความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่อยู่ในฐานข้อมูล จากรูปแบบที่เป็นแนวความคิดซึ่งเข้าใจได้ยากให้อยู่ในรูปแบบที่เข้าใจได้ง่ายขึ้นและสะดวกในการใช้ อาจจะแบ่งแบบจำลองฐานข้อมูลออกได้เป็น 2 ประเภท คือ

1. แบบจำลองฐานข้อมูลเชิงแนวคิด (Conceptual Data Models) เป็นแบบจำลองที่ประกอบด้วยสัญลักษณ์ที่ใช้อธิบายแทนตัวข้อมูล คุณสมบัติของข้อมูล และความสัมพันธ์ของข้อมูล ความสัมพันธ์ต่าง ๆ ใช้อธิบายภาพหรือลักษณะโดยรวมของข้อมูลทั้งหมดในระบบโดยแสดงในรูปของแผนภาพ ซึ่งประกอบด้วย Entity ต่าง ๆ ความสัมพันธ์ระหว่าง Entity ของแบบจำลองเชิงแนวคิดนี้ ต้องการนำเสนอให้เกิดความเข้าใจระหว่างผู้ออกแบบและผู้ใช้งาน ผู้ที่สามารถเข้าไปแก้ไขโครงสร้างข้อมูลนี้ได้ คือ ผู้บริหารฐานข้อมูล หรือโปรแกรมเมอร์

แบบจำลองฐานข้อมูลเชิงแนวคิดมักถูกนำไปใช้ในการออกแบบฐานข้อมูล เพื่อต้องการอธิบายให้เห็นว่าภายในฐานข้อมูลจะประกอบด้วยข้อมูลอะไรบ้าง และแต่ละข้อมูลนั้นมีความสัมพันธ์กันอย่างไร ตัวอย่างแบบจำลองฐานข้อมูลประเภทนี้ เช่น แบบจำลองฐานข้อมูล E-R (Entity-relationship model) และแบบจำลองฐานข้อมูลเชิงวัตถุ (Object-oriented model) เป็นต้น

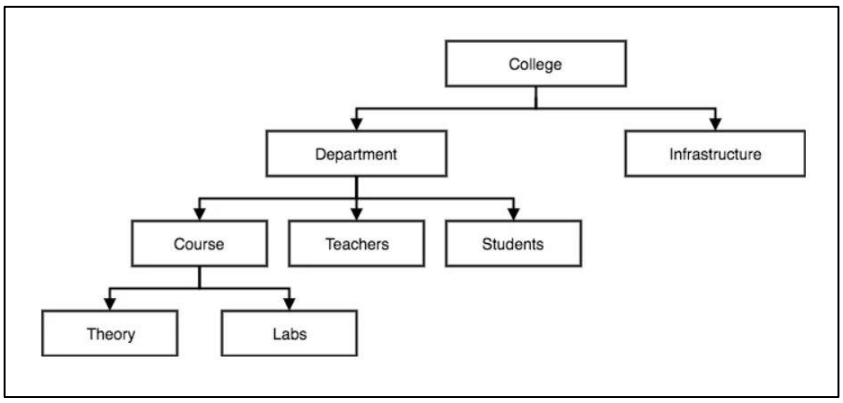
2. แบบจำลองฐานข้อมูลแบบนำไปใช้ (Implementation data models) เป็นแบบจำลองฐานข้อมูลที่อธิบายถึงโครงสร้างข้อมูลของฐานข้อมูลที่แสดงถึงรูปแบบที่อิงกับระบบการจัดการฐานข้อมูลที่เลือกใช้งาน และเป็นแบบจำลองที่ถูกนำมาใช้ในขั้นตอนการสร้างฐานข้อมูล ตัวอย่างแบบจำลองฐานข้อมูลประเภทนี้ เช่น แบบจำลองฐานข้อมูลแบบลำดับชั้น (Hierarchical

model) แบบจำลองฐานข้อมูลแบบเครือข่าย (Network model) และ แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational model) เป็นต้น

ชนิดแบบจำลองฐานข้อมูล (Types of database models)

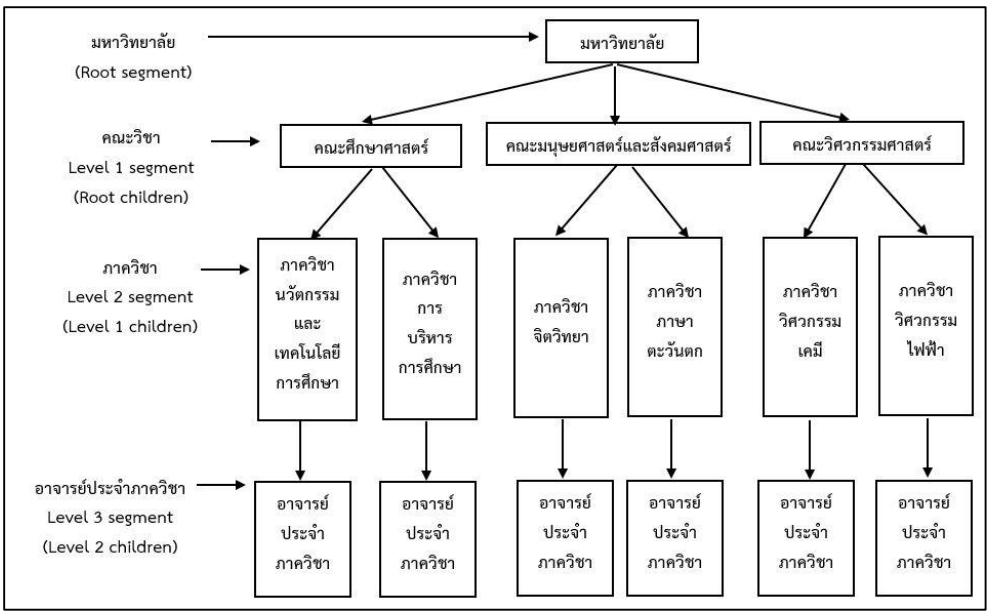
1. แบบจำลองฐานข้อมูลลำดับชั้น (Hierarchical Database Model)

แบบจำลองฐานข้อมูลลำดับชั้น หรือบางครั้งเรียกว่าแบบจำลองฐานข้อมูลโครงสร้างต้นไม้ (Tree-based data model) เป็นแบบจำลองฐานข้อมูลที่พัฒนามาจากแนวความคิดในการจัดเก็บข้อมูลเพื่อลดการซ้ำซ้อนของข้อมูล (Data redundancy) เป็นแบบจำลอง ต้นแบบสำหรับการแก้ไขปัญหาการเก็บข้อมูลด้วยระบบแฟ้มข้อมูล โดยนำเอาข้อมูลในแต่ละส่วน หรือแต่ละ Part มาจัดเก็บเป็นกลุ่มหรือเป็น Component แล้วรวมแต่ละกลุ่มเป็นกลุ่มใหญ่เรียกว่า Final component โดยมีโครงสร้างอยู่ในรูปแบบของต้นไม้ที่มีลักษณะคล้ายต้นไม้ที่คว่ำหัวลงเป็นลำดับชั้น (Upside-down tree) แตกออกเป็นกิ่งก้านสาขา เป็นการจัดเก็บข้อมูลในลักษณะความสัมพันธ์แบบ พ่อแม่-ลูก (Parent-child relationship Type หรือ PCR)



ภาพที่ 3.12 ตัวอย่างแบบจำลองฐานข้อมูลแบบลำดับชั้น

โครงสร้างข้อมูลพื้นฐานของแบบจำลองฐานข้อมูลลำดับชั้น ประกอบด้วย ระเบียน (Record) หรือกลุ่มค่าของข้อมูลของ Entity หนึ่ง ๆ หรือของความสัมพันธ์ระหว่าง Entity ระเบียนข้อมูลชนิดเดียวกันเมื่อนำมารวมกลุ่มกัน เรียกว่าชนิดระเบียน (Record type) (Entity หมายถึงวัตถุใด ๆ ทั้งที่เป็นนามธรรมและรูปธรรมที่บรรจุไว้เป็นข้อมูล ประกอบด้วยคุณสมบัติ หรือ Attribute ต่าง ๆ เช่น ฐานข้อมูลของคณะวิชาอาจจะประกอบด้วย Entity อาจารย์ นักศึกษา เจ้าหน้าที่ งบประมาณ ภาควิชา และ หลักสูตร เป็นต้น Entityอาจารย์ อาจประกอบด้วย ชื่อ-สกุล วันเดือนปีเกิด ตำแหน่งวิชาการ คุณวุฒิการศึกษา เลขประจำตัวบัตรประชาชน ที่อยู่ เบอร์โทรศัพท์ เป็นต้น ข้อมูลของ หน่วยงานเช่น มหาวิทยาลัย คณะวิชา ภาควิชา จะเหมาะสมกับแบบจำลองฐานข้อมูลเชิงลำดับชั้นมากที่สุด



ภาพที่ 3.12 ตัวอย่างแบบจำลองฐานข้อมูลแบบลำดับชั้น

1.1 ลักษณะของแบบจำลองฐานข้อมูลลำดับชั้น มีลักษณะดังนี้

1.1.1 ข้อมูลทั้งหมดอยู่ในฐานข้อมูลร่วมกัน มีความสัมพันธ์แบบ 1:M 2) มีความสัมพันธ์แบบพ่อแม่-ลูก

1.1.2 ระเบียนที่อยู่แถวบน (ระดับสูงสุด) เรียกว่า ระเบียนพ่อแม่ (Parent record) ส่วนระเบียนในแถวถัดลงมาเรียกระเบียนลูก (Child record)

1.1.3 ระเบียนพ่อแม่ สามารถมีระเบียนลูกได้มากว่า 1 ระเบียน แต่ระเบียน ลูกแต่ละระเบียนจะมีระเบียนพ่อแม่ได้เพียง 1 ระเบียนเท่านั้น

1.1.4 เป็นการจัดโครงสร้างแบบบนลงล่าง (Top-down) มีลักษณะคล้ายโครงสร้างต้นไม้

1.1.5 เป็นสถาปัตยกรรมฐานข้อมูลที่เก่าแก่ที่สุด ยากต่อการพัฒนาแอปพลิเคชัน (Application) เพื่อใช้งาน

1.1.6 โครงสร้างมีความยืดหยุ่นน้อย

1.1.7 ไม่สามารถกำหนดความสัมพันธ์แบบกลุ่มต่อกลุ่มได้

1.2 ข้อดีแบบจำลองฐานข้อมูลลำดับชั้น

1.2.1 เป็นโครงสร้างที่เข้าใจง่าย มีประสิทธิภาพในการค้นหา

1.2.2 ค้นหาได้รวดเร็วเพราะมีการจัดลำดับชั้นความสัมพันธ์เอาไว้

1.2.3 เหมาะกับข้อมูลที่มีการเรียงลำดับแบบต่อเนื่อง

1.2.5 มีโครงสร้างเหมาะสมกับระบบคอมพิวเตอร์ที่มีขนาดใหญ่

1.2.6 โครงสร้างมีความซับซ้อนน้อย เหมาะสมกับข้อมูลที่มีความสัมพันธ์แบบ 1: M

1.2.7 สามารถกำหนดกฎเกณฑ์ที่ใช้ควบคุมความถูกต้องให้กับข้อมูลในฐานข้อมูลได้ง่าย

1.2.8 การรักษาความปลอดภัยควบคุมได้โดยระบบการจัดการฐานข้อมูล

1.2.9 ระบบการจัดการฐานข้อมูลสามารถสร้างสภาวะความเป็นอิสระของข้อมูล การแก้ไข การบำรุงรักษาโปรแกรมที่เขียนขึ้นไม่มีผลต่อข้อมูล

1.2.10 ค่าใช้จ่ายในการจัดสร้างฐานข้อมูลน้อย

1.3 ข้อจำกัดแบบจำลองฐานข้อมูลลำดับชั้น

ถึงแม้ว่าระบบการจัดการฐานข้อมูลแบบจำลองฐานข้อมูลแบบลำดับชั้น จะช่วยแบ่งเบาในปัญหาเรื่องความเป็นอิสระของข้อมูล แต่ระบบจัดการฐานข้อมูลยังคงต้องการความรู้ในระดับกายภาพของการจัดเก็บข้อมูล

1.3.1 ไม่มีข้อมูลจริงของระเบียนใด ๆ ยกเว้นรากต้นไม้ (Root) ที่จะดำรงอยู่ได้โดยไม่ติดอยู่กับระเบียนพ่อแม่

1.3.2 ถ้าระเบียนลูกมีระเบียนพ่อตั้งแต่ 2 ตัวขึ้นไป จากระเบียนชนิดเดียวกัน ระเบียนลูกจำเป็นต้องทำซ้ำภายในแต่ละระเบียนพ่อแม่

1.3.3 การเปลี่ยนแปลงโครงสร้างของข้อมูลส่งผลให้ต้องมีการปรับเปลี่ยนและแก้ไขโปรแกรมประยุกต์ทั้งหมดที่ใช้ฐานข้อมูล

1.3.4 รองรับเฉพาะกับข้อมูลที่มีความสัมพันธ์ในแบบหนึ่งต่อกลุ่ม ไม่สามารถรองรับข้อมูลที่มีความสัมพันธ์แบบกลุ่มต่อกลุ่มได้

1.3.5 จะต้องมีการกำหนดลักษณะความสัมพันธ์ของข้อมูลทุกตัว ก่อนเพื่อจะนำมาจัดเป็นโครงสร้าง

1.3.6 การค้นหาข้อมูลมีจำกัด ถ้าข้อมูลมีจำนวนมาก การเข้าถึงข้อมูลจะใช้เวลานานในการค้นหา เนื่องจากจะต้องเข้าถึงต้นกำเนิดของข้อมูล

1.3.7 การออกแบบฐานข้อมูลเป็นเรื่องที่ซับซ้อนยากทำให้ยากต่อการพัฒนา เพราะจะต้องเข้าใจโครงสร้างทางกายภาพของข้อมูล

1.3.8 การเรียกใช้ข้อมูลจะต้องผ่านจาก Root เสมอ

1.3.9 ไม่สะดวกในการค้นหาข้อมูลในระดับล่าง เพราะการค้นหาข้อมูลในระดับล่างต้องค้นหาทั้งแฟ้มข้อมูล

1.3.10 ไม่มีความเป็นอิสระในโครงสร้างและข้อมูล เนื่องจากการจัดการข้อมูลจะขึ้นอยู่กับชนิดของอุปกรณ์ที่ใช้จัดเก็บข้อมูล

1.3.11 มีการจัดการที่ซับซ้อนและขาดความยืดหยุ่น อาจจะเป็นผลทำให้เกิดความเสียหายที่คาดไม่ถึงได้

1.3.12 ขาดมาตรฐานการรองรับที่ชัดเจน

2. แบบจำลองฐานข้อมูลเครือข่าย (Network database model)

แบบจำลองฐานข้อมูลเครือข่าย พัฒนามาจากแบบจำลองฐานข้อมูลลำดับชั้น มีวัตถุประสงค์เพื่อกำหนดให้เป็นรูปแบบโครงสร้างข้อมูลที่เป็นมาตรฐาน และสามารถรองรับข้อมูลที่มีความสัมพันธ์ในแบบกลุ่มต่อกลุ่มได้ ลักษณะแบบจำลองฐานข้อมูลเครือข่ายจะคล้ายกับลักษณะแบบจำลองฐานข้อมูลลำดับชั้น แต่จะแตกต่างกันตรงที่แบบจำลองฐานข้อมูลเครือข่ายจะมีความยืดหยุ่นที่สูงกว่า เช่น สามารถมีต้นกำเนิดของข้อมูลได้มากกว่า 1 และยินยอมให้ระดับชั้นที่อยู่เหนือกว่า มีได้หลายแฟ้มข้อมูลถึงแม้ว่าระดับชั้นถัดลงมาจะมีเพียงแฟ้มข้อมูลเดียว ในขณะที่แบบจำลองฐานข้อมูลลำดับชั้นจะมีเพียงพ่อเดียวหรือ แฟ้มข้อมูลเท่านั้น

แบบจำลองเครือข่ายนี้จะใช้อุปกรณ์เก็บข้อมูลสำหรับตัวบ่งชี้ หรือพอยน์เตอร์ (Pointer) เป็นตัวโยงความสัมพันธ์ระหว่างระเบียนในไฟล์ต่าง ๆ รวมทั้งสนับสนุนความสัมพันธ์ทั้งแบบหนึ่งต่อกลุ่ม และความสัมพันธ์แบบกลุ่มต่อกลุ่ม ทำให้โครงสร้างของข้อมูลแต่ละแฟ้มข้อมูลมีความสัมพันธ์

คล้าย ๆ ร่างแห เช่น พนักงานขาย 1 คน สามารถออกใบส่งของได้หลายใบ แต่ละใบจะมีชื่อพนักงานขายเพียงชื่อเดียว หรือ ลูกค้าคนหนึ่งอาจจะมีการซื้อสินค้าได้หลายครั้ง จึงอาจจะมีใบส่งของได้หลายใบ และแต่ละใบจะมีชื่อผู้ซื้อได้เพียง 1 ชื่อเท่านั้น เป็นต้น

2.1 ลักษณะของแบบจำลองฐานข้อมูลเครือข่ายมีลักษณะดังนี้

2.1.1 สมาชิกของเซตหนึ่ง ๆ สามารถเป็นสมาชิกของเซตอื่นได้

2.1.2 ข้อมูลมีความสัมพันธ์กันแบบ 1: 1, 1: M และ N: M)

2.1.3 นำเสนอข้อมูลหลายรายการ (Multilist)

2.1.4 โครงสร้างเป็นเซตของระเบียน (Record set)

2.1.5 เซตของระเบียนประกอบ ด้วย Owner และ Member record

2.1.6 Member record สามารถมี Owner record ได้หลายระเบียน

2.2 ข้อดีแบบจำลองฐานข้อมูลเครือข่าย

2.2.1 สนับสนุนความสัมพันธ์แบบกลุ่มต่อกลุ่มได้

2.2.2 ข้อมูลมีความซ้ำซ้อนน้อยกว่าแบบจำลองแบบลำดับชั้น

2.2.3 สามารถเชื่อมโยงข้อมูลแบบไปกลับได้

2.2.4 สะดวกในการค้นหาเพราะไม่ต้องไปเริ่มค้นหาตั้งแต่ต้นกำเนิด

2.2.5 มีความยืดหยุ่นในการค้นหาข้อมูล เข้าถึงข้อมูลได้ง่าย

2.2.6 สนับสนุนให้โปรแกรมมีความเป็นอิสระจากข้อมูล

2.3 ข้อจำกัดแบบจำลองฐานข้อมูลเครือข่าย

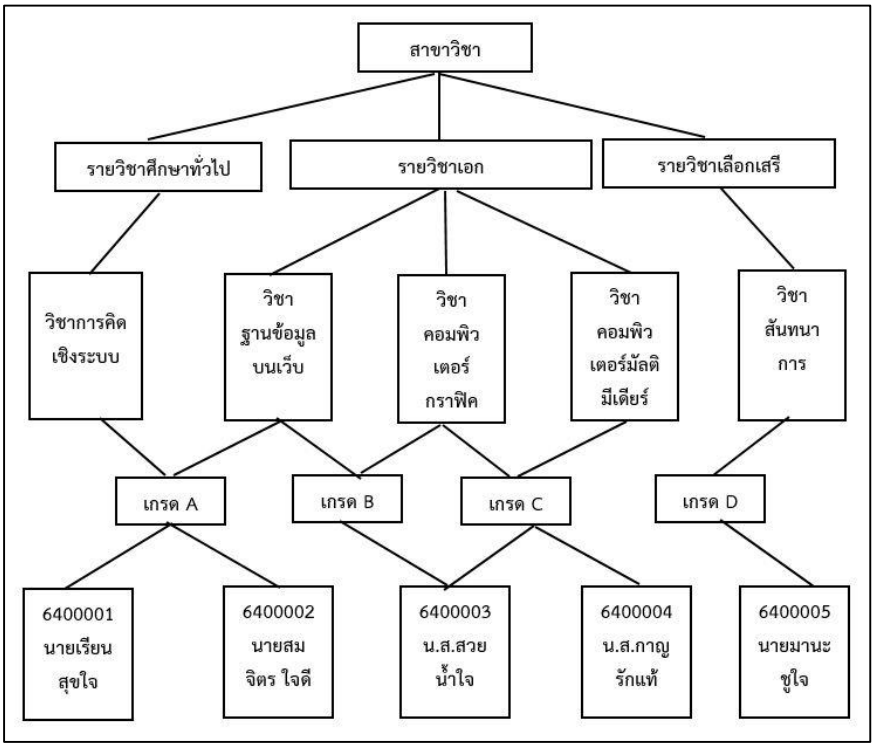
2.3.1 การป้องกันความปลอดภัยของข้อมูลมีน้อย เพราะสามารถเข้าถึงระเบียนได้โดยตรง

2.3.2 สิ้นเปลืองเนื้อที่หน่วยความจำในการเก็บพอยน์เตอร์

2.3.3 การออกแบบและการเปลี่ยนแปลงข้อมูลทำได้ค่อนข้างยากเพราะต้องกำหนดความสัมพันธ์ให้ครอบคลุมทุกข้อมูล

2.3.4 ความสัมพันธ์ข้อมูลที่เชื่อมโยงกันไปมายากต่อการใช้งาน

2.3.5 การป้องกันความลับของข้อมูลทำได้ยากเพราะมีโครงสร้างแบบง่าย ๆ ไม่ซับซ้อน



ภาพที่ 3.13 ตัวอย่างแบบจำลองฐานข้อมูลแบบเครือข่าย

3. แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational database model)

เป็นแบบจำลองที่นิยมแพร่หลายมากที่สุดในปัจจุบัน เพราะใช้งานง่าย ไม่ยุ่งยากซับซ้อน คิดค้นโดย Edgar Frank Codd (จาก บริษัท IBM ในปี ค.ศ.197O) เป็นแบบจำลองฐานข้อมูลที่แสดงข้อมูลในรูปแบบของตาราง ภายในตารางจะประกอบด้วยแถว และคอลัมน์ ข้อมูลของแต่ละแถวเรียกว่า ทูเพิล (Tuple) คอลัมน์ในตารางจะหมายถึงลักษณะหรือคุณสมบัติหรือ Attribute ของตาราง แบบจำลองฐานข้อมูลนี้สามารถแสดงความสัมพันธ์ข้อมูลได้ทั้งแบบหนึ่งต่อหนึ่ง หนึ่งต่อกลุ่ม และแบบกลุ่มต่อกลุ่ม และใช้ทั้งคีย์หลัก (Primary key) และ คีย์รอง (Secondary key) ในการอ้างอิงกับตารางอื่น

ข้อแตกต่างแบบจำลองเชิงสัมพันธ์กับแบบจำลองลำดับชั้น และแบบจำลองเครือข่าย คือ ในส่วนรายละเอียดของการจัดเก็บข้อมูลจะถูกซ่อนไว้จากผู้ใช้ การเรียกใช้ข้อมูลจะผ่านทางภาษาสอบถาม (Query language) ซึ่งเป็นภาษาที่มีประสิทธิภาพสูงและใช้ง่าย ภาษาที่ได้รับความนิยมในกลุ่มผู้ใช้ในปัจจุบัน คือภาษา SQL (Structured query language) ซึ่งสร้างขึ้นโดยใช้พื้นฐานของแบบจำลองเชิงสัมพันธ์

การเชื่อมโยงข้อมูลระหว่างตารางจะใช้ Attribute ที่มีอยู่ทั้ง 2 ตารางเป็นตัวเชื่อมโยงข้อมูลที่สามารถสื่อสัมพันธ์กับผู้ใช้ได้เข้าใจง่ายที่สุด ข้อมูลที่จัดเก็บอยู่ในตาราง สามารถจัดเก็บข้อมูลในส่วนของตน และมีความสัมพันธ์กับตารางอื่น ๆ ได้

3.1 ลักษณะของแบบจำลองฐานข้อมูลเชิงสัมพันธ์มีลักษณะดังนี้

3.1.1 ข้อมูลถูกเก็บในรูปของตาราง 2 มิติ คือ แถวและคอลัมน์

3.1.2 ข้อมูลในแต่ละแถวต้องมีชื่อกำกับ และชื่อจะต้องไม่ซ้ำกัน

3.1.3 ค่าของข้อมูลแต่ละคอลัมน์ต้องเป็นค่าของข้อมูลชนิดเดียวกัน

3.1.4 การเรียงลำดับของข้อมูลแต่ละแถวไม่เป็นประเด็นสำคัญ

3.1.5 การเรียงลำดับของข้อมูลแต่ละคอลัมน์ไม่เป็นประเด็นสำคัญ

3.1.6 ข้อมูลในแต่ละแถวหนึ่ง ๆ จะบรรจุข้อมูลได้เพียงค่าเดียวเท่านั้น

3.1.7 ค่าของข้อมูลในแต่ละคอมลัมน์จะต้องเป็นไปตามข้อกำหนดของโดเมน (Domain) หรือการกำหนดขอบเขตค่าข้อมูล และชนิดของข้อมูลในคอมลัมน์นั้น

ภาพที่ 3.14 ตัวอย่าง Relation ในแบบจำลองฐานข้อมูลเชิงสัมพันธ์

ดีกรี (Degree) คือ จำนวน Attribute ใน Relation นั้น ๆ

คาร์ดินัลลิตี (Cardinality) คือ จำนวนทูเพิลใน Relation

3.2 ข้อดีแบบจำลองฐานข้อมูลเชิงสัมพันธ์

3.2.1 รูปแบบโครงสร้างสามารถสื่อสารและเข้าใจได้ง่าย

3.2.2 สนับสนุนความสัมพันธ์แบบกลุ่มต่อกลุ่ม

3.2.3 มีความยืดหยุ่นในการเข้าถึงข้อมูล ความซ้ำซ้อนข้อมูลมีน้อย

3.2.4 มีข้อมูลความคงสภาพ (Data integrity) และเป็นมาตรฐาน

3.2.5 โครงสร้างข้อมูลมีความเป็นอิสระจากโปรแกรม

3.2.6 สามารถเลือกแสดงข้อมูลตามเงื่อนไขได้หลาย Key field

3.2.7 มีระบบรักษาความปลอดภัยที่ดี

3.3 ข้อจำกัดแบบจำลองฐานข้อมูลเชิงสัมพันธ์

3.3.1 การแก้ไขปรับปรุงแฟ้มข้อมูลทำได้ยาก

3.3.2 ระบบยังมีความซับซ้อน และมีข้อจำกัดด้านประสิทธิภาพ

3.3.3 จะต้องสร้างตารางขึ้นมาใหม่เมื่อมีการประมวลผล

3.3.4 ใช้ฮาร์ดแวร์และซอฟท์แวร์ที่มีความสามารถสูงซึ่งมีราคาแพง

ภาพที่ 3.15 ความสัมพันธ์ของแบบจำลองฐานข้อมูลเชิงสัมพันธ์

4. แบบจำลองฐานข้อมูลเชิงวัตถุ (Object-oriented database model)

แบบจำลองฐานข้อมูลเชิงวัตถุ เป็นเทคโนโลยีที่พัฒนาต่อจากแบบจำลองฐานข้อมูลเชิงสัมพันธ์ โดยมีแนวคิดหลักที่ประกอบไปด้วยวัตถุ (Object) การปกปิดความลับของวัตถุ (Encapsulation) คลาส (Class) การสืบทอด (Inheritance) คุณลักษณะ (Identity) และโดเมน (Domain) เป็นต้น

ฐานข้อมูลเชิงวัตถุมีจุดกำเนิดขึ้นมาในช่วงกลางปี ค.ศ. 1980 (Kim, 1990) โดยมีแนวคิดมาจากการเขียนโปรแกรมเชิงวัตถุ (Object oriented program หรือ OOP) ที่ได้รับความนิยมในการใช้งานเป็นอย่างมาก โดยมองทุกอย่างเป็นวัตถุและแต่ละวัตถุจะประกอบด้วยข้อมูลและโอเปอเรชั่น หรือการปฏิบัติงานที่มี Class หรือต้นแบบข้อมูลเป็นตัวกำหนดคุณสมบัติของวัตถุ ซึ่งจะคล้าย ๆ กับ Attribute ที่ใช้อธิบาย Entity ในแบบจำลองฐานข้อมูลเชิงสัมพันธ์นั่นเอง และฐานข้อมูลเชิงสัมพันธ์จะไม่เหมาะกับโครงสร้างฐานข้อมูลที่มีขนาดใหญ่และมีความซับซ้อน

แบบจำลองฐานข้อมูลเชิงวัตถุ เป็นแบบจำลองฐานข้อมูลที่แตกต่างไปจากแบบจำลองฐานข้อมูลแบบเชิงสัมพันธ์ เช่น ฐานข้อมูลเชิงวัตถุจะเก็บข้อมูลไว้ในวัตถุ แต่แบบจำลองฐานข้อมูลแบบเชิงสัมพันธ์จะเก็บข้อมูลต่าง ๆ ไว้ในตาราง และจากคุณสมบัติที่สำคัญของฐานข้อมูลเชิงวัตถุดังที่กล่าวมา จะช่วยให้การจัดเก็บข้อมูลมีความซ้ำซ้อนลดน้อยลง ฐานข้อมูลมีความคงสภาพมากขึ้น ลดความขัดแย้งของข้อมูล และไม่มีการสูญเสียประสิทธิภาพของระบบจัดการฐานข้อมูล

4.1 ลักษณะแบบจำลองฐานข้อมูลเชิงวัตถุมีลักษณะสำคัญดังนี้

4.1.1 มองทุกสิ่งเป็นวัตถุ วัตถุประกอบด้วยข้อมูลและโอเปอเรชั่น

4.1.2 มีคลาสเป็นตัวกำหนดรายละเอียดของวัตถุ

4.1.3 โครงสร้างของเชิงวัตถุกำหนดโดยใช้คุณสมบัติของคลาส

4.1.4 มีคุณสมบัติ โพลีมอร์ฟิสม (Polymorphism) หรือการออกแบบเดียวแต่สามารถตอบสนองได้หลายรูปแบบ

4.1.5 การปกปิดความลับของวัตถุ ป้องกันไม่ให้วัตถุอื่นที่อยู่ภายนอกเข้าถึงวัตถุได้อย่างอิสระ

4.2 ข้อดีแบบจำลองฐานข้อมูลเชิงวัตถุ

4.2.1 คุณสมบัติการสืบทอด ทำให้ข้อมูลมีความคงสภาพสูง

4.2.2 การนำเสนอแบบ Visual ทำให้อธิบายหัวข้อความหมายได้ดี

4.2.3 สามารถจัดการกับข้อมูลที่มีความซับซ้อนได้เป็นอย่างดี

4.2.4 สนับสนุนคุณสมบัติของการนำกลับมาใช้ใหม่ (Reusable)

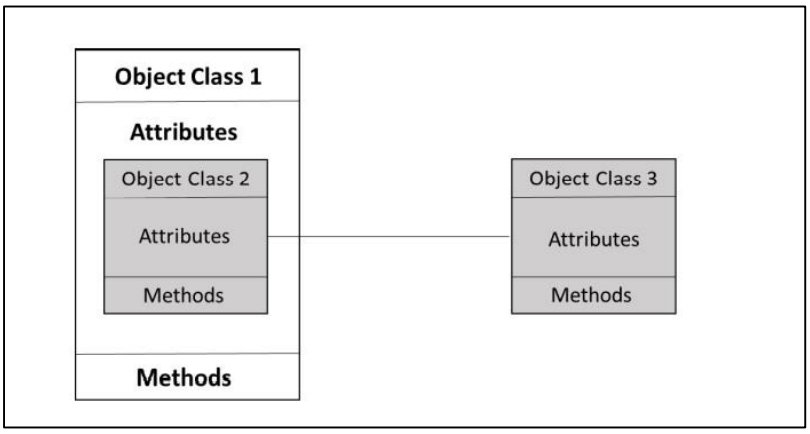
4.2.5 การเข้าถึงข้อมูลในแบบจำลองฐานข้อมูลเชิงวัตถุ ผู้ใช้ไม่จำเป็นต้องรู้วิธีการทำงานภายในของแต่ละโอเปอเรชั่นของวัตถุ

4.3 ข้อจำกัดแบบจำลองฐานข้อมูลเชิงวัตถุ

4.3.1 ค่าใช้จ่ายเกี่ยวกับระบบแบบจำลองฐานข้อมูลเชิงวัตถุค่อนข้างสูง

4.3.2 ต้องอาศัยบุคลากรที่มีความรู้ ความเชี่ยวชาญด้านเทคโนโลยีเชิงวัตถุในการจัดการกับฐานข้อมูลเชิงวัตถุ

4.3.3 ยังไม่มีมาตรฐานรองรับที่ชัดเจน



ภาพที่ 3.16 ความสัมพันธ์ของแบบจำลองฐานข้อมูลเชิงวัตถุ

5. แบบจำลองฐานข้อมูลมัลติไดเมนชัน (Multidimensional database model)

แบบจำลองฐานข้อมูลมัลติไดเมนชัน เป็นแบบจำลองฐานข้อมูลที่พัฒนาจากแบบจำลองฐานข้อมูลเชิงสัมพันธ์ เป็นการแสดงข้อมูลหลายมิติ การจัดเก็บข้อมูลและการแสดงแถวและคอลัมน์ข้อมูลจะมีลักษณะเป็นลูกบาศก์นิยมใช้กับคลังข้อมูลทำให้มองเห็นและวิเคราะห์ข้อมูลได้ 2 ทาง ทำให้การวิเคราะห์ข้อมูลกระทำได้ง่ายขึ้น มองเห็นปัญหาและสร้างวิธีการแก้ไขปัญหาได้ดียิ่งขึ้น

5.1 ลักษณะของแบบจำลองฐานข้อมูลมัลติไดเมนชัน มีลักษณะดังนี้

5.1.1 แถวและคอลัมน์จะมีลักษณะเป็นรูปลูกบาศก์ (Data cube)

5.1.2 มองข้อมูลได้ 2 ทาง ทำให้เห็นปัญหาและสร้างวิธีแก้ปัญหาได้

5.1.3 ใช้แบบจำลอง Star schema ในการออกแบบ

5.1.4 มี Fact table และ Dimension table ทำงานร่วมกันในการจัดเก็บข้อมูล

5.2 ข้อดีแบบจำลองฐานข้อมูลมัลติไดเมนชั่น

5.2.1 เป็นแบบจำลองที่นำเสนอมุมมองข้อมูลได้หลายมิติ

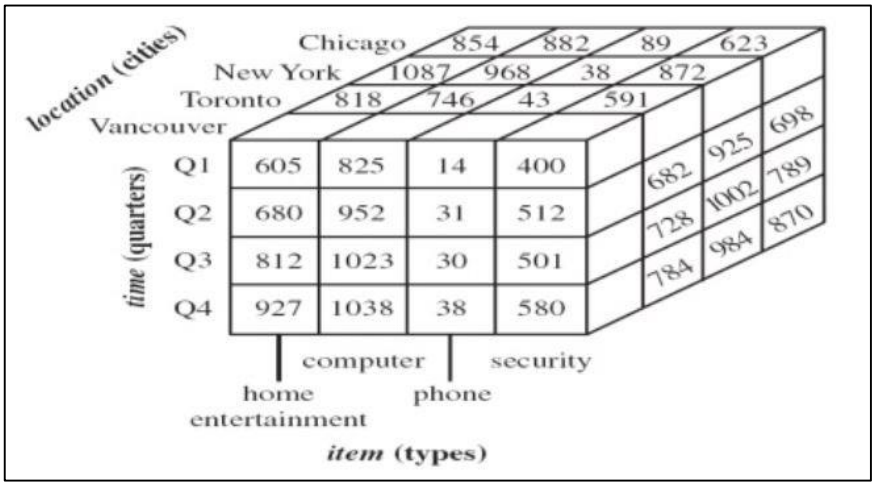
5.2.2 สามารถนำไปประยุกต์ใช้ในการตัดสินใจ การวางแผนและการแก้ปัญหาทางธุรกิจได้

5.3 ข้อจำกัดแบบจำลองฐานข้อมูลมัลติโดเมนชั่น

5.3.1 การใช้งานต้องอาศัยผู้เชี่ยวชาญเฉพาะด้าน

5.3.2 การวิเคราะห์ข้อมูลมีค่าใช้จ่ายสูง

5.3.3 เป็นแบบจำลองฐานข้อมูลที่เหมาะกับธุรกิจขนาดใหญ่



ภาพที่ 3.17 ความสัมพันธ์ของแบบจำลองฐานข้อมูลมัลติไดเมนชั่น (Han, Kamber & Pei, 2012)

คุณลักษณะพื้นฐานสำคัญของแบบจำลองฐานข้อมูล

ปัจจุบันการออกแบบจำลองฐานข้อมูลจะมุ่งเน้นที่การแก้ไขจุดอ่อนของแบบจำลองก่อนหน้า เช่น แบบจำลองฐานข้อมูลแบบเครือข่าย ได้พัฒนาขึ้นมาเพื่อแทนที่แบบจำลองฐานข้อมูลแบบลำดับชั้น โดยที่แบบจำลองฐานข้อมูลแบบเครือข่าย จะสามารถปรับเปลี่ยนความสัมพันธ์ที่ซับซ้อนให้มีความง่ายมากขึ้น แบบจำลองฐานข้อมูลเชิงสัมพันธ์ จะมีประโยชน์มากกว่าแบบจำลองแบบเครือข่ายในหลาย ๆ ด้าน เช่น การแสดง ข้อมูลด้วยวิธีการที่ง่าย ความไม่สอดคล้องของข้อมูลมีน้อยกว่า และการประยุกต์ใช้ภาษาคิวรีในการเข้าถึงข้อมูล และแบบจำลองฐานข้อมูลเชิงวัตถุได้พัฒนาต่อจากแบบจำลองฐานข้อมูลเชิงสัมพันธ์ โดยการเพิ่มเติมแนวความคิดเชิงวัตถุ ที่จะช่วยให้สามารถจัดการกับข้อมูลที่มีความซับซ้อนได้

คุณลักษณะของแบบจำลองฐานข้อมูลที่ดี ควรมีลักษณะดังนี้

1. มีความเข้าใจได้ง่าย แบบจำลองฐานข้อมูลที่ดีควรที่จะต้องใช้กฎเกณฑ์ ทั่ว ๆ ไป จะทำให้เข้าใจได้ง่าย โดยมีข้อมูล Attribute ที่อธิบายในรายละเอียดของแต่ละ Entity

2. มีสาระสำคัญและไม่ซ้ำซ้อน Attribute ในแต่ละ Entity จะต้องมีสาระสำคัญ ไม่ควรมีข้อมูลที่ซ้ำซ้อนกัน

3. มีความยืดหยุ่นและง่ายต่อการปรับปรุง แบบจำลองฐานข้อมูลที่ดีต้องมีความยืดหยุ่นและง่ายต่อการปรับปรุง หรือการพัฒนาในอนาคต และไม่ควรขึ้นอยู่กับแอปพลิเคชันโปรแกรม

4. มีความเป็นอิสระของข้อมูล คือผู้ใช้สามารถเปลี่ยนแปลงข้อมูลในระดับแนวความคิด (เชิงตรรกะ) หรือระดับภายใน (เชิงกายภาพ) ได้โดยที่ไม่กระทบกับโปรแกรม และผู้ใช้ยังคงสามารถใช้งานได้ และมองเห็นโครงสร้างข้อมูลในระดับภายนอกได้เหมือนเดิม

สรุป

วัตถุประสงค์สำคัญของแบบจำลองฐานข้อมูลก็เพื่อนำแนวความคิดต่าง ๆ มาเสนอให้เกิดเป็นแบบจำลองฐานข้อมูล ซึ่งเป็นเครื่องมือสำคัญที่ใช้ในการอธิบายถึงโครงสร้างข้อมูล และความสัมพันธ์ระหว่างข้อมูลภายในฐานข้อมูลจากรูปแบบที่เป็นแนวคิดที่เข้าใจยาก ให้อยู่ในรูปแบบที่เข้าใจง่าย เพราะแบบจำลองฐานข้อมูลจะประกอบไปด้วยรูปภาพสัญลักษณ์ต่าง ๆ ที่แสดงให้เห็นถึงการทำงานของระบบ ทำให้การสื่อสารระหว่างผู้ออกแบบฐานข้อมูล ผู้เขียนโปรแกรมและผู้ใช้ระบบฐานข้อมูลให้เข้าใจถูกต้องตรงกันมากขึ้น

ประเภทแบบจำลองข้อมูล ดังนี้ 1) แบบจำลองฐานข้อมูลลำดับชั้น (Hierarchical data model) 2) แบบจำลองฐานข้อมูลเครือข่าย (Network data model) 3) แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational database model) 4) แบบจำลองฐานข้อมูลเชิงวัตถุ (Object-oriented database model) 5) แบบจำลองฐานข้อมูลมัลติไดเมนชัน (Multidimensional database model)

บทที่ 4

แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relation database model)

แบบจำลองข้อมูลมีลักษณะเป็นแผนภาพที่ใช้แสดงโครงสร้างที่ซับซ้อนของฐานข้อมูล มีหน้าที่ในการช่วยให้ผู้ออกแบบฐานข้อมูล สามารถเข้าใจความซับซ้อนของข้อมูลที่ถูกใช้ในองค์กร นอกจากนั้น แบบจำลองข้อมูลมักจะแสดงถึงโครงสร้างของข้อมูลในฐานข้อมูลและคุณลักษณะของข้อมูลเหล่านั้น และบทที่ 3 ได้กล่าวถึงชนิดของแบบจำลองฐานข้อมูล 5 ชนิด ได้แก่ 1) แบบจำลองฐานข้อมูลลำดับชั้น (Hierarchical database model) 2) แบบจำลองฐานข้อมูลเครือข่าย (Network database model) 3) แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relation database model) 4) แบบจำลองฐานข้อมูลเชิงวัตถุ (Object-oriented database model) และ 5) แบบจำลองฐานข้อมูลมัลติไดเมนชัน (Multidimensional database model) แต่ในปัจจุบัน พบว่า ฐานข้อมูลเชิงสัมพันธ์เป็นรูปแบบฐานข้อมูลที่ง่ายสำหรับผู้ใช้งานทั่วไป โดยเฉพาะกับผู้ที่ไม่ใช่นักวิเคราะห์ หรือนักออกแบบระบบ จึงทำให้ฐานข้อมูลนี้เป็นที่นิยมใช้กันมากกว่าฐานข้อมูลแบบอื่น ๆ ดังนั้น ในบทนี้จะกล่าวถึงแบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relation database model) โดยนำเสนอเนื้อหาสำหรับการศึกษาดังต่อไปนี้

- ความหมายของ “ฐานข้อมูลเชิงสัมพันธ์”
- กฎ 12 ข้อ ของ E.F Codd (Codd's 12 Rules)
- คุณสมบัติของ Relation (Properties of relations)
- Relation คีย์ (Relation key)
- กฎความคงสภาพข้อมูล (Data integrity rules)
- ข้อดีและข้อจำกัดของแบบจำลองฐานข้อมูลเชิงสัมพันธ์

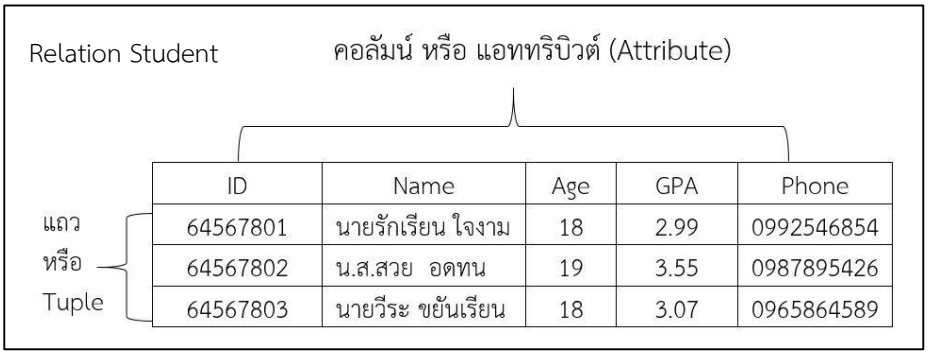
ฐานข้อมูลเชิงสัมพันธ์

เนื่องจากแบบจำลองฐานข้อมูลแบบลำดับชั้น และแบบจำลองฐานข้อมูลแบบเครือข่าย มีข้อจำกัดในการจัดการข้อมูลภายใน เช่น การเปลี่ยนแปลงโครงสร้างของข้อมูลในแบบจำลองฐานข้อมูลแบบลำดับชั้นส่งผลให้ต้องมีการปรับเปลี่ยน และแก้ไขโปรแกรมจะต้องมีการเขียนโปรแกรมจัดการในด้านนี้ โดยเฉพาะโดยผู้ที่มีความเชี่ยวชาญในการเขียนโปรแกรม และการเปลี่ยนแปลงข้อมูลในแบบจำลองฐานข้อมูลแบบเครือข่าย จะทำได้ค่อนข้างยาก เนื่องจากต้องกำหนดความสัมพันธ์ให้ครอบคลุมทุกข้อมูลในฐานข้อมูล จะเห็นได้ว่าการจัดการข้อมูลในฐานข้อมูลทั้ง 2 แบบ มีความยุ่งยากและซับซ้อนมาก

เหตุผลจากข้อจำกัดดังกล่าว ในปี ค.ศ.1970 Edgar Frank Codd จึงได้คิดค้นและพัฒนาแบบจำลองฐานข้อมูลแบบใหม่ขึ้นมา คือ แบบจำลองฐานข้อมูลเชิงสัมพันธ์ ซึ่งเป็นแบบจำลองที่ใช้หลักพื้นฐานจากทฤษฎีทางคณิตศาสตร์เกี่ยวกับเซต (Set) มากำหนดส่วนประกอบของแบบจำลองในส่วนของโครงสร้างข้อมูลส่วนควบคุมความถูกต้องของข้อมูลและส่วนจัดการข้อมูล

แบบจำลองฐานข้อมูลเชิงสัมพันธ์ เป็นแบบจำลองฐานข้อมูลที่นิยมใช้กันมากที่สุดในปัจจุบัน เพราะการออกแบบระบบฐานข้อมูลมีทฤษฎีรองรับ สามารถลดความซ้ำซ้อนของข้อมูลได้ มีการเก็บข้อมูลเป็นตาราง 2 มิติ ทำให้เห็นภาพข้อมูลชัดเจน ผู้ใช้เรียนรู้และประยุกต์การใช้งานได้ง่าย สามารถเรียกใช้หรือเชื่อมโยงข้อมูลได้ง่ายโดย ไม่จำเป็นต้องทราบว่าข้อมูลนั้นถูกจัดเก็บจริงอยู่ที่ตำแหน่งใดในสื่อบันทึกข้อมูล และกลุ่มคำสั่งหรือภาษาที่ใช้ในการจัดการข้อมูลเป็นภาษา SQL (Structured query language) ซึ่งมีประสิทธิภาพสูงและเข้าใจง่าย

ฐานข้อมูลเชิงสัมพันธ์ (Relation database) เป็นรูปแบบของฐานข้อมูลที่ได้รับความนิยมมากที่สุด (ประภัสสร สิเนหะวัฒนะ, 2555) ฐานข้อมูลประเภทนี้จะมีลักษณะของการจัดเก็บข้อมูลไว้ในรูปแบบของตาราง (Table) ซึ่งแทนความสัมพันธ์ (Relation) และภายในตารางก็จะประกอบไปด้วย แถวแทนเรคคอร์ดของแต่ละรายการข้อมูล ซึ่งเรียกว่า Tuple และคอลัมน์แทนลักษณะของข้อมูล เรียกว่า แอททริบิวต์ (Attribute)



ID	Name	Age	GPA	Phone
64567801	นายรักเรียน ใจงาม	18	2.99	0992546854
64567802	น.ส.สวย อดทน	19	3.55	0987895426
64567803	นายวีระ ขยันเรียน	18	3.07	0965864589

ภาพที่ 4.1 ส่วนประกอบภายในตารางของฐานข้อมูลเชิงสัมพันธ์

ฐานข้อมูลเชิงสัมพันธ์ (Relation database) ประกอบด้วยกลุ่มของ Relation โดยแต่ละ Relation คือ Table หรือตารางข้อมูล ที่แต่ละแถวของตาราง คือตัวแทนค่าของข้อมูลที่สัมพันธ์กัน ซึ่งข้อมูลเหล่านั้น คือ ข้อเท็จจริงที่เกิดขึ้นในระบบ เช่น Table “Student data” จากภาพที่ 4.1 แสดงข้อเท็จจริงของนักศึกษา โดยแต่ละแถวของ Table คือ ข้อเท็จริงของศึกษาแต่ละคน เป็นต้น

โครงสร้างฐานข้อมูลเชิงสัมพันธ์

โครงสร้างของฐานข้อมูลเชิงสัมพันธ์ประกอบไปด้วย Relation, Attribute, Domain, Tuple, Degree และ Cardinality ดังนี้

Relation คือ ความสัมพันธ์ระหว่างข้อมูลที่อยู่ในรูปตาราง 2 มิติ แสดงข้อมูลในแนวแถวและคอลัมน์

Attribute คือ ชื่อคอลัมน์ของ Relation

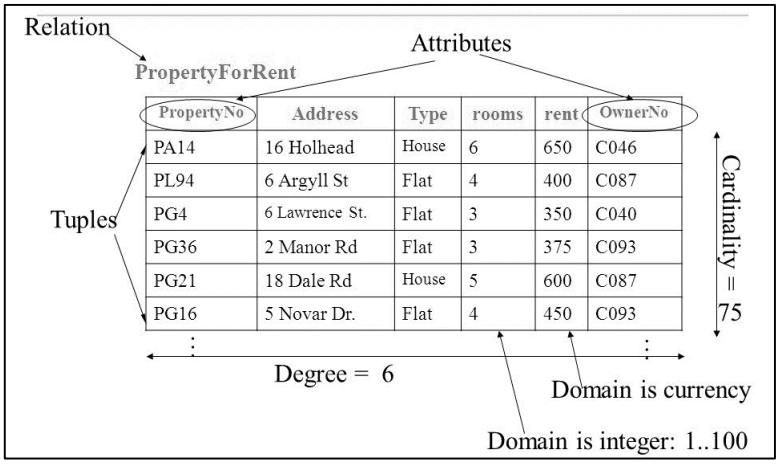
Domain คือ กลุ่มของค่าข้อมูลที่เป็นไปได้ของแต่ละ Attribute

Tuple คือ แถวของ Relation

Degree คือ จำนวนของ Attribute ในแต่ละ Relation

Cardinality คือ จำนวนของ Tuple ในแต่ละ Relation

Domain คือ ค่าที่เป็นไปได้ของข้อมูลแต่ละ Attribute



ภาพที่ 4.2 ตัวอย่างโครงสร้างฐานข้อมูลเชิงสัมพันธ์ (Software School of Hunan University, 2006)

1. Relation

Relation คือ หน่วยที่ใช้จัดเก็บข้อมูลของฐานข้อมูลเชิงสัมพันธ์ มีลักษณะเป็นตาราง (Table) ขนาด 2 มิติ ที่ประกอบด้วย แถว (Row) และคอลัมน์ (Column) บางครั้งมีการใช้คำว่า “Relation” และ “Table” ในความหมายเดียวกัน ลักษณะของ Relation เป็นดังภาพที่ 4.2

คุณสมบัติของ Relation มีดังนี้

1.1 Row

- แถวของ Relation เรียกว่า “Tuple” เป็นชื่อที่ใช้อ้างถึงกลุ่มข้อมูลที่เกิดขึ้นในแนวแถว (Occurrence/ Object instance)

- Tuple คือ ข้อมูล 1 รายการ ซึ่งเทียบเท่ากับ Record ในระบบแฟ้มข้อมูล

- จำนวน Tuple ทั้งหมดที่อยู่ใน Relation ถือว่าเป็น “Cardinality” ของ Relation

- แต่ละคอลัมน์ที่ตัดกับแต่ละแถว คือข้อมูลแต่ละส่วนย่อยที่เมื่อนำมาประกอบกันแล้วทำให้เป็น Tuple

- จากตัวอย่าง Relation ในภาพที่ 4.2 มี Tuple ทั้งหมด 6 Tuples

1.2 Column

- คอลัมน์ของ Relation คือคุณลักษณะ (Attribute) ต่าง ๆ ของข้อมูลในแต่ละแถว ซึ่งเทียบเท่ากับ Field ของ Record ในระบบแฟ้มข้อมูล

- ชื่อคอลัมน์คือ “Attribute” ซึ่งไม่อนุญาตให้ซ้ำกัน

- หากใน Attribute ใดไม่มีค่าข้อมูล จะถือว่า Attribute นั้นมีค่าเป็น Null

- ค่าข้อมูลของแต่ละ Attribute จะต้องอยู่ในช่วงข้อมูลที่ถูกกำหนดไว้ ซึ่งเรียกว่า “Domain” ของข้อมูล

- จากตัวอย่าง Relation ในภาพที่ 4.2 มี Attribute ทั้งหมด 6 Attributes

1.3 Domain

เป็นการนิยามขอบเขตของค่าที่เป็นไปได้ให้กับข้อมูลแต่ละ Attribute เพื่อป้องกันไม่ให้เกิดการป้อนข้อมูลเกินขอบเขตที่กำหนด เช่น ค่าของเงินเดือนอาจารย์จะต้องมากกว่า 0 เพื่อป้องกันไม่ให้ผู้ใช้ป้อนจำนวนเงินเป็น 0 หรือมีค่าติดลบ หรือการกำหนดให้เพศของนักศึกษามีค่าได้ 2 ค่า คือ ชาย (M) หรือหญิง (F) เป็นต้น

1.4 Degree

Degree ของ Relation คือ จำนวนของ Attribute ของ Relation นั้น จากภาพที่ 4.2 Relation “PropertyForRent” มี Degree = 6 เนื่องจากมีทั้งหมด 6 Attribute ซึ่งหมายความว่าแต่ละแถวของ Relation ประกอบไปด้วยค่าข้อมูลทั้งหมด 6 ค่า

1.5 Cardinality

คือ จำนวนของ Tuple ใน Relation หนึ่งที่มีความสัมพันธ์กับ Tuple ในอีก Relation หนึ่ง ซึ่งจะนำไปใช้กำหนดประเภทของความสัมพันธ์ระหว่างข้อมูลทั้ง 2 Relation ได้ เช่น จากภาพที่ 4.3 จะเห็นว่าข้อมูลใน Relation “Faculty” ที่มีความสัมพันธ์กับ Relation “Student” จะมี Cardinality เท่ากับ 1: M เนื่องจากสามารถมีนักศึกษาได้ตั้งแต่ 1 คนขึ้นไป และนักศึกษาแต่ละคนจะสังกัดอยู่ในคณะวิชาได้เพียงคณะวิชาเดียว

Student

ID	Name	CodeFaculty
64567801	นายรักเรียน ใจงาม	101
64567802	น.ส.สวย อดทน	101
64567803	นายวีระ ขยันเรียน	103

Faculty

CodeFaculty	FacultyName
101	คณะศึกษาศาสตร์
102	คณะมุษยศาสตร์
103	คณะนิติศาสตร์

ภาพที่ 4.3 แสดงตัวอย่างข้อมูลที่ไม่อิสระจาก Relation อื่น

คุณสมบัติของ Relation

Relation จะต้องมีคุณสมบัติ ดังนี้

1. เนื่องจาก Relation ในฐานข้อมูลเชิงสัมพันธ์ อยู่ในรูปแบบของเซตข้อมูลทางคณิตศาสตร์ ที่ภายในเซตจะต้องประกอบด้วยสมาชิกที่มีค่าไม่ซ้ำกัน ดังนั้น ภายใน Relation ใด ๆ จึงต้องมี Attribute ที่จะทำให้แต่ละ Tuple มีข้อมูลที่ไม่ซ้ำกัน ซึ่งก็คือ Attribute ที่เป็น Primary Key (อธิบายในหัวข้อถัดไป) เช่น ใน Relation “Student” หากไม่มี Attribute “ID” หรือรหัสนักศึกษา เป็น Primary key จะทำให้ผลลัพธ์ที่ได้เมื่อค้นหาชื่อนักศึกษา คือ มีนักศึกษาที่มีชื่อและนามสกุลเหมือนกันหลายคน จนทำให้ไม่ทราบว่าเป็นคนเดียวกันหรือคนละคน แต่หากมี Attribute “ID” ก็ทราบได้ว่าเป็นคนละคนกัน เป็นต้น

Student

ID	Name	Age	GPA	Phone
64567801	นายรักเรียน ใจงาม	18	2.99	0992546854
64567802	น.ส.สวย อดทน	19	3.55	0987895426
64567803	นายวีระ ขยันเรียน	18	3.07	0965864589

ภาพที่ 4.4 แสดงตัวอย่าง Relation “Student”

2. เนื่องจาก Relation ประกอบไปด้วยเซตของ Tuple ซึ่งในทางคณิตศาสตร์แล้ว สมาชิกในเซตสามารถสลับตำแหน่งกันได้ ดังนั้น ลำดับของ Tuple ใน Relation ใด ๆ จะไม่มีผลต่อ Relation นั้น ๆ เพราะข้อมูลของแต่ละ Tuple ถูกจำแนกความแตกต่างด้วย Primary Key แล้ว จึงไม่จำเป็นต้องกำหนดลำดับให้กับแต่ละ Tuple รวมถึงไม่จำเป็นต้องกำหนดลำดับให้กับ Attribute ด้วย เพราะ Attribute ใน Relation จะไม่มีชื่อที่ซ้ำกันเลย

3. ค่าข้อมูลในทุก Attribute ของ Relation จะต้องมีคุณสมบัติ “Atomic” ซึ่งเป็นคุณสมบัติที่กำหนดให้ค่าของข้อมูลในแต่ละ Attribute ของ Relation จะต้องมีความหมายใดความหมายหนึ่งเพียงความหมายเดียว หรือกล่าวคือ Attribute จะต้องมีค่าข้อมูลเพียงค่าเดียว (Single Value) ไม่สามารถมีหลายค่าได้ (Multi-valued Attribute)

ลักษณะของข้อมูลที่จัดเก็บ

ฐานข้อมูลเชิงสัมพันธ์ประกอบด้วยข้อมูลของ Attribute ต่าง ๆ ที่ถูกจัดเก็บอยู่ในรูปตาราง 2 มิติ คือ แถว (Row) และคอลัมน์ (Column) คุณลักษณะของจัดเก็บของ Relation เป็นดังนี้

1. ข้อมูลในแต่ละแถวจะไม่ซ้ำกัน

ข้อมูลที่จัดเก็บในแต่ละแถวจะไม่มีข้อมูลที่ซ้ำกัน ทั้งนี้เพราะการจัดการฐานข้อมูลที่ดีไม่ควรจะมีข้อมูลที่ซ้ำร้อนปรากฏอยู่ใน Relation โดยระบบจัดการฐานข้อมูลจะมีกลไกที่ใช้ในการควบคุมไม่ให้มีความซ้ำซ้อนเกิดขึ้น

ตัวอย่างเช่น จากรูป 4.5 มีข้อมูลของนักศึกษาหัสเป็นคีย์และมีรหัส 1001 ซ้ำกัน ระบบจะมีข้อความเตือนว่า รหัสนักศึกษาซ้ำซ้อนกัน เป็นต้น

ID	Name	Age	GPA	Phone
64567801	นายรักเรียน ใจงาม	18	2.99	0992546854
64567802	น.ส.สวย อดทน	19	3.55	0987895426
64567803	นายวีระ ขยันเรียน	18	3.07	0965864589
64567803	นางสาวสมพร ใจดี	17	2.98	096585625

ภาพที่ 4.5 ข้อมูลซ้ำซ้อนกันใน Relation

2. การเรียงลำดับของข้อมูลในแต่ละแถวไม่เป็นสาระสำคัญ

การจัดเก็บของข้อมูลใน Relation จะถูกจัดเรียงตามลำดับลงบนสื่อที่เก็บข้อมูล แต่การเรียกใช้ข้อมูลใน Relation สามารถเรียกใช้ตามที่ผู้ใช้ต้องการ

3. การเรียงลำดับของ Attribute จะเรียงลำดับก่อนหลังอย่างไรก็ได้

การเรียงลำดับของ Attribute แต่ละ Attribute ของ Relation จะเรียงลำดับอย่างไรก็ได้ ไม่มีการระบุว่า Attribute ซ้ายสุด คือ Attribute แรก หรือ Attribute ขวาสุด คือ Attribute สุดท้าย ทั้งนี้ เพราะการอ้างอิงถึง Attribute ใด จะใช้ชื่อของ Attribute นั้น ๆ ในการอ้างอิง ไม่ใช่ลำดับที่ Attribute นั้น ปรากฏอยู่

4. ค่าของข้อมูลในแต่ละ Attribute ของ Tuple บรรจุข้อมูลได้เพียงค่าเดียว (Single value)

ข้อมูลในแต่ละ Attribute ของ Tuple จะต้องบรรจุข้อมูลเพียงค่าเดียวไม่ใช่กลุ่มของข้อมูลที่แสดงค่าที่มากกว่า 1 แถว (Repeating group)

ID	Name	Age	GPA	Phone
64567801	นายรักเรียน ใจงาม	18	2.99	0992546854 0992541256
64567802	น.ส.สวย อดทน	19	3.55	0987895426
64567803	นายวีระ ขยันเรียน	18	3.07	0965864589
64567803	นางสาวสมพร ใจดี	17	2.98	096585625

ภาพที่ 4.6 Attribute ที่มีค่ามากกว่า 1 ค่า

จากรูป 4.6 ข้อมูลของนักศึกษารหัส 64567801 มีค่าของ Phone ที่มากกว่า 1 ค่า (Repeating group) ซึ่งไม่ใช่คุณลักษณะในการจัดเก็บข้อมูลของ Relation

5. ค่าของข้อมูลในแต่ละแอททริบิวต์จะบรรจุค่าของข้อมูลประเภทเดียวกัน

ข้อมูลในแต่ละ Attribute จะเป็นค่าของข้อมูลประเภทเดียวกัน ซึ่งถูกกำหนดค่าตามโดเมน (Domain) ตัวอย่างเช่น Attribute วันเกิด จะมีค่าของข้อมูลเป็นวันเดือนปีเกิดต่าง ๆ ในปีปฏิทิน หรือ Attribute สถานภาพจะมีค่าของข้อมูลที่แสดงว่าเป็นโสดหรือแต่งงาน โดยขอบเขตของคำที่ถูกกำหนดไว้ คือ S แทนค่า โสด และ M แทนค่าว่า แต่งงานแล้ว เป็นต้น (ศิริลักษณ์ โรจนกิจอำนวย, 2552)

Relation key

คุณสมบัติของ Relation ที่ว่าข้อมูลในแต่ละแถวจะต้องไม่ซ้ำกัน นั่นคือ ไม่มี Tuple คู่ใด ๆ ที่ซ้ำกัน และปัญหาจากความซ้ำซ้อนของข้อมูลที่จัดเก็บอยู่ในรูปแบบของแฟ้มข้อมูลดังที่ได้กล่าวมาแล้วในบทต้นๆ จำเป็นอย่างยิ่งต้องมีเครื่องมือเข้ามาช่วยควบคุมไม่ให้ข้อมูลเกิดความซ้ำซ้อน เครื่องมือดังกล่าว คือ คีย์ หรือ Relation key ซึ่งจะช่วยในการแยกความแตกต่างของข้อมูลในแต่ละแถว และยังใช้ในการเชื่อมโยงข้อมูลระหว่างตาราง ดังนั้น เราสามารถที่ระบุค่าใน Attribute ใด Attribute หนึ่ง ซึ่งไม่ซ้ำกันในแต่ละ Tuple ได้ นอกจากนี้ คีย์ยังเป็นเครื่องมือที่ช่วยในการกำหนด

ความสัมพันธ์ของ Entity คีย์ที่เกี่ยวข้องกับ Relation ที่รู้จักกันดี เช่น คีย์หลัก (Primary key) คีย์ร่วม (Composite หรือ Compound key) และคีย์นอก (Foreign key) เป็นต้น

Key เป็นเขตข้อมูลหรือฟิลด์ที่ใช้เป็นตัวเข้าถึงข้อมูลในแต่ละระเบียน หรือบ่งชี้ว่า ระเบียนใดเป็นระเบียนใด ถ้ากำหนดไว้ในตารางจะทำให้สามารถนำไปเชื่อมโยงกับตารางอื่นได้ หรืออาจจะสรุปได้ว่า คีย์ หมายถึง Attribute หรือกลุ่มของ Attribute ที่สามารถใช้ในการบ่งบอกความแตกต่างของแต่ละ Tuple นั่นเอง ดังนั้น ในการอ้างอิงการประมวลผล การค้นหา การแก้ไขข้อมูล หรือการกำหนดความสัมพันธ์ระหว่างข้อมูล จะดำเนินการได้อย่างสะดวกและมีประสิทธิภาพ จะต้องกำหนดคีย์ ให้กับตารางก่อน นอกจากนี้ ยังจะช่วยประหยัดพื้นที่ในการจัดเก็บข้อมูลอีกด้วย

1. ประเภทของคีย์ (Types of keys)

เนื่องจากต้องใช้คีย์ในการเชื่อมโยงตาราง ต่าง ๆ เข้าด้วยกัน คีย์แต่ละชนิดจึงทำหน้าที่แตกต่างกันในแต่ละตารางจะมีคีย์หลายชนิดหรือชนิดเดียวก็ได้ คีย์หลักๆ ที่ใช้กับฐานข้อมูลมีดังนี้

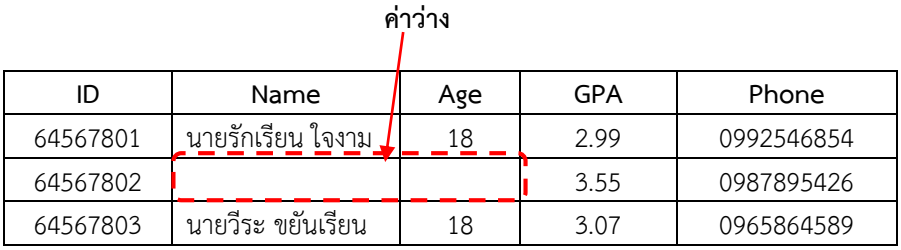
1.1 ซูเปอร์คีย์ (Super key) หมายถึง “Attribute หรือกลุ่มของ Attribute ที่เป็นไปได้ทั้งหมดที่สามารถใช้จำแนกความแตกต่างของแต่ละ Tuple ใน Relation ได้ ดังนั้น ใน 1 Relation สามารถมีได้หลายซูเปอร์คีย์ ถ้า Attribute หรือกลุ่ม Attribute ใด ที่มีค่าไม่ซ้ำกันเลยในแต่ละแถว (Unique) ก็จัดได้ว่า Attribute หรือกลุ่มของ Attribute นั้นเป็นซูเปอร์คีย์ได้” เช่น จากภาพที่ 4.7 Relation นักศึกษาจะมีซูเปอร์คีย์ ได้ 13 แบบ ดังนี้

รหัสนักศึกษา
รหัสนักศึกษา + ชื่อ
รหัสนักศึกษา + สกุล
รหัสนักศึกษา + คณะวิชา
รหัสนักศึกษา + สาขาวิชา
รหัสนักศึกษา + ชื่อ + สกุล
รหัสนักศึกษา + ชื่อ + คณะวิชา
รหัสนักศึกษา + ชื่อ + สาขาวิชา
รหัสนักศึกษา + สกุล + คณะวิชา
รหัสนักศึกษา + สกุล + สาขาวิชา
รหัสนักศึกษา + ชื่อ + สกุล + คณะวิชา
รหัสนักศึกษา + ชื่อ + สกุล + สาขาวิชา
รหัสนักศึกษา + ชื่อ + สกุล + คณะวิชา + สาขาวิชา

ภาพที่ 4.7 แสดงตัวอย่าง Super Key

1.2 คีย์หลัก (Primary key) หมายถึง Attribute หรือกลุ่มของ Attribute ของ Relation หนึ่ง ๆ ที่มีข้อมูลที่เป็นเอกลักษณ์ คือมีข้อมูลที่ไม่ซ้ำกัน และจะต้องไม่เป็นค่าว่าง *(ค่าที่ปล่อยให้ว่างเปล่าไว้ในบาง Attribute ของ Tuple อาจจะเป็นเพราะไม่มีค่าหรือยังไม่ทราบค่าในขณะนั้น ส่วนการกำหนดข้อมูลเป็นเลขศูนย์ 0 หรือ เครื่องหมายขีด - ไม่ใช่ค่าว่าง)* คุณสมบัติจะ

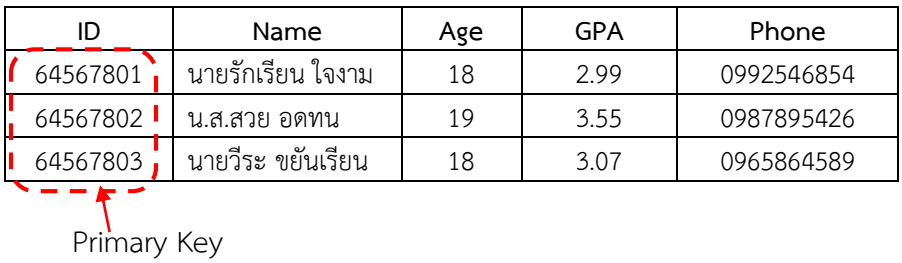
สามารถระบุว่าข้อมูลนั้นเป็นของ Tuple ใด ในทุก Relation จะต้องกำหนด Attribute ให้เป็นคีย์หลักของแต่ละ Relation เสมอ แต่ละ Relation ควรมีคีย์หลักเพียงคีย์เดียว คีย์หลักจะใช้ประโยชน์ในการระบุข้อมูลที่ต้องการเข้าถึงเพื่อใช้ในการค้นหา การจัดเรียงรายการข้อมูลที่มีอยู่เป็นจำนวนมากได้อย่างรวดเร็วคือดัชนี (Index) ตัวอย่างของคีย์หลัก เช่น รหัสนักศึกษา จะต้องไม่มีนักศึกษาคนใดใช้รหัสซ้ำกัน



ID	Name	Age	GPA	Phone
64567801	นายรักเรียน ใจงาม	18	2.99	0992546854
64567802			3.55	0987895426
64567803	นายวีระ ขยันเรียน	18	3.07	0965864589

ภาพที่ 4.8 แสดงค่าว่าง (NULL)

คีย์หลัก อาจเป็นคีย์ที่ประกอบขึ้นจากหลาย ๆ Attribute มารวมกันทำหน้าที่เป็นคีย์หลักของ Relation โดยทั่ว ๆ ไป เรามักจะกำหนดคีย์หลักมาจากคีย์คู่แข่ง แต่ถ้าใน Relation มีคีย์คู่แข่งมากกว่า 1 คีย์ จะพิจารณาเลือกคีย์คู่แข่งที่มีขนาดเล็กที่สุดมาเป็นคีย์หลักของ Relation



ID	Name	Age	GPA	Phone
64567801	นายรักเรียน ใจงาม	18	2.99	0992546854
64567802	น.ส.สวย อดทน	19	3.55	0987895426
64567803	นายวีระ ขยันเรียน	18	3.07	0965864589

ภาพที่ 4.9 การกำหนด Primary Key

สรุป: คีย์หลัก มีคุณลักษณะดังนี้

1) มีค่าไม่ซ้ำกันในแต่ละแถว
2) ไม่มีค่าที่เป็นค่าว่าง
3) จะถูกเปลี่ยนแปลงค่าไม่ได้
4) แต่ละตารางจะมี 1 คีย์หลัก เท่านั้น
5) สร้างจากฟิลด์เดียว หรือจากหลาย ๆ ฟิลด์ก็ได้

1.3 คีย์ร่วม เป็นการนำเอาหลาย ๆ ฟิลด์มารวมกันเป็นคีย์ เพื่อทำให้มีคุณสมบัติเป็นเอกลักษณ์ คือ ข้อมูลไม่ซ้ำกัน และไม่เป็นข้อมูลว่าง เนื่องจากในบางครั้ง การสร้างคีย์หลักจากคอลัมน์เดียว อาจมีโอกาสที่จะเกิดข้อมูลซ้ำกันได้

ตัวอย่างจากภาพที่ 4.10 Relation “Student” ซึ่งประกอบด้วย ID + Faculty + Major สามารถจัดเป็นคีย์ร่วมได้หลายแบบ เช่น

ID + Name

ID + Faculty

ID + Major

ID + Name + Faculty

ID	Name	Faculty	Major	Phone
64567801	นายรักเรียน ใจงาม	101	EdTech	0992546854
64567802	น.ส.สวย อดทน	101	EdEng	0987895426
64567803	นายวีระ ขยันเรียน	102	EdSci	0965864589

คีย์ร่วม

ภาพที่ 4.10 การกำหนดคีย์ร่วม

1.4 คีย์คู่แข่ง (Candidate Key) เป็นคีย์ที่เล็กที่สุดมีคุณสมบัติใกล้เคียงกับคีย์หลัก สามารถแยกความแตกต่างของข้อมูลแต่ละ Tuple ได้และสามารถใช้แทนคีย์หลักได้ คีย์คู่แข่งบางคีย์อาจประกอบด้วยหนึ่ง Attribute หรือมากกว่าก็ได้ ที่ระบุแต่ละ Tuple ได้อย่างไม่ซ้ำกันในแต่ละ Relation หรือที่มีคุณสมบัติครบถ้วนจนนำมาใช้เป็นคีย์หลัก การกำหนดคีย์คู่แข่งควรให้มีจำนวน Attribute ที่น้อยที่สุดที่ ประกอบกันเป็นคีย์ 1 คีย์ ในแต่ละ Relation สามารถมีคีย์คู่แข่งได้มากกว่า 1 คีย์ คุณลักษณะของคีย์คู่แข่ง

1) เป็นซูเปอร์คีย์ ที่มีคอลัมน์เดียว และไม่มีค่าว่าง

2) เป็นซูเปอร์คีย์ ที่ประกอบไปด้วยหลาย Attribute และมีค่าไม่ซ้ำกันในแต่ละแถว จากตัวอย่างภาพที่ 4.1 Relation “Student” สามารถจัดเป็น คีย์คู่แข่ง ได้ดังนี้ ID ตรงตามคุณลักษณะข้อที่ 1 คือ เป็นซูเปอร์คีย์ ที่มีคอลัมน์เดียว Name + Faculty + Major ตรงตามคุณลักษณะข้อที่ 2 คือไม่มีค่าซ้ำกันในแต่ละแถว Relation “Student” คีย์หลักคือ ID ส่วน Name, Faculty, Major จึงจัดเป็นคีย์คู่แข่ง

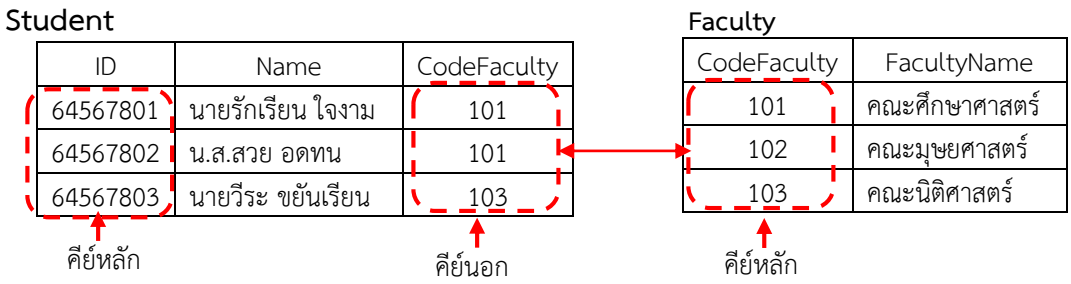
คีย์หลัก

คีย์คู่แข่ง

ID	Name	Faculty	Major	Phone
64567801	นายรักเรียน ใจงาม	101	EdTech	0992546854
64567802	น.ส.สวย อดทน	101	EdEng	0987895426
64567803	นายวีระ ขยันเรียน	102	EdSci	0965864589

ภาพที่ 4.11 การกำหนดคีย์คู่แข่ง

1.5 คีย์นอก (Foreign Key) เป็นคีย์ที่ใช้เชื่อมโยงหรือแสดงความสัมพันธ์ของ Relation ที่เกี่ยวข้องกันอย่างน้อย 2 Relation และใช้ในการอ้างอิงถึง Attribute เดียวกันนี้ในอีก Relation หนึ่ง คีย์นอก สามารถมีค่าซ้ำกันหรือเป็นค่าว่างก็ได้ และไม่จำเป็นต้องมีชื่อเหมือนกับคีย์หลักของอีก Relation หนึ่ง ที่มีความสัมพันธ์กัน



ภาพที่ 4.12 แสดงตัวอย่างคีย์นอก

Relation หรือตารางหนึ่ง ๆ อาจจะมีหรืออาจจะไม่มีคีย์นอกก็ได้ เช่น Relation “Student” ซึ่งกำหนดให้ ID เป็นคีย์หลัก และ Relation “Faculty” ซึ่งกำหนดให้ CodeFaculty เป็นคีย์หลัก ถ้าต้องการทราบชื่อรหัสคณะวิชา และชื่อเต็มคณะวิชา สามารถทำได้กำหนดคีย์นอก (Foreign key) ขึ้นเพื่อเชื่อมความสัมพันธ์ของทั้ง 2 Relation เข้าด้วยกัน โดยการกำหนดให้ฟิลด์ CodeFaculty ใน Relation “Student” เป็นคีย์นอกที่ไปสัมพันธ์กับฟิลด์ CodeFaculty ใน Relation “Faculty” ที่เป็นคีย์หลัก

1.6 คีย์สำรอง หรือดัชนี (Secondary/ Alternate key or index)

ในกรณีที่มีคีย์คู่แข่งหลายคีย์ คีย์สำรองจะหมายถึงคีย์คู่แข่งอื่น ๆ ที่เหลือ ที่ไม่ได้ถูกเลือกให้เป็นคีย์หลักของ Relation ดังนั้น คีย์สำรองจึงเป็นคีย์ที่สามารถใช้แยกแยะข้อมูลได้เช่นเดียวกันกับคีย์หลัก แต่จะไม่ใช้คีย์สำรองในการเชื่อมโยงข้อมูลระหว่างตาราง เช่น Relation Student มีฟิลด์รหัสนักศึกษา เลขบัตรประชาชน ชื่อ นามสกุล คณะวิชา จะเห็นได้ว่าฟิลด์รหัสนักศึกษา และฟิลด์รหัสบัตรประชาชน เป็นฟิลด์ที่มีคุณสมบัติที่มีข้อมูลที่เป็นเอกลักษณ์จะเป็นค่าว่างไม่ได้ ซึ่งฟิลด์เหล่านี้ สามารถถูกเลือกเป็นคีย์หลักได้ หากเลือกให้ฟิลด์รหัสนักศึกษาเป็นคีย์หลัก ฟิลด์รหัสบัตรประชาชนจะกลายเป็นคีย์สำรอง หรือในทางเดียวกันหากเลือกให้ฟิลด์รหัสบัตรประชาชนเป็นคีย์หลัก ฟิลด์รหัสนักศึกษาก็จะกลายเป็นคีย์สำรอง เป็นต้น

ID	IDCard	Name	Faculty	Major	Phone
64567801	1254678952134	นายรักเรียน ใจงาม	101	EdTech	0992546854
64567802	2154685695412	น.ส.สวย อดทน	101	EdEng	0987895426
64567803	3215468512353	นายวีระ ขยันเรียน	102	EdSci	0965864589

คีย์หลัก คีย์สำรอง

ภาพที่ 4.13 แสดงตัวอย่างคีย์สำรอง

คุณสมบัติของ Relation (Properties of relations)

1. องค์ประกอบแบบจำลองฐานข้อมูล ประกอบด้วย 3 ส่วน ดังนี้

1.1 ส่วนโครงสร้างของข้อมูล (Data Structure) เป็นส่วนการจัดเก็บข้อมูลในรูปแบบของตารางที่ประกอบด้วยคอลัมน์และแถว

1.2 ส่วนจัดการข้อมูล (Data Manipulation) เป็นส่วนของคำสั่งที่ใช้จัดการข้อมูลเก็บอยู่ในฐานข้อมูล

1.3 ส่วนควบคุมความคงสภาพของข้อมูล (Data Integrity) เป็นข้อกำหนดหรือกฎเกณฑ์ถูกสร้างขึ้นมา เพื่อใช้ควบคุมความคงสภาพของข้อมูล

2. คุณสมบัติของ Relation (Properties of Relations)

Relation หรือตารางจะอยู่ในลักษณะตาราง 2 มิติ ประกอบด้วย แถวและคอลัมน์ โดยทั่วไป Relation หนึ่ง ๆ จะมีคุณสมบัติต่าง ๆ ดังนี้

2.1 Relation ต้องมีชื่อกำกับ แต่ละชื่อต้องแตกต่างกันจะซ้ำกันไม่ได้

2.2 ค่าของข้อมูลในแต่ละ Attribute จะเป็นค่าประเภทเดียวกันและเป็นค่า ๆ เดียว (Atomic) ทำให้ข้อมูลในตารางมีความเป็นบรรทัดฐาน

2.3 ค่าของข้อมูลที่อยู่ในแต่ละ Attribute จะต้องอยู่ในขอบเขตของโดเมนที่กำหนดไว้สำหรับ Attribute นั้น

2.4 ชื่อของแต่ละ Attribute จะต้องมีชื่อเรียกที่แตกต่างกัน

2.5 การเรียงลำดับของ Attribute และ Tuple ไม่มีความสำคัญ

2.6 ข้อมูลในแต่ละแถวจะไม่ซ้ำกัน (ไม่มี Tuple คู่ใด ๆ ที่ซ้ำกัน)

2.7 ฐานข้อมูลจะมีกี่ Relation ขึ้นอยู่กับความซับซ้อนของฐานข้อมูล

3. รูปแบบของ Relation ประกอบด้วย:

3.1 ชื่อของ Relation

3.2 รายชื่อของ Attribute ของ Relation นั้นจะอยู่ในวงเล็บ โดยใช้เครื่องหมายจุลภาค (,) คั่นระหว่าง Attribute แต่ละตัว ดังนี้ รูปแบบ: ชื่อ Relation (ชื่อ Attribute 1, ชื่อ Attribute 2, ชื่อ Attribute สุดท้าย) ตัวอย่าง: Relation “Student” (ID, Name, Age, GPA, Phone)

ID	Name	Age	GPA	Phone
64567801	นายรักเรียน ใจงาม	18	2.99	0992546854
64567802	น.ส.สวย อดทน	19	3.55	0987895426
64567803	นายวีระ ขยันเรียน	18	3.07	0965864589

ภาพที่ 4.14 แสดงตัวอย่าง Relation student

4. ประเภทของ Relation

4.1 Relation หลัก (Base relation) เป็น Relation ที่ถูกกำหนดขึ้นเพื่อเก็บข้อมูล และเพื่อนำข้อมูลไปใช้เมื่อมีการสร้าง Relation โดยใช้ภาษาในการนิยามข้อมูล เช่น ในภาษา SQL คำสั่ง Create table เป็นการสร้าง Relation หลัก คำสั่ง Alter table เป็นคำสั่งในการแก้ไข Relation หลัก เป็นต้น หลังจากนั้นก็จะทำการเก็บข้อมูลเพื่อการเรียกใช้ข้อมูลในภายหลัง Relation หลักจะเป็นตารางที่ใช้จัดเก็บข้อมูลจริงไว้

4.2 Relation เสมือน (Virtual relation) หรือ วิว (View) เป็นตาราง ที่สร้างขึ้นตามความต้องการใช้ข้อมูลของผู้ใช้แต่ละคน เนื่องจากผู้ใช้แต่ละคนอาจจะต้องการใช้ข้อมูลในลักษณะที่แตกต่างกัน เพื่อความสะดวกในการใช้ข้อมูล และได้ข้อมูลที่ตรงกับความต้องการของผู้ใช้ จึงมีการกำหนด View ของผู้ใช้แต่ละคนขึ้นมาจาก Relation หลัก นอกจากนั้น ยังเป็นการช่วยรักษา

ความปลอดภัยของฐานข้อมูลไว้ด้วย อย่างไรก็ตาม View จะเป็นเพียงตารางสมมุติเท่านั้น จะไม่มีการจัดเก็บข้อมูลจริงลงในระบบฐานข้อมูลแต่อย่างใด

2. กฎ 12 ข้อ ของ E.F Codd (Codd's 12 Rules)

แบบจำลองฐานข้อมูลเชิงสัมพันธ์เสนอ โดย E.F. Codd นักวิทยาศาสตร์คอมพิวเตอร์ชาวอังกฤษ ในปี ค.ศ. 1970 โดยใช้ทฤษฎีของเซทซึ่งเป็นทฤษฎีทางคณิตศาสตร์มาใช้ในการจัดการกับข้อมูล หรือกลุ่มข้อมูลเชิงเลขคณิต (Mathematical CONCEPT OF RELATION SETS) ด้วยการนิยามทฤษฎีต่าง ๆ ให้กับโครงสร้างข้อมูลในรูปแบบของคณิตศาสตร์ เพื่อแก้ไขข้อบกพร่องของฐานข้อมูลแบบเดิมที่ใช้กันอยู่ในขณะนั้น คือแบบจำลองฐานข้อมูลแบบลำดับชั้น และแบบจำลองแบบเครือข่าย โดยที่เซทจะหมายถึงกลุ่มของสิ่งต่าง ๆ เช่น กำหนดให้ A เป็นเซตของสัตว์ปีก, A = นก ไก่ กา ห่าน หงส์ จำนวนของสมาชิก A หรือ n (A) = 5 และกำหนดให้ B เป็นเซตของผลไม้, B = (มะม่วง ทุเรียน มังคุด เงาะ ล้ม ลำใย) จำนวนของสมาชิก B หรือ n (B) = 6

Codd ได้ศึกษาและกำหนดคุณลักษณะที่เป็นบรรทัดฐานของระบบฐานข้อมูลแบบจำลองฐานข้อมูลเชิงสัมพันธ์ และได้สรุปเป็นเกณฑ์หรือกฎ (Rules) ของ แบบจำลองนี้ว่าจะต้องมีคุณสมบัติ 12 ข้อ ที่รู้จักกันดีในนาม กฎ 12 ข้อของ Codd (Codd’s 12 rules) และจากกฎ 12 ข้อนี้ระบบปฏิบัติการฐานข้อมูล จึงได้ถูกพัฒนาขึ้น มาอย่างจริงจัง

กฎ 12 ข้อของ Codd ประกอบกลุ่มของกฎ 13 ข้อ (ข้อที่ 0 ถึง 12) ดังนี้

Rule 0: The foundation rule: กฎพื้นฐาน

“ระบบฐานข้อมูลใด ๆ ที่จะจัดเป็นแบบจำลองฐานข้อมูลเชิงสัมพันธ์ระบบฐานข้อมูลนั้นจะต้องสามารถที่จะจัดการข้อมูลทั้งหมดด้วยความสามารถทาง Relation”

Rule 1: The information rule: กฎสารสนเทศ

“สารสนเทศหรือข้อมูลทั้งหมดในระบบฐานข้อมูล ต้องแสดงอยู่ในรูปแบบเดียวกันเท่านั้น คือ รูปแบบของตาราง และสามารถกำหนดรูปแบบความสัมพันธ์ของข้อมูลได้”

Rule 2:The guaranteed access rule: กฎการรับประกันการเข้าถึงข้อมูล

“ข้อมูลทุกอย่างที่ถูกจัดเก็บไว้ในฐานข้อมูล ต้องสามารถอ้างอิงหรือเข้าถึงได้ด้วยการระบุชื่อตารางค่าของคีย์หลัก (Primary key) และชื่อคอลัมน์ (Column) ที่ต้องการ”

Rule 3:Systematic treatment of null values: กฎการจัดการค่าว่างอย่างเป็นระบบ

“ระบบจัดการฐานข้อมูลจะต้องมีการรองรับการแทนค่าสำหรับข้อมูลบางอย่างที่ยังไม่มีค่าด้วยค่าว่าง เช่น ข้อมูลที่ขาดหายไป หรือไม่มีข้อมูลที่เหมาะสมอย่างเป็นระบบ”

Rule 4: Active online catalog based on the relation model: แอคทีฟออนไลน์แคตตาล็อก ที่อยู่บนพื้นฐานของแบบจำลองฐานข้อมูลเชิงสัมพันธ์

“ระบบจัดการฐานข้อมูลต้องสนับสนุนให้ผู้ใช้สามารถเข้าถึงได้โดยที่มีสิทธิ์ผ่านทางภาษาแบบสอบถาม”

Rule 5: The Comprehensive Data Sublanguage Rule: กฎภาษาข้อมูล

“ระบบแบบจำลองฐานข้อมูลเชิงสัมพันธ์” จะต้องรองรับภาษาที่เป็น Relation อย่างน้อย 1 ภาษาที่มีคุณสมบัติดังต่อไปนี้

1. สามารถนิยามโครงสร้างระบบฐานข้อมูล

2. นิยาม View
3. การจัดการข้อมูล (การใช้คำสั่งแบบโต้ตอบและการเขียนโปรแกรม)
4. ความถูกต้องมั่นคงของข้อมูล และเงื่อนไขอินทิกริต
5. การจัดการสิทธิ์การใช้งาน
6. มีการทำงานในรูปแบบของทรานแซกชัน (Transaction)

Rule 6: The view updating rule: กฎการแก้ไขข้อมูลผ่านทาง View

"ระบบจะต้องรองรับความสามารถในการแก้ไข View ทุก View ที่สามารถเปลี่ยนแปลงแก้ไขข้อมูลได้"

Rule 7: High-level insert, update, and delete: กฎความสามารถในการเพิ่ม การลบ และการแก้ไขข้อมูล

"ระบบจะต้องมีความสามารถในการแทรก แก้ไข และลบข้อมูลในลักษณะของเซตได้ด้วยคำสั่งเดียว"

Rule 8: Physical data independence: กฎความเป็นอิสระของข้อมูลเชิงกายภาพ

"ระบบจะต้องรองรับการจัดการการเปลี่ยนแปลงในระดับโครงสร้าง (Schema) ทางกายภาพได้ โดยที่ไม่มีผลกับผู้ใช้งาน ๆ ไม่จำเป็นจะต้องรับรู้เกี่ยวกับการเปลี่ยนแปลงวิธีการจัดเก็บและการเข้าถึงข้อมูล"

Rule 9: Logical Data Independence: กฎความเป็นอิสระของข้อมูลในเชิงตรรกะ

"ระบบจะต้องรองรับการจัดการการเปลี่ยนแปลงโครงสร้างทางกายภาพโดยไม่มีผลกระทบต่อโครงสร้างของโปรแกรมประยุกต์" (การเปลี่ยนแปลงข้อมูลในระดับตรรกะ รวมถึงการเปลี่ยนแปลงโครงสร้างข้อมูล มีผลต่อคำสั่งและโปรแกรมมากกว่าปลงทางกายภาพ)

Rule 10: Integrity independence: กฎความเป็นอิสระของอินทิกริตี

"เงื่อนไขของอินทิกริที่ต้องมีการกำหนดแยกต่างหากจากโปรแกรมประยุกต์ และจัดเก็บไว้ในแคตตาล็อก ทั้งนี้เพื่อให้เกิดความเป็นไปได้ในการที่จะเปลี่ยนแปลงเงื่อนไขของอินทิกริตีโดยไม่มีผลกระทบต่อโปรแกรมประยุกต์ที่มีอยู่" (การควบคุมข้อมูลให้มีความคงสภาพและเป็นไปตามกฎที่ตั้งไว้ ต้องสามารถระบุได้จากโปรแกรมประยุกต์ และเก็บในรายการซึ่งเงื่อนไขต่าง ๆ สามารถเปลี่ยนแปลงได้ตามความเหมาะสม โดยไม่กระทบต่อโปรแกรมประยุกต์ที่ใช้งานอยู่)

Rule 11: Distribution independence: กฎความเป็นอิสระของการกระจาย

"การกระจายข้อมูลเป็นคุณสมบัติประการหนึ่งของฐานข้อมูลที่ต้องการจะเพิ่มพื้นที่ในการเก็บข้อมูล" ผู้ใช้ไม่ต้องให้ความสนใจว่าข้อมูลจะถูกจัดเก็บอย่างไร หรือกระบวนการทำงานเป็นอย่างไร ระบบฐานข้อมูลจะทำการกระจายผ่านโปรแกรมประยุกต์

Rule 12: The non subversion rule: กฎการไม่มีเวอร์ชันย่อย

กฎความคงสภาพข้อมูล (Data integrity rules)

เนื่องจากโครงสร้างแบบจำลองฐานข้อมูลเชิงสัมพันธ์ จะประกอบด้วยหลาย ๆ Relation ดังนั้น จำเป็นต้องมีการควบคุมข้อมูลให้มีความถูกต้องเป็นจริง และสามารถนำมาใช้เชื่อมโยงกันได้ กฎของความคงสภาพของข้อมูล จะเป็นข้อกำหนดที่ใช้ในการควบคุมความถูกต้องของฐานข้อมูล ทำให้ข้อมูลในระบบฐานข้อมูลมีความถูกต้องอยู่เสมอ การเปลี่ยนแปลงข้อมูลจากผู้ใช้ฐานข้อมูล หรือ

โดยระบบจัดการฐานข้อมูล จะต้องมีผลให้ข้อมูลและความสัมพันธ์ระหว่างข้อมูลถูกต้องตามโครงสร้างฐานข้อมูล

การสูญเสียความคงสภาพของข้อมูล มีหลายสาเหตุด้วยกัน เช่น การเพิ่ม การลบ การแก้ไขข้อมูลใน Relation หนึ่งที่มีความสัมพันธ์กับอีก Relation หนึ่ง ทำให้เกิดความไม่สอดคล้องกันของข้อมูลระหว่าง Relation ที่เกี่ยวข้องกัน การรักษาความถูกต้องของข้อมูลในฐานข้อมูลจะต้องเป็นไปตามกฎเกณฑ์ ได้แก่ กฎเกณฑ์เรื่องความถูกต้องของตัวข้อมูลเอง กฎเกณฑ์เรื่องความสัมพันธ์ระหว่างข้อมูล และกฎเกณฑ์การประมวลผล รายการเปลี่ยนแปลงในฐานข้อมูล ซึ่งจะป้องกันไม่ให้ข้อมูลผิดไปจากความเป็นจริง ความถูกต้องของข้อมูลจะถูกบังคับโดยกฎเกณฑ์ซึ่งมีระบบจัดการฐานข้อมูลคอยดูแลและตรวจสอบ

1. กฎความคงสภาพของ Entity (Entity integrity rule)

กฎความคงสภาพของ Entity มีใจความสำคัญคือ “ค่าของคีย์หลักจะเป็นค่าว่างไม่ได้” และ “คีย์หลักของ Relation จะต้องมีค่าข้อมูลที่ไม่ซ้ำกัน” เนื่องจากคุณสมบัติของคีย์หลักจะเป็น Attribute ที่ใช้ในการเจาะจงแถวข้อมูลใด ๆ ใน Relation ดังนั้น ค่าข้อมูลของคีย์หลักจะเป็นค่าว่างไม่ได้ และคีย์หลักของ Relation จะต้องมีค่าข้อมูลที่ไม่ซ้ำกัน คือจะต้องมีคุณสมบัติที่เป็นเอกลักษณ์สามารถระบุข้อมูล Attribute อื่น ๆ ที่อยู่ใน Tuple เดียวกันได้ เช่น Relation “นักศึกษา” มี Attribute “รหัสนักศึกษา” ทำหน้าที่เป็น คีย์หลัก จากกฎความคงสภาพของข้อมูลข้อนี้ จึงทำให้ Attribute “รหัสนักศึกษา” ไม่สามารถมีค่าว่างได้ และทุก Relation ต้องมี 1 คีย์หลัก

2. กฎความคงสภาพของการอ้างอิง (Referential integrity rule)

กฎข้อนี้จะเกี่ยวข้องกับสร้างความสัมพันธ์ระหว่าง Relation 2 Relation ใน ฐานข้อมูลเชิงสัมพันธ์ ซึ่งสามารถกำหนดด้วยคีย์นอก โดยมีนิยาม ดังนี้ “ค่าของคีย์นอกใน Relation จะต้องมีข้อมูลอยู่ในอีก Relation หนึ่งที่คีย์นอกของ Relation นั้นอ้างอิงถึง” หมายความว่าถ้า Relation ใดมี Attribute ที่เป็นคีย์นอกอยู่ ข้อมูลที่เป็นคีย์นอกนั้นจะต้องมีค่าตรงกับข้อมูลของคีย์หลักของอีก Relation ที่มีความสัมพันธ์กัน หรืออาจให้ข้อมูลที่เป็นคีย์นอกนั้นจะต้องมีค่าเป็นค่าว่าง

ตัวอย่างเช่น ถ้ามีการแก้ไขหรือลบข้อมูลของ Relation ใน Attribute ซึ่งมีความสัมพันธ์อยู่กับ Relation อื่น ๆ จะทำให้ความสัมพันธ์ของข้อมูลเสียหาย การแก้ไขหรือการลบฟิลด์ข้อมูลที่เป็นคีย์หลักใน Relation ที่สัมพันธ์กันจะไปมีผลกระทบถึงคีย์นอกในอีก Relation หนึ่ง จึงต้องเลือกการกระทำเพื่อไม่ให้ความสัมพันธ์ของข้อมูลสูญเสียไป ระบบจัดการฐานข้อมูลจะยอมให้สามารถกระทำได้ใน 3 เงื่อนไข ดังนี้

เงื่อนไขที่ 1 การกระทำแบบมีข้อจำกัด (Restriction) เป็นโอเปอเรเตอร์ ทางคณิตศาสตร์ที่ใช้ในการเลือกข้อมูลจาก Relation หนึ่ง ๆ ที่มีเงื่อนไขตรงตามที่ระบุไว้ เช่น = (เท่ากับ) < (น้อยกว่า) > (มากกว่า) และ <> (ไม่เท่ากับ) เป็นต้น ในกรณีที่มีเงื่อนไขมากกว่า 1 เงื่อนไขอาจใช้คำว่า หรือ (Or) และ (And) ประกอบกันเป็นเงื่อนไขที่ต้องการได้ โดยไม่ยอมให้กระทำการแก้ไข (Update) หรือลบ (Delete) ข้อมูลในฟิลด์ที่เป็นคีย์หลัก นอกจากจะมีเงื่อนไขกำหนดไว้จึงจะกระทำได้ ดังนี้

การแก้ไขข้อมูล (Update)

- ห้ามทำการแก้ไขข้อมูลใน Relation ที่ถูกอ้างถึง เนื่องจากจะทำให้ข้อมูลใน Relation ที่อ้างอิงมา ไม่สามารถอ้างอิงข้อมูลได้

• อนุญาตให้ทำการแก้ไขข้อมูลใน Relation ที่ถูกอ้างอิงถึงได้ แต่จะต้องตามไปแก้ไขข้อมูลใน Relation ที่อ้างอิงมาให้ตรงกับข้อมูลที่แก้ไขใหม่ทั้งหมด

• อนุญาตให้ทำการแก้ไขข้อมูลใน Relation ที่ถูกอ้างอิงถึงได้ โดยการแก้ไขข้อมูลใน Relation ที่อ้างอิงมาให้มีค่าเป็นค่าว่าง การลบข้อมูล (Delete)

• ห้ามทำการลบข้อมูลใน Relation ที่ถูกอ้างถึงนั้น เนื่องจากจะทำให้ข้อมูลใน Relation ที่อ้างอิงมา ไม่สามารถอ้างอิงข้อมูลได้ อนุญาตให้ทำการลบข้อมูลใน Relation ที่ถูกอ้างอิงถึงได้ แต่จะต้องตามไปลบข้อมูลใน Relation ที่อ้างอิงมาทั้งหมด

• อนุญาตให้ทำการลบข้อมูลใน Relation ที่ถูกอ้างอิงถึงได้ โดยการแก้ไขข้อมูลใน Relation ที่อ้างอิงมาให้มีค่าเป็นค่าว่าง

เงื่อนไขที่ 2 การกระทำแบบต่อเนื่อง (Cascade) ยอมให้กระทำการเปลี่ยนแปลงหรือลบข้อมูลในฟิลด์ที่เป็นคีย์หลักได้ แต่ต้องกระทำอย่างต่อเนื่องในอีก Relation หนึ่งที่มีความสัมพันธ์กัน

เงื่อนไขที่ 3 การกระทำโดยใส่ค่าว่าง (Nullify) ยอมให้กระทำการเปลี่ยนแปลงหรือลบข้อมูลในฟิลด์ที่เป็นคีย์หลักได้โดยการกำหนดค่าของคีย์นอกให้เป็นค่าว่าง

วิว (View)

View เป็นตารางที่ถูกสร้างขึ้นโดยการนำข้อมูลของ Relation หลักหรือ View อื่น ๆ ไปดำเนินการเพื่อให้ได้ข้อมูลตามที่ผู้ใช้ต้องการ View เป็น Relation เสมือน (Virtual Relation) มีข้อมูลจัดเก็บอยู่ในฐานข้อมูลเหมือน Relation หลัก ข้อมูลใน View อาจจะไม่คงที่ (Dynamic) หากมีการเปลี่ยนแปลงข้อมูลใน Relation ข้อมูลของ View ที่มาจาก Relation หลักที่มีการเปลี่ยนแปลงก็จะเปลี่ยนแปลงตามไปด้วย

การสร้าง View มีวัตถุประสงค์ ดังนี้คือ

1. การสร้าง View ทำให้มีความเป็นอิสระของข้อมูล (Data independence) หากโครงสร้างของฐานข้อมูลมีการเปลี่ยนแปลง

ตัวอย่างเช่น การสร้าง View ทำให้เกิดความเป็นอิสระของข้อมูลในระดับตรรกะ (Logical data independence) กล่าวคือ เมื่อมีการปรับการออกแบบในระดับแนวคิด (Conceptual level) เช่น หากมีการเพิ่มแอททริบิวต์ใน Relation ผู้ใช้จะไม่เห็นข้อมูลของแอททริบิวต์ดังกล่าว หาก View ของผู้ใช้นั้นไม่ได้รวมแอททริบิวต์ที่เพิ่มรวมอยู่ด้วย

2. เพื่อรักษาความปลอดภัยของฐานข้อมูล โดยผู้ใช้เห็นเพียงข้อมูลใน View ของตนเองเท่านั้น การสร้าง View ทำให้ผู้ใช้สามารถเข้าถึงข้อมูลตามที่ต้องการ ในขณะเดียวกันผู้ใช้คนอื่น ๆ ก็สามารถเข้าถึงฐานข้อมูลเดียวกันนี้ได้ เพียงแต่อาจจะเป็น View ที่ต่างกันเท่านั้น

3. ข้อมูลที่เกิดจากการเชื่อมโยงข้อมูลระหว่าง Relation หรือการใช้ฟังก์ชั่นหรือการจัดกลุ่ม ฯลฯ ซึ่งมีความซับซ้อนในการได้มาของข้อมูล หากมีการสร้าง View ด้วยการเชื่อมโยงข้อมูลระหว่าง Relation ดังกล่าว จะเป็นการง่ายสำหรับผู้ใช้ในการเรียกข้อมูลผ่าน View มาใช้งานง่ายขึ้น

ข้อจำกัดของ View

การสร้าง View เพื่อใช้งานควรจะต้องทำอย่างระมัดระวังรอบคอบ เพราะ View มีข้อจำกัดในหลายประเด็น ดังนี้

1. เมื่อมีการสร้าง View สามารถเรียกใช้ข้อมูล (Select) View ได้ ซึ่งการเรียกใช้ View จะต้องเรียกรายละเอียดของ View จากพจนานุกรมข้อมูล (Data dictionary)

2. มีข้อจำกัดในการใช้ภาษาจัดดำเนินการข้อมูลกับ View (Data manipulation language: DML) View ที่สร้างจาก Relation เดียว (Single relation/Table) จะไม่มีปัญหาในการปรับปรุง (Update) การลบ (Delete) หรือการเพิ่มข้อมูล (Insert) เสมือนหนึ่งเป็น Relation แต่ View ที่ถูกสร้างจากการเชื่อมโยงข้อมูลระหว่าง Relation (Join) การใช้ฟังก์ชั่นทางสถิติหรือ Group by หรือการใช้ Distinct จะมีข้อจำกัด ดังนั้น การปรับปรุง ลบ หรือเพิ่มข้อมูลของ View จะทำได้ภายใต้เงื่อนไขดังนี้

ก. การลบข้อมูล จะทำได้ต่อเมื่อ View นั้น ถูกสร้างภายใต้เงื่อนไขต่อไปนี้

- View ที่สร้างจาก Relation เดียว
- ไม่มีการใช้ Aggregate function เช่น SUM, AVERAGE
- ไม่มีการใช้ Group by
- ไม่มีการใช้ Distinct

ข. การปรับปรุงข้อมูลจะทำได้ต่อเมื่อ View นั้น ถูกสร้างภายใต้เงื่อนไขต่อไปนี้

- View ที่สร้างจาก Relation เดียว
- ไม่มีการใช้ Aggregate function เช่น SUM, AVERAGE
- ไม่มีการใช้ Group by
- ไม่มีการใช้ Distinct
- ไม่มีแอททริบิวต์ใดที่ถูกกำหนดโดยใช้นิพจน์ (Expression)

ค. การเพิ่มข้อมูล จะทำได้ต่อเมื่อ View นั้น ถูกสร้างภายใต้เงื่อนไขต่อไปนี้

- View ที่สร้างจาก Relation เดียว
- ไม่มีการใช้ Aggregate function เช่น SUM, AVERAGE
- ไม่มีการใช้ Group by
- ไม่มีการใช้ Distinct
- ไม่มีแอททริบิวต์ใดที่ถูกกำหนดโดยใช้นิพจน์ (Expression)
- ไม่มีแอททริบิวต์ใดที่เป็นค่าว่าง

ข้อดีและข้อจำกัดของแบบจำลองฐานข้อมูลเชิงสัมพันธ์

ข้อดีของแบบจำลองฐานข้อมูลเชิงสัมพันธ์

1. เป็นรูปแบบจำลองของฐานข้อมูลที่เข้าใจง่าย เพราะเป็นกลุ่มของ Relation หรือตารางที่ข้อมูลได้จัดเก็บเป็นแถวและคอลัมน์ทำให้ผู้ใช้เห็นภาพของข้อมูลได้ง่าย สะดวกต่อการใช้งานและการเขียนโปรแกรม

2. มีเครื่องมือที่ช่วยให้ผู้ใช้สามารถจัดการกับข้อมูลได้ด้วยคำสั่งง่ายๆ โดยผู้ใช้ไม่ต้องทราบรายละเอียดของการจัดเก็บข้อมูลอย่างไร

3. สามารถใช้ภาษาที่ง่ายในการเรียกดูข้อมูล เช่น ภาษา SQL เป็นภาษาที่มีลักษณะคล้ายภาษาอังกฤษ ไม่จำเป็นต้องเขียนเป็นลำดับขั้นตอนของคำสั่งอย่างในภาษาอื่น

4. การเรียกใช้หรือเชื่อมโยงข้อมูลทำได้ง่าย โดยการใช้โอเปอเรเตอร์ทางคณิตศาสตร์ ไม่จำเป็นต้องใช้พอยท์เตอร์

5. มีระบบความปลอดภัยที่ดี เนื่องจากโครงสร้างนี้ผู้ใช้งานจะไม่ทราบถึงกระบวนการจัดเก็บข้อมูลภายในฐานข้อมูลแท้จริงว่าเป็นอย่างไร

6. ระบบการจัดการฐานข้อมูลที่พัฒนาในปัจจุบันล้วนรองรับเทคโนโลยีฐานข้อมูลแบบเชิงสัมพันธ์

ข้อจำกัดของแบบจำลองฐานข้อมูลเชิงสัมพันธ์

1. มีการลงทุนสูงเนื่องจากต้องใช้ฮาร์ดแวร์และซอฟท์แวร์และที่มีความสามารถสูง

2. แนวคิดฐานข้อมูลเชิงสัมพันธ์ในภาพรวมนั้นง่ายต่อการนำไปใช้งาน ดังนั้น บุคลากรที่ไม่ได้รับการฝึกอบรมหรือผู้ที่มีความรู้ไม่ดีพอได้นำเครื่องมือไปใช้งานที่ผิด ทำให้ระบบที่ดีต้องแย่ลง และหากไม่ได้รับการตรวจสอบ อาจทำให้เกิดข้อมูลซ้ำซ้อนได้เช่นเดียวกับระบบแฟ้ม

สรุป

แบบจำลองฐานข้อมูลแบบลำดับชั้น และแบบจำลองฐานข้อมูลแบบเครือข่าย ต่างก็มีข้อจำกัดในการจัดการข้อมูลภายใน และการจัดการข้อมูลในฐานข้อมูลมีความยุ่งยากและซับซ้อนมาก จากข้อจำกัดของแบบจำลองฐานข้อมูลทั้ง 2 แบบดังกล่าว จึงนำไปสู่การพัฒนาระบบฐานข้อมูลแบบจำลองฐานข้อมูลเชิงสัมพันธ์ขึ้นมา แบบจำลองฐานข้อมูลเชิงสัมพันธ์ เป็นแบบจำลองที่ใช้พื้นฐานทฤษฎี คณิตศาสตร์เกี่ยวกับเซ็ทมากำหนดส่วนประกอบของแบบจำลองฐานข้อมูล ได้แก่ ส่วนของโครงสร้างข้อมูล ส่วนควบคุมความถูกต้องของข้อมูล และส่วนจัดการข้อมูล แบบจำลองฐานข้อมูลนี้ได้ออกแบบมาเพื่อลดความซ้ำซ้อนของการเก็บข้อมูล การนำเสนอมุมมองของข้อมูลในลักษณะตาราง 2 มิติ คือแถวและคอลัมน์ และมีความอิสระในโครงสร้าง ทำให้เห็นภาพข้อมูลได้ชัดเจน ง่ายต่อการเข้าใจและการประยุกต์ใช้ รวมทั้งสามารถเรียกใช้ข้อมูลได้อย่างมีประสิทธิภาพ กฎเกณฑ์ที่ใช้ในฐานข้อมูลเชิงสัมพันธ์ คือ ทุก Relation จะต้องมีแอททริบิวต์ที่เป็นคีย์หลัก ซึ่งมีคุณสมบัติเป็นค่าไม่ซ้ำ และต้องไม่เป็นค่าว่าง นอกจากนี้ การเชื่อมโยงระหว่าง Relation จะใช้ Attribute ที่เป็นคีย์นอกเชื่อมโยงกับ Attribute ที่เป็นคีย์หลักของอีกตารางหนึ่ง เพื่อเรียกข้อมูลที่ต้องการได้ และกลุ่มคำสั่งที่ใช้ในการจัดการข้อมูลเป็นภาษา SQL (Structured query language) ซึ่งมีประสิทธิภาพสูง เข้าใจง่าย ทั้งยังตอบสนองความต้องการของผู้ใช้ในเรื่องการจัดการฐานข้อมูลของระบบได้อย่างมีประสิทธิภาพ ทำให้เป็นที่นิยมใช้กันมากในปัจจุบัน

View เป็นตารางที่สร้างขึ้นตามความต้องการใช้งาน ไม่มีข้อมูลจริง ประโยชน์ของการสร้าง View เพื่อให้ผู้ใช้ในระบบใช้งานข้อมูลแทนการเรียกจากข้อมูลจาก Relation ทำให้ฐานข้อมูลมีความปลอดภัย และความเป็นอิสระของข้อมูล

บทที่ 5

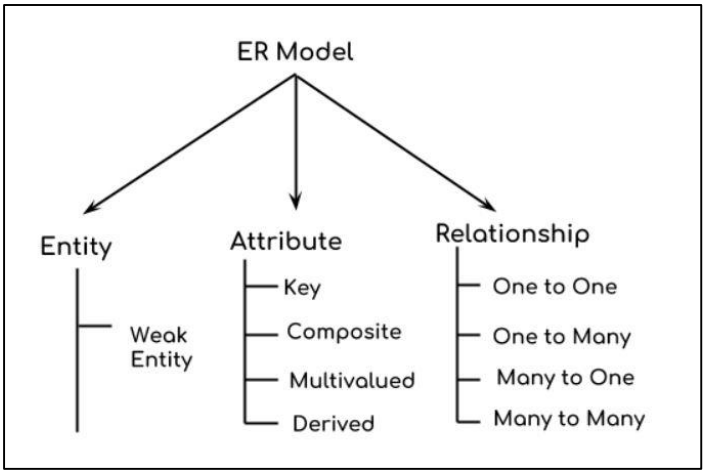
แบบจำลอง อี-อาร์ (Entity relationship model)

แบบจำลอง E-R มีความสำคัญในการเป็นสื่อกลางเพื่อสื่อสารกับบุคลากรต่าง ๆ ที่เกี่ยวข้องกับระบบฐานข้อมูล ไม่ว่าจะเป็นในระดับผู้บริหาร นักเขียนโปรแกรม และผู้ใช้ในระดับปฏิบัติการ ทำให้เข้าใจระบบได้อย่างถูกต้องตรงกัน เนื่องจากมีการแสดงภาพรวมของระบบในลักษณะของรูปภาพหรือแบบจำลอง ทำให้เข้าใจง่าย ดังนั้น ระบบที่ออกแบบมาจึงมีความถูกต้องและเป็นไปตามวัตถุประสงค์ขององค์กร จากเนื้อหาในบทที่ผ่านได้กล่าวถึงแบบจำลองข้อมูลเชิงสัมพันธ์ ซึ่งเครื่องมือสำคัญที่นำมาใช้ออกแบบฐานข้อมูลเชิงแนวคิดภายใต้ระบบฐานข้อมูลเชิงสัมพันธ์นั่นคือ แบบจำลอง E-R ดังนั้น ในในบทนี้ จะกล่าวถึงแบบจำลอง อี-อาร์ (Entity relationship model) โดยนำเสนอเนื้อหาสำหรับการศึกษาดังต่อไปนี้

- ♦ องค์ประกอบของแบบจำลอง E-R
- ♦ สัญลักษณ์ที่ใช้แบบจำลอง E-R
- ♦ ความสัมพันธ์ระหว่าง Entity (Entity relationship)
- ♦ ขั้นตอนการออกแบบฐานข้อมูลโดยใช้ แบบจำลอง E-R
- ♦ นอร์มอลฟอร์ม (Normal forms)

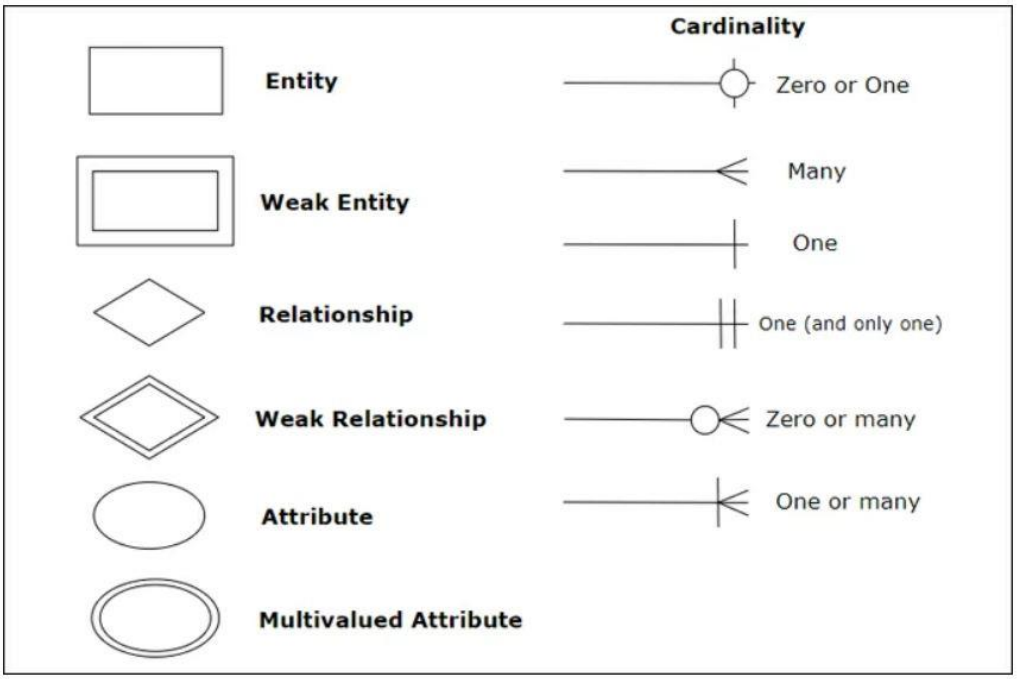
องค์ประกอบของแบบจำลอง E-R

แบบจำลอง E-R ประกอบด้วยส่วนประกอบที่สำคัญ คือ Entity, Attribute, และความสัมพันธ์ (Relationship) ดังภาพที่ 5.1



ภาพที่ 5.1 องค์ประกอบของแบบจำลอง E-R

แบบจำลอง E-R มีการใช้สัญลักษณ์แทนความหมายของ Entity, Attribute, และความสัมพันธ์ (Relationship) ดังภาพที่ 5.2



ภาพที่ 5.2 สัญลักษณ์แทนความหมายแบบจำลอง E-R

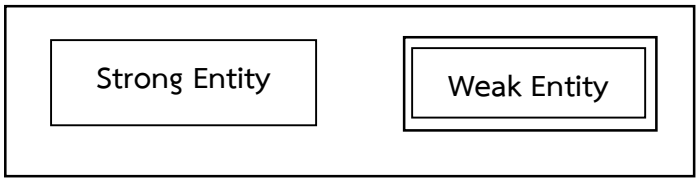
1. Entity

Entity หมายถึงสิ่งที่มีอยู่จริง จับต้องได้ หรืออาจจะเป็นจินตภาพที่แสดงความเป็นหนึ่งเดียว ซึ่งเมื่อกล่าวถึงแล้วทุกคนเข้าใจตรงกัน แต่โดยทั่วไปแล้ว Entity มักจะอยู่ในรูปของนามซึ่งอาจเป็นสิ่งที่เป็นรูปธรรมคือสามารถมองเห็นได้ด้วยตา หรืออยู่ในรูปของนามธรรมคือไม่สามารถมองเห็นได้ด้วยตาก็ได้ เช่น บุคคล (นักศึกษา อาจารย์ ลูกค้า พนักงาน คนไข้) สถานที่ (โรงเรียน ห้องเก็บสินค้า คลังสินค้า ร้านค้า) วัตถุ (หนังสือ เครื่องจักรกล เครื่องยนต์ สินค้า วัตถุดิบ) เหตุการณ์ (การยืม การคืน ลงทะเบียน การฝาก-การถอนเงิน) หรือ แนวคิด (วัน วิชา บัญชี หลักสูตร ความชำนาญ) ที่ทำให้เกิดกลุ่มของข้อมูลที่ต้องการจัดเก็บรวมทั้งสามารถบ่งชี้ถึงความเป็นเอกลักษณ์เฉพาะตัวได้ Entity มีคุณสมบัติได้หลายอย่าง และ Entity เป็นสิ่งที่ผู้ใช้งานฐานข้อมูลจะต้องเกี่ยวข้องด้วย เมื่อมีการออกแบบระบบฐานข้อมูลสัญลักษณ์รูปสี่เหลี่ยมผืนผ้าแทน Entity 1 Entity และมีชื่อ Entity กำกับอยู่ภายในชื่อที่ใช้จะต้องเป็นคำนาม และโดยปกติมักจะเขียนตัวพิมพ์ใหญ่

Entity สามารถแบ่งออกได้หลายประเภทด้วยกัน เช่น

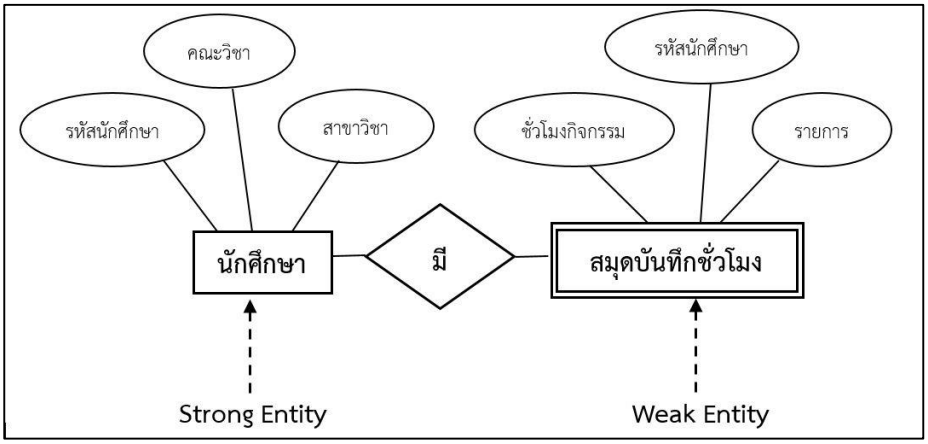
1.1 Entity ปกติ (Strong entity หรือ Regular entity) เป็น Entity ที่มีคุณสมบัติเฉพาะ (Identity) ในตัวเอง สามารถเกิดขึ้นได้ด้วยตัวเอง การคงอยู่ของ Entity จะไม่ขึ้นกับ Entity อื่น เช่น นักศึกษา หรืออาจารย์ หรือสินค้า สัญลักษณ์ที่ใช้ คือ รูปสี่เหลี่ยมผืนผ้า

1.2 Entity อ่อนแอ เป็น Entity ที่ไม่สามารถเกิดขึ้นได้ตามลำพัง จะขึ้นอยู่กับ Entity อื่น จะมีคีย์หลักจากการสืบทอด Entity ที่อ้างอิงอยู่มา ใช้เป็นคีย์หลักหรือส่วนหนึ่งของคีย์หลัก และจะถูกลบออกไปด้วยเมื่อ Entity หลักถูกลบ สัญลักษณ์ที่ใช้ คือ รูปสี่เหลี่ยมผืนผ้าเส้นคู่



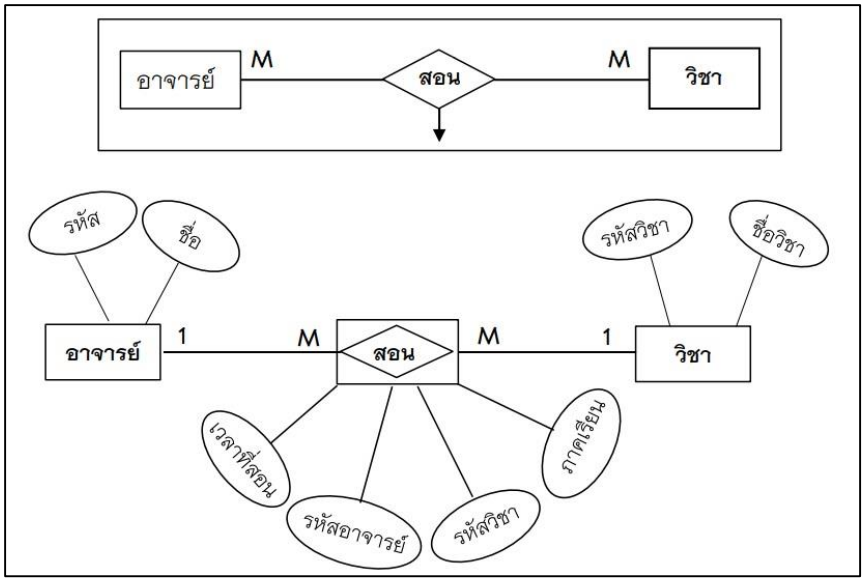
ภาพที่ 5.3 สัญลักษณ์ Entity ปกติ และ Entity อ่อนแอ

Entity ทั้ง 2 ประเภทอธิบายได้ดังนี้ เช่น ในมหาวิทยาลัยแต่ละแห่งหนึ่ง นักศึกษาแต่ละคนจะมีรหัสนักศึกษา ชื่อ-สกุล หมายเลขโทรศัพท์ที่ไม่ซ้ำกัน ดังนั้น Entity นักศึกษาจะจัดเป็น Entity ปกติ และนักศึกษาในมหาวิทยาลัยแต่ละคนจะมีสมุดบันทึกชั่วโมงกิจกรรม นักศึกษาแต่ละคนอาจจะมีรายการกิจกรรม หรือมีชั่วโมงกิจกรรมที่เหมือนกัน หรือไม่เหมือนกันก็ได้ ดังนั้น ถ้าไม่มี Entity นักศึกษาก็จะไม่สามารถทราบว่านักศึกษาคนใดทำกิจกรรมอะไร จำนวนชั่วโมงกิจกรรมเป็นเท่าใด ดังนั้น Entity สมุดบันทึกชั่วโมงกิจกรรมจึงจัดเป็น Entity อ่อนแอ เพราะ Entity นี้จะคงอยู่ได้ต้องอาศัย Entity นักศึกษา



ภาพที่ 5.4 ตัวอย่าง Entity ปกติ และ Entity อ่อนแอ

1.3 คอมโพสิต Entity หรือ Entity เชิงความสัมพันธ์ (Composite/ Associate entity) เป็น Entity ที่สร้างขึ้นมาแปลงความสัมพันธ์ของ Entity 2 Entity ที่มีความสัมพันธ์แบบ M: N ให้เป็นแบบ 1: M เพื่อให้เข้าใจได้ง่าย โดยการนำเอาคีย์หลักของ Entity ทั้ง 2 มารวมกันกับ Attribute อื่น ๆ ที่สนใจ สัญลักษณ์ที่ใช้คือสี่เหลี่ยมผืนผ้าที่ภายในมีรูปสี่เหลี่ยมขนมเปียกปูน เช่น อาจารย์ 1 คนสอนได้หลายวิชา และวิชา 1 รายวิชา มีอาจารย์สอนได้หลายคน



ภาพที่ 5.5 ตัวอย่าง คอมโพสิต Entity

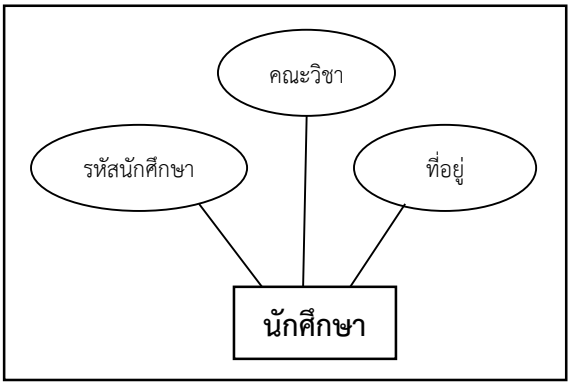
2. แอททริบิวท์ (Attribute) หรือ พรอพเพอตี (Property)

Entity หนึ่ง ๆ จะประกอบไปด้วยคุณสมบัติหรือลักษณะที่แตกต่างกันออกไป ข้อมูลที่แสดงถึงคุณลักษณะของ Entity เรียกว่า Attribute ดังนั้น ความหมายของ Attribute คือคุณสมบัติของ Entity หรือข้อมูลของ Entity ที่เราสนใจจะเก็บหรืออาจจะหมายถึงกลุ่มของค่าความจริงใด ๆ ที่เป็นรายละเอียดของ Entity ซึ่งแสดงลักษณะและคุณสมบัติของ Entity แต่ละ Entity สามารถ

มีคุณสมบัติต่าง ๆ ได้มากมาย ขึ้นอยู่กับว่าต้องการจัดเก็บข้อมูลอะไรบ้าง เช่น คุณสมบัติหรือข้อมูลของนักศึกษา ประกอบไปด้วย ชื่อ ที่อยู่ หมายเลขโทรศัพท์ รหัสนักศึกษา คณะวิชา สาขาวิชา วัน เดือนปีเกิด เพศ ส่วนสูง น้ำหนัก ฯลฯ Attribute สามารถแบ่งออกได้หลายประเภท เช่น

2.1 Simple attribute เป็น Attribute แบบง่าย ๆ มีเพียงองค์ประกอบเดียว หรือข้อมูลเดียว ภายใน Attribute ไม่สามารถแบ่งย่อยได้อีก เช่น ข้อมูล เพศ อายุ ส่วนสูง น้ำหนัก รหัสนักศึกษา คณะวิชา สาขาวิชา เป็นต้น จะประกอบด้วยค่าข้อมูลเดียว สัญลักษณ์ที่ใช้แทน Attribute ชนิดนี้ คือ วงรีที่มีเส้นเชื่อมต่อไปยัง Entity ของ Attribute นั้น ๆ โดยมีชื่อ Attribute อยู่ภายในวงรี

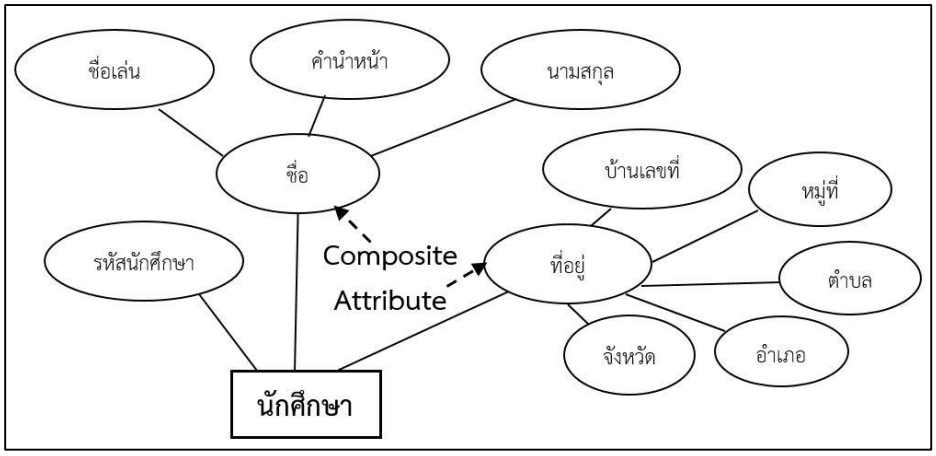
ตัวอย่าง Attribute “รหัสนักศึกษา” “คณะวิชา” “ที่อยู่” ของ Entity “นักศึกษา”



ภาพที่ 5.6 ตัวอย่าง Simple Attribute

2.2 Composite Attribute หมายถึง Attribute ที่มีองค์ประกอบหรือข้อมูล ซึ่งจะตรงกันข้ามกับ Simple attribute ค่าภายใน Attribute สามารถแบ่งย่อยออกมาได้อีก แต่มีอิสระต่อกัน เช่น

- Attribute “ที่อยู่” จะประกอบด้วย บ้านเลขที่ หมู่ที่ ตำบล อำเภอ และจังหวัด
- Attribute “ชื่อ” จะประกอบด้วย คำนำหน้าชื่อ ชื่อเล่นและนามสกุล



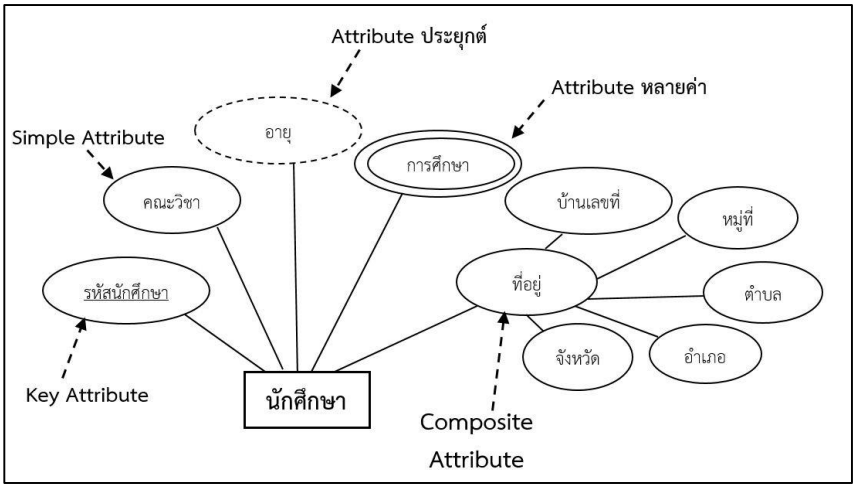
ภาพที่ 5.7 ตัวอย่าง Composite attribute

2.3 Attribute ค่าเดียว (Single-value attribute) คือ Attribute ที่มีค่าข้อมูลที่เก็บเพียงค่าเดียว เช่น รหัสนักศึกษา ชื่อ ภาควิชา ใน Entity “นักศึกษา” ซึ่งนักศึกษาแต่ละคนก็จะมีค่าข้อมูลเหล่านี้ เพียงค่าเดียวเท่านั้น หรือ Entity “เพศ” ก็จะมีค่าข้อมูลที่เก็บเพียง 1 ค่า คือ หญิง หรือชายเท่านั้น

2.4 Attribute หลายค่า (Multivalued Attribute) คือ Attribute ที่สามารถมีได้หลายค่า เช่น คน 1 คนสามารถมี “วุฒิการศึกษา” ได้หลายระดับ เช่น ระดับปริญญาตรี ระดับปริญญาโท และระดับปริญญาเอก หรือคน 1 คนอาจจะมี “หมายเลขโทรศัพท์” ได้หลายหมายเลข หรือข้อมูล “โรงเรียนที่เรียนจบ” ตั้งแต่ระดับ อนุบาล ประถมศึกษา มัธยมศึกษา และอุดมศึกษา อาจจะมาจากหลายโรงเรียน เป็นต้น สัญลักษณ์ของ Attribute หลายค่า คือ รูปวงรี 2 วงซ้อนกัน

2.5 Attribute ประยุกต์ (Derived attribute) หมายถึง Attribute ที่ไม่ได้เก็บอยู่ในฐานข้อมูล แต่ได้จากการประยุกต์ใช้ Attribute อื่น เช่น Attribute รายได้รวมของอาจารย์ ได้มาจากการรวมเงินเดือน ค่าสอนพิเศษ ของแต่ละเดือน หรือ Attribute อายุ ได้จากการคำนวณจากวันเดือนปีเกิด สัญลักษณ์ของ Attribute ประยุกต์เป็นรูปวงรีแบบเส้นประ

2.6 คีย์ Attribute (Key attributes) หรือ คีย์หลัก หรือ กุญแจหลัก เป็น Attribute ที่บอกเอกลักษณ์ของ Entity นั้น ๆ ได้ จะมีค่าของข้อมูลที่ไม่ซ้ำกัน เช่น รหัสนักศึกษา จัดเป็นคีย์ Attribute ใช้ระบุความแตกต่างของแต่ละคนใน Entity สัญลักษณ์ของคีย์หลักคือรูปวงรีภายในมีชื่อ Attribute ที่มีการขีดเส้นใต้แทนคีย์หลัก



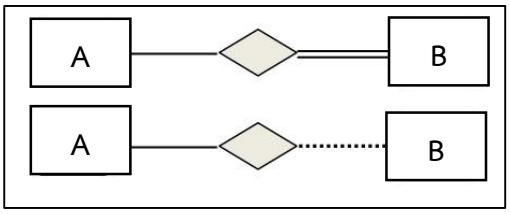
ภาพที่ 5.8 แสดง Attribute ชนิดต่าง ๆ

ความสัมพันธ์ระหว่าง Entity (Entity relationship)

ความสัมพันธ์ หรือ Relationship จะหมายถึง ความสัมพันธ์ระหว่าง Entity หรือความสัมพันธ์ระหว่างข้อมูล เป็นลักษณะของความสัมพันธ์ระหว่าง Entity หนึ่งกับตัวของมันเองหรือกับ Entity อื่น (มากกว่า 2 Entity ก็ได้) ว่ามีความสัมพันธ์ของข้อมูลเป็นแบบใด ความสัมพันธ์ระหว่าง Entity ในแบบจำลอง E-R จะแทนด้วยรูปสี่เหลี่ยมข้าวหลามตัดที่มีชื่อของความสัมพันธ์อยู่ภายในและมีเส้นเชื่อมระหว่าง Entity และความสัมพันธ์ระหว่าง Entity

ความสัมพันธ์ระหว่างข้อมูลที่เราสนใจ ซึ่งพิจารณาได้ 2 แบบ คือ ความสัมพันธ์แบบจำเป็นหรือแบบบังคับ (Mandatory) หมายถึงความสัมพันธ์ที่จะต้องมี หรือจะต้องเกิดขึ้นอย่างแน่นอน

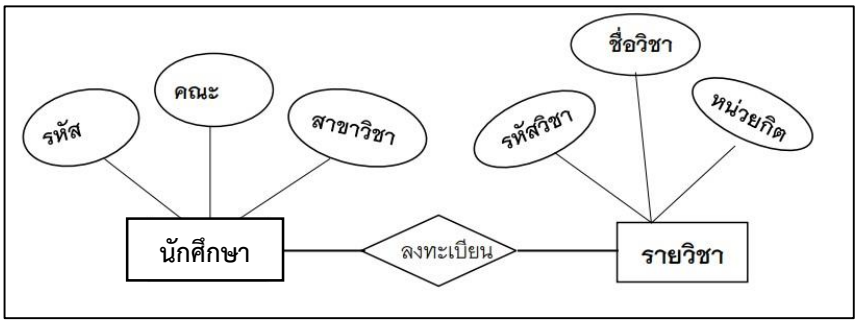
นิยมแทนด้วยสัญลักษณ์เส้นตรงทึบ หรือเส้นตรง 2 เส้น และความสัมพันธ์แบบไม่จำเป็น (Optional) เป็นความสัมพันธ์ที่อาจจะมีหรือไม่มีก็ได้ นิยมแทนด้วยสัญลักษณ์เส้นตรงประหรือเส้นตรง 1 เส้น



ภาพที่ 5.9 แสดงความสัมพันธ์ระหว่าง Entity หรือระหว่างข้อมูล

จากภาพที่ 5.9 อธิบายได้ ดังนี้: ถ้าข้อมูล A มีความสัมพันธ์กับข้อมูล B อย่างแน่นอน หรืออย่างน้อยจะต้องมีสมาชิก (Element) ของข้อมูล A หนึ่งสมาชิกที่มีความสัมพันธ์กับสมาชิกของข้อมูล B และแทนความสัมพันธ์ด้วยเส้นตรงทึบออกจากข้อมูล A ไปยังข้อมูล B แต่ในขณะเดียวกันข้อมูล B อาจจะมีความสัมพันธ์กับข้อมูล A หรือไม่มีก็ได้ เส้นตรงที่ออกจากข้อมูล B จึงแทนด้วยเส้นประ

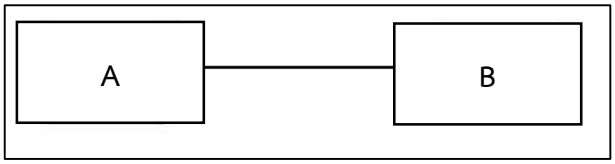
ถ้ากำหนดให้ A และ B เป็นข้อมูล สัญญลักษณ์ที่ใช้แทนชนิดความสัมพันธ์ระหว่างข้อมูล A กับข้อมูล B คือ เส้นตรงเส้นเดียวหรือจะใช้ลูกศร 1 หัว กับ ลูกศร 2 หัวก็ได้



ภาพที่ 5.10 แสดงตัวอย่างความสัมพันธ์ของข้อมูล

ชนิดความสัมพันธ์ระหว่างข้อมูล

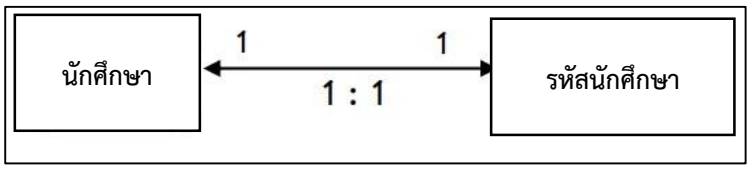
1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One to ONE RELATIONSHIPS:1: 1) เป็นการแสดงถึงความสัมพันธ์ของข้อมูลของ Entity หนึ่งว่ามีความสัมพันธ์กับข้อมูลในอีก Entity หนึ่งในลักษณะหนึ่งต่อหนึ่ง หมายถึงข้อมูลหนึ่งข้อมูลของ Entity ที่ 1 มีความสัมพันธ์กับข้อมูลใน Entity ที่ 2 เพียงข้อมูลเดียวเท่านั้น ซึ่งใช้ตัวเลขกำกับที่เส้นเพื่อระบุความสัมพันธ์ เช่น ข้อมูล A จะเกี่ยวข้องสัมพันธ์กับข้อมูล B ได้เพียง 1 ค่า เท่านั้น และข้อมูล B จะเกี่ยวข้องกับข้อมูล A ได้เพียง 1 ค่า เช่นกัน



ภาพที่ 5.11 แสดงความสัมพันธ์หนึ่งต่อหนึ่ง หรือแบบ 1:1

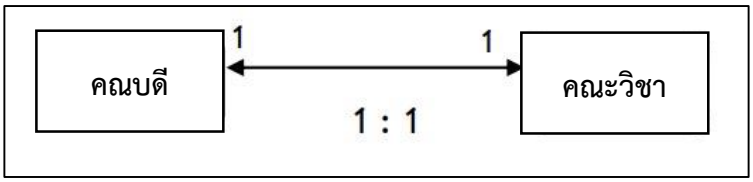
ตัวอย่างที่ 1 เช่น มี Entity 2 Entity คือ Entity “อาจารย์” และ Entity “คณะวิชา” มีความสัมพันธ์กันชื่อ “บริหาร” แบบ 1: 1 ซึ่งหมายถึง อาจารย์ 1 คน จะสามารถบริหาร หรือเป็นคณบดีได้พียง 1 คณะวิชาเท่านั้น และในขณะเดียวกันแต่ละคณะวิชาก็จะมีคณบดีได้เพียง 1 คน เช่นกัน หรือความสัมพันธ์ระหว่างนักศึกษากับรหัสนักศึกษา จะเป็นแบบ 1: 1 เพราะนักศึกษา 1 คน จะมีรหัสนักศึกษาได้เพียง 1 รหัสเท่านั้น และในขณะเดียวกันรหัสนักศึกษา 1 รหัส จะใช้กับนักศึกษาได้เพียง 1 คน เช่นกัน

♦ ความสัมพันธ์ระหว่างข้อมูลนักศึกษากับรหัสนักศึกษา: นักศึกษา 1 คน จะมีรหัสนักศึกษาได้เพียง 1 รหัสเท่านั้น ในขณะเดียวกันรหัสนักศึกษา 1 รหัส จะใช้กับนักศึกษาได้เพียง 1 คน เช่นกัน



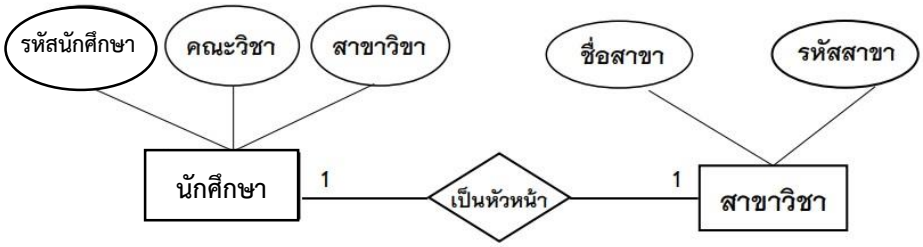
ภาพที่ 5.12 ความสัมพันธ์ระหว่างข้อมูลนักศึกษากับรหัสนักศึกษา แบบ 1:1

♦ ความสัมพันธ์ระหว่างคณบดีกับคณะวิชา: คณบดี 1 คน จะบริหารคณะวิชาได้เพียง 1 คณะวิชาเท่านั้น ในขณะเดียวกัน 1 คณะวิชาจะมีคณบดีบริหารได้เพียง 1 คน เช่นกัน



ภาพที่ 5.13 ความสัมพันธ์ระหว่างคณบดีกับคณะวิชา แบบ 1:1

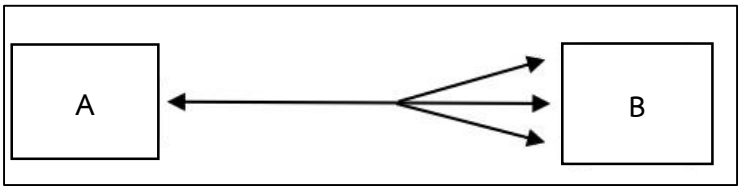
♦ ความสัมพันธ์ของ Entity “นักศึกษา” และ Entity “สาขาวิชา” ซึ่งกำหนดให้แต่ละสาขาวิชามีนักศึกษาเป็นหัวหน้าห้องได้เพียงคนเดียวเท่านั้น



ภาพที่ 5.14 แสดงตัวอย่างความสัมพันธ์แบบหนึ่งต่อหนึ่ง หรือ 1: 1

2. ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One to many relationships: 1: N หรือ 1: M) เป็นความสัมพันธ์ระหว่าง Entity หนึ่ง ไปมีความสัมพันธ์กับอีก Entity หนึ่ง ได้มากกว่า 1 รายการในลักษณะหนึ่งต่อกลุ่ม (ข้อมูลหนึ่งข้อมูลของ Entity ที่ 1 มีความสัมพันธ์กับข้อมูลใน Entity ที่ 2 หลายข้อมูล) เช่น ข้อมูล “A” มีความสัมพันธ์กับข้อมูล “B” ได้มากกว่า 1 รายการ

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

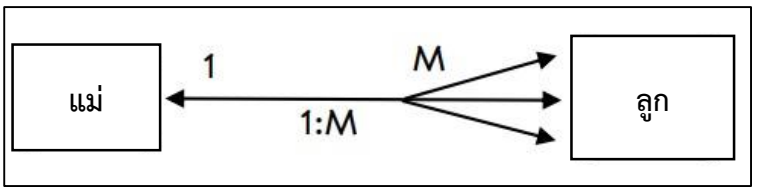


ภาพที่ 5.15 แสดงความสัมพันธ์แบบหนึ่งต่อกลุ่ม หรือ 1: M

ตัวอย่างที่ 2 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม

♦ ความสัมพันธ์ระหว่างแม่กับลูก

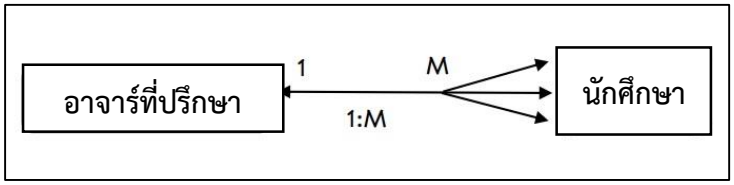
แม่ 1 คนอาจจะมีลูกได้หลายคน แต่ลูก 1 คนจะมีแม่ได้เพียงคนเดียวเท่านั้น



ภาพที่ 5.16 แสดงความสัมพันธ์ระหว่างแม่กับลูกแบบ 1: M

♦ ความสัมพันธ์ระหว่างอาจารย์ที่ปรึกษากับนักศึกษา

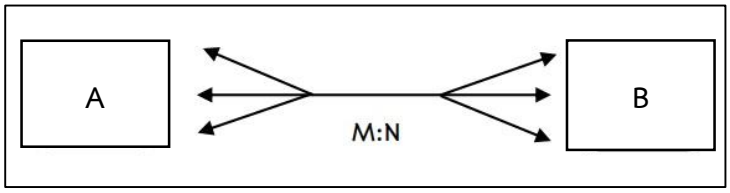
อาจารย์ที่ปรึกษา 1 คนจะมีนักศึกษาได้หลายคน แต่นักศึกษาคน 1 จะมีอาจารย์ที่ปรึกษาได้เพียง 1 คน



ภาพที่ 5.17 แสดงความสัมพันธ์ระหว่างอาจารย์ที่ปรึกษากับนักศึกษา

3. ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many to many relationships, M: N) เป็นความสัมพันธ์แบบหลายรายการระหว่าง 2 Entity เป็นความสัมพันธ์ของข้อมูลของ Entity หนึ่งมีความสัมพันธ์กับข้อมูลในอีก Entity หนึ่ง ในลักษณะกลุ่มต่อกลุ่ม เช่น ข้อมูล “A” มีความสัมพันธ์กับอีกข้อมูล “B” ได้มากกว่า 1 รายการ ในขณะเดียวกันแต่ละรายการของข้อมูล “B” ก็มีความสัมพันธ์กับข้อมูล “A” ได้มากกว่า 1 รายการ เช่นเดียวกัน ตัวอย่างเช่น นักศึกษา 1 คน สามารถลงทะเบียนเรียนได้มากกว่า 1 รายวิชา และ 1 รายวิชาจะมีนักศึกษาลงทะเบียนเรียนได้มากกว่า 1 คน

ความสัมพันธ์แบบกลุ่มต่อกลุ่มนี้ ข้อมูล A จะเกี่ยวข้องกับข้อมูล B ได้หลายค่า และในทำนองเดียวกันข้อมูล B เกี่ยวข้องกับข้อมูล A ได้มากกว่า 1 ค่า เช่นกัน ดังตัวอย่าง

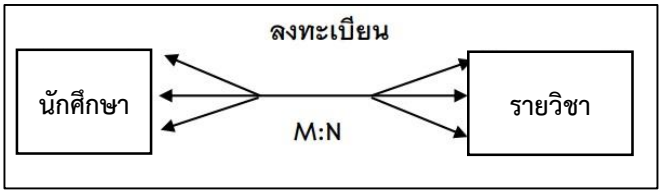


ภาพที่ 5.18 แสดงความสัมพันธ์แบบกลุ่มต่อกลุ่ม หรือ แบบ M:N

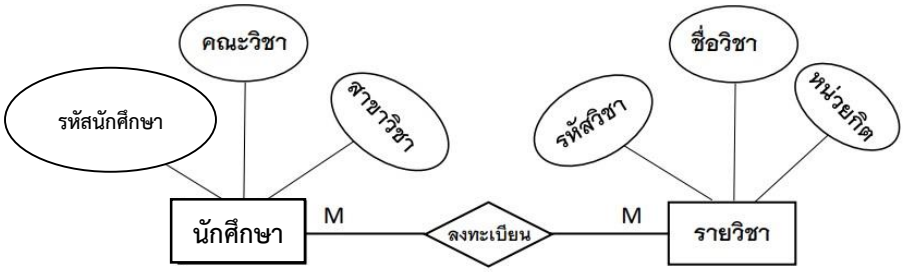
ตัวอย่างที่ 3 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม

♦ ความสัมพันธ์ระหว่างนักศึกษากับรายวิชาที่เปิดสอน

นักศึกษา 1 คน ลงทะเบียนเรียนได้หลายวิชาและวิชาหนึ่ง ๆ จะมีนักศึกษาลงทะเบียนได้มากกว่า 1 คน



ภาพที่ 5.19 ความสัมพันธ์ระหว่างนักศึกษากับรายวิชาที่เปิดสอน



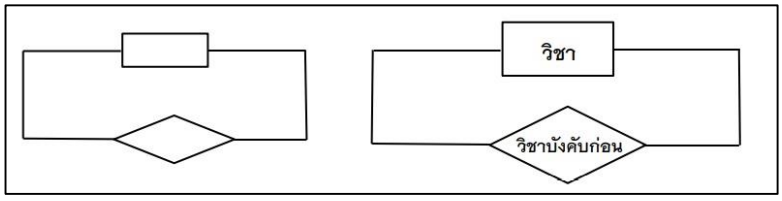
ภาพที่ 5.20 แสดงความสัมพันธ์แบบกลุ่มต่อกลุ่ม หรือ M: N

3. ดีกรีของความสัมพันธ์ (Degree of relationship)

ดีกรีความสัมพันธ์ หรือระดับชั้นของความสัมพันธ์ หมายถึงจำนวน Entity ที่มีส่วนร่วมกับความสัมพันธ์นั้น ๆ ดีกรีความสัมพันธ์แบ่งออกได้เป็น 3 ประเภท คือ

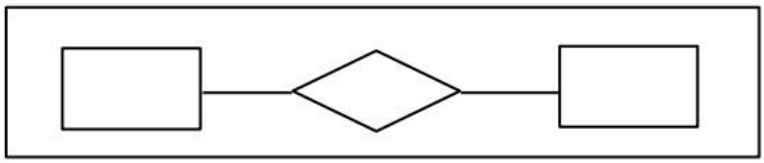
3.1 ความสัมพันธ์แบบยูนารี (Unary relationship) หรือความสัมพันธ์แบบรีเคอซีพ (Recursive relationship) หรือความสัมพันธ์ Entity เดียว เป็นความสัมพันธ์ระหว่าง Entity ชนิดเดียวกัน หรือความสัมพันธ์ในตัวเอง เช่น

♦ ความสัมพันธ์ระหว่าง Entity “วิชาที่บังคับเรียนก่อน” และ “วิชา” (วิชาเรียนบางวิชาบังคับให้ต้องเรียนบางวิชาก่อนจึงจะสามารถเรียนวิชาดังกล่าวได้)



ภาพที่ 5.21 ความสัมพันธ์แบบยูนารี

3.2 ความสัมพันธ์แบบไบนารี หรือความสัมพันธ์ระหว่าง 2 Entity (Binary Relationship) เป็นความสัมพันธ์เป็นความสัมพันธ์แบบ 2 ทางระหว่าง Entity 2 Entity คือ Entity หนึ่งมีความสัมพันธ์กับ Entity อื่นที่ต่างประเภทกัน เป็นความสัมพันธ์ที่พบได้บ่อยที่สุด

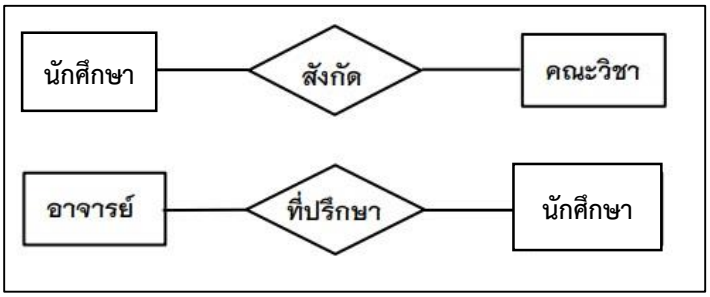


ภาพที่ 5.22 แสดงความสัมพันธ์แบบไบนารี

ตัวอย่างที่ 4

ความสัมพันธ์แบบไบนารี

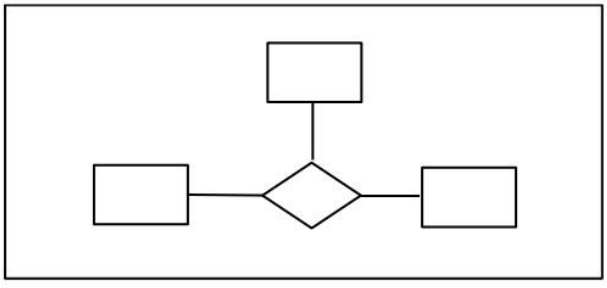
- ♦ นักศึกษา <สังกัด> คณะวิชา
- ♦ อาจารย์ <ที่ปรึกษา> นักศึกษา



ภาพที่ 5.23 แสดงตัวอย่างความสัมพันธ์แบบไบนารี

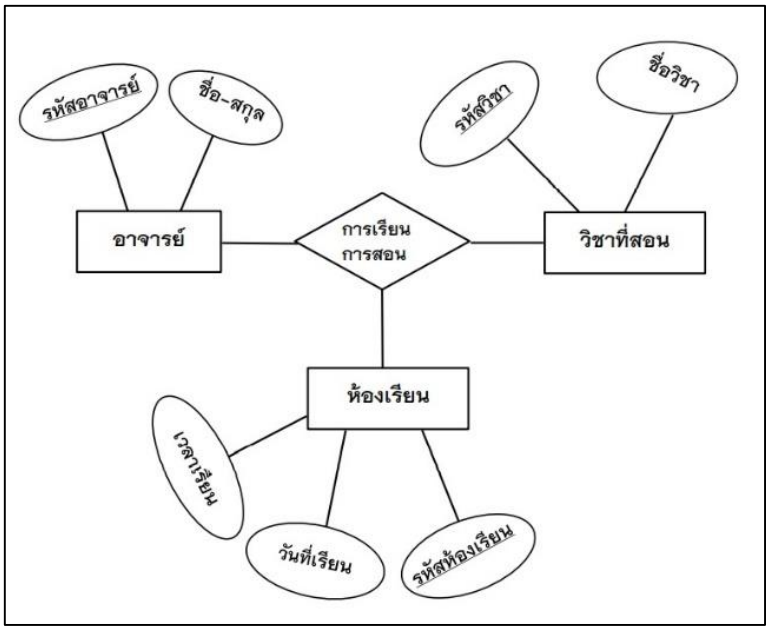
3.3 ความสัมพันธ์ที่มี Entity มาเกี่ยวข้องตั้งแต่สาม Entity ขึ้นไป (N-ary Relationship) เป็นความสัมพันธ์แบบ 3 ทาง หรือเป็นความสัมพันธ์ระหว่าง Entity ต่างประเภทกัน ตั้งแต่ 3 Entity ขึ้นไป โดยที่ N จะหมายถึงจำนวน Entity ที่มาสัมพันธ์กับความสัมพันธ์หนึ่ง เช่น

♦ ความสัมพันธ์แบบสามทางหรือเทอนารี (Ternary relationship) เป็นความสัมพันธ์ระหว่างสามเทอนารี อธิบายได้ ดังนี้



ภาพที่ 5.24 แสดงความสัมพันธ์แบบ 3 ทาง

ตัวอย่างที่ 5 ความสัมพันธ์แบบ 3 ทางหรือเทอนารี ความสัมพันธ์ตารางเรียนแสดงความสัมพันธ์ระหว่าง Entity อาจารย์ Entity วิชาที่สอน และ Entity ชั้นเรียน



ภาพที่ 5.25 แสดงตัวอย่างความสัมพันธ์แบบเทอนารี ตารางเรียน

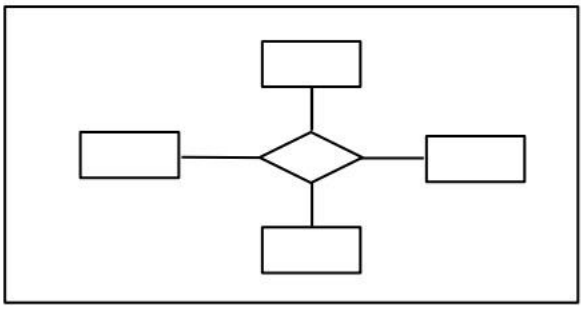
จากภาพที่ 5.25 ความสัมพันธ์ระหว่าง Entity ทั้ง 3 Entity คือ

1) ความสัมพันธ์คู่ที่ 1 คือ ระหว่าง Entity อาจารย์กับ Entity วิชาที่สอนด้วยความสัมพันธ์การเรียนการสอน โดยอาจารย์มีหน้าที่รับผิดชอบการสอนในแต่ละรายวิชา

2) ความสัมพันธ์คู่ที่ 2 คือ ระหว่าง Entity อาจารย์กับ Entity ห้องเรียนด้วยความสัมพันธ์การเรียนการสอน โดยอาจารย์จะทำการสอนรายวิชาภายในห้องเรียน

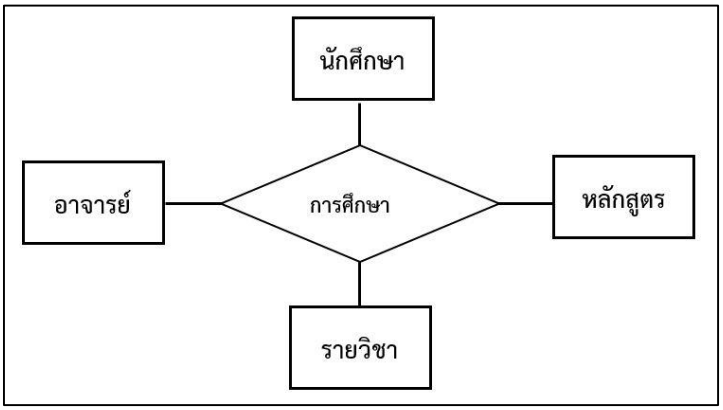
3) ความสัมพันธ์คู่ที่ 3 คือ ระหว่าง Entity วิชาที่สอนกับ Entity ห้องเรียนด้วยความสัมพันธ์การเรียนการสอน โดยแต่ละวิชาที่เปิดสอนจะต้องมีห้องเรียน

♦ ความสัมพันธ์แบบ 4 ทาง หรือควาเทอร์นารี (Quaternary relationship) เป็นความสัมพันธ์ระหว่างสี่ Entity



ภาพที่ 5.26 แสดงความสัมพันธ์แบบควาเทอร์นารี

ตัวอย่างที่ 6 ความสัมพันธ์แบบควาเทอร์นารี



ภาพที่ 5.27 แสดงตัวอย่างความสัมพันธ์แบบควาเทอร์นารี

การออกแบบฐานข้อมูลโดยใช้ E-R โมเดล

การออกแบบฐานข้อมูลโดยใช้ E-R โมเดล มีขั้นตอนที่เกี่ยวข้องหลายขั้นตอน และกระบวนการในการออกแบบ E R โมเดลเป็นกระบวนการที่จะต้องพิจารณาซ้ำ ๆหลายครั้งเพื่อให้ได้ E-R โมเดลที่เหมาะสมและมีข้อมูลครบถ้วน ขั้นตอนในการออกแบบข้อมูลโดยใช้ E-R โมเดล ประกอบด้วยขั้นตอนดังนี้ คือ

1. ศึกษาถึงลักษณะหน้าที่งานของระบบ (Business function) ว่า มีรายละเอียดของการทำงานและข้อมูลที่เกี่ยวข้องอะไรบ้าง มีข้อสมมุติฐาน (Business rule) ของงานต่าง ๆ อะไรบ้าง

2. กำหนด Entity ที่ควรจะมีอยู่ในฐานข้อมูล ฐานข้อมูลหนึ่ง ๆ ประกอบด้วยหลาย Entity ในการกำหนด Entity ที่ควรจะมีอยู่ในฐานข้อมูลหนึ่ง ๆ จะต้องคำนึงรวมไปถึงว่า Entity นั้น ๆ เป็น Entity ในประเภทใดบ้าง เช่น Entity ประเภทอ่อนแอ (Weak entity)

3. กำหนดประเภทของความสัมพันธ์ระหว่าง Entity ว่ามีความสัมพันธ์อย่างไรบ้าง รวมถึงข้อกำหนดของความสัมพันธ์ โดยพิจารณาจากข้อสมมุติฐานของความสัมพันธ์ระหว่างเ Entity ที่ได้ศึกษามาในข้อ 1 และข้อ 2

4. กำหนดคุณลักษณะของ Entity ว่าควรจะมีรายละเอียดอะไรบ้าง ซึ่งการกำหนดคุณลักษณะของ Entity จะพิจารณาว่ารายละเอียดต่าง ๆ เป็นรายละเอียดที่มีคุณสมบัติ เป็นคีย์ หรือเป็นรายละเอียดที่แปลค่ามา หรือเป็นรายละเอียดที่ประกอบด้วยรายละเอียดที่เป็นข้อมูลหลายอย่าง เช่น ที่อยู่ ประกอบด้วย บ้านเลขที่ ถนน เขต ตำบล จังหวัด รหัสไปรษณีย์ เป็นต้น

5. กำหนดคีย์ของแต่ละ Entity ว่า จะใช้รายละเอียดของข้อมูลใดเป็นคีย์ของ Entity นั้น ๆ ซึ่งจะต้องเป็นรายละเอียดของข้อมูลที่มีค่าเป็นเอกลักษณ์ หรือค่าเฉพาะไม่ซ้ำซ้อนใน Entity นั้น ๆ

6. นำรายละเอียดตั้งแต่ขั้นตอนที่ 2 ถึง 5 มาเขียน E-R โมเดล โดยใช้สัญลักษณ์ที่มาข้างต้นเป็นการอธิบายความสัมพันธ์ระหว่างเอนทิตี้ หลังจากนั้นให้ทำการทบทวน E-R โมเดลว่าจะปรับเปลี่ยนเค้าร่างใหม่หรือไม่ (Refinement primitive) เพื่อให้ได้ E-R โมเดลที่สมบูรณ์

นอร์มอลฟอร์ม (Normal forms)

กระบวนการหนึ่งที่สำคัญในการออกแบบฐานข้อมูลคือ กระบวนการ Normalization ซึ่งเป็นกระบวนการในการออกแบบฐานข้อมูลเชิงสัมพันธ์เพื่อให้มีการปรับปรุงเปลี่ยนแปลงแบบแผนเชิงสัมพันธ์ให้อยู่ในรูปแบบที่เหมาะสม คือการกำจัดความซ้ำซ้อนในความสัมพันธ์และลดปัญหาที่จะตามมา อันได้แก่ความผิดปกติอันเนื่องมาจากการเพิ่ม ลบ และปรับปรุงข้อมูล เป็นต้น ขั้นตอน Normal forms เพื่อการออกแบบที่ดีที่สุดในการทำ Normalization กระบวนการจะเป็นแบบ Step-wise ซึ่งเป็นลำดับขั้นตอนแต่ละขั้นจะแปลงสกีมาของความสัมพันธ์เพื่อให้อยู่ในรูปของ Normal Form ที่สูงขึ้น แต่ละ Normal form จะประกอบด้วย Normal form ก่อนหน้านี้ทั้งหมด และมีการเพิ่มประสิทธิภาพโดยเพิ่มเติมบางอย่างซึ่งทำให้มีประสิทธิภาพมากกว่าเดิม

1. นอร์มอลฟอร์ม ระดับที่ 1 (First normal form (1NF))

ความสัมพันธ์ของรีเลชันในอันดับแรกจะถือว่าความสัมพันธ์นั้นอยู่ในรูปของ First normal form โดยแอตทริบิวต์ของทั้งหมด มีโดเมนที่ไม่สามารถแบ่งออกเป็นส่วน ๆ ได้ หรือ การทำให้เป็นหน่วยที่เล็กที่สุด แนวคิดของหน่วยที่เล็กที่สุดสำหรับแอตทริบิวต์ เพื่อให้แน่ใจว่าไม่มี ‘กลุ่มซ้ำ’ เนื่องจากระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ คือความสามารถในการจัดเก็บค่าเดียวเท่านั้นที่จุดตัดของแถวและคอลัมน์ กลุ่มซ้ำ คือเมื่อกลุ่มเราพยายามที่จะเก็บค่าหลายค่าที่จุดตัดของแถวและคอลัมน์ และตารางที่จะประกอบด้วยเป็นค่าที่ซ้ำกันโดยไม่มีการจำกัดความสัมพันธ์อย่างเคร่งครัด

1NF เป็นการทำ normalization ขั้นที่ 1 โดยกำหนดว่า table ใด ๆ จะเป็น 1NF จะต้องเป็น table ที่ไม่มีค่า multivalued column หรือ repeating group

ตัวอย่างกำหนดตารางเก็บข้อมูลนักศึกษาดังนี้

Student

St_ID	St_Name	Major	Faculty	Sub_ID	Sub_Name	Grade
64000001	นายสมใจ รักเรียน	Technology	Education	422359	Database	C
				426589	Computer	B
64000045	นางสาวขยัน ใจดี	Mathematic	Science	422359	Database	B
				426589	Computer	A
				438569	Media	B

ตารางนี้ไม่เป็น 1NF เนื่องจากมีข้อมูลในฟิลด์ Sub_ID, Sub_Name และ Grade ที่เป็น repeating group ดังนั้น ถ้าจะปรับตารางนี้ให้เป็น 1NF ก็จะต้องขจัด repeating group นี้ออกไปดังนี้

Student

St_ID	St_Name	Major	Faculty	Sub_ID	Sub_Name	Grade
64000001	นายสมใจ รักเรียน	Technology	Education	422359	Database	C
64000001	นายสมใจ รักเรียน	Technology	Education	426589	Computer	B
64000045	นางสาวขยัน ใจดี	Mathematic	Science	422359	Database	B
64000045	นางสาวขยัน ใจดี	Mathematic	Science	426589	Computer	B
64000045	นางสาวขยัน ใจดี	Mathematic	Science	438569	Media	B

2. นอร์มอลฟอร์ม ระดับที่ 2 (Second normal form (2NF))

ความสัมพันธ์ของตารางจะถือว่าอยู่ในรูปของ Second normal Form เมื่ออยู่ใน 1NF และไม่มี Non-key attribute ที่ขึ้นอยู่กับบางส่วนของ Candidate key แต่ขึ้นอยู่กับทั้งหมดของ Candidate key จากข้อกำหนดข้างต้น ความสัมพันธ์ที่มีแอตทริบิวต์เดียวเป็น Candidate key จะถือว่าอยู่ในรูป 2NF เสมอ

ตัวอย่าง

จากตาราง Student หลังจากปรับให้เป็น 1NF แล้วจะเห็นว่าข้อมูลใน Student ประกอบด้วยข้อมูล 2 เรื่องปะปนกัน คือ ข้อนิสิตและข้อมูลผลการเรียน ดังนั้น ถ้าเราทำการแยกตารางออกเป็น 2 ตาราง โดยให้ตารางหนึ่งเก็บนิสิต และอีกตารางเก็บข้อมูลผลการเรียนจะได้ตารางใหม่ 2 ตาราง ดังนี้

Student

St_ID	St_Name	Major	Faculty
64000001	นายสมใจ รักเรียน	Technology	Education
64000001	นายสมใจ รักเรียน	Technology	Education
64000045	นางสาวขยัน ใจดี	Mathematic	Science
64000045	นางสาวขยัน ใจดี	Mathematic	Science
64000045	นางสาวขยัน ใจดี	Mathematic	Science

Grade

St_ID	Sub_ID	Sub_Name	Grade
64000001	422359	Database	C
64000001	426589	Computer	B
64000045	422359	Database	B
64000045	426589	Computer	B
64000045	438569	Media	B

จากตาราง Firth semester/ Grade มีฟิลด์ St_ID และ Sub_ID เป็น Key ซึ่งถ้ากำหนดค่าขึ้นมา 1 ชุด จะสามารถอ้างถึงค่าของฟิลด์ Grade ได้ 1 ค่า ดังนั้น เขียนแสดงความสัมพันธ์ของ Functional dependency ชุดนี้ได้เป็น

St_ID, Sub_ID ⟶Grade

นอกจากนั้น ถ้ากำหนดค่าของฟิลด์ Sub_ID ขึ้นมา 1 ค่าก็จะสามารถกำหนดค่าของฟิลด์ Sub_Name ได้ 1 ค่า เช่นกัน ดังนั้นความสัมพันธ์ของ Functional dependency ของ 2 ฟิลด์นี้ คือ

Sub_ID ⟶ Sub_Name

ซึ่งทำให้ตาราง Firth semester/ Grade นี้ยังไม่เป็น 2NF เพราะมี non-key column ในที่นี้คือ ฟิลด์ Sub_Name ที่ขึ้นกับบางส่วนของ Key ในที่นี้คือ ฟิลด์ Sub_ID

วิธีการปรับให้ตารางเป็น 2NF ก็คือ การแตกตารางเป็น 2 ตาราง โดยแยกตารางที่มี Non-key column ที่ขึ้นกับบางส่วนของ key ออกมาจากตารางหลัก ดังนั้น ทำการแยกตารางประกอบด้วยฟิลด์ Sub_ID และ Sub_Name ออกมาอีก 1 ตาราง ดังนี้

Subject

Sub_ID	Sub_Name
422359	Database
426589	Computer
438569	Media

ส่วนตารางที่เหลือ คือ ตัดฟิลด์ที่เป็น Non-key column ที่ขึ้นกับบางส่วนของ Key ออกไป

Grad

St_ID	Sub_ID	Grade
64000001	422359	C
64000001	426589	B
64000045	422359	B
64000045	426589	B
64000045	438569	B

3. นอร์มอลฟอร์ม ระดับที่ 3 (Third normal form (3NF)

ความสัมพันธ์ที่อยู่ในรูปของ Third normal form หมายถึงตารางที่อยู่ในรูปของ 2NF และไม่มี Non-key attribute ที่ขึ้นกับ Candidate transitively key ที่มีแอตทริบิวต์ทุกตัวขึ้นโดยตรงกับ Primary key และไม่ผ่าน Transitive relation ที่แอตทริบิวต์ Z อาจขึ้นกับ Non-key

attribute Y และ Y ขึ้นกับ Primary key X ตามลำดับ โดยอ้างถึงกฎความสัมพันธ์แบบ Transitivity หมายถึง เมื่อ X⟶Y และ Y⟶Z แล้ว X⟶Z

จากความสัมพันธ์ 3NF แสดงให้เห็นว่า Non-key แอตทริบิวต์จะเป็นอิสระต่อกันและกัน

ตัวอย่าง

จากตาราง Student ในขั้นตอน 2NF จะเห็นว่ามีฟิลด์ Major และ Faculty ที่ขึ้นต่อกัน หมายความว่า ถ้ากำหนดค่าของฟิลด์ Major ขึ้นมา 1 ค่าจะสามารถกำหนดค่าของฟิลด์ Faculty ขึ้นมาได้ 1 ค่า เช่นกัน

Student

St_ID	St_Name	Major	Faculty
64000001	นายสมใจ รักเรียน	Technology	Education
64000045	นางสาวขยัน ใจดี	Mathematic	Science

แต่ว่าฟิลด์ Major และ Faculty ไม่ได้เป็น Key ดังนั้น จึงทำให้ตารางนี้ยังไม่เป็น 3NF

วิธีการปรับให้ตารางเป็น 3NF ก็คือ การแตกตารางเป็น 2 ตาราง โดยแยกตารางที่มีคู่ของ Non-key column ที่ขึ้นต่อกันออกมา ดังนี้

Faculty

Major	Faculty
Technology	Education
Mathematic	Science

ส่วนตารางที่เหลือ คือ ตัดฟิลด์ที่เป็น Dependency ของคู่ non-key column นั้นออกไป ดังนี้

Student

St_ID	St_Name	Major
64000001	นายสมใจ รักเรียน	Technology
64000045	นางสาวขยัน ใจดี	Mathematic

สรุป หลังจากทำ Normalization มาถึง 3NF แล้วจะได้ตารางข้อมูลทั้งหมด 4 ตารางดังนี้

Student (St_ID, St_Name, Major)

Faculty (Major, Faculty)

Subject (Sub_ID, Sub_Name)

Grade (ST_ID, Sub_ID, Grad)

สรุป

การออกแบบฐานข้อมูลด้วยแบบจำลอง E-R (Entity relationship model) เป็นการอธิบายโครงร่างของฐานข้อมูลผ่านทางแบบจำลอง E-R Diagram เปรียบเสมือนเป็นพิมพ์เขียวของระบบฐานข้อมูล ซึ่งประกอบด้วยองค์ประกอบสำคัญ คือ Entity, Attribute, และ Relationship

Entity คือ สิ่งที่มีอยู่จริง จับต้องได้ หรือเป็นจินตภาพที่แสดงความเป็นหนึ่งเดียว เมื่อกล่าวถึงแล้ว ทุกคนเข้าใจตรงกัน เช่น อาจารย์ ผู้บริหาร นักศึกษา เป็นต้น

Attribute หรือ พรอพเพอตี้ (Property) คือกลุ่มของค่าความจริงใด ๆ ที่เป็นรายละเอียดของ Entity ซึ่งแสดงลักษณะและคุณสมบัติของเอนทิตี้ เช่น รหัสนักศึกษา ชื่อ-นามสกุล อายุ สาขาวิชา นอกจากนั้น ยังมีการระบุด้วยว่า Attribute เป็นคีย์กำหนดกฎข้อบังคับต่าง ๆ Entity และ Relationship

Relationship คือ ความสัมพันธ์ซึ่งเป็นลักษณะการเกี่ยวพันกันระหว่าง Entity หนึ่งกับตัวเองหรือ Entity อื่น อาจเป็นความสัมพันธ์ที่มากกว่า 2 Entity ก็ได้ เช่น นักศึกษาลงทะเบียนเรียน

นอร์มอลฟอร์ม (Normal forms) เป็นกระบวนการในการออกแบบฐานข้อมูลเชิงสัมพันธ์เพื่อให้มีการปรับปรุงเปลี่ยนแปลงแบบแผนเชิงสัมพันธ์ให้อยู่ในรูปแบบที่เหมาะสม คือการกำจัดความซ้ำซ้อนในความสัมพันธ์และลดปัญหาที่จะตามมา อันได้แก่ความผิดปกติอันเนื่องมาจากการเพิ่ม ลบ และปรับปรุงข้อมูล

บทที่ 6
ภาษา SQL เบื้องต้น

ภาษา Structured query language (SQL) หรือเรียกว่า เอส-คิว-แอล เป็นภาษาพื้นฐานที่ช่วยให้ผู้ใช้สามารถจัดการกับข้อมูลในฐานข้อมูลเชิงสัมพันธ์ ภาษา SQL มีข้อดีคือผู้ใช้งานสามารถระบุข้อมูลที่สนใจ หรือข้อมูลที่ต้องการเรียกใช้ได้ โดยไม่จำเป็นต้องเข้าใจ หรือรู้ถึงวิธีการที่จะเรียกใช้ข้อมูล ซึ่งระบบการจัดการฐานข้อมูลจะทำหน้าที่ในการหากรรมวิธีในการเข้าถึงข้อมูลที่ผู้ใช้ต้องการผ่านทางคำสั่ง SQL โดยที่คำสั่ง SQL นั้นจะทำงานกับข้อมูลที่เป็นตาราง สามารถทำงานได้กับตารางอย่างน้อย 1 ตารางหรือมากกว่าเป็นต้นไป

จากความสำคัญของของภาษา SQL ในการสร้างฐานข้อมูล ในบทนี้จะกล่าวถึงภาษา SQL เบื้องต้น โดยนำเสนอเนื้อหาสำหรับการศึกษาดังต่อไปนี้

- ความเป็นมาของภาษา SQL
- ภาษาสำหรับการนิยามข้อมูล (Data definition language: DDL)
- ภาษาสำหรับการจัดการข้อมูล (Data manipulation language: DML)
- ภาษาสำหรับการควบคุมข้อมูล (Data control language: DCL)
- ตัวอย่างการแปลงแบบจำลอง E-R มาเป็นตารางโดยใช้ภาษา SQL

ภาษา SQL

SQL ย่อมาจาก structured query language คือภาษาที่ใช้ในการเขียนโปรแกรมเพื่อจัดการกับฐานข้อมูลโดยเฉพาะ เป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์และเป็นระบบเปิด (open system) หมายถึงเราสามารถใช้คำสั่ง SQL กับฐานข้อมูลชนิดใดก็ได้ และคำสั่งงานเดียวกันเมื่อสั่งงานผ่านระบบฐานข้อมูลที่แตกต่างกันจะได้ผลลัพธ์เหมือนกัน ทำให้เราสามารถเลือกใช้ฐานข้อมูลชนิดใดก็ได้โดยไม่ติดยึดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง นอกจากนี้แล้ว SQL ยังเป็นชื่อโปรแกรมฐานข้อมูล ซึ่งโปรแกรม SQL เป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง โปรแกรม SQL จึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์ และเป็นภาษาหนึ่งซึ่งแบ่งการทำงานได้เป็น 4 ประเภท 1) Select query ใช้สำหรับดึงข้อมูลที่ต้องการ 2) Update query ใช้สำหรับแก้ไขข้อมูล 3) Insert query ใช้สำหรับการเพิ่มข้อมูล 4) Delete query ใช้สำหรับลบข้อมูลออกไป

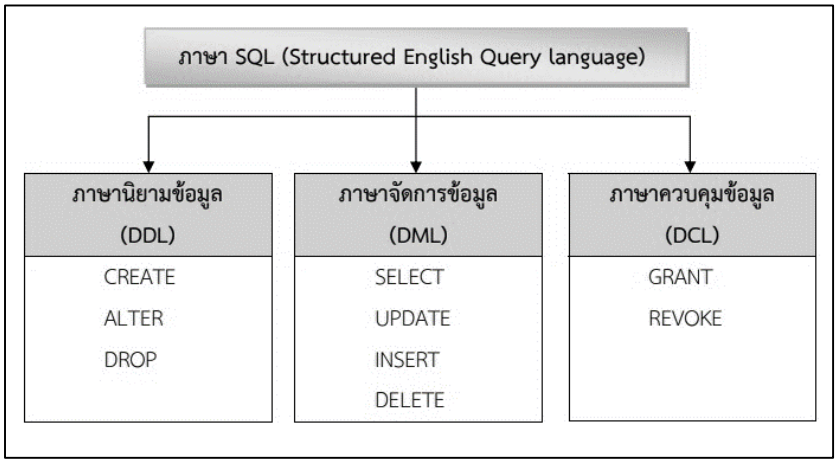
ปัจจุบันมีซอฟต์แวร์ระบบจัดการฐานข้อมูล (DBMS) ที่สนับสนุนการใช้คำสั่ง SQL เช่น Oracle , DB2, MS-SQL, MS-Access นอกจากนี้ ภาษา SQL ถูกนำมาใช้เขียนร่วมกับโปรแกรมภาษาต่าง ๆ เช่น ภาษา c/C++ , Visual Basic และ Java

1. ความเป็นมาของภาษา SQL

ในปีค.ศ 1974 ศูนย์วิจัย IBM ได้พัฒนาRDBMS (Relational database management system) ที่ชื่อว่า System R เพื่อทดลองใช้งาน และแสดงให้เห็นถึงความง่ายของการใช้งานฐานข้อมูลเชิงสัมพันธ์จึงได้ทำการพัฒนาภาษาที่เรียกว่า SEQUEL (Structured english query language) เป็นส่วนประกอบของ RDBMS ที่สามารถรองรับการสืบค้นข้อมูลจากหลายตาราง และการเข้าถึงฐานข้อมูลโดยผู้ใช้หลายคน และต่อมาได้ถูกเปลี่ยนชื่อเป็น SQL ซึ่งในปัจจุบันได้มี RDBMS หลายตัวถูกสร้างขึ้นมา โดยมีภาษา SQL เป็นองค์ประกอบที่สำคัญ เนื่องจากความหลากหลายของ RDBMS ที่ถูกสร้างโดยบริษัทต่าง ๆ สถาบันมาตรฐานแห่งชาติอเมริกา (American National Standards institute–ANSI) และองค์กรมาตรฐานระหว่างประเทศ (International Standard Organization-ISO) จึงได้กำหนดมาตรฐานของภาษา SQL ขึ้น เพื่อให้ทุก DBMS มีภาษา SQL ที่เป็นมาตรฐานเดียวกัน ในปี 1980 จึงได้เกิดภาษา SQL-89 หรือที่เรียกว่า SQL1 ขึ้น และในปี 1992 ก็ได้มีการพัฒนาปรับปรุง SQL-89 ให้มีความสามารถเพิ่มขึ้นอีก และพัฒนาเป็นภาษา SQL-92 หรือ SQL2 ซึ่งเป็นเวอร์ชันที่กลุ่มผู้ผลิต DBMS ใช้เป็นมาตรฐาน SQL ในปัจจุบัน ในปี ค.ศ.1999 มีการพัฒนาเป็นภาษา SQL-99 หรือ SQL3 ที่มีการรวมแนวคิดเชิงวัตถุ (Object-oriented concept) เข้าไปด้วยแต่ SQL-99 ก็ไม่เป็นที่นิยมในกลุ่มผู้ผลิต DBMS

2. องค์ประกอบของภาษา SQL

ภาษา SQL (Structured english query language) เป็นภาษาที่มีรูปแบบเป็นภาษาอังกฤษง่ายต่อการเรียนรู้และการเขียนโปรแกรม ซึ่งเป็นภาษาที่มีอยู่ใน DBMS มีความสามารถใช้นิยามโครงสร้างตารางภายในฐานข้อมูล การจัดการข้อมูล รวมไปถึงการควบคุมสิทธิ์การใช้งานฐานข้อมูล SQL จะประกอบด้วยภาษา 3 รูปแบบด้วยกัน แต่ละแบบก็จะมีหน้าที่เฉพาะแตกต่างกันไป แสดงดังภาพที่ 6.1 ได้แก่ 1) ภาษาสำหรับการนิยามข้อมูล (Data definition language: DDL) 2) ภาษาสำหรับการจัดการข้อมูล (Data manipulation language: DML) และ 3) ภาษาสำหรับการควบคุมข้อมูล (Data control language: DCL)



ภาพที่ 6.1 แสดงองค์ประกอบของภาษา SQL

2.1 ภาษาสำหรับการนิยามข้อมูล (Data Definition Language: DDL)

การออกแบบฐานข้อมูลจะทำให้ทราบถึงโครงสร้างของฐานข้อมูลว่าประกอบด้วย Relation ใดบ้างแต่ละ Relation มีความสัมพันธ์กันอย่างไร คีย์ใดเป็นคีย์หลักของ Relation ขั้นตอนต่อไปคือการสร้างฐานข้อมูลลงบนเครื่องคอมพิวเตอร์โดยการแปลง Relation ต่าง ๆ ให้อยู่ในรูปของภาษาสำหรับการนิยามข้อมูล หรือ DDL ซึ่งคำสั่งต่าง ๆ ในภาษาสำหรับนิยามข้อมูล มีดังนี้

ตารางที่ 6.1 คำสั่งต่าง ๆ ในภาษาสำหรับนิยามข้อมูล

คำสั่ง	ความหมาย
CREATE DATABASE	สร้างฐานข้อมูลใหม่
CREATE TABLE	นิยามโครงสร้างข้อมูลในตารางที่สร้างขึ้น
DROP TABLE	ลบโครงสร้างตารางข้อมูลออกจากระบบ
ALTER TABLE	แก้ไขปรับปรุงโครงสร้างตาราง
CREATE INDEX	สร้างดัชนีของตารางโดยอาจใช้ 1 คอลัมน์หรือมากกว่า
DROP INDEX	ลบดัชนีของตารางออกจากระบบ
CREATE VIEW	กำหนดโครงสร้างวิวของผู้ใช้
DROP VIEW	ลบโครงสร้างวิวออกจากระบบ

2.1.1 การกำหนดฐานข้อมูล

หมายถึงคำสั่งในการสร้างฐานข้อมูล และลบฐานข้อมูล ซึ่งมีรูปแบบคำสั่งดังนี้

1) คำสั่งในการสร้างฐานข้อมูล

คำสั่ง CREATE DATABASE เป็นคำสั่งที่ใช้สร้างฐานข้อมูล โดยการระบุชื่อฐานข้อมูลที่ต้องการสร้าง

รูปแบบคำสั่ง

```
CREATE DATABASE <Database Name>;
```

Database Name คือ ชื่อฐานข้อมูลที่ต้องการสร้าง

2) คำสั่งในการลบฐานข้อมูล

คำสั่ง DROP DATABASE เป็นคำสั่งที่ใช้ลบฐานข้อมูลที่เคยสร้างไว้โดยการระบุชื่อฐานข้อมูลที่ต้องการลบ

รูปแบบคำสั่ง

```
DROP DATABASE <Database Name>;
```

Database Name คือ ชื่อฐานข้อมูลที่ต้องการลบ

2.1.2 การสร้างโครงสร้างตาราง

คำสั่ง CREATE TABLE เป็นคำสั่งที่ใช้สร้างตาราง หรือ Relation โดยการระบุชื่อ Attribute ต่าง ๆ ขนาดความกว้างของ Attribute ประเภทของข้อมูลที่จัดเก็บใน Attribute นั้น รวมถึงการระบุว่า Attribute ใด เป็นคีย์หลัก หรือคีย์นอก

รูปแบบคำสั่ง

```
CREATE TABLE <Table Name>
(<COLUMN NAME> < DATA TYPE > [<WIDTH >],
.....
[PRIMARY KEY <ชื่อคอลัมน์>]
[FOREIGN KEY <ชื่อคอลัมน์>]);
```

COLUMN NAME คือ เป็นชื่อของ Attribute ของ Relation

DATA TYPE คือ ประเภทค่าของข้อมูล ตัวอย่างประเภทของข้อมูล

WIDTH คือ ขนาดความกว้างของข้อมูล

PRIMARY KEY คือ กำหนดคีย์หลัก

FOREIGN KEY คือ กำหนดคีย์นอก

ชนิดข้อมูล

ตารางที่ 6.2 ชนิดข้อมูล

ชนิด	ความหมาย
SMALLINT	เป็นประเภทของข้อมูลที่เป็นเลขจำนวนเต็ม
INT	เป็นประเภทของข้อมูลที่เป็นเลขจำนวนเต็มที่มีขนาดกว้างกว่า SMALLINT
NUMBER (X, Y)	เป็นประเภทของตัวเลขที่มีเครื่องหมายและจุดทศนิยมรวม X จำนวน และ Y คือ จำนวนเลขหลังจุดทศนิยม
FLOAT (X, Y)	เป็นประเภทของตัวเลขที่กำหนดในรูปแบบยกกำลัง
CHARACTER (N)	เป็นสตริงที่มีขนาดคงที่ตามที่ระบุ
VARCHAR	สตริงขนาดไม่คงที่
DATE	เป็นข้อมูลประเภท เดือน/วัน/ปี
LOGICAL	เป็นข้อมูลที่มีค่าเป็นนตรรกะที่เป็น “T” (True) หรือ “F” (False)

2.1.3 คำสั่งการเปลี่ยนแปลงโครงสร้างตาราง

คำสั่ง ALTER TABLE เป็นคำสั่งที่ใช้ในการเปลี่ยนแปลงโครงสร้างที่มีการนิยามไว้แล้ว เช่น อาจต้องการเพิ่ม Attribute หรือคอลัมน์ในตาราง หรือต้องการเปลี่ยนประเภทของข้อมูล รวมไปถึงการเปลี่ยนแปลงคำสั่งควบคุมความคงสภาพ โดยมีรูปแบบของคำสั่ง ดังนี้

รูปแบบคำสั่ง

```
ALTER TABLE <TABLE NAME>
 ADD (<COLUMN NAME><DATA
TYPE><WIDTH>[,<COLUMN NAME>
 <DATA TYPE>< WIDTH>,...])
```

2.1.4 การลบโครงสร้างตาราง

คำสั่ง DROP TABLE เป็นคำสั่งที่ใช้ในการลบตาราง มีรูปแบบดังนี้

รูปแบบคำสั่ง

```
DROP TABLE <TABLE NAME>;
```

2.1.5 คำสั่งสร้างดัชนี

ดัชนี (Index) มีความสำคัญต่อฐานข้อมูลเชิงสัมพันธ์เนื่องจาก DBMS ของฐานข้อมูลประเภทนี้จะใช้ดัชนีในการค้นหาแถวข้อมูลที่ต้องการได้อย่างรวดเร็ว นอกจากนี้ ยังสามารถนำมาใช้ในการควบคุมคอลัมน์ที่นำมาสร้างเป็นดัชนีให้มีการเก็บข้อมูลไม่ซ้ำกันได้อีกด้วย

คำสั่ง CREATE INDEX เป็นคำสั่งที่ใช้ในการสร้างดัชนีมีรูปแบบ ดังนี้

รูปแบบคำสั่ง

```
CREATE [UNIQUE] INDEX <INDEX NAME>
 ON<TABLE NAME><COLUMN NAME>[,COLUMN NAME]);
```

2.1.6 คำสั่งลบดัชนี

คำสั่ง DROP INDEX เป็นคำสั่งที่ใช้ในการลบ INDEX มีรูปแบบดังนี้

รูปแบบคำสั่ง

```
DROP INDEX <INDEX NAME>;
```

2.1.7 คำสั่งสร้างวิว

คำสั่ง CREATE VIEW เป็นคำสั่งที่ใช้ในการสร้าง VIEW มีรูปแบบ ดังนี้

รูปแบบคำสั่ง

```
CREATE VIEW <VIEW NAME>
 AS SELECT Statement;
```

2.1.8 คำสั่งลบวิว

คำสั่ง DROP VIEW เป็นคำสั่งในการลบ VIEW มีรูปแบบ ดังนี้

รูปแบบคำสั่ง

```
DROP VIEW <VIEW NAME>;
```

2.2 ภาษาสำหรับการจัดการข้อมูล (Data manipulation language: DML)

การใช้คำสั่งสำหรับการนิยามข้อมูล (DDL) ในการสร้างโครงสร้างของฐานข้อมูลที่ประกอบด้วยตารางต่าง ๆ และดัชนีของตาราง ซึ่งตารางที่ถูกสร้างนี้ยังไม่มีข้อมูล การป้อนข้อมูลเข้าไปในตาราง เพิ่มเติมข้อมูล เรียกค้นข้อมูล ลบแถวข้อมูล จะต้องใช้คำสั่งสำหรับการจัดการข้อมูลหรือเรียกย่อ ๆ ว่า DML คำสั่งต่าง ๆ ที่มีในภาษาสำหรับการจัดการข้อมูล มีดังนี้

คำสั่ง	ความหมาย
SELECT	เรียกค้นข้อมูลในตาราง
INSERT	เพิ่มแถวข้อมูลลงไปในตาราง
UPDATE	ปรับปรุงแถวข้อมูลในตาราง
DELETE	ลบแถวข้อมูลในตาราง

รูปแบบคำสั่ง

```
SELECT [*|DISTINCT] <TARGET LIST>
 FROM <TABLE NAME> [<ALIASES.]
 [WHERE PREDICATE]
 [GROUP BY <COLUMN LIST>] [HAVING PREDICATE]
 [ORDER BY <COLUMN LIST>];
```

*|DISTINCT เป็นการสั่งให้เลือกข้อมูลที่ต้องการทั้งหมด ซึ่งสามารถใช้เครื่องหมาย * แทนได้ แต่ถ้าใช้ DISTINCT จะเลือกข้อมูลที่เป็นไปตามเงื่อนไขเฉพาะ Tuple ที่มีข้อมูลไม่ซ้ำกัน

TARGET LIST เป็นชื่อของ Attribute และนิพจน์ที่ต้องการให้แสดงค่าออกมาตามที่ต้องการ

FROM <TABLE NAME[ALIASES]> เป็นการระบุถึงชื่อ Relation ที่ต้องการให้ดึงข้อมูลออกมา ส่วน ALIASES คือชื่อของ Relation ที่ตั้งขึ้นใหม่

[WHERE <PREDICATE>] หมายถึง เงื่อนไขที่ระบุขึ้นเพื่อดึงข้อมูลของ Tuple ตามที่ต้องการในส่วนของเงื่อนไขนี้

[GROUP BY <COLUMN LIST>] [HAVING PREDICATE] เป็นการจัดกลุ่มตามเงื่อนไขที่ระบุ และภายใต้การจัดกลุ่ม ก็จะมีเงื่อนไขที่จะให้แสดงผลที่เกิดจากการจัดกลุ่มโดยใช้ HAVING

[ORDER BY <COLUMN LIST>] เป็นการระบุให้เรียงลำดับข้อมูลตาม Attribute ที่ต้องการ ถ้ามี ASC ต่อท้ายชื่อของ Attribute เป็นการเรียงจากน้อยไปหามาก และถ้าเป็น DESC เป็นการเรียงจากมากไปหาน้อย

ในการเรียกดูข้อมูล ข้อมูลที่ใช้เรียกดูจากตารางต่าง ๆ อาจจะไม่มีเงื่อนไขหรือมีระบุเงื่อนไขโดยเงื่อนไขที่ระบุหลัง WHERE สามารถทำได้หลายรูปแบบ อาจมีเพียงเงื่อนไขเดียว หรือหลายเงื่อนไข นอกจากนี้ ยังสามารถระบุเงื่อนไขโดยเขียนเป็นข้อคำถามย่อย (Sub query) ได้เช่นกัน

DML (Data Manipulation Language) คือภาษาสำหรับจัดการข้อมูลที่จัดเก็บอยู่ในตารางข้อมูล ซึ่งในกลุ่มภาษา DML นั้น จะครอบคลุมการจัดการข้อมูลทั้งหมด เช่น การเพิ่ม แก้ไข ค้นหา และลบข้อมูล โดยคำสั่งต่าง ๆ มีดังนี้

1. SELECT คือคำสั่งสำหรับสืบค้นข้อมูล หรือค้นหาข้อมูลที่จัดเก็บอยู่ในตารางข้อมูล โดยมักนิยมใช้งานรวมกับคำสั่ง WHERE เพื่อใช้ในการสร้างเงื่อนไขในการแสดงผลข้อมูล โดยมีคำสั่งโดยย่อดังนี้

```
SELECT * FROM <ชื่อตาราง> WHERE ( <เงื่อนไข> )
```

2. INSERT คือคำสั่งสำหรับการเพิ่มข้อมูลลงไปในตารางข้อมูล เช่น การเพิ่มข้อมูลพนักงาน สินค้า เป็นต้น โดยมีคำสั่งโดยย่อดังนี้

```
INSERT INTO <ชื่อตาราง> VALUES ( <ค่าข้อมูล>, <ค่าข้อมูล>, ... N )
```

3. UPDATE คือคำสั่งสำหรับการปรับปรุง หรือแก้ไขข้อมูลในตารางข้อมูล โดยสามารถใช้งานรวมกับคำสั่ง WHERE เพื่อสร้างเงื่อนไขในการแก้ไขข้อมูล

```
UPDATE <ชื่อตาราง> SET <ชื่อ Field> = <ค่าข้อมูล> WHERE ( <เงื่อนไข> )
```

4. DELETE คือคำสั่งในการลบข้อมูลในตารางข้อมูล โดยสามารถใช้งานรวมกับคำสั่ง WHERE เพื่อสร้างเงื่อนไขสำหรับการลบข้อมูล

```
DELETE FROM <ชื่อตาราง> WHERE ( <เงื่อนไข> )
```

โดยทั้งหมดคือความหมาย และคำสั่งโดยย่อของกลุ่มภาษา DML (Data manipulation language)

2.3 ภาษาสำหรับการควบคุมข้อมูล (Data Control Language: DCL) คือกลุ่มภาษาที่ใช้สำหรับการกำหนดสิทธิ์ในการเข้าถึงข้อมูลที่จัดเก็บอยู่ในตาราง

ข้อมูล เช่น กำหนดให้นาย A สามารถเข้าถึงเฉพาะตารางข้อมูลนี้ ห้ามเข้าถึงตารางข้อมูลอื่น หรือกำหนดให้ นาย B เข้าถึงได้ เฉพาะ Field ที่ต้องการเป็นต้น โดยมีคำสั่งต่าง ๆ โดยย่อ ดังนี้

2.3.1 GRANT คือคำสั่งสำหรับการให้สิทธิ์กับบุคคลต่าง ๆ เพื่อเข้าถึงข้อมูลในระดับที่เรากำหนดโดยมีรูปแบบคำสั่งโดยย่อ ดังนี้

```
1. GRANT <privilege>
2. ON <object>
3. TO <user>
4.<WITH GRANT OPTION>
```

2.3.2 REVOKE คือคำสั่งที่มีไว้สำหรับการยกเลิกสิทธิ์ในการเข้าถึงข้อมูล ของบุคคลต่าง ๆ โดยมีรูปแบบคำสั่งโดยย่อดังนี้

```
1. REVOKE <GRANT OPTION FOR><permission>
2. ON <object>
3. FROM <user><CASCADE>
```

อธิบายเพิ่มเติมคำสั่งต่าง ๆ ที่สำคัญ ดังนี้

<privilege> คือสิทธิ์ในการเข้าถึงข้อมูล โดยสามารถระบุคำสั่งของกลุ่ม DML ลงไปได้ เช่น SELECT, UPDATE, DELETE และ INSERT

<object> คือชื่อตารางข้อมูลที่เราต้องการกำหนดสิทธิ์
<user> คือชื่อผู้เข้าใช้งาน

หลักการในการแปลงแบบจำลอง E-R มาเป็นตารางโดยใช้ภาษาเอสคิวแอล

วิเชียร เปรมชัยสวัสดิ์ (2555) ได้เสนอหลักการการแปลงแบบจำลอง E-R มาเป็นตาราง โดยใช้ภาษา SQL มีดังนี้

1. Entity สามารถแปลงเป็นตารางได้โดยตรง โดยการใช้คำสั่ง CREATE TABLE

2. คอมโพสิต Entity ที่มี Relationship แบบไบนารี สามารถแปลงเป็นตารางได้โดยตรง โดยการใช้คำสั่ง CREATE TABL โดยการนำเอาคีย์หลักของทั้ง 2 Entity ที่เกี่ยวข้องมาเป็นคีย์นอกในตารางนี้ด้วย

. **3. Entity ที่มี Relationship แบบ One to Many** ให้คัดลอกเอาคีย์หลักของตารางของ Entity ที่อยู่ข้างที่เป็น One (1) ไปใส่เป็นคีย์นอกของตารางของ Entity ที่อยู่ข้าง Many (M)

4. Entity ที่มี Relationship แบบรีเคอร์ซีฟ มีหลักการเช่นเดียวกับในข้อ 3

5. คอมโพสิต Entity ที่มี Relationship แบบแบบเทอร์นารี สามารถแปลงเป็นตารางได้โดยตรง โดยการใช้คำสั่ง CREATE TABLE โดยการนำเอาคีย์หลักของทั้งสาม Entity ที่เกี่ยวข้องมาเป็นคีย์นอกในตารางนี้ด้วย

6. Attribute ของ Entity สร้างเป็น Attribute ของตารางนั้น ๆ

7. Super Entity สร้างเหมือน Entity ทั่วไป

8. Sub type Entity แปลงเป็นตารางโดยตรงเช่นเดียวกับ Entity ทั่วไป แต่คีย์หลักของตารางของ Entity ที่เป็นซูเปอร์ไทป์จะถูกสืบทอดมาเป็นคีย์นอกในตารางนี้ด้วย

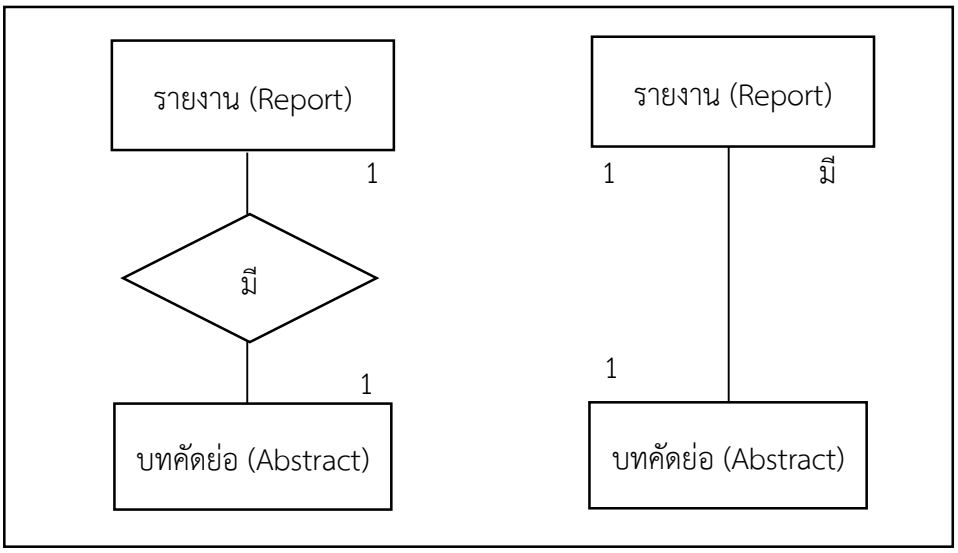
9. Relationship ที่เป็นแบบบังคับ (mandatory) ที่เป็นแบบ One to Many คีย์นอกในตารางข้าง Many ต้องถูกกำหนดให้มีคำเป็น Not Null ด้วย เนื่องจากค่า Default ในภาษา SQL อนุญาตให้มีค่าเป็น Null ได้

ตัวอย่างการแปลงแบบจำลอง E-R มาเป็นตารางโดยใช้ภาษา SQL

1. Relationship แบบ One to one และมีความสัมพันธ์แบบบังคับทั้ง 2 ข้าง

ตัวอย่าง

รายงานแต่ละฉบับต้องมีบทคัดย่อ 1 เรื่อง และบทคัดย่อ 1 เรื่อง ต้องเป็นของรายงาน 1 ฉบับเท่านั้น เขียนเป็นแบบจำลอง E-R ดังภาพที่ 6.2 นี้



ภาพที่ 6.2 แบบจำลองแบบแบบ One to one และมีความสัมพันธ์แบบบังคับทั้ง 2 ข้าง

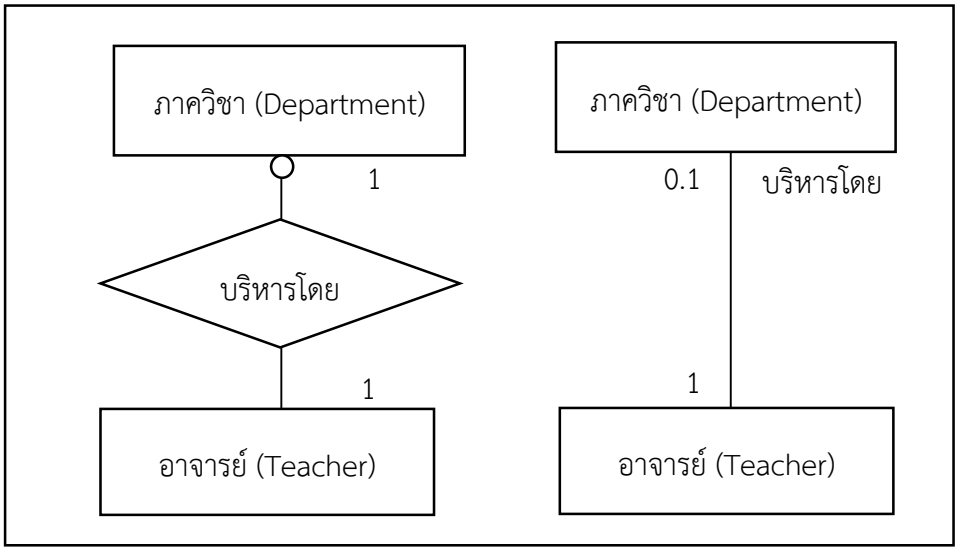
แปลงแบบจำลอง E-R มาเป็นตารางโดยใช้ภาษา SQL ดังนี้

```
CREATE TABLE report
      (report_no number(6),
      report_name varchar2(200),
      primary key (report_no));
CREATE TABLE abstract
   (abt_no number(6),
   report_no number(6) not null unique,
   primary key (abt_ no),
   foreign key (report_no) references report
      on delete cascade
      on update cascade);
```

2. Relationship แบบ One to one และมีความสัมพันธ์แบบข้างหนึ่งเป็นแบบเลือกได้ และอีกข้างหนึ่งเป็นแบบบังคับ

ตัวอย่าง

แต่ละภาควิชามีหัวหน้าภาคได้เพียง 1 คน และอาจารย์สามารถเป็นหัวหน้าภาควิชาได้เพียง 1 คนเท่านั้น



ภาพที่ 6.3 แบบจำลองแบบ One to one และมีความสัมพันธ์แบบข้างหนึ่งเป็นแบบเลือกได้ และอีกข้างหนึ่งเป็นแบบบังคับ

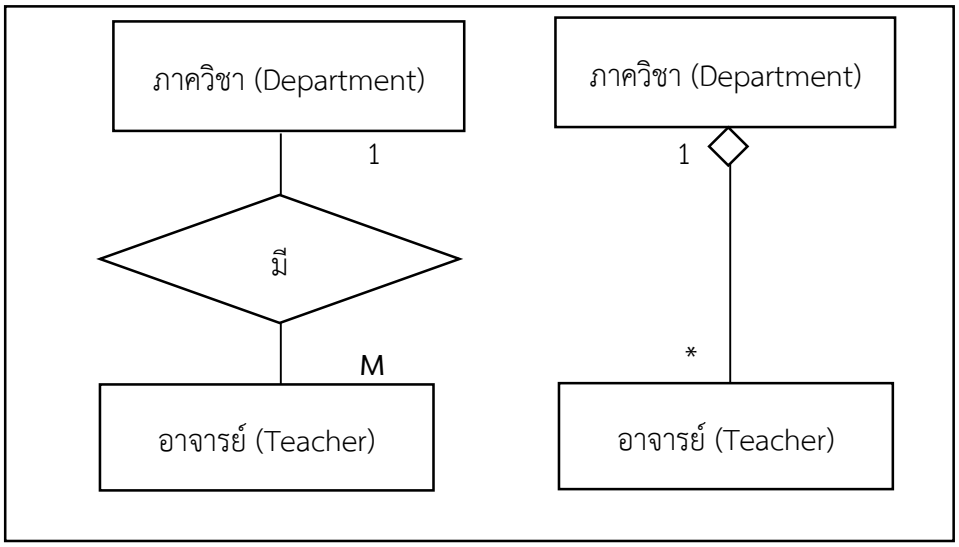
แปลงแบบจำลอง E-R มาเป็นตารางโดยใช้ภาษา SQL ดังนี้

```
CREATE TABLE staff
   (teacher_no        number(10),
   teacher_name varchar2(50),
   primary key (teacher_no);

CREATE TABLE department
   (dept_no number(6),
   dept_name varchar2(30),
   mgr_no number(10) not null unique,
   primary key (dept_no),
   foreign key (mgr_no) references staff
      on delete set default
      on update cascade);
```

3. Relationship แบบ One to One และมีความสัมพันธ์แบบบังคับทั้ง 2 ข้าง

ตัวอย่าง

อาจารย์ต้องมีสังกัดและสังกัดภาควิชาเดียวเท่านั้น และแต่ละภาควิชาต้องมีอาจารย์อย่างน้อย 1 คน

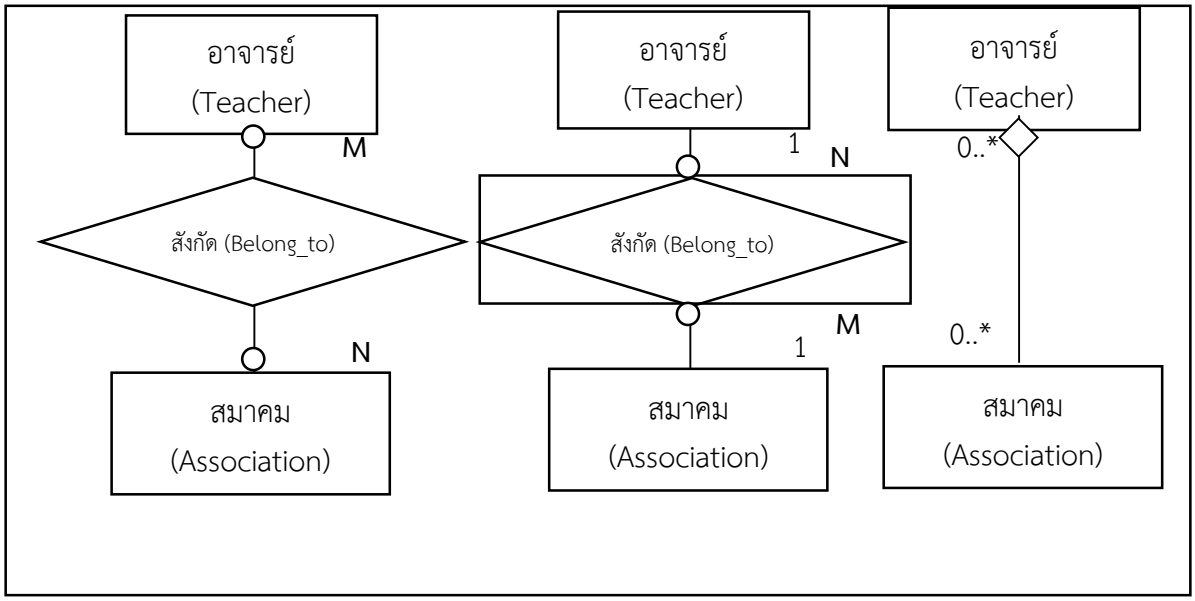


ภาพที่ 6.4 แบบจำลองแบบ One to One และมีความสัมพันธ์แบบบังคับทั้ง 2 ข้าง

```
CREATE TABLE department
        number(6), (dept_no
        dept_name varchar2(30),
        primary key (dept_no);

CREATE TABLE staff
        (teacher_no number(10),
        teacher_name varchar2(30),
        dept_no number(6) not null,
        primary key (teacher_no),
        foreign key (dept_no) references department
                on delete set default
                on update cascade);
```

4. Relationship แบบ Many to Many และมีความสัมพันธ์แบบทั้ง 2 ข้างแบบเลือกตัวอย่าง

1 สมาคม อาจมีสมาชิกได้หลายคนหรือไม่มีเลย และพนักงาน 1 คน สามารถเป็นสมาชิกได้หลายสมาคม หรือไม่เป็นสมาชิกสมาคมใดเลยก็ได้



ภาพที่ 6.5 แบบจำลองแบบ แบบ Many to Many และมีความสัมพันธ์แบบทั้ง 2 ข้างแบบเลือก

```
CREATE TABLE staff
        (teacher_no number(10),
        teacher_name varchar2(256)
        primary key (teacher_no));

CREATE TABLE association
        (assoc_no number(10),
        assoc_name varchar2(256)
        primary key (assoc_name));

CREATE TABLE belong_to
        (emp_no number(10),
        assoc_no number(10),
        primary key (teacher_no, assoc_no),
        foreign key (teacher_no) references employee
                on delete cascade
                on update cascade,
        foreign key (assoc_no) references association
                on delete cascade
                on update cascade);
```

สรุป

ภาษาเอสคิวแอล SQL หรือ Structured query language เป็นภาษาที่ใช้ในการจัดการข้อมูลที่อยู่ภายในฐานข้อมูลเชิงสัมพันธ์สามารถแบ่งหน้าที่ของภาษา SQL ตามองค์ประกอบหลักของภาษาได้ 3 องค์ประกอบ คือ ภาษาสำหรับการนิยามข้อมูล (Data definition language: DDL) ภาษาสำหรับการจัดการข้อมูล (Data manipulation language: DML) และภาษาสำหรับการควบคุมข้อมูล (Data control language: DCL)

ภาษาสำหรับการนิยาม SQL เป็นภาษาที่ผู้ออกแบบฐานข้อมูลใช้ในการสร้างโครงสร้างตารางต่าง ๆ ภายในฐานข้อมูล ประกอบด้วยคำสั่งที่ใช้ในการสร้างโครงสร้างตาราง การเพิ่ม Attribute ในตาราง การลบตาราง การสร้างดัชนีการลบดัชนีการสร้างและลบวิว เป็นต้น ภาษาสำหรับการจัดการข้อมูล เป็นภาษาที่ใช้ในการบันทึกข้อมูลเข้าไปเก็บไว้ในตาราง ประกอบด้วยคำสั่งที่ใช้ในการเรียกค้นข้อมูล คำสั่งเพิ่มแถวข้อมูล คำสั่งปรับปรุง และลบแถวข้อมูล ส่วนภาษาสำหรับการควบคุมข้อมูล เป็นภาษาที่ใช้ในการกำหนดสิทธิ์ในการใช้ข้อมูลในตาราง และกำหนดระดับของการใช้งานประกอบด้วยคำสั่งที่ใช้ในการกำหนดสิทธิ์ให้แก่ผู้ใช้และการยกเลิกสิทธิ์แก่ผู้ใช้

บทที่ 7

ระบบจัดการฐานข้อมูล MySQL และโปรแกรม phpMyAdmin

ระบบจัดการฐานข้อมูล (Database management system: DBMS) ได้ ได้ถูกพัฒนาขึ้นมาเพื่อแก้ไขปัญหาด้านความเป็นอิสระของข้อมูล (Data independence) ที่ไม่มีในระบบแฟ้มข้อมูล ทำให้มีความเป็นอิสระจากทั้งส่วนของฮาร์ดแวร์และส่วนของข้อมูลภายในฐานข้อมูล เพราะการทำงานของโปรแกรมระบบจัดการฐานข้อมูล จะใช้ภาษาสอบถาม (Query Language) ซึ่งเป็นภาษาคอมพิวเตอร์มีรูปแบบคำสั่งคล้ายกับประโยคในภาษาอังกฤษ ที่ใช้สำหรับสอบถาม หรือจัดการกับข้อมูลในฐานข้อมูล และการทำงานของระบบจัดการฐานข้อมูลจะไม่ขึ้นอยู่กับรูปแบบของฮาร์ดแวร์ที่นำมาใช้ และไม่ขึ้นอยู่กับโครงสร้างทางกายภาพของข้อมูล ผู้ใช้สามารถเรียกใช้ข้อมูลจากฐานข้อมูลได้โดยไม่จำเป็นต้องทราบถึงประเภทหรือขนาดของข้อมูล

ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ที่มีการใช้งานกันอย่างแพร่หลายในปัจจุบัน คือ MySQL ซึ่งสามารถนำมาสร้างฐานข้อมูลอย่างง่ายโดยไม่ต้องเขียนโปรแกรมภาษา SQL ในระบบระบบจัดการฐานข้อมูลมีโปรแกรมที่สำคัญคือโปรแกรม phpMyAdmin ซึ่งทำหน้าที่เป็นส่วน Graphic User Interface (GUI) หรือ ระบบการติดต่อสื่อสารระหว่างผู้ใช้งานกับเครื่องคอมพิวเตอร์โดยใช้สัญลักษณ์ภาพ โดยโปรแกรม phpMyAdmin ทำหน้าที่สร้างและนำเข้ารหัส SQL (SQL Code) สู่ระบบจัดการฐานข้อมูล MySQL ที่ดำเนินงานอยู่ในเบื้องหลังโดยอัตโนมัติ ในบทนี้จะกล่าวถึงระบบจัดการฐานข้อมูล MySQL และโปรแกรม phpMyAdmin โดยนำเสนอเนื้อหาสำหรับการศึกษาดังต่อไปนี้

- ระบบจัดการฐานข้อมูล (Database management system: DBMS)
- องค์ประกอบของระบบฐานข้อมูล
- ข้อดีข้อเสียของการระบบจัดการฐานข้อมูล
- ภาษา MySQL
- การสร้างฐานข้อมูลด้วยโปรแกรม phpMyAdmin

ระบบจัดการฐานข้อมูล (Database management system: DBMS)

ระบบจัดการฐานข้อมูล คือชุดคำสั่ง หรือ โปรแกรม หรือ ซอฟท์แวร์ที่สร้างขึ้นมา เพื่อทำหน้าที่บริหารจัดการฐานข้อมูล เช่น รวบรวมข้อมูลให้เป็นระบบ สะดวกและง่ายต่อการจัดการเกี่ยวกับระบบแฟ้มข้อมูลภายในฐานข้อมูล (การเก็บรักษา การเรียกใช้ การแก้ไข การเข้าถึงข้อมูล) รวมถึงการที่จะนำมาปรับปรุงให้ทันสมัย ระบบจัดการฐานข้อมูลจะทำหน้าที่เป็นเครื่องมือ หรือเป็นตัวกลางระหว่างผู้ใช้ชุดคำสั่งต่าง ๆ ที่เกี่ยวข้องกับฐานข้อมูล เพื่อจัดการและควบคุมความถูกต้อง ความซ้ำซ้อนและความสัมพันธ์ระหว่างข้อมูลที่อยู่ภายในฐานข้อมูล รวมถึงความเสียหาย โดยที่ผู้ใช้ไม่จำเป็นต้องทราบถึงรายละเอียดภายในโครงสร้างของฐานข้อมูล ตัวอย่างของระบบจัดการฐานข้อมูล ที่นิยมใช้กันอย่างแพร่หลาย เช่น MySQL, PostgreSQL, Microsoft Access, SQL Server, FileMaker, Oracle, Sybase, dBASE, Clipper และ FoxPro เป็นต้น

1. หน้าที่ระบบจัดการฐานข้อมูล

หน้าที่หลักของระบบจัดการฐานข้อมูล คือการเก็บข้อมูลลงไว้ในฐานข้อมูล และการนำข้อมูลออกมาจากฐานข้อมูล โดยการกระบวนการทำงานของระบบต้องเป็นไปอย่างมีประสิทธิภาพ สะดวก คล่องตัว และทำให้ผู้ใช้เกิดความมั่นใจในความถูกต้อง ความสอดคล้องกัน (Consistency) ของข้อมูลต่าง ๆ ภายในฐานข้อมูล ดังนั้น สามารถสรุปหน้าที่หลักของระบบจัดการฐานข้อมูล ได้ดังนี้

1.1 การจัดการพจนานุกรมข้อมูล (Data dictionary management)

พจนานุกรมข้อมูลเป็นสิ่งที่มีความสำคัญอย่างยิ่งต่อการจัดเก็บรายละเอียดของข้อมูลทั้งหมด หรือที่เรียกกันว่า meta data ไว้อย่างเป็นระบบ เมื่อมีการเข้าถึงข้อมูลในฐานข้อมูล ๆ จะทำการค้นหาโครงสร้างข้อมูลและรายละเอียดของข้อมูลนั้น ๆ ในพจนานุกรมข้อมูลนี้ก่อนและใช้ข้อมูลนี้ในการเข้าถึงข้อมูลจริง ถ้ามีการเปลี่ยนแปลงข้อมูลใด ๆ เกิดขึ้นกับข้อมูลในฐานข้อมูล ก็จะมีการปรับปรุงพจนานุกรมข้อมูลนี้ด้วย

1.2 การจัดการเก็บข้อมูลและการแปลงข้อมูล (Data storage management and transformation)

ระบบจัดการฐานข้อมูล จะทำหน้าที่จัดเก็บข้อมูลอย่างเป็นระบบจัดหมวดหมู่ข้อมูลและแปลงคำสั่งที่ใช้จัดการกับข้อมูลภายในฐานข้อมูล ให้อยู่ในรูปแบบที่ฐานข้อมูลเข้าใจ ซึ่งจะทำให้ผู้จัดเก็บทำงานได้สะดวกมากขึ้น และป้องกันความผิดพลาดได้

1.3 การจัดการด้านความปลอดภัย (Security management)

ระบบฐานข้อมูลที่มีการใช้งานฐานข้อมูลร่วมกันจะสร้างระบบรักษาความปลอดภัยของข้อมูล โดยกำหนดรายชื่อผู้มีสิทธิ์เข้าใช้ระบบ ระบบจัดการฐานข้อมูล จัดการเรื่องนี้โดยการเก็บรายละเอียดเกี่ยวกับผู้ใช้งานแต่ละคนไว้ในพจนานุกรมข้อมูล เช่น มีใครบ้างที่สามารถเข้ามาใช้งานฐานข้อมูลได้ มีรหัสผ่านอย่างไร สามารถใช้งานได้ในระดับใด ด้วยวิธีการดังกล่าวจะทำให้มั่นใจได้ว่าเมื่อมีผู้ใช้หลาย ๆ คนเข้าถึงข้อมูลพร้อมกัน จะยังคงความถูกต้องของข้อมูลในฐานข้อมูลได้

1.4 การจัดการความคงสภาพของข้อมูล (Data integrity management)

เนื่องจากมีการเก็บรายละเอียดเกี่ยวกับข้อมูลต่าง ๆ ไว้ในพจนานุกรม ข้อมูลทั้งหมดการบันทึกหรือแก้ไขข้อมูลใด ๆ แต่ละครั้ง ระบบจัดการฐานข้อมูลจะทำการตรวจสอบและยอมรับให้มีการบันทึกหรือแก้ไขข้อมูลนั้นในขอบเขตที่กำหนด

1.5 การจัดการสำรองข้อมูลและการกู้คืนข้อมูล (Backup and recovery management)

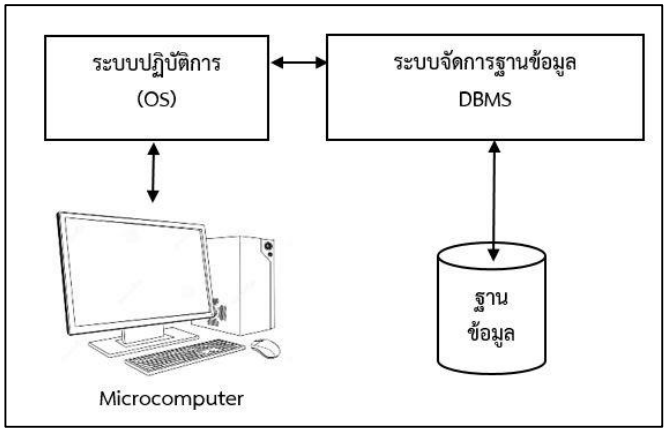
ระบบจัดการฐานข้อมูล จะมีโปรแกรมหรือเครื่องมืออำนวยความสะดวกให้กับฐานข้อมูลเพื่อช่วยในการสำรองข้อมูลและการกู้คืนข้อมูล เพื่อควบคุมความปลอดภัยหรือความมั่นคงและความคงสภาพของข้อมูล

1.6 ภาษาที่ใช้ในการเข้าถึงฐานข้อมูลและการเชื่อมต่อกับโปรแกรมประยุกต์

ระบบการจัดการฐานข้อมูลสนับสนุนหรือเลือกเส้นทางการเข้าถึงข้อมูล โดยผ่านภาษาสำหรับสอบถาม ซึ่งเป็นภาษาหรือคำสั่งเข้าใจง่ายในการค้นคืนข้อมูลจากฐานข้อมูล

1.7 การควบคุมการเข้าถึงข้อมูลของผู้ใช้หลายคน (Multi user access management)

ระบบการจัดการฐานข้อมูลจะใช้หลักการออกแบบโปรแกรมที่เหมาะสม เพื่อให้แน่ใจว่าผู้ใช้หลายคนสามารถเข้าใช้ฐานข้อมูลพร้อมกันได้ในเวลาเดียวกัน โดยไม่ทำให้เกิดความขัดข้องของข้อมูลและข้อมูลมีความถูกต้องสมบูรณ์



ภาพที่ 7.1 ระบบการจัดการฐานข้อมูล

1.8 การติดต่อสื่อสารกับฐานข้อมูล (Database Communication)

ระบบการจัดการฐานข้อมูลที่ทันสมัยจะต้องสนับสนุนการใช้งานฐานข้อมูลผ่านทางเครือข่ายอินเทอร์เน็ตได้ เพื่อให้ผู้ใช้สามารถเขียนคำสั่งด้วยโปรแกรมที่ทำงานบน Web Browser ได้

1.9 การพัฒนาระบบงานได้รวดเร็ว (High Productivity Tools)

ระบบการจัดการฐานข้อมูล เป็นเครื่องมือที่ใช้ในการพัฒนาระบบงานได้รวดเร็วในเวลาอันสั้น ช่วยลดค่าใช้จ่ายในพัฒนาได้ แต่อาจจะไม่ลดค่าใช้จ่ายในการบำรุงรักษา

2. ส่วนประกอบระบบการจัดการฐานข้อมูล

ระบบการจัดการฐานข้อมูล จะมีส่วนประกอบที่สำคัญ 3 ส่วนได้แก่

2.1 ภาษาคำนิยามของข้อมูล [Data Definition Language (DDL)] ในส่วนนี้เป็นส่วนประกอบของระบบการจัดการฐานข้อมูลว่าข้อมูลแต่ละส่วนประกอบด้วยอะไรบ้าง (Data element) ในฐานข้อมูล ซึ่งเป็นภาษาทางการที่นักเขียนโปรแกรมใช้ในการสร้างเนื้อหาข้อมูลและโครงสร้างข้อมูลก่อนที่ข้อมูลดังกล่าวจะถูกแปลงเป็นแบบฟอร์มที่ต้องการของโปรแกรมประยุกต์ หรือในส่วนของ DDL จะประกอบด้วยคำสั่งที่ใช้ในการกำหนดโครงสร้างข้อมูลว่ามีคอลัมน์อะไร แต่ละคอลัมน์เก็บข้อมูลประเภทใด รวมถึงการเพิ่มคอลัมน์ การกำหนดดัชนี เป็นต้น

2.2 ภาษาการจัดการฐานข้อมูล (Data Manipulation Language (DML) เป็นภาษาเฉพาะที่ใช้ในการจัดการระบบฐานข้อมูลซึ่งอาจจะเป็นการเชื่อมโปรแกรมภาษาในยุคที่ 3 และยุคที่ 4

เข้าด้วยกันเพื่อจัดการข้อมูลในฐานข้อมูลภาษานี้มักจะประกอบด้วยคำสิ่งที่อนุญาตให้ผู้ใช้สามารถสร้างโปรแกรมพิเศษขึ้นมา รวมถึงข้อมูลต่าง ๆ ในปัจจุบันที่นิยมใช้ ได้แก่ ภาษา SQL(Structure Query Language) แต่ถ้าหากเป็นเครื่องคอมพิวเตอร์ขนาดใหญ่ DBMS มักจะสร้างด้วยภาษาโคบอล (COBOL language) ภาษาฟอร์แทน (FORTRAN) และภาษาอื่นในยุคที่ 3

2.3 พจนานุกรมข้อมูล (Data Dictionary) เป็นเครื่องมือสำหรับการเก็บและการจัดข้อมูลสำหรับการบำรุงรักษาในฐานข้อมูล โดยพจนานุกรมจะมีการกำหนดชื่อของสิ่งต่าง ๆ (Entity) และระบุไว้ในโปรแกรมฐานข้อมูล เช่น ชื่อของฟิลด์ ชื่อของโปรแกรมที่ใช้รายละเอียดของข้อมูล ผู้มีสิทธิ์ใช้และผู้ที่รับผิดชอบ แสดงส่วนประกอบของระบบการจัดการฐานข้อมูล

3. องค์ประกอบของระบบฐานข้อมูล

องค์ประกอบระบบฐานข้อมูลโดยทั่วไปจะแบ่งออกเป็น 5 ส่วน ซึ่งสอดคล้องกับระบบสารสนเทศ ได้แก่ ฮาร์ดแวร์ ซอฟต์แวร์ ข้อมูล บุคลากร และขั้นตอนการปฏิบัติงาน มีรายละเอียดดังนี้

3.1 ฮาร์ดแวร์ (Hardware)

อุปกรณ์ต่าง ๆ ที่เกี่ยวกับการระบบฐานข้อมูลที่พร้อมอำนวยความสะดวกให้กับผู้ใช้ในการเรียกใช้ข้อมูลในฐานข้อมูล ซึ่งจะมีประสิทธิภาพมากน้อยเพียงใดขึ้นกับขนาดของหน่วยความจำหลัก ความเร็วในการประมวลผลของซีพียู อุปกรณ์รับข้อมูลเข้า/ออก รวมถึงหน่วยความจำสำรองเพื่อใช้เก็บข้อมูล และอุปกรณ์เครือข่ายคอมพิวเตอร์ เพื่อรองรับการเชื่อมโยง (Link) ไปยังคอมพิวเตอร์ทุก ๆ เครื่อง ให้สามารถแบ่งปันทรัพยากรร่วมกันได้

3.2 ซอฟต์แวร์ (Software)

เป็นส่วนที่ทำหน้าที่ประสานการทำงานให้ผู้ใช้สามารถติดต่อกับฐานข้อมูล ซึ่งจะมีซอฟต์แวร์ฐานข้อมูล 2 ประเภท ได้แก่

3.2.1 ซอฟต์แวร์ประยุกต์ (Application Software)

เป็นโปรแกรมที่พัฒนาขึ้นมา อาจพัฒนาเองหรือโปรแกรมสำเร็จรูป ซึ่งการเลือกใช้โปรแกรมจะแตกต่างกันออกไป ขึ้นอยู่กับผู้ใช้งานข้อมูลแต่ละคนว่าจะใช้โปรแกรมอะไรทำงานกับข้อมูลในฐานข้อมูล เช่น โปรแกรมลูกค้าสัมพันธ์ทำให้สามารถเข้าถึงข้อมูลของลูกค้า ตรวจสอบดูประวัติของลูกค้า ที่อยู่ลูกค้า การให้ส่วนลด การให้เครดิตกับลูกค้า สามารถติดต่อสื่อสารกับลูกค้าได้สะดวกขึ้น เป็นต้น

3.2.2 ระบบจัดการฐานข้อมูล (Database Management System: DMBS)

ทำหน้าที่ควบคุมดูแล จัดการใช้ข้อมูล และรักษาความปลอดภัยของข้อมูลที่เก็บไว้ในฐานข้อมูล ซึ่งถือว่าเป็นสื่อกลางระหว่างผู้ใช้โปรแกรมประยุกต์ และฐานข้อมูล เพื่อให้ผู้ใช้สามารถทำงานได้สะดวกมากยิ่งขึ้น และจัดการการเข้าใจข้อมูลว่า ใครจะได้เข้าใช้ก่อนหลัง อาจใช้เทคนิคการจัดแบ่งเวลา (Time sharing) ในการเข้าใช้ข้อมูลพร้อม ๆ กันของผู้ใช้หลาย ๆ คน

3.3 ข้อมูล (Data)

ข้อมูลต่าง ๆ ที่จัดเก็บลงในฐานข้อมูล จะเป็นข้อมูลที่เก็บรวบรวมไว้อยู่ที่ศูนย์กลาง (Server) ซึ่งข้อมูลในฐานข้อมูลนั้นสามารถเข้าถึง (Retrieve) และเรียกใช้ (Query) จากหลาย ๆ คนที่เชื่อมต่อเข้าระบบเครือข่ายได้ ซึ่งจะสามารถเรียกใช้ข้อมูลร่วมกันหรือพร้อม ๆ กันได้ (Concurrency) ในเวลาเดียวกัน แต่ผู้ใช้แต่ละคนจะมองเห็นข้อมูลในฐานข้อมูลในมุมมอง (View)

ที่แตกต่างกันออกไป

3.4 บุคลากร (People)

บุคลากรที่เกี่ยวข้องกับระบบฐานข้อมูล มีดังนี้

3.4.1 ผู้ใช้ทั่วไป (User)

เป็นบุคคลที่เข้ามาใช้ข้อมูลจากฐานข้อมูล การเข้าใช้งานในฐานข้อมูลถ้าเป็นบุคคลที่เกี่ยวข้องกับข้อมูลนั้นโดยตรงก็สามารถจะดู หรือผู้บริหารฐานข้อมูล (DBA) อาจกำหนดสิทธิให้สามารถแก้ไข เพิ่ม หรือ ลบข้อมูลได้ แต่ถ้าเป็นบุคคลภายนอกอาจเข้ามาดูข้อมูลหรือพิมพ์ออกมาเป็นรูปของเอกสารได้ ตัวอย่าง เช่น ระบบการจองตั๋วเครื่องบินออนไลน์ ผู้ใช้ทั่วไปอาจเป็นพนักงานรับจองตั๋ว หรือลูกค้าที่ต้องการจองตั๋ว และเมื่อผ่านกระบวนการจองเสร็จเรียบร้อยก็สามารถพิมพ์เป็นเอกสารเพื่อใช้เป็นหลักฐานการจองได้ทันทันที

3.4.2 พนักงานปฏิบัติการ (Operator)

เป็นผู้ที่ทำหน้าที่ป้อนข้อมูลเข้าระบบฐานข้อมูล หรือการตรวจสอบความถูกต้องของข้อมูลก่อนนำเข้าสู่ระบบ ผู้ควบคุมเครื่อง เป็นต้น

3.4.3 นักวิเคราะห์และออกแบบระบบ (System analyst)

เป็นผู้ศึกษาความเป็นไปได้ของการพัฒนาระบบฐานข้อมูลที่ใช้ในองค์กร และทำการวิเคราะห์การจัดเก็บข้อมูล รวบรวมความต้องการของผู้ใช้ (User requirement) วิเคราะห์ค่าใช้จ่ายและจุดคุ้มทุนของการพัฒนาหรือนำระบบฐานข้อมูลมาใช้ วางแผนการดำเนินงานตามกรอบของการวิเคราะห์และออกแบบระบบ ซึ่งจะวิเคราะห์ระบบงานเดิม จากนั้นวิเคราะห์และออกแบบระบบงานใหม่ก่อนส่งต่อไปยังโปรแกรมเมอร์

3.4.4 โปรแกรมเมอร์ (Programmer)

เป็นบุคคลที่ทำหน้าที่เขียนโปรแกรมประยุกต์ (Application program) ตามที่นักวิเคราะห์และออกแบบระบบได้ทำเอาไว้ ซึ่งโปรแกรมเมอร์ หรือนักเขียนโปรแกรมจะเลือกใช้โปรแกรมในการพัฒนาระบบตามความเหมาะสม การมีประสบการณ์ในการเขียนโปรแกรมมาก ๆ จะทำให้เลือกใช้เทคนิคที่ดีจะทำให้ประมวลผลข้อมูลได้รวดเร็ว แม่นยำ และต้องเขียนให้ถูกตรรกะ (Logic) จึงจะส่งผลให้ผลลัพธ์หรือสารสนเทศที่ได้มีความถูกต้อง

3.4.5 ผู้บริหารฐานข้อมูล (Database administrator: DBA)

เป็นบุคคลที่ทำหน้าที่ดูแล ควบคุมการบริหารจัดการงานต่าง ๆ ของระบบฐานข้อมูลทั้งหมด โดยความรับผิดชอบของ DBA มีดังนี้

1) ตัดสินใจว่าจะรวมข้อมูลเข้าไว้ในระบบฐานข้อมูลใดบ้าง

2) วางแผน วิเคราะห์ และตัดสินใจว่าควรจะจัดเก็บข้อมูลด้วยวิธีใด และใช้เทคนิคในการเรียกใช้ข้อมูล

3) ประสานงานกับผู้ใช้ (User Interface) คอยให้คำแนะนำช่วยเหลือแก่ผู้ใช้เมื่อประสบปัญหาในการใช้งานฐานข้อมูล หรือไม่สามารถเรียกข้อมูลขึ้นมาดูได้

4) กำหนดระบบรักษาความปลอดภัย (Data Security) และคงสภาพข้อมูล (Data integrity)

5) กำหนดแผนการในการสร้างระบบข้อมูลสำรอง (Backup system) และการฟื้นสภาพข้อมูล (Discovery)

6) สำรวจดูผลการทำงาน และตรวจสอบความต้องการของผู้ใช้

3.5 ขั้นตอนการปฏิบัติงาน (Procedure)

การใช้ข้อมูลในระบบฐานข้อมูลจะมีขั้นตอนในการใช้งาน ซึ่งการพัฒนาระบบฐานข้อมูลขึ้นมาใหม่จะยังไม่เคยมีใครใช้งานมาก่อน ทำให้ต้องเริ่มเรียนรู้และฝึกใช้งานระบบฐานข้อมูล ดังนั้น ผู้พัฒนาระบบฐานข้อมูล จึงควรจัดทำเอกสารที่ระบุถึงขั้นตอนการทำงานของหน้าที่งานต่าง ๆ ให้ชัดเจน ทั้งในการใช้งานระบบฐานข้อมูลที่เป็นสภาวะปกติ และในสภาวะที่ระบบอาจเกิดปัญหาหรือใช้งานไม่ได้ (Failure) ซึ่งจะเป็นขั้นตอนสำหรับการปฏิบัติงานของบุคลากรในองค์กรทุกระดับ

4. ข้อดีของการระบบจัดการฐานข้อมูล

จากการจัดเก็บข้อมูลรวมกันเป็นฐานข้อมูลจะก่อให้เกิดประโยชน์ ดังนี้

4.1 ความเป็นอิสระของโปรแกรมและข้อมูล (Program and Data Independence)

เนื่องจากโปรแกรมที่ใช้ในแต่ละไฟล์ข้อมูล มักจะมีรูปแบบและรายละเอียดของข้อมูลที่เก็บภายในฐานข้อมูลเอง ข้อมูลกับโปรแกรมประยุกต์จะแยกกันทำให้เกิดความอิสระของโปรแกรมและข้อมูล เมื่อมีการเปลี่ยนแปลงข้อมูลในระดับใดระดับหนึ่งก็จะไม่ส่งผลกระทบต่อข้อมูลในระดับอื่น ความเป็นอิสระของโปรแกรมและข้อมูล จะเป็นการลดค่าใช้จ่ายในการบำรุงรักษา ความเป็นอิสระของข้อมูลแบ่งออกได้เป็น 2 ประเภทด้วยกัน คือ

4.1.1 ความเป็นอิสระของข้อมูลเชิงกายภาพ (Physical data independence) หมายถึงการเปลี่ยนแปลงโครงสร้างของข้อมูลในระดับภายใน ไม่ส่งผลกระทบต่อโครงสร้างของข้อมูลในระดับแนวคิดหรือระดับตรรกะ เช่น การเปลี่ยนแปลงโครงสร้างการจัดเก็บข้อมูลจากหรืออุปกรณ์จัดเก็บข้อมูลแบบอื่น จะไม่ส่งผลกระทบต่อโครงสร้างข้อมูลในฐานข้อมูล

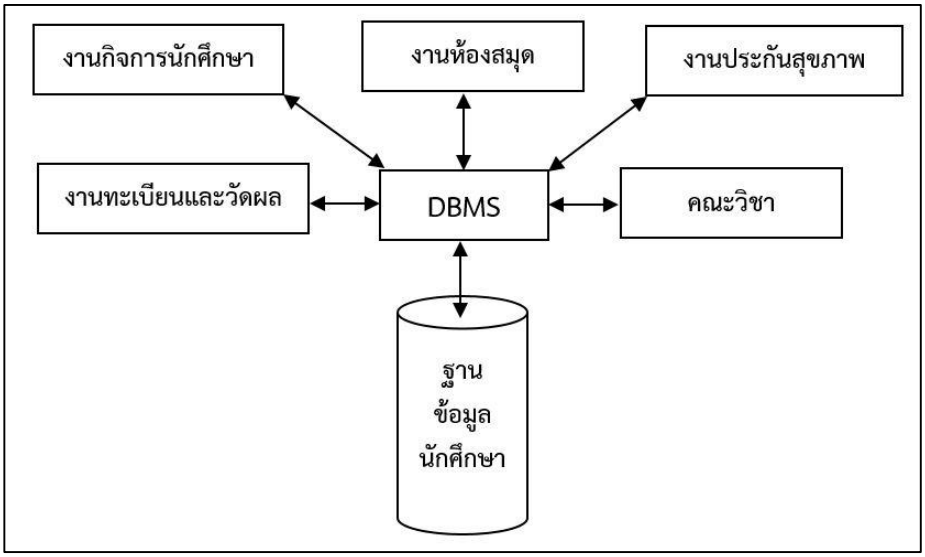
4.1.2 ความเป็นอิสระของข้อมูลเชิงตรรกะ (Logical Data Independence) เป็นการเปลี่ยนแปลงโครงสร้างของข้อมูลในระดับแนวคิดหรือระดับตรรกะ แต่ไม่ส่งผลกระทบต่อโครงสร้างของข้อมูลในระดับภายนอก เช่น โปรแกรมจัดการข้อมูลร้านขายของสะดวกซื้อ ได้มีการปรับปรุงข้อมูลของสินค้า โดยการเพิ่มฟิลด์บางฟิลด์เข้าไปในโครงสร้างข้อมูลสินค้า จะไม่ส่งผลกระทบต่อการทำงานของโปรแกรมที่เกี่ยวข้อง

4.2 ลดความซ้ำซ้อนของข้อมูล (Minimal data redundancy)

การจัดเก็บข้อมูลแบบแฟ้มข้อมูล ทำให้ข้อมูลประเภทเดียวกันถูกเก็บไว้ในหลายแหล่ง เพราะมีผู้ใช้หลายคน การจัดเก็บแบบนี้ก่อให้เกิดความซ้ำซ้อน การแก้ไขเปลี่ยนแปลงข้อมูลทำได้ไม่สะดวก วัตถุประสงค์หลักของการจัดทำระบบฐานข้อมูลคือลดความซ้ำซ้อนของข้อมูล โดยจัดเก็บข้อมูลให้รวมอยู่ในที่ ๆ เดียวกันโดยมีความสัมพันธ์กันของแต่ละตารางที่เกี่ยวข้องกันโดยข้อมูลของระเบียนหนึ่ง ๆ จะไม่ซ้ำซ้อนในอีกตารางหนึ่ง ทำให้การปรับปรุง แก้ไข หรือการควบคุมข้อมูลกระทำได้ง่าย และสามารถแบ่งข้อมูลกันใช้ได้ระหว่างผู้ใช้หลาย ๆ คน รวมทั้งการใช้ข้อมูล เดียวกันในเวลาพร้อม ๆ กันได้อีกด้วย ดังนั้น การที่นำข้อมูลทั้งหมดมาเก็บไว้ที่เดียวกัน ภายในระบบการจัดการเดียวกัน จะเป็นการ “ลด” ความซ้ำซ้อนลงไปได้

4.3 การใช้ข้อมูลร่วมกัน (Improved Data Sharing)

ระบบการจัดการฐานข้อมูล ได้ถูกออกแบบมาเพื่อให้สามารถแบ่งปันการใช้งานโดยจัดเก็บข้อมูลไว้ด้วยกันที่แหล่งเดียวกัน ผู้ใช้งานหลาย ๆ คน สามารถใช้ข้อมูลร่วมกันในเวลาเดียวกันได้ รายละเอียดที่แสดงจะมีเพียงบางส่วนสำหรับผู้ใช้คนหนึ่งเท่านั้น หรือข้อมูลที่แสดงนั้นมาจากตารางต่าง ๆ ที่เกี่ยวข้อง ซึ่งตารางต่าง ๆ จะก็บไว้ในแหล่งเดียวกันเพื่อใช้งานร่วมกัน ดังนั้น ผู้ใช้ไม่ว่าจะอยู่ที่แผนกใดต่างก็สามารถเข้ามาใช้ข้อมูลต่าง ๆ ในฐานข้อมูลได้



ภาพที่ 7.2 แสดงการใช้ข้อมูลจากระบบฐานข้อมูลร่วมกัน

ระบบการจัดการฐานข้อมูล จะจัดเก็บข้อมูลอยู่เพียงที่เดียวหรือแหล่งเดียว เป็นการลดความซ้ำซ้อนของข้อมูล ทำให้ข้อมูลมีความคงที่และมีความสอดคล้องตรงกันมากขึ้น เมื่อมีการเปลี่ยนแปลงข้อมูลใด ๆ ก็จะมีการแก้ไขที่จุดเดียวทำให้ข้อมูลมีความถูกต้องเสมอ และระบบการจัดการฐานข้อมูลจะตรวจสอบความเป็นไปได้ของข้อมูลก่อนที่จะเก็บลงในฐานข้อมูลเพื่อให้ได้ข้อมูลที่ถูกต้องตรงตาม กฎเกณฑ์ผู้ออกแบบระบบฐานข้อมูลสามารถใส่กฎเกณฑ์เพื่อควบคุมดูแลให้ข้อมูลดังกล่าวถูกต้องตามกฎเกณฑ์ และยังเป็นการลดค่าใช้จ่ายในการพัฒนาระบบงาน เนื่องจากกฎเกณฑ์ต่าง ๆ จะเก็บไว้ในฐานข้อมูล ไม่ได้เก็บไว้ในโปรแกรม

4.5 ความเป็นมาตรฐานเดียวกัน (Enforcement of standards)

การเก็บข้อมูลไว้ในฐานข้อมูลเดียวกันจะทำให้สามารถกำหนดมาตรฐาน รวมทั้งมาตรฐาน ต่าง ๆ ในการจัดเก็บข้อมูลให้เป็นไปในลักษณะเดียวกัน หรือให้เป็นไปตามมาตรฐานสากลทำให้ข้อมูลนั้น ๆ นำไปใช้แลกเปลี่ยนข้อมูลระหว่างระบบได้ เช่น การกำหนดรูปแบบ ให้เป็นไปในลักษณะเดียวกัน เช่น วัน/เดือน/ปี หรือ ปี/เดือน/วัน ทั้งนี้จะมีผู้บริหารฐานข้อมูล (Database administrator: DBA) ทำหน้าที่ในการกำหนดสิทธิการใช้งาน และเป็นผู้กำหนดมาตรฐานต่าง ๆ รวมทั้งการบังคับใช้ให้เป็นไปตามมาตรฐาน

4.6 ข้อมูลมีคุณภาพมากขึ้น (Improved data quality)

การประมวลผลในระบบฐานข้อมูล จะมีเครื่องมือหรืออุปกรณ์ที่เกี่ยวข้องเป็นจำนวนมาก ปัจจัยสำคัญที่เกี่ยวข้องที่ทำให้ข้อมูลมีคุณภาพมากขึ้น คือ

4.6.1 ผู้ออกแบบฐานข้อมูล (database designers) เป็นผู้ที่สามารถกำหนด หรือบังคับใช้ระบบจัดการฐานข้อมูล โดยไม่ให้ผู้ที่ไม่มีสิทธิ์ใช้งานเข้าไปทำลายข้อมูลการจัดการดังกล่าวทำให้ข้อมูลมีความถูกต้องและมีความปลอดภัยมากขึ้น

4.6.2 คลังข้อมูล (data warehouse) ถือเป็นแหล่งข้อมูลขนาดใหญ่ที่จัดเก็บไว้เพื่อใช้ในการประกอบการตัดสินใจ ก่อนที่จะนำข้อมูลเข้าไปจัดเก็บไว้ในคลังข้อมูล จะต้องมีการคัด

กรองข้อมูลก่อน ทำให้ข้อมูลที่จัดเก็บในคลังข้อมูลเป็นข้อมูลที่มีคุณภาพ ทำให้การนำเสนอข้อมูลที่ต้องการนำมาวิเคราะห์มีความชัดเจน และน่าเชื่อถือ

4.7 เพิ่มคุณประโยชน์ในการพัฒนาและบำรุงรักษาโปรแกรม (Increased productivity of program development and maintenance)

ข้อมูลที่จัดเก็บมักมีการเปลี่ยนแปลงอยู่เสมอ ทั้งการเปลี่ยนแปลงขนาดหรือรูปแบบของข้อมูล เช่น ธนาคารเดิมมีเพียงการดำเนินการในการรับฝาก-ถอนเงิน และการกู้เงินเป็นหลัก แต่ในปัจจุบันมีการดำเนินการในเรื่องของการทำธุรกรรมด้านประกันภัยเข้ามาด้วย ทำให้ต้องมีการปรับเปลี่ยนแก้ไขฐานข้อมูลไปพร้อม ๆ กับพัฒนาโปรแกรมที่ใช้ดำเนินงาน

จะเห็นได้ว่าถ้าการเก็บข้อมูลใช้วิธีเก็บแบบแฟ้มข้อมูลจะมีความยุ่งยากมากในการเขียนโปรแกรมเพื่อพัฒนาหรือปรับปรุง แต่สำหรับการเก็บข้อมูลโดยวิธีแบบฐานข้อมูล การเปลี่ยนแปลงในโครงสร้างข้อมูลจะเป็นการเปลี่ยนแปลงของข้อมูลในระดับความคิด เช่น การเพิ่มฟิลด์ หรือการเปลี่ยนแปลงชนิดของข้อมูลในฟิลด์ จะไม่ส่งผลกระทบต่อโครงสร้างการทำงานของโปรแกรมที่เกี่ยวข้อง ทำให้โปรแกรมเมอร์สามารถที่จะทำการแก้ไขปรับปรุงหรือพัฒนาโปรแกรมเพิ่มเติมโดยไม่ต้องกังวลกับการออกแบบข้อมูล

5. ข้อเสียของระบบฐานข้อมูล

5.1 ต้นทุนระบบจัดการฐานข้อมูลมีราคาสูง (Cost of DBMS)

ต้นทุนจากการนำระบบจัดการฐานข้อมูลมาใช้ จะขึ้นอยู่กับสภาวะแวดล้อมต่าง ๆ ราคาของระบบจัดการฐานข้อมูล จะเพิ่มขึ้นตามจำนวนเครื่องลูกข่ายที่ให้บริการ ระบบฐานข้อมูลขนาดใหญ่จะมีค่าใช้จ่ายในการซ่อมบำรุงรักษามากกว่าระบบฐานข้อมูลขนาดเล็ก รวมทั้งฮาร์ดแวร์ ซอฟท์แวร์ที่ใช้ในการจัดการระบบฐานข้อมูล ค่าจ้างบุคลากรในการปฏิบัติงาน เป็นต้น

5.2 ความซับซ้อน (Complexity)

การทำงานของระบบจัดการฐานข้อมูลมีความซับซ้อนมากกว่าการจัดการข้อมูลแบบแฟ้มข้อมูล ยิ่งฐานข้อมูลมีขนาดใหญ่ การวิเคราะห์ การออกแบบ และการใช้งานก็จะมีความยุ่งยากและซับซ้อนมากขึ้นตามไปด้วย

5.3 ผลกระทบต่อความเสียหายสูง (Higher impact of a failure)

ฐานข้อมูลมีการทำงานแบบศูนย์กลาง (Centralize database) คือ ข้อมูล จัดเก็บรวมไว้ที่เดียวกัน หากมีข้อผิดพลาดเกิดขึ้น จะทำให้การทำงานของระบบหยุดชะงัก อาจจะทำให้สูญเสียข้อมูลทั้งหมดในฐานข้อมูล ทำให้ผู้ใช้ไม่สามารถเข้ามาใช้ข้อมูลในขณะนั้นได้ ดังนั้น การจัดทำฐานข้อมูลที่ดีจึงต้องมีการสำรองข้อมูลไว้เสมอ

5.4 ขนาดความจุเพิ่มขึ้น (Increased size)

ระบบจัดการฐานข้อมูลที่มีประสิทธิภาพสูง ความซับซ้อนของระบบจัดการฐานข้อมูลนั้นก็จะมีความซับซ้อนมากขึ้นตามไปด้วย เช่น ต้องการขนาดของฮาร์ดดิสก์ในการติดตั้งระบบจัดการฐานข้อมูลที่มีขนาดใหญ่ขึ้น รวมถึงความต้องการใช้ขนาดของแรมเพิ่มขึ้นตามไปด้วย เป็นต้น

ระบบจัดการฐานข้อมูลมายเอสคิวแอล (MySQL)

MySQL เป็นระบบจัดการฐานข้อมูลโดยใช้ภาษา SQL แม้ว่า MySQL เป็นซอฟต์แวร์โอเพนซอร์ส แต่แตกต่างจากซอฟต์แวร์โอเพนซอร์สทั่วไป โดยมีการพัฒนาภายใต้บริษัท MySQL AB ในประเทศสวีเดน โดยจัดการ MySQL ทั้งในแบบที่ให้ใช้ฟรี และแบบที่ใช้ในเชิงธุรกิจ

ความสามารถและการทำงานของโปรแกรม MySQL มีดังต่อไปนี้

1. MySQL เป็นระบบจัดการฐานข้อมูล มีลักษณะเป็นโครงสร้างของการเก็บรวบรวมข้อมูลในการเพิ่มข้อมูล เข้าถึงฐานข้อมูล หรือประมวลผลข้อมูลที่เก็บในฐานข้อมูลจำเป็นจะต้องอาศัยระบบจัดการฐานข้อมูล MySQL จะทำหน้าที่เป็นตัวกลางในการจัดการกับข้อมูลในฐานข้อมูลทั้งสำหรับการใช้งานเฉพาะ และรองรับการทำงานของแอพลิเคชันอื่น ๆ ที่ต้องการใช้งานข้อมูลในฐานข้อมูลเพื่อให้ได้รับความสะดวกในการจัดการกับข้อมูลจำนวนมาก MySQL ทำหน้าที่เป็นทั้งตัวฐานข้อมูลและระบบจัดการฐานข้อมูล

2. MySQL เป็นระบบจัดการฐานข้อมูลแบบ Relational

ฐานข้อมูลแบบ relational จะทำการเก็บข้อมูลทั้งหมดในรูปแบบของตารางแทนการเก็บข้อมูลทั้งหมดลงในไฟล์เพียงไฟล์เดียว ทำให้ทำงานได้รวดเร็วและมีความยืดหยุ่น นอกจากนั้นแต่ละตารางที่เก็บข้อมูลสามารถเชื่อมโยงเข้าหากันทำให้สามารถรวมหรือจัดกลุ่มข้อมูลได้ตามต้องการ โดยอาศัยภาษา SQL ที่เป็นส่วนหนึ่งของโปรแกรม MySQL ซึ่งเป็นภาษามาตรฐานในการเข้าถึงฐานข้อมูล

3. MySQL เป็นโปรแกรมแบบ Open Source ทุกคนสามารถใช้งานและปรับแต่งการทำงานได้ตามต้องการ สามารถดาวน์โหลดโปรแกรม MySQL ได้จากอินเทอร์เน็ตและนำมาใช้งานโดยไม่มีค่าใช้จ่ายใด ๆ

โปรแกรม phpMyAdmin

phpMyAdmin คือ โปรแกรมที่ถูกพัฒนาโดยใช้ภาษา PHP เพื่อใช้ในการบริหารจัดการฐานข้อมูลด้วย MySQL การใช้งานโปรแกรม phpMyAdmin เพื่อแทนการคีย์คำสั่ง เนื่องจากถ้าใช้ฐานข้อมูลที่เป็น MySQL บางครั้งจะมีความลำบากและยุ่งยากในการใช้งาน ดังนั้น จึงมีเครื่องมือในการจัดการฐานข้อมูล MySQL ขึ้นมา นั่นคือโปรแกรม phpMyAdmin ที่ใช้งานด้ง่ายและสะดวกยิ่งขึ้น

phpMyAdmin เป็นส่วนต่อประสานที่สร้างโดยภาษา PHP ซึ่งใช้จัดการฐานข้อมูล MySQL ผ่านเว็บเบราว์เซอร์ โดยสามารถที่จะทำการสร้างฐานข้อมูลใหม่ หรือทำการสร้าง Table ใหม่ ๆ และยังมี function ที่ใช้สำหรับการทดสอบการ query ข้อมูลด้วยภาษา SQL พร้อมกันนั้นยังสามารถทำการ insert delete update หรือแม้กระทั่งใช้คำสั่งต่าง ๆ เหมือนกับกันการใช้ภาษา SQL ในการสร้างตารางข้อมูล

phpMyAdmin เป็นโปรแกรมประเภท MySQL Client ตัวหนึ่งที่ใช้ในการจัดการข้อมูล MySQL ผ่าน web browser ได้โดยตรง phpMyAdmin เป็นโปรแกรมทำงานบน Web server เป็น PHP Application ที่ใช้ควบคุมจัดการ MySQL Server

ความสามารถของ phpMyAdmin คือ

1. สร้างและลบฐานข้อมูล

2. สร้างและจัดการ Table เช่น แทรก Record ลบ Record แก้ไข Record ลบ Table แก้ไข Field

3. โหลดเท็กซ์ไฟล์เข้าไปเก็บเป็นข้อมูลในตารางได้

4. หาผลสรุป (Query) ด้วยคำสั่ง SQL

การติดตั้งโปรแกรม phpMyAdmin บน Server จำลองบนคอมพิวเตอร์ด้วย AppServ

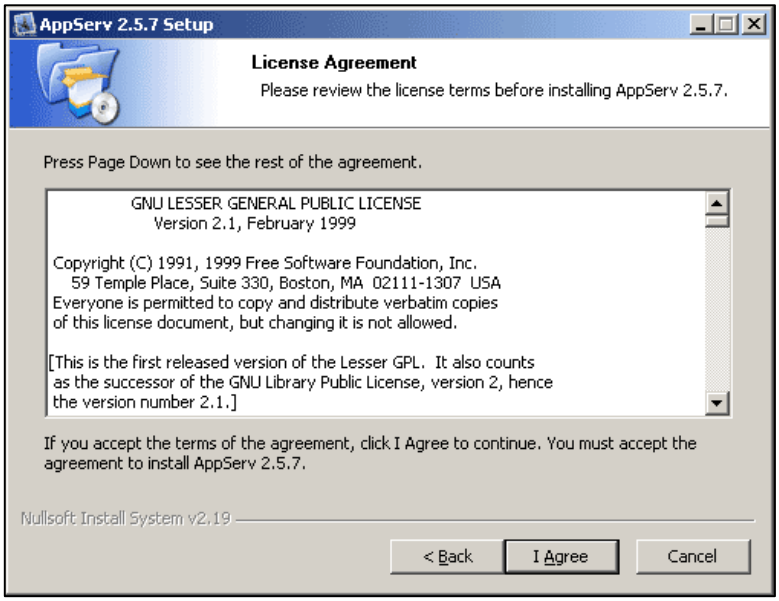
1. ดาวน์โหลดโปรแกรม AppServ จากเว็บไซต์ http://www.appserv.org โดยเลือกเวอร์ชันที่ต้องการ

ภาพที่ 7.3 เว็บไซต์ http://www.appserv.org

2. เมื่อ Download โปรแกรมเสร็จสมบูรณ์ เข้าสู่การติดตั้ง โดยดับเบิ้ลคลิกไฟล์ appserv-win32-x.x.x.exe จะปรากฏหน้าจอตามภาพที่ 7.4

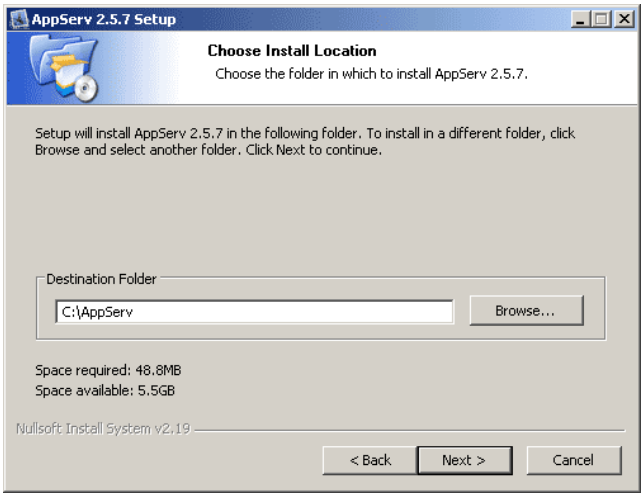
ภาพที่ 7.4 ขั้นตอนที่ 1 การติดตั้งโปรแกรม AppServ

3. เข้าสู่ขั้นตอนเงื่อนไขการใช้งานโปรแกรม โดยโปรแกรม AppServ ได้เผยแพร่ในรูปแบบ GNU License หากผู้ติดตั้งอ่านเงื่อนไขต่าง ๆ เสร็จสิ้นแล้ว หากยอมรับเงื่อนไขให้กด Next เพื่อเข้าสู่การติดตั้งในขั้นต่อไป แต่หากว่าไม่ยอมรับเงื่อนไข ให้กด Cancel เพื่อออกจากการติดตั้งโปรแกรม AppServ ดังภาพตัวอย่างที่ 7.5



ภาพที่ 7.5 แสดงรายละเอียดเงื่อนไขการ GNU License

4. เข้าสู่ขั้นตอนการเลือกปลายทางที่ต้องการติดตั้ง โดยค่าเริ่มต้นปลายทางที่ติดตั้งจะเป็น C:AppServ หากต้องการเปลี่ยนปลายทางที่ติดตั้ง ให้กด Browse แล้วเลือกปลายทางที่ต้องการ ตามภาพที่ 7.6 เมื่อเลือกปลายทางเสร็จสิ้นให้กดปุ่ม Next เพื่อเข้าสู่ขั้นตอนการติดตั้งขั้นต่อไป



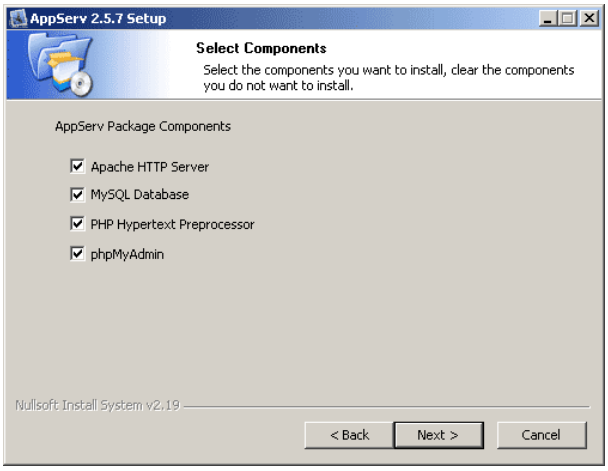
ภาพที่ 7.6 เลือกปลายทางการติดตั้งโปรแกรม AppServ

5. เลือก Package Components ที่ต้องการติดตั้ง โดยค่าเริ่มต้นนั้นจะให้เลือกลงทุก Package แต่หากว่าผู้ใช้งานต้องการเลือกลงเฉพาะบาง Package ก็สามารถเลือกตามข้อที่ต้องการออก โดยรายละเอียดแต่ละ Package มีดังนี้

- Apache HTTP Server คือ โปรแกรมที่ทำหน้าเป็น Web Server
- MySQL Database คือ โปรแกรมที่ทำหน้าเป็น Database Server
- PHP Hypertext Preprocessor คือ โปรแกรมที่ทำหน้าประมวลผลการทำงาน

ของภาษา PHP

– phpMyAdmin คือ โปรแกรมที่ใช้ในการบริหารจัดการฐานข้อมูล MySQL ผ่านเว็บไซต์เมื่อทำการเลือก Package ตามภาพที่ 7.7 เรียบร้อยแล้ว ให้กด Next เพื่อเข้าสู่ขั้นตอนการติดตั้งต่อไป



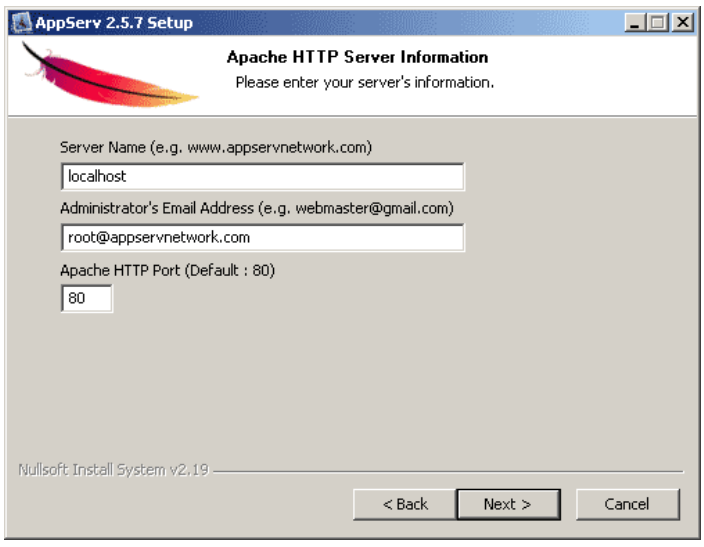
ภาพที่ 7.7 เลือก Package Components ที่ต้องการติดตั้ง

6. กำหนดค่าคอนฟิกของ Apache Web Server มีอยู่ด้วยกันทั้งหมด 3 ส่วน ตามภาพที่ 7.8 ดังนี้

Server Name หรับป้อนข้อมูลชื่อ Web Server ของท่าน เช่น www.appserv.org

Admin Email สำหรับป้อนข้อมูล อีเมล์ผู้ดูแลระบบ เช่น root@appserv.org

HTTP Port สำหรับระบุ Port ที่จะเรียกใช้งาน Apache web server โดยทั่วไปแล้ว Protocol HTTP นั้นจะมีค่าหลักคือ 80 หากว่าท่านต้องการหลีกเลี่ยงการใช้ Port 80 ก็สามารถแก้ไขได้ หากมีการเปลี่ยนแปลง Port การเข้าใช้งาน Web server แล้ว ทุกครั้งที่เรียกใช้งานเว็บไซต์จำเป็นที่ต้องระบุหมายเลข Port ด้วย เช่น หากเลือกใช้ Port 99 ในการเข้าเว็บไซต์ทุกครั้งต้องใช้ http://www.appserv.org:99 จึงจะสามารถเข้าใช้งานได้

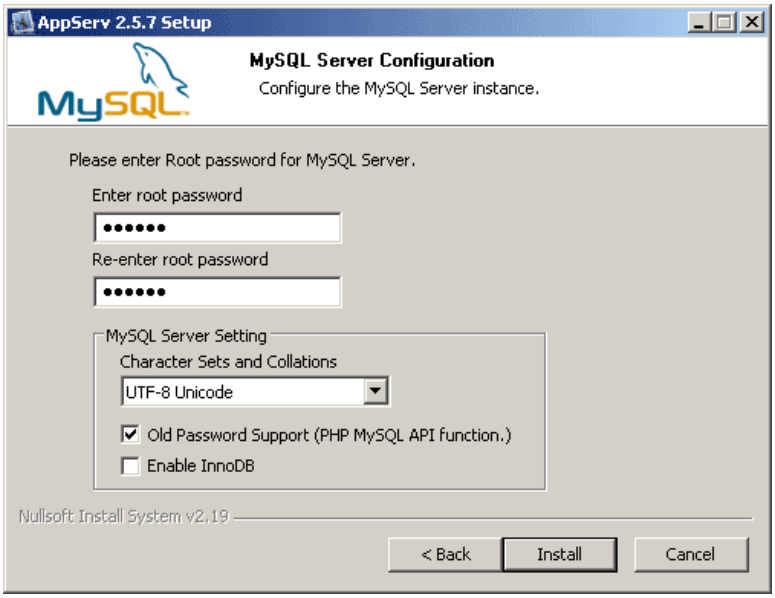


ภาพที่ 7.8 แสดงการกำหนดค่าคอนฟิกค่า Apache web server

7. กำหนดค่าคอนฟิกของ MySQL Database มีอยู่ด้วยกันทั้งหมด 3 ส่วน ตามภาพที่ 7.9 คือ Root password คือช่องสำหรับป้อนรหัสผ่านการเข้าใช้งานฐานข้อมูลของ Root หรือผู้ดูแลระบบ ทุกครั้งที่เข้าใช้งานฐานข้อมูลในลักษณะที่เป็นผู้ดูแลระบบ ให้ระบุ User คือ Root Character sets ใช้ในการกำหนดค่าระบบภาษาที่ใช้ในการจัดเก็บฐานข้อมูล เรียงลำดับฐานข้อมูล Import ฐานข้อมูลExport ฐานข้อมูล ติดต่อฐานข้อมูล

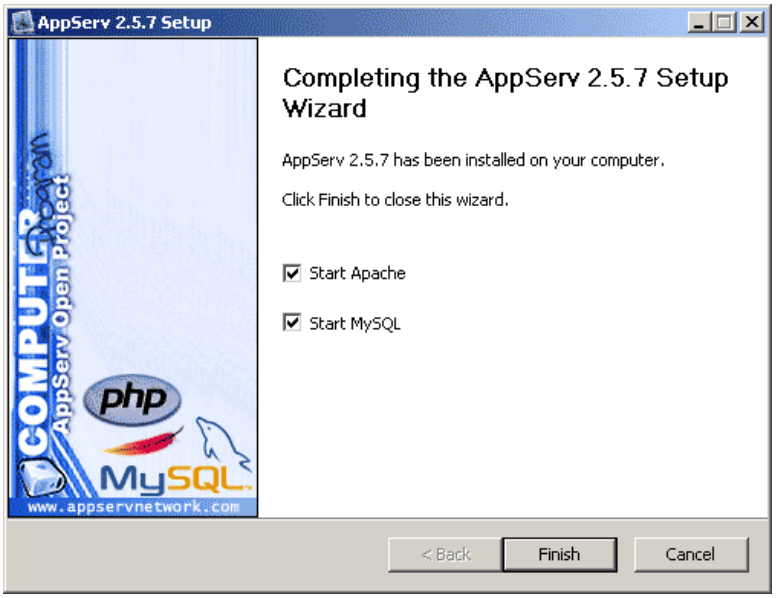
Old password หากมีปัญหาเกี่ยวกับการใช้งาน PHP กับ MySQL API เวอร์ชันเก่า โดยเจอ Error client does not support authentication protocol requested by server; consider upgrading MySQL client ให้เลือกในส่วนของ Old password เพื่อหลีกเลี่ยงปัญหานี้

Enable MyISAM ต้องการใช้งานฐานข้อมูลในรูปแบบ MyISAM ให้เลือกในส่วนนี้ด้วย



ภาพที่ 7.9 แสดงการกำหนดค่าคอนฟิกของ MySQL Database

8. สิ้นสุดขั้นตอนการติดตั้งโปรแกรม AppServ สำหรับขั้นตอนสุดท้ายนี้จะมีให้เลือกว่าต้องการสั่งให้มีการรัน Apache และ MySQL ทันทีหรือไม่ จากนั้นกดปุ่ม Finish เพื่อเสร็จสิ้นการติดตั้งโปรแกรม AppServ



ภาพที่ 7.10 แสดงหน้าจอขั้นตอนสิ้นสุดการติดตั้งโปรแกรม AppServ

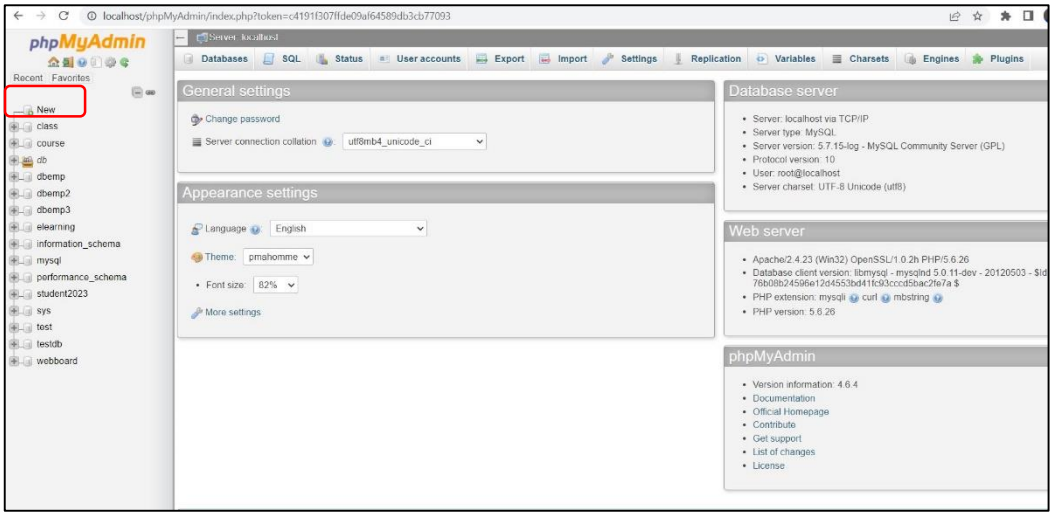
ตัวอย่าง การใช้โปรแกรม phpMyAdmin สำหรับสร้างฐานข้อมูล

การเข้าใช้งาน phpMyAdmin บน Server จำลองบนคอมพิวเตอร์ ทำได้โดยเปิดเว็บบราวเซอร์และพิมพ์ลงไปที่ช่อง URL โดยพิมพ์ localhost/phpMyAdmin/ หรือ http://127.0.0.1/phpMyAdmin/

1. กรอก username และ password ที่ได้กำหนดไว้ในขั้นตอนการติดตั้ง และคลิกที่ปุ่ม Go เพื่อเข้าสู่การทำงาน

ภาพที่ 7.11 แสดงหน้าจอล๊อคอินเข้าโปรแกรม phpMyAdmin

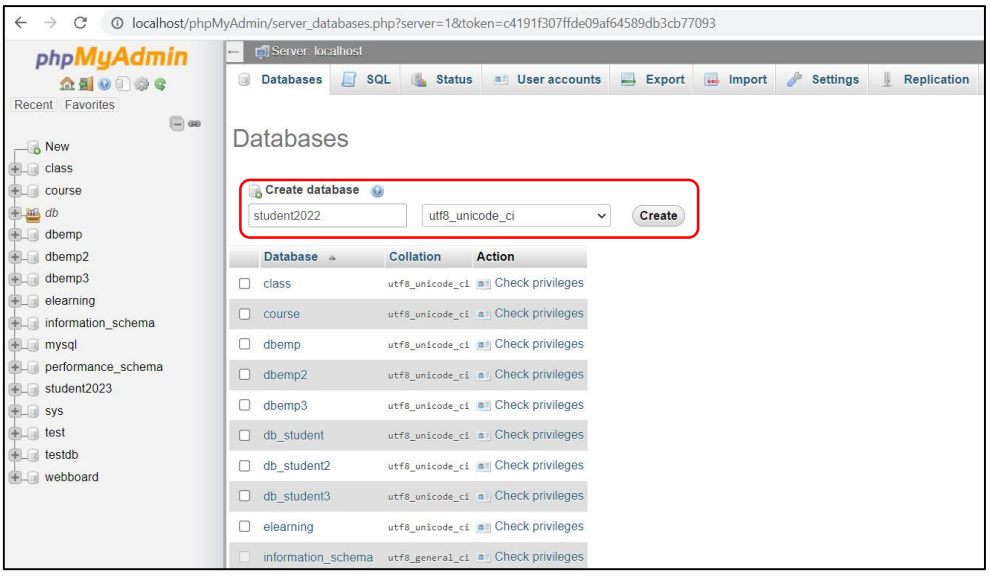
เมื่อ Long in สำเร็จจะปรากฏหน้าจอดังภาพที่ 7.12



ภาพที่ 7.12 แสดงหน้าจอหลักโปรแกรม phpMyAdmin

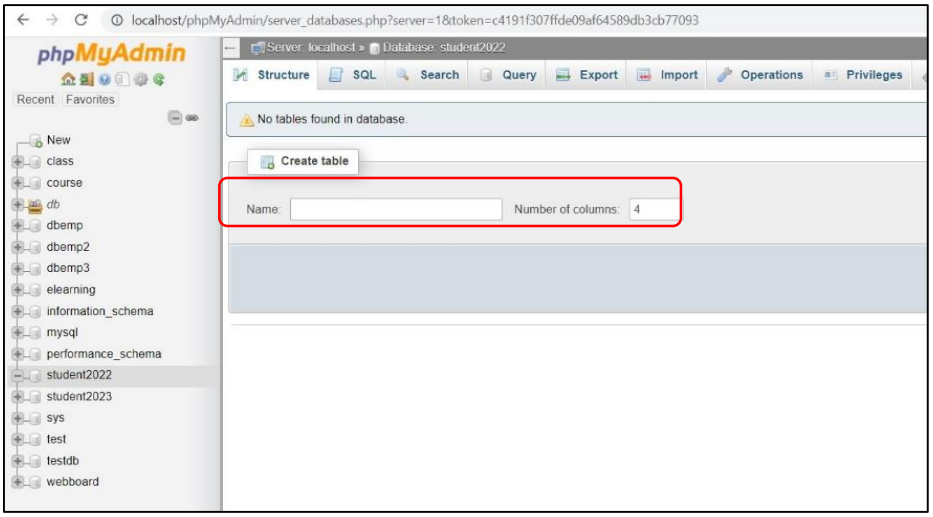
2. เมื่อล๊อคอินเข้าสู่โปรแกรมสำเร็จ สามารถสร้างฐานข้อมูลได้

3. เริ่มต้นสร้างฐานข้อมูลคลิกที่ New ระบบจะปรากฏหน้าต่างเพื่อให้สร้างฐานข้อมูลใหม่ (Create database) จากนั้นตั้งชื่อฐานข้อมูลที่ช่อง Database name และคลิกที่ปุ่ม Create



ภาพที่ 7.13 แสดงขั้นตอนการตั้งชื่อฐานข้อมูล

4. ตั้งชื่อตารางหรือ Relation และกำหนดจำนวน Columns หรือ Attribute จากนั้นคลิก Go ดังภาพที่ 7.14



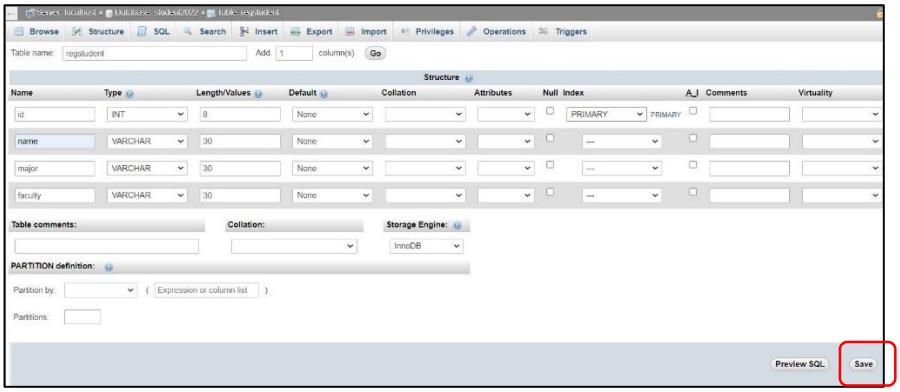
ภาพที่ 7.14 แสดงขั้นตอนการตั้งชื่อตารางและกำหนดจำนวน Columns

5. ตั้งชื่อ Columns ตั้งค่าชนิดของข้อมูล และการกำหนดคีย์หลัก ตัวอย่างการสร้างฐานข้อมูลชื่อ student2022 โดยมีจำนวน 1 ตาราง คือ regstudent ซึ่งในตาราง regstudent ประกอบด้วย 4 attribute ประกอบด้วย ดังตารางที่ 7.1 ดังนี้

ตารางที่ 7.1 โครงสร้างตาราง Regstudent

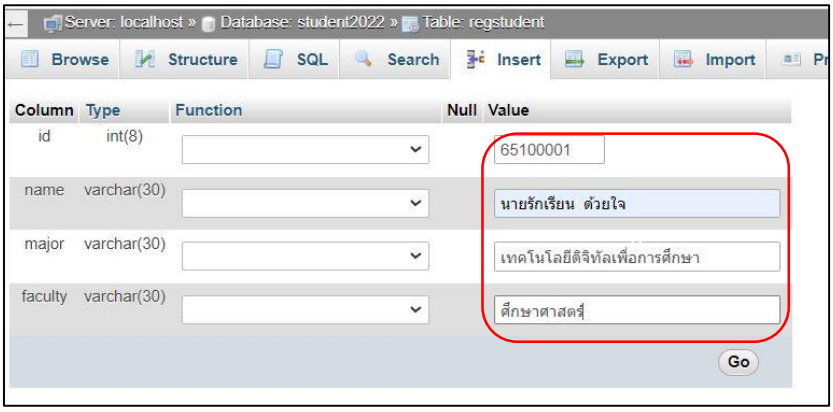
ชื่อ Attribute	ชนิดข้อมูล	ความยาวข้อมูล	กำหนดเพิ่มเติม
id	INT	8	คีย์หลัก
name	VACHARE	30	
major	VACHARE	30	
faculty	VACHARE	30	

จากข้อมูลในตาราง 7.1 นำไปกำหนดในโปรแกรม phpMyAdmin จากนั้นคลิก Save ดังภาพที่ 7.15 ดังนี้



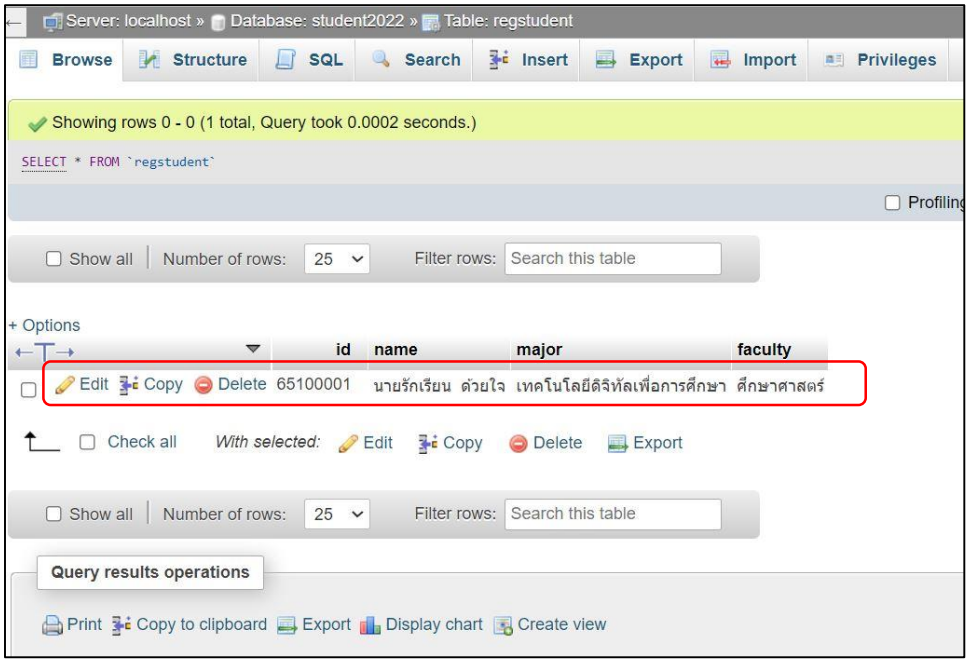
ภาพที่ 7.15 แสดงตั้งชื่อ Columns ตั้งค่าชนิดของข้อมูล และการกำหนดคีย์หลัก

6. การเพิ่มข้อมูลเข้าฐานข้อมูล คลิกเมนู Insert และป้อนข้อมูล จากนั้นคลิกปุ่ม Go



ภาพที่ 7.16 การเพิ่มข้อมูลเข้าฐานข้อมูลด้วยโปรแกรม phpMyAdmin

หลังจากเพิ่มข้อมูลเข้าฐานข้อมูลเสร็จสมบูรณ์ สามารถตรวจสอบข้อมูลที่เพิ่มเข้าฐานข้อมูลโดยคลิกเมนู Browse จะปรากฏข้อมูลดังภาพที่ 7.17



ภาพที่ 7.17 แสดงข้อมูลในตาราง regstudent

หลังจากขั้นตอนการสร้างฐานข้อมูลและตารางเสร็จแล้ว ในการเพิ่ม แก้ไข หรือลบข้อมูลนอกจากทำในโปรแกรม phpMyAdmin โดยตรงแล้ว ผู้ดูแลระบบฐานข้อมูลสามารถทำงานบน Web Browser โดยการเขียนโปรแกรมเชื่อมต่อกับฐานข้อมูล MySQL ด้วยภาษา PHP ซึ่งจะนำเสนอในบทที่ 8 เรื่องการเขียนโปรแกรมภาษา PHP เชื่อมระบบฐานข้อมูล MySQL

ข้อดีของโปรแกรม phpMyAdmin

1. ทำงานได้ทุกระบบปฏิบัติการ เนื่องจากโปรแกรมอยู่ในรูปแบบของเว็บไซต์
2. สามารถใช้งานได้ฟรี
3. โปรแกรมมีประสิทธิภาพสูง

4. ความสามารถของโปรแกรมครบ เพียงพอต่อการใช้งานในระดับหนึ่ง
5. โปรแกรมมีขนาดเล็ก ทำงานได้รวดเร็ว และมีการอัพเดทต่อเนื่อง

สรุป

ระบบฐานจัดการฐานข้อมูล มีวัตถุประสงค์เพื่อช่วยให้การพัฒนาฐานและจัดการฐานข้อมูลให้สามารถใช้งานได้ง่าย มีความรวดเร็ว ข้อมูลมีความถูกต้องและทันสมัย และลดค่าใช้จ่ายในการสร้างดูแลและการบำรุงรักษา DBMS ทำให้ข้อมูลมีมาตรฐาน มีความปลอดภัยที่สูง มีระบบควบคุมความถูกต้อง มีความเป็นอิสระของข้อมูล ทำให้โปรแกรมเป็นอิสระจากโครงสร้างข้อมูลทางด้านกายภาพ และด้านตรรกภาพ รวมทั้งมีระบบสำรองข้อมูลและการฟื้นสภาพที่มีประสิทธิภาพ องค์ประกอบระบบฐานข้อมูลโดยทั่วไปจะแบ่งออกเป็น 5 ส่วน ได้แก่ ฮาร์ดแวร์ ซอฟต์แวร์ ข้อมูล บุคลากร และขั้นตอนการปฏิบัติงาน

MySQL เป็นระบบจัดการฐานข้อมูลโดยใช้ภาษา SQL โปรแกรม phpMyAdmin เป็นเครื่องมือที่สำคัญของ MySQL เพราะเป็นโปรแกรมที่คำสั่งแสดงในรูปแบบ GUI ทำให้ใช้งานสะดวกโดยไม่ต้องเขียนโค้ช SQL ในการสร้างฐานข้อมูล หน้าที่ของโปรแกรม phpMyAdmin เป็นตัวกลาง นำเข้ารหัส SQL (SQL Code) สู่ระบบจัดการฐานข้อมูล MySQL ที่ทำงานอยู่ในเบื้องหลังโดยอัตโนมัติ

บทที่ 8

การเขียนโปรแกรมภาษา PHP เชื่อมระบบฐานข้อมูล MySQL

ในอดีตภาษาที่นิยมใช้ในการพัฒนาเว็บไซต์ (Web Site) คือ ภาษา HTML (Hypertext Markup Language) โดยภายในหน้าเว็บเพจ (Web page) จะประกอบไปด้วย ตัวอักษร ภาพ หรือ อื่น ๆ ที่ไม่สามารถเปลี่ยนแปลงข้อมูลได้ด้วยตัวเอง ไม่สามารถคำนวณ ประมวลผลข้อมูล หรือ เชื่อมต่อกับฐานข้อมูลได้ เว็บไซต์ที่เขียนด้วยภาษา HTML เป็นแค่ส่วนของการแสดงผลของหน้าเว็บเพจเท่านั้น หรือเรียกว่า หน้าเว็บเพจแบบคงที่ (Static Web page) ต่อมานักพัฒนาโปรแกรมได้มีการพัฒนาภาษาที่สามารถนำเสนอข้อมูลที่ถูกเปลี่ยนแปลงโดยอัตโนมัติตามเงื่อนไขต่าง ๆ ที่ผู้เขียนกำหนดไว้ หรือเรียกว่าไดนามิกเว็บเพจ (Dynamic Web page) ในปัจจุบันมีหลายภาษาที่ถูกพัฒนาขึ้นมามีลักษณะเป็นภาษาสคริปต์ (Script Language) ที่สามารถติดต่อกับผู้ใช้ฐานข้อมูลเพื่อการคำนวณหรือประมวลผลตามเงื่อนไขต่าง ๆ ได้ และ 1 ในภาษาสคริปต์ที่ได้รับความนิยมในปัจจุบันก็คือ ภาษาพีเอชพี ในชื่อภาษาอังกฤษว่า PHP ซึ่งใช้เป็นคำย่อจากคำว่า PHP Hypertext Preprocessor หรือชื่อเดิม Personal Home Page ในบทนี้จะกล่าวถึงภาษา PHP เบื้องต้นสำหรับการเชื่อมต่อและประยุกต์ใช้งานร่วมกับระบบฐานข้อมูล MySQL โดยนำเสนอเนื้อหาสำหรับการศึกษาดังต่อไปนี้

- ♦ ความเป็นมาภาษา PHP
- ♦ หลักการทำงานของภาษา PHP
- ♦ โครงสร้างภาษา PHP
- ♦ การเขียนโปรแกรมภาษา PHP เข้าถึงฐานข้อมูล MySQL

ประวัติความเป็นมาของภาษา PHP

ภาษา PHP ถูกสร้างขึ้นในปี ค.ศ.1994 โดย Rasmus Lerdorf เป็นผู้เริ่มต้นพัฒนา จุดเริ่มต้นมาจากความต้องการที่จะบันทึกข้อมูลผู้ที่เข้ามาเยี่ยมชมเว็บไซต์ส่วนตัวที่สร้างขึ้น โดยแนวคิดคือการเขียนโปรแกรมด้วยภาษา C แต่ต้องการแยกส่วนที่เป็นภาษา HTML ออกจากภาษา C และเป็นเหตุผลสำคัญทำให้สร้างโค้ด (Code) ของภาษา HTML ขึ้นมาใหม่ และตั้งชื่อว่า Personal Home Page (PHP-Tools)

หลังจากพัฒนาภาษา PHP ขึ้นมาแล้วได้เผยแพร่ให้บุคคลทั่วไปใช้งาน แต่ในช่วงแรกภาษา PHP ยังไม่มีความสามารถมากนัก ต่อมาในปี ค.ศ.1995 มีการพัฒนาเพิ่มขีดความสามารถให้กับภาษา PHP ให้สามารถรับข้อมูลที่ส่งมาจากฟอร์ม (Form) ของภาษา HTML รวมทั้งสามารถติดต่อกับฐานข้อมูล MySQL

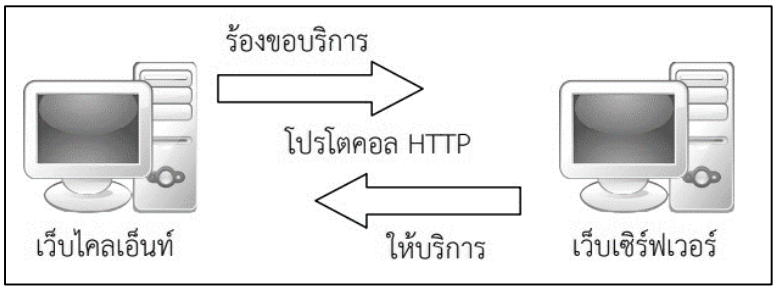
ในปี ค.ศ.1997 ได้เปิดให้ผู้สนใจเข้าร่วมพัฒนาภาษา PHP และได้ผู้ร่วมพัฒนาเพิ่มอีก 2 คน คือ Zeev Suraski และ Andi Gutmans ได้ช่วยกันปรับปรุงและพัฒนาโค้ดขึ้นมาใหม่ให้ดีขึ้นในหลาย ๆ ด้าน ทั้งแก้ไขข้อบกพร่อง เพิ่มประสิทธิภาพและเพิ่มฟังก์ชัน (Function) ใหม่ ๆ มากขึ้น และสามารถเขียนโปรแกรมเชิงวัตถุที่สมบูรณ์แบบ สามารถทำงานร่วมกับเว็บเซิร์ฟเวอร์บนระบบปฏิบัติการหลายชนิด จนเป็นที่นิยมในปัจจุบัน และเปลี่ยนชื่อใหม่จากเดิม Personal home page (PHP-Tools) เป็น PHP hypertext preprocessor

หลักการทำงานของภาษา PHP

ภาษา PHP นั้น ทำงานอยู่ในฝั่งเว็บเซิร์ฟเวอร์ จะไม่สามารถเห็นสคริปต์ของภาษา PHP ที่เครื่องคอมพิวเตอร์ในฝั่งไคลเอ็นท์ เพราะระหว่างที่ใช้งานเว็บบราวเซอร์เรียกชมเว็บไซต์นั้น ในฝั่งของเว็บเซิร์ฟเวอร์จะทำการประมวลผลภาษา PHP และส่งผลลัพธ์ที่ได้เป็นภาษา HTML หลังจากนั้นจะส่ง HTML ที่ได้จากการประมวลผลแล้วนั้นมายังเว็บบราวเซอร์ของไคลเอ็นท์ แล้วแสดงผลให้เห็นเป็นรูปร่างของเว็บไซต์ และเมื่อดูซอร์สโค้ด (Source Code) ในฝั่งไคลเอ็นท์จะเห็นเฉพาะ HTML tags ของภาษา HTML เพียงอย่างเดียว ไม่เห็นสคริปต์ PHP สามารถสรุปขั้นตอนและหลักการทำงานระหว่างไคลเอ็นท์และเว็บเซิร์ฟเวอร์ ได้ดังต่อไปนี้

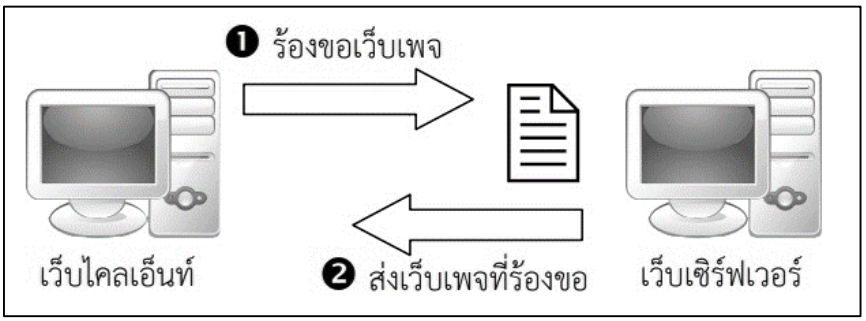
1. เว็บเซิร์ฟเวอร์ คือ โปรแกรมที่อยู่และทำงานบนเครื่องฝั่งเซิร์ฟเวอร์ ทำหน้าที่ในการรับคำสั่งจากการร้องขอของไคลเอ็นท์ (โดยผู้ใช้จะเรียกชมหน้าเว็บเพจได้โดยใช้โปรโตคอล HTTP ผ่านทางเว็บบราวเซอร์) และประมวลผลการทำงานจากการร้องขอดังกล่าว แล้วส่งข้อมูลกลับไปยังเครื่องของไคลเอ็นท์ที่ร้องขอ โดยสรุปคือเว็บเซิร์ฟเวอร์จะคอยให้บริการแก่ไคลเอ็นท์ที่ร้องขอข้อมูลเข้ามาโดยผ่านเว็บบราวเซอร์ หรือจากการร้องขอผ่านโปรโตคอล HTTP โดยเว็บเพจที่เขียนก็สามารถเขียนได้หลายภาษา หลายรูปแบบ การเขียนสคริปต์ PHP ระบบจะสามารถทำงานได้นั้นจำเป็นต้องมีเว็บเซิร์ฟเวอร์เป็นตัวเรียกใช้งาน Engine ของภาษา PHP ดังนั้น ถ้าต้องการให้คอมพิวเตอร์สามารถประมวลผลและแปลภาษา PHP ได้นั้น จำเป็นต้องมีเว็บเซิร์ฟเวอร์หรือกรณีศึกษานี้สามารถจำลองเครื่องคอมพิวเตอร์ธรรมดาให้ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ รวมถึงรูปแบบและวิธีการเข้าถึงข้อมูลจะเรียกใช้ผ่านบริการจากโปรโตคอล HTTP เพื่อให้ผู้ใช้สามารถอ่านข้อมูล ภาพ เสียง หรือข้อมูลอื่น ๆ ตามต้องการ

2. หลักการทำงานของเว็บเซิร์ฟเวอร์ มีลักษณะเช่นเดียวกันกับบริการอื่น ๆ บนระบบอินเทอร์เน็ต (Internet) คือ อยู่ในรูปแบบไคลเอ็นท์-เซิร์ฟเวอร์ (Client-Server) โดยมีโปรแกรมเว็บไคลเอ็นท์ (Web Client) ทำหน้าที่เป็นผู้ร้องขอบริการ และมีโปรแกรมเว็บเซิร์ฟเวอร์ ทำหน้าที่เป็นผู้ให้บริการ โปรแกรมเว็บไคลเอ็นท์ก็คือโปรแกรมเว็บบราวเซอร์ในเครื่องของผู้ใช้ สำหรับโปรแกรมเว็บ เซิร์ฟเวอร์นั้นจะถูกติดตั้งไว้ในเครื่องของผู้ให้บริการเว็บไซต์ (มักเรียกเครื่องคอมพิวเตอร์ที่ติดตั้งโปรแกรมเว็บเซิร์ฟเวอร์เพื่อให้บริการว่า “เว็บเซิร์ฟเวอร์”) การติดต่อระหว่างโปรแกรมเว็บบราวเซอร์กับโปรแกรม เว็บเซิร์ฟเวอร์จะกระทำผ่านโปรโตคอล HTTP ดังภาพที่ 8.1



ภาพที่ 8.1 รูปแบบการติดต่อสื่อสารระหว่างเว็บไคลเอ็นท์และเว็บเซิร์ฟเวอร์ผ่านโปรโตคอล HTTP

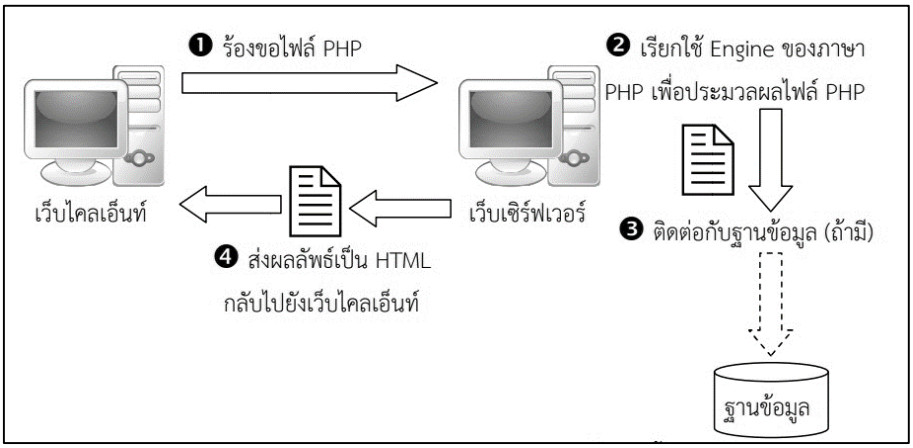
3. กลไกการทำงานเมื่อร้องขอเว็บเพจธรรมดา โดยปกติจะมีนามสกุลของเอกสารเป็น htm หรือ html นั้น เมื่อใช้เว็บไคลเอ็นท์เปิดดูหรือเรียกดูเว็บเพจใด เว็บเซิร์ฟเวอร์ก็จะส่งเว็บเพจนั้นกลับมายังเว็บไคลเอ็นท์ จากนั้นเว็บไคลเอ็นท์จะแสดงผลตามคำสั่งภาษา HTML ที่อยู่ในเอกสารนั้น ๆ



ภาพที่ 8.2 รูปแบบการติดต่อสื่อสารระหว่างเว็บไคลเอ็นท์และเว็บเซิร์ฟเวอร์เพื่อเรียกเว็บเพจธรรมดา

จะเห็นได้ว่าเว็บเพจ จากภาพที่ 8.2 เป็นเว็บเพจแบบคงที่ กล่าวคือ ผู้ใช้จะพบกับเว็บเพจหน้าตาเดิม ๆ ทุกครั้ง จนกว่าผู้ดูแลเว็บจะทำการปรับปรุงเว็บเพจนั้นใหม่ ภาษา HTML เป็นภาษาที่ใช้อธิบายหน้าตาของเว็บเพจ (ภาษา HTML จัดเป็นภาษาในกลุ่มที่เรียกว่า page description language) หรือ กล่าวอีกนัยหนึ่งคือ ภาษา HTML สามารถกำหนดให้เว็บเพจมีหน้าตาอย่างที่ต้องการได้ ในปัจจุบันมีการพัฒนาโปรแกรมที่ใช้ในการจัดการเว็บเพจให้มีความน่าสนใจมากยิ่งขึ้น

4. กลไกการทำงานเมื่อร้องขอเว็บเพจที่ใช้ภาษา PHP จะมีรูปแบบการประมวลผลและแสดงผลเป็นแบบไดนามิกเว็บเพจ สามารถติดต่อกับผู้ใช้ฐานข้อมูล คำนวณ และประมวลผลตามเงื่อนไขต่าง ๆ ได้ ลักษณะการประมวลผล ดังภาพที่ 10.3



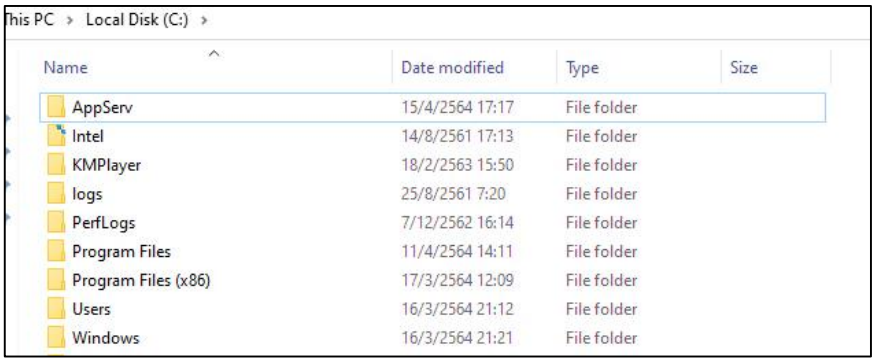
ภาพที่ 8.3 กระบวนการประมวลผลภาษา PHP

จากภาพที่ 10.3 เป็นกระบวนการประมวลผลภาษา PHP โดยเริ่มต้นเมื่อเว็บไคลเอ็นท์ร้องขอไฟล์ หรือเอกสารที่ใช้ภาษา PHP ในการพัฒนาร่วมกับภาษา HTML มายังเว็บเซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์จะเรียกใช้ Engine ของภาษา PHP หรือตัวแปลคำสั่งของภาษา PHP กรณีที่มีการใช้คำสั่งในการเชื่อมต่อฐานข้อมูลระบบก็จะติดต่อไปยังฐานข้อมูลตามการร้องขอ มีการตรวจสอบสิทธิ์การเข้าถึงฐานข้อมูลและอื่น ๆ ตามลำดับขั้นตอนวิธี ลักษณะการแปลคำสั่งของภาษา PHP นั้นจะแปลและเปลี่ยนคำสั่งจากภาษา PHP เป็นภาษา HTML แล้วส่งกลับไปแสดงผลลัพธ์ที่เว็บไคลเอ็นท์ที่ร้องขอ

การสร้างเว็บเซิร์ฟเวอร์ (Web server) บนคอมพิวเตอร์

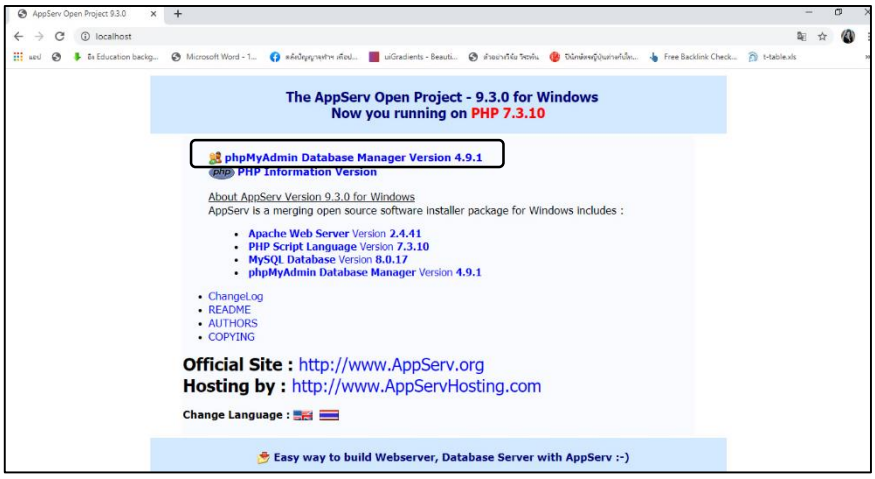
การแสดงผลจากการเขียนสคริปต์ PHP ได้นั้นจำเป็นต้องมีเว็บเซิร์ฟเวอร์เป็นตัวเรียกใช้งาน Engine ของภาษา PHP ปัจจุบันสามารถดาวโหลดชุดโปรแกรมสำหรับสร้างเว็บเซิร์ฟเวอร์บนเครื่องคอมพิวเตอร์ตั้งโต๊ะหรือโน๊ทบุค และรองรองรับการเขียนสคริปต์ PHP, การสร้างฐานข้อมูล MySQL ด้วยโปรแกรม phpMyAdmin เช่น AppServ หรือ XAMPP เป็นต้น โดยหนังสือเล่มนี้ผู้เขียนใช้ชุดโปรแกรม AppServ ในการสาธิตการติดตั้งโปรแกรม และทำงานต่าง ๆ ซึ่งได้นำเสนอในบทที่ 7 เรื่อง ระบบจัดการฐานข้อมูลด้วย MySQL ด้วย phpMyAdmin ดังนั้น จึงขอนำเสนอการใช้งานต่อจากการติดตั้งโปรแกรม AppServ ในบทที่ 7 ดังนี้

1. วิธีตรวจสอบว่าเว็บเซิร์ฟเวอร์ AppServ ติดตั้งสมบูรณ์และพร้อมใช้งานได้หรือไม่ ในกรณีที่ติดตั้งบนไดร์ C ให้ไปที่ไดร์ C ตรวจสอบมี Folder ชื่อ AppServ หรือไม่



ภาพที่ 8.4 การตรวจสอบ Folder AppServ

2. เปิด Web Brower พิมพ์ชื่อเว็บไซต์ (URL) ชื่อ localhost หรือ 127.0.0.1 Web Brower จะแสดง ดังนี้

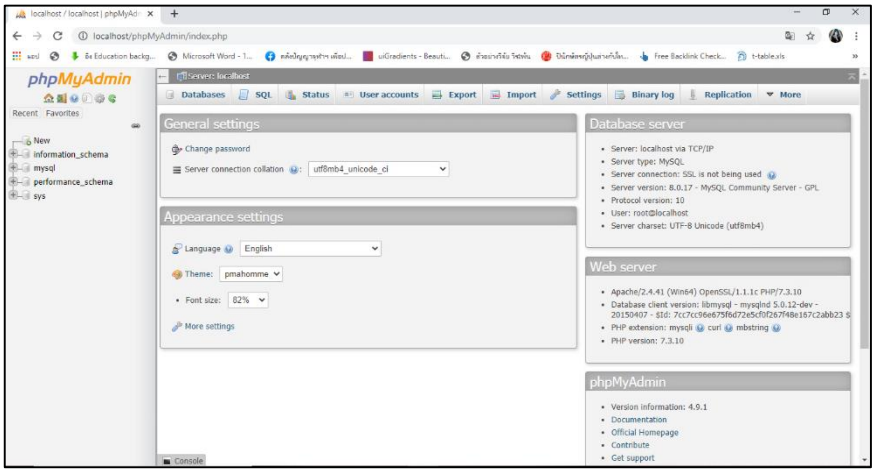


ภาพที่ 8.5 หน้าเวป localhost

3. ทดสอบการทำงานของโปรแกรม phpMyAdmin โดยเข้าลิงก์ phpMyAdmin Database Manager จะปรากฏช่องให้กรอก Username และ Password

ภาพที่ 8.6 หน้าเวปล๊อคอินเข้า phpMyAdmin

4. Username และ Password ถูกต้องจะปรากฏดังภาพที่ 10.5



ภาพที่ 8.7 โปรแกรม phpMyAdmin

5. เมื่อติดตั้งเว็บเซิร์ฟเวอร์ AppServ เรียบร้อยก็สามารถเขียนสคริปต์ PHP เพื่อติดต่อกับฐานข้อมูล โดยโปรแกรมสำหรับเขียนสคริปต์ PHP มีหลายโปรแกรมทั้งแบบมีลิขสิทธิ์และไม่มีลิขสิทธิ์ ในหนังสือเล่มนี้ผู้เขียนใช้โปรแกรมไม่มีลิขสิทธิ์ชื่อ Notepad++ ในการเขียนสคริปต์ PHP สามารถดาวโหลดที่ https://notepad-plus-plus.org/

6. เมื่อเขียนสคริปต์ PHP เสร็จให้บันทึกไฟล์ไว้ที่เว็บเซิร์ฟเวอร์ AppServ บันทึกไฟล์ .php ไปที่: C > AppServ > www

7. การเปิดเว็บแอพลิเคชั่นที่เขียนสคริปต์ PHP ให้ทำการเปิด Web Brower พิมพ์ช่อง URL ด้วยชื่อ localhost หรือ 127.0.0.1 ตามด้วยเครื่องหมาย / และตามด้วยชื่อไฟล์ที่ต้องการเรียกดู เช่น localhost/database.php หรือ 127.0.0.1/database.php

ภาษา PHP เบื้องต้น

หลังจากเรียนรู้การใช้งานโปรแกรม PhpMyAdmin เพื่อจัดการฐานข้อมูล MySQL ถึงแม้ว่าโปรแกรม PhpMyAdmin จะสามารถใช้จัดการฐานข้อมูล MySQL ได้ง่าย แต่ในทางปฏิบัติแล้ว ผู้ดูแลระบบคงไม่อนุญาตให้ผู้ใช้สามารถเข้ามาใช้โปรแกรม PhpMyAdmin เพื่อจัดการฐานข้อมูล MySQL เพราะมิฉะนั้นแล้วผู้ใช้อาจจะทำการเพิ่ม ลบ หรือ แก้ไขข้อมูลต่าง ๆ ในฐานข้อมูล โดยปกติในการจัดการฐานข้อมูลจะเป็นหน้าที่ของผู้ดูแลระบบ (Administrator) เท่านั้น วิธีแก้ปัญหาดังกล่าวสามรถเขียนโปรแกรมภาษา PHP เพื่อให้สามารถติดต่อและเข้าถึงฐานข้อมูล MysQL ดังนั้น จึงขอสรุปคำสั่งพื้นฐานของภาษา PHP เพื่อนำไปประยุกต์ใช้กับฐานข้อมูล MysQL ต่อไป

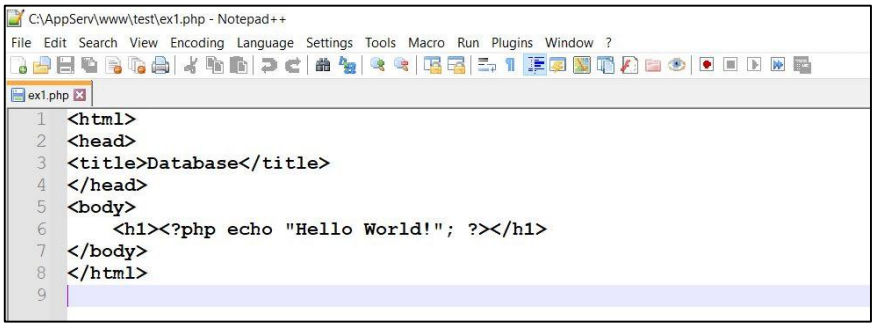
1. การแทรกคำสั่งภาษา PHP ในไฟล์ HTML

การแทรกคำสั่งภาษา PHP โดยการใช้ PHP tags ซึ่งมีไว้เพื่อเป็นการบ่งบอกให้รู้ส่วนที่เป็นคำสั่งของภาษา PHP ที่อยู่ในไฟล์ HTML การใช้ PHP tags นั้นสามารถทำได้หลายรูปแบบดังต่อไปนี้

- <?php ... ?> (XML style)

- <? ?> (short style)
- <script language=`php'>...</script> (Script style)
- <% ... %> (ASP style)

ตัวอย่างการแทรกคำสั่งภาษา PHP ในไฟล์ HTML



ผลการเขียน PHP ในไฟล์ HTML ดังนี้

ภาพที่ 8.8 ผลการเขียน PHP ในไฟล์ HTML

2. รูปแบบภาษา PHP และคำสั่ง

2.1 Comments

Comments ในภาษา PHP เอารูปแบบมาจาก comments ในภาษา C++ และภาษา XML Comments มีไว้เพื่อเป็นบันทึกช่วยให้ผู้เขียนหรือผู้อ่านโปรแกรมเข้าใจโปรแกรม

ตัวอย่างของ comments มีดังนี้

1) แบบบรรทัดเดียว มี 2 แบบ

// Comment

Comment

2) แบบหลายบรรทัด

/* Comment

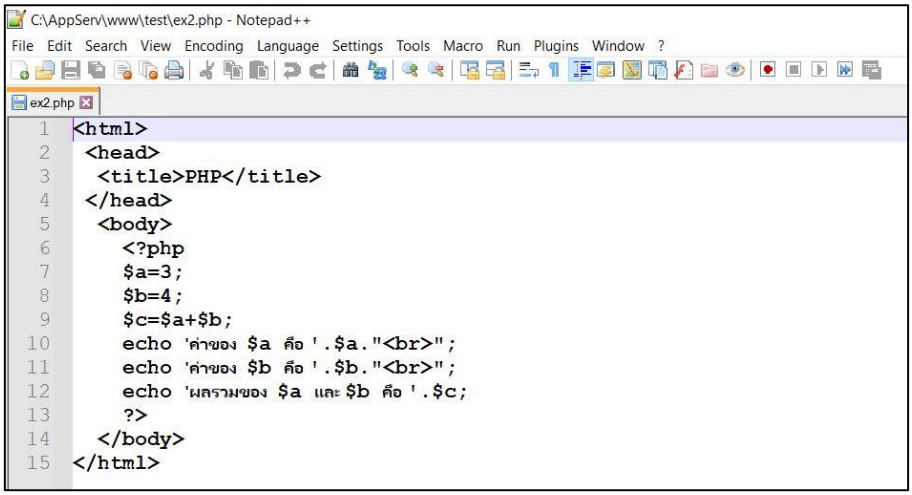
and Comment */.

3) <!-- file: hello.php (comments in xml style) -->

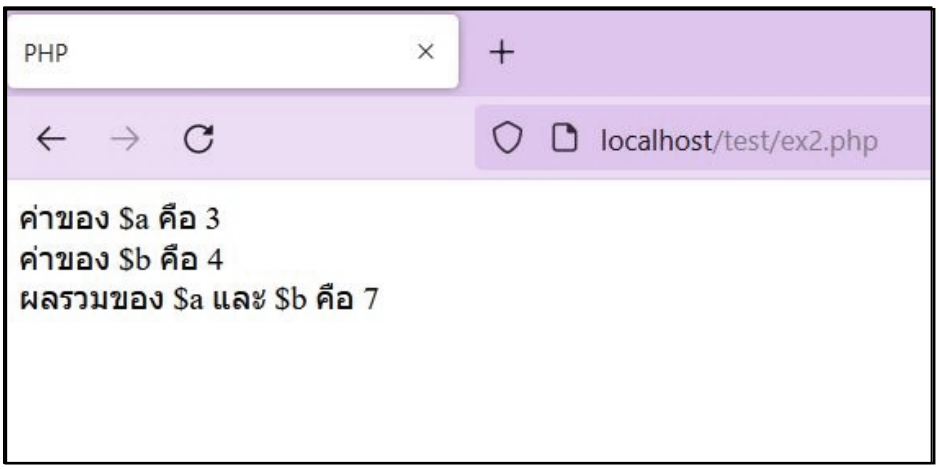
2.2 Variables (ตัวแปร)

ตัวแปรในภาษา PHP จะเหมือนกับในภาษา Perl คือเริ่มต้นด้วยเครื่องหมาย dollar ($) โดยเราไม่จำเป็นต้องกำหนดแบบของข้อมูล (data type) อย่างเจาะจงเหมือนในภาษา C เพราะว่า ตัวแปลภาษาจะจำแนกเองโดยอัตโนมัติว่า ตัวแปรดังกล่าว ใช้ข้อมูลแบบใด ในช่วงเวลานั้น ๆ เช่น ข้อความ จำนวนเต็ม จำนวนที่มีเลขจุดทศนิยม ตรรกะ เป็นต้น

ตัวอย่างการกำหนดตัวแปร



ผลลัพธ์ที่ได้จากการเขียนสคริปต์ PHP ดังนี้



ภาพที่ 8.9 ผลการเขียน PHP การกำหนดตัวแปร

2.3 ตัวดำเนินการ (Operators)

Operators ที่สามารถใช้ได้ในภาษา PHP มีดังนี้

1) ตัวดำเนินการทางคณิตศาสตร์

+ (บวก)
- (ลบ)
* (คูณ)
/ (หาร)
% (หาร เอาค่า เศษ)

2) ตัวดำเนินการ การกำหนดค่า

= (เช่น $num = 5 มีค่า $num = 5)
+= (เช่น $num += 5 มีค่า $num = $num + 5)
-= (เช่น $num -= 5 มีค่า $num = $num - 5)
*= (เช่น $num *= 5 มีค่า $num = $num * 5)
/= (เช่น $num /= 5 มีค่า $num = $num / 5)
%= (เช่น $num %= 5 มีค่า $num = $num % 5)
&= (เช่น $num &= 5 มีค่า $num = $num & 5)
|= (เช่น $num |= 5 มีค่า $num = $num | 5)
^= (เช่น $num ^= 5 มีค่า $num = $num ^ 5)
.= (เช่น $num .= 5 มีค่า $num = $num . 5)

3) ตัวดำเนินการเพิ่มค่าหรือลดค่า

++$ ตัวแปร คือ การเพิ่มค่าตัวแปรขึ้น 1 ก่อน แล้วจึงค่อยคืนค่า
$ ตัวแปร++ คือ คืนค่าตัวแปรก่อน แล้วจึงค่อยเพิ่มค่าตัวแปรขึ้น 1
--$ ตัวแปร คือ การลดค่าตัวแปรลง 1 ก่อน แล้วจึงค่อยคืนค่า
$ ตัวแปร-- คือ คืนค่าตัวแปรก่อน แล้วจึงค่อยลดค่าตัวแปรลง 1

4) ตัวดำเนินการเชิงเปรียบเทียบ

== (เท่ากับ)
!= (ไม่เท่ากับ)
<> (ไม่เท่ากับ)
> (มากกว่า)
>= (มากกว่าหรือเท่ากับ)
< (น้อยกว่า)
<= (น้อยกว่าหรือเท่ากับ)
=== (เท่ากันทุกประการ ทั้งค่าและชนิด)
!== (ไม่เท่ากันทุกประการ ทั้งค่าและชนิด)

5) ตัวดำเนินการเชิงตรรกะ

&& (and)
AND (and)
|| (or)
OR (or)
^ (xor)
! (not)

6) ตัวดำเนินการเชิงข้อความ

. (dot ใช้เชื่อมข้อความเข้าด้วยกัน)

7) ตัวดำเนินการเชิงอาร์เรย์

+ (บวก)

== (เท่ากับ คือ value เท่ากัน)

=== (เท่ากันทุกประการ คือ ทั้ง value และ index)

<> (ไม่เท่ากัน คือ value ไม่เท่ากัน)

!= (ไม่เท่ากัน คือ value ไม่เท่ากัน)

!== (ไม่เท่ากันทุกประการ คือ ทั้ง value และ index)

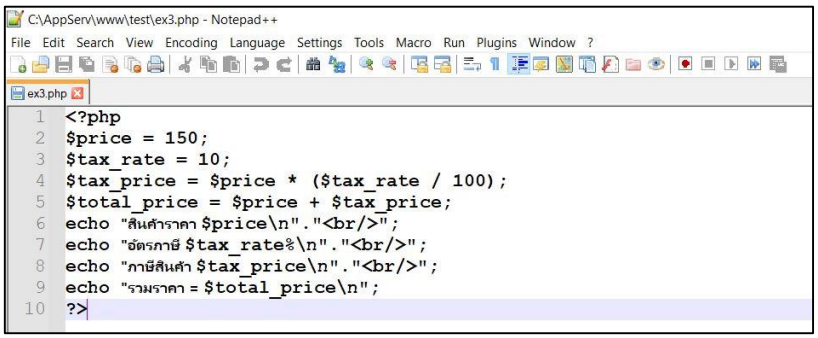
8) ตัวดำเนินการควบคุมความผิดพลาด

@ (เช่น นำไปวางหน้าตัวแปร เช่น @$number)

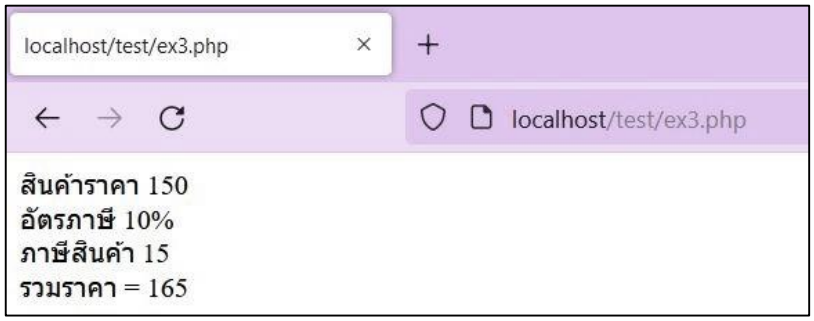
9) ตัวดำเนินการตรวจสอบ class ของ object

instanceof (เช่น $rooney instanceof Footballer
คือ ตรวจสอบว่า $rooney สร้างมาจาก class ที่กำหนดหรือไม่)

ตัวอย่างการเขียนสคริปต์ PHP ใช้ตัวดำเนินการ



ผลลัพธ์ที่ได้จากการเขียนสคริปต์ PHP ดังนี้



ภาพที่ 8.10 ผลการเขียน PHP ใช้ตัวดำเนินการ

2.4 คำสั่ง if/else

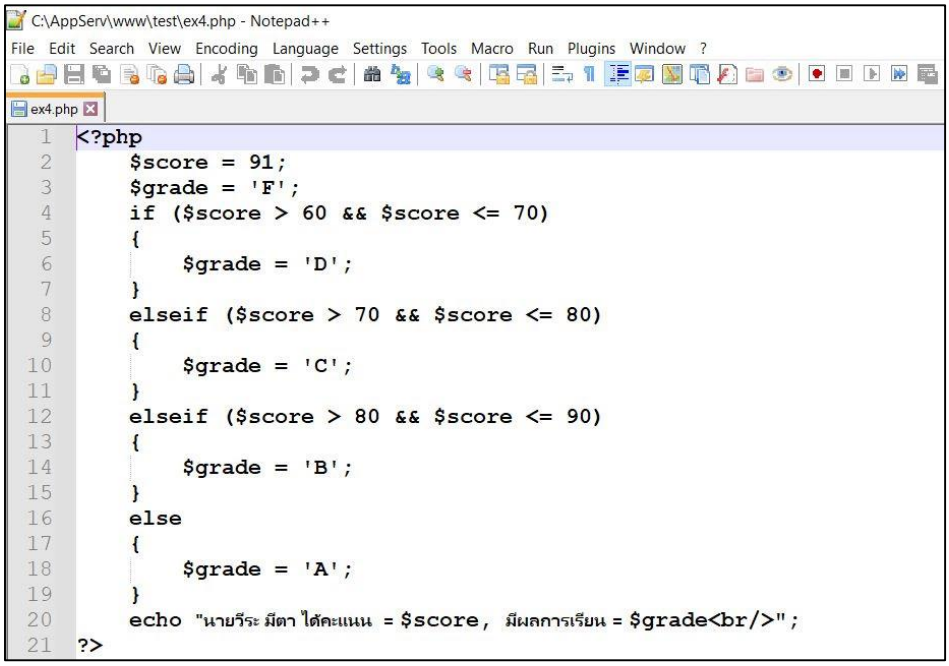
คำสั่ง if/else เป็นคำสั่งที่เลือกการทำงานตามเงื่อนไข รูปแบบของคำสั่งมีดังต่อไปนี้

1) if (เงื่อนไข) { คำสั่ง }

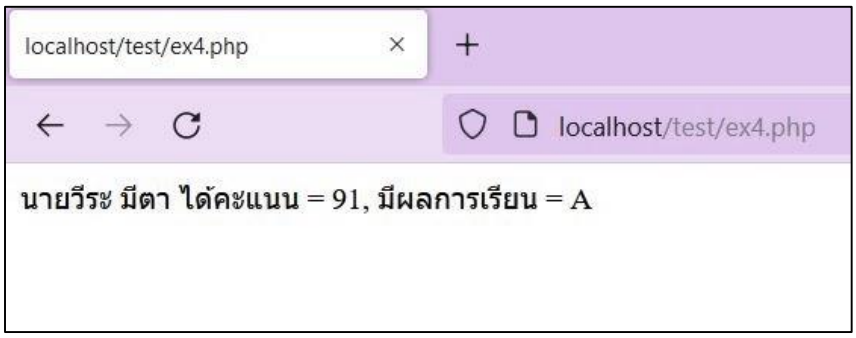
2) if (เงื่อนไข) { คำสั่ง } else { คำสั่ง }

3) if (เงื่อนไข) { คำสั่ง } elseif { คำสั่ง } else { คำสั่ง }

ตัวอย่างของการใช้คำสั่ง if/else



ผลลัพธ์ที่ได้จากการเขียนสคริปต์ PHP ดังนี้



ภาพที่ 8.11 ตัวอย่างผลของการใช้คำสั่ง if/else

2.5 คำสั่ง switch/case

คำสั่ง switch/case เป็นคำสั่งที่เลือกการทำงานตามเงื่อนไขเช่นเดียวกับคำสั่ง if/else แต่ว่า นิพจน์ของ if เป็น expression อย่างเช่น (a == 2) หรือ (a*b > 5) ได้ ในขณะที่นิพจน์ของ case ต้องเป็นค่าคงที่ที่เป็นจำนวนเต็มเท่านั่น (อย่างเช่น 2, ‘a’, 2.7 แต่ไม่ใช่ 2.5) และตรวจสอบเฉพาะว่าตัวแปรมีค่าเท่ากับจำนวนเต็มที่ให้มาหรือไม่ (เช่น ไม่สามารถเช็คได้ว่า a > 2 หรือไม่) อีกประการหนึ่ง คำสั่งใน witch ถ้าหากว่าถูกกระทำแล้วจะหยุดก็ต่อเมื่อเจอ break

ตัวอย่างการใช้คำสั่ง switch/case

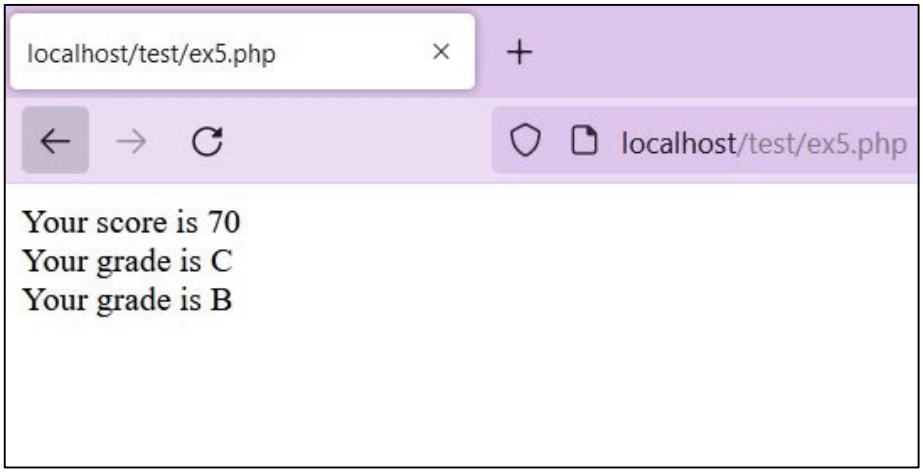
C:\AppServ\www\test\ex5.php - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

ex5.php

```php
<?php
    $score = 70;
    $grade = 'F';
    echo "Your score is $score<br/>";
    switch ($score)
    {
        case 60:
            echo "Your grade is D<br/>";
        case 70:
            echo "Your grade is C<br/>";
        case 80:
            echo "Your grade is B<br/>";
            break;
        case 90:
            echo "Your grade is A<br/>";
    }
?>
```

ผลลัพธ์ที่ได้จากการเขียนสคริปต์ PHP ดังนี้



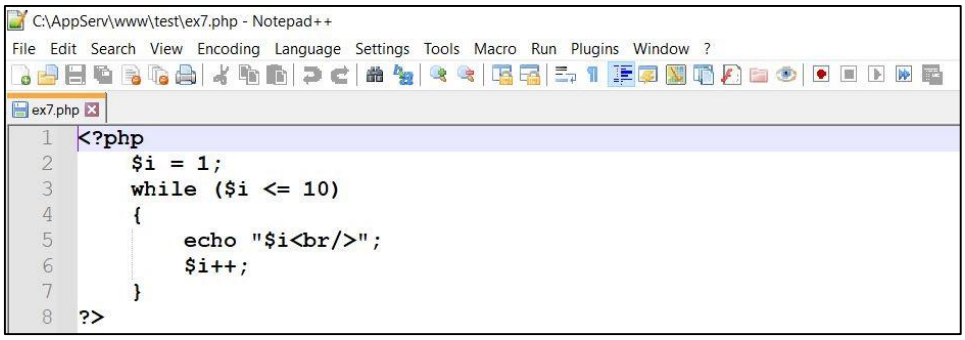
ภาพที่ 8.12 ตัวอย่างผลของใช้คำสั่ง Switch/ Case

2.6 คำสั่ง while

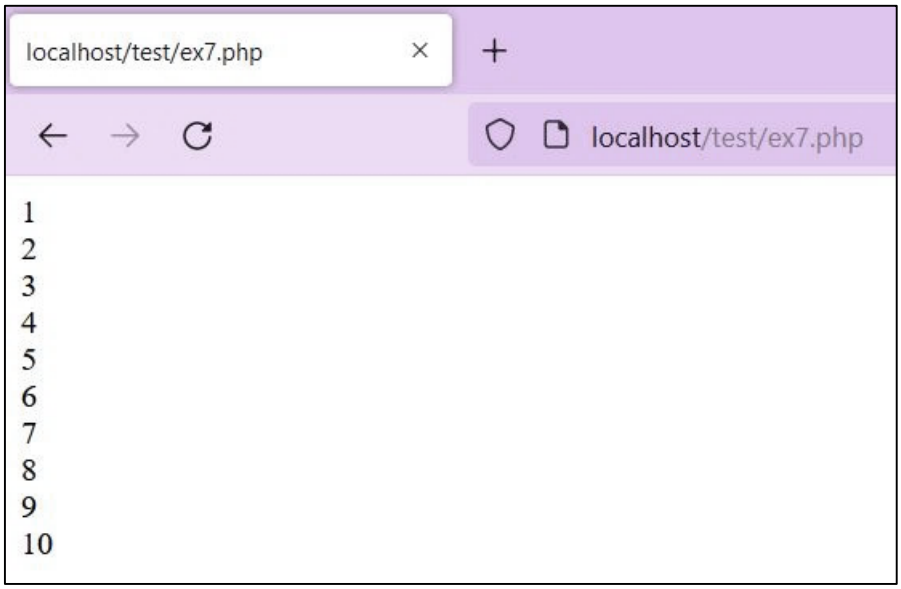
คำสั่ง while เป็นคำสั่งวนลูปตามเงื่อนไข รูปแบบของคำสั่งคือ

while (เงื่อนไข) { คำสั่ง }

ตัวอย่างการใช้คำสั่ง while



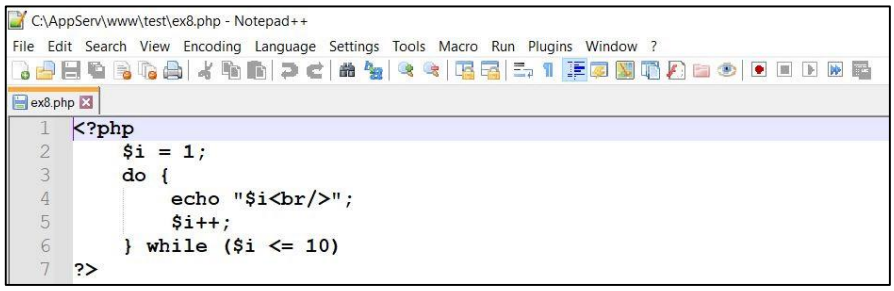
ผลลัพธ์ที่ได้จากการเขียนสคริปต์ PHP ดังนี้



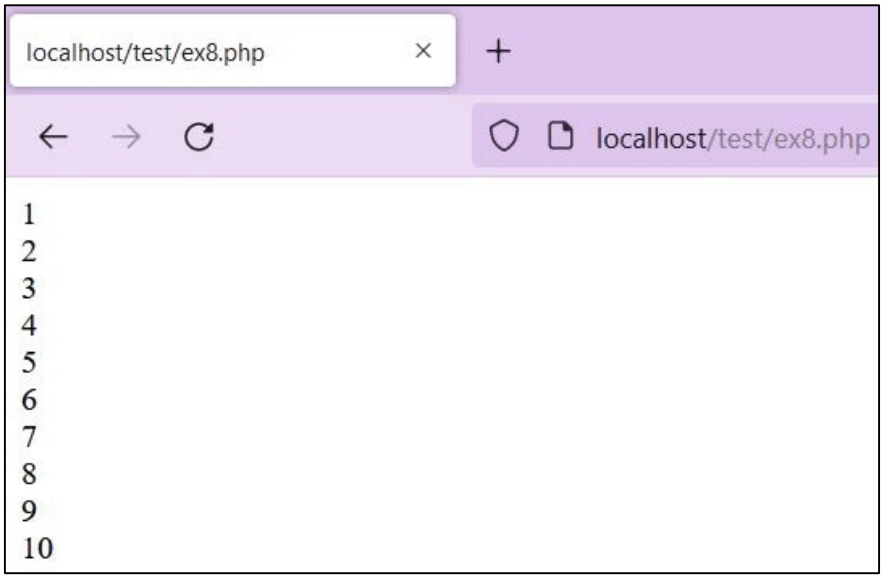
ภาพที่ 8.13 ตัวอย่างผลของใช้คำสั่ง while

2.7 คำสั่ง do/while

คำสั่ง do/while เป็นคำสั่งวนลูป ทำงานตามเงื่อนไข รูปแบบของคำสั่งคือ do { คำสั่ง } while (เงื่อนไข)



ผลลัพธ์ที่ได้จากการเขียนสคริปต์ PHP ดังนี้

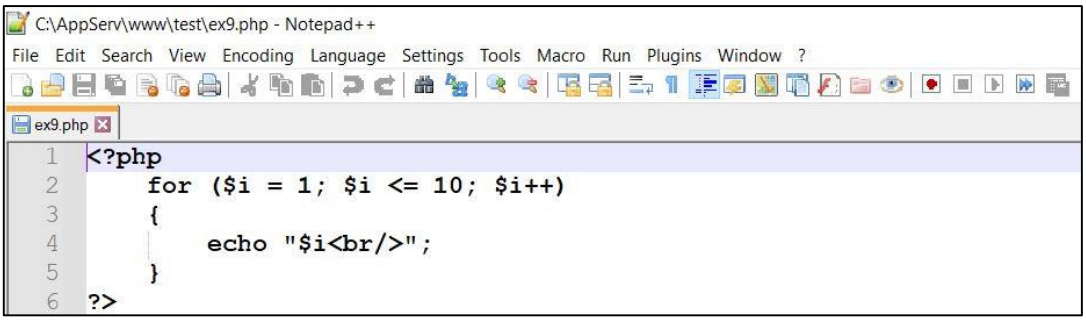


ภาพที่ 8.14 ตัวอย่างผลของใช้คำสั่ง do/while

2.8 คำสั่ง for

เป็นคำสั่งที่เลือกการทำงานตามเงื่อนไข รูปแบบของคำสั่งคือ

for (ค่าเริ่มต้น; ค่าสิ้นสุด; เงื่อนไข) { คำสั่ง }



ผลลัพธ์ที่ได้จะเป็นเช่นเดียวกับคำสั่ง do/while ตามภาพที่ 8.14

การเขียนโปรแกรม PHP เข้าถึงฐานข้อมูล MySQL

การเขียนโปรแกรมเพื่อเข้าถึงฐานข้อมูล MySQL ด้วยภาษา PHP มีความสำคัญต่อผู้ดูแลระบบ และผู้นำข้อมูลไปใช้ นอกจากนั้นทำให้ฐานข้อมูลมีความปลอดภัยขึ้น เนื่องจากสามารถเข้าถึงฐานข้อมูลฝ่านเว็บไซต์บนอุปกรณ์ต่าง ๆ ได้ เช่น คอมพิวเตอร์ตั้งโต๊ะ คอมพิวเตอร์โน้ตบุ๊ก แท็บเล็ต ไอแพด หรือสมาร์ทโฟน เป็นต้น

จากบทที่ 7 ได้นำเสนอเกี่ยวกับการสร้างฐานข้อมูลด้วย MySQLด้วย phpMyAdmin จะพบว่า ปกติสามารถสร้างฐานข้อมูลและตารางจากโปรแกรม phpMyAdmin ได้โดยตรงและมีความสะดวกต่อผู้สร้างฐานข้อมูล แต่นอกจากการสร้างฐานข้อมูลด้วยโปรแกรม phpMyAdmin

แล้ว เราสามารถเขียนโปรแกรมภาษา PHP เพื่อสร้างฐานข้อมูลได้เช่นเดียวกัน คำสั่งที่ใช้ คือ mysqli_query0 โดยคำสั่งนี้จะรับ SQL แล้วนำไปประมวลผลต่อไป

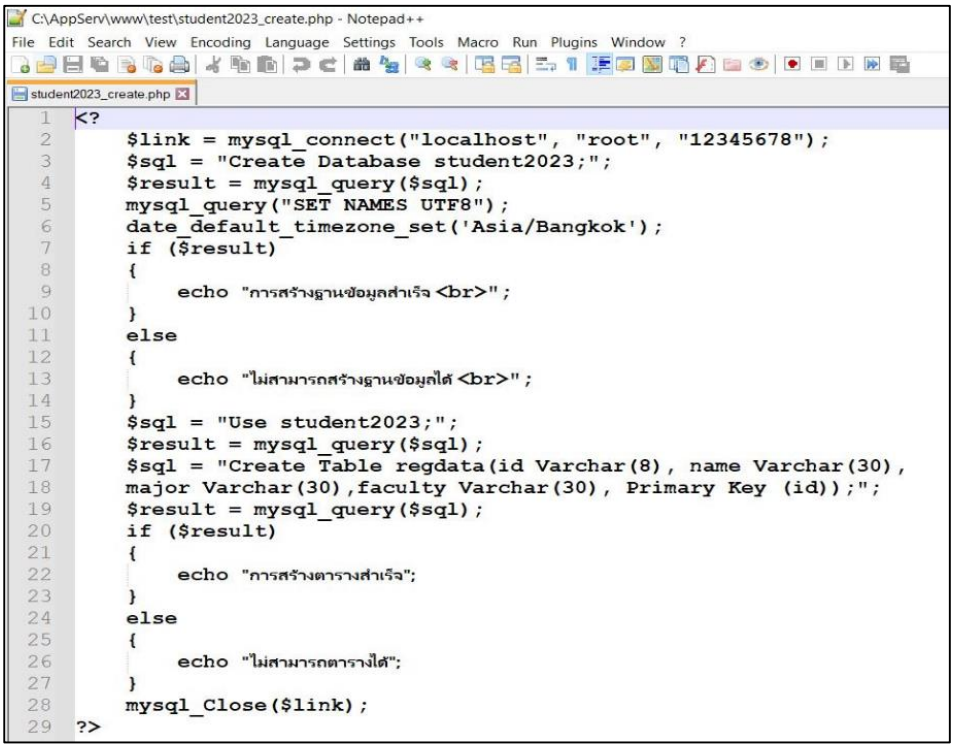
กรณีตัวอย่าง

1. การสร้างฐานข้อมูลและตารางด้วยโปรแกรมภาษา PHP

ในกรณีที่เราต้องการสร้างฐานข้อมูลชื่อ student2023 และฐานข้อมูลมีตารางชื่อ regdata ซึ่งมีรายละเอียดตารางเก็บข้อมูล คือ

ชื่อ Attribute	ชนิดข้อมูล	ความยาวข้อมูล	กำหนดเพิ่มเติม
id	INT	8	คีย์หลัก
name	VACHARE	30	
major	VACHARE	30	
faculty	VACHARE	30	

1.1 ใช้โปรแกรม Text editor เช่น Visual Studio Code หรือ Notepad++ เขียนโปรแกรมภาษา PHP สำหรับสร้างฐานข้อมูล กรณีตัวอย่างใช้โปรแกรม Notepad++ เขียนโค้ดโปรแกรมเพื่อสร้างฐานข้อมูล โดยประยุกต์โค้ดโปรแกรมจาก (ชาญชัย ศุภอรรถกร, 2561) ดังนี้

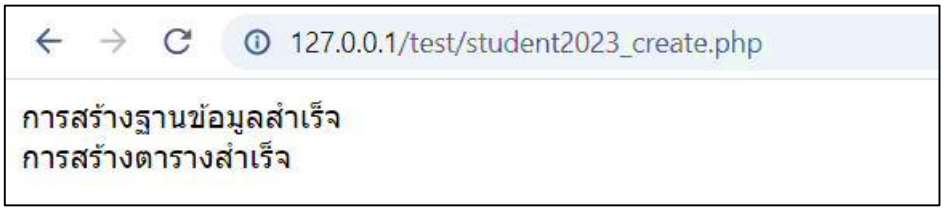


ภาพที่ 8.15 โค้ดภาษา PHP สร้างฐานข้อมูลและตาราง

1.2 เมื่อเขียนโปรแกรมเสร็จสมบูรณ์ ให้บันทึกไฟล์โปรแกรมนามสกุล .php ในกรณีตัวอย่างตั้งชื่อเป็น student2023_create (สามารถตั้งชื่ออื่นได้แต่ต้องระบุนามสกุลไฟล์เป็น .php) โดยบันทึกไฟล์ student2023_create.php ไปเก็บไว้ในเซิฟเวอร์ (Server) จำลองบนคอมพิวเตอร์ โดยเปิดไดร์ C ต่อด้วยโฟลเด้อ AppServ ต่อด้วยโฟลเด้อ www หรือ (C > AppServ > www)

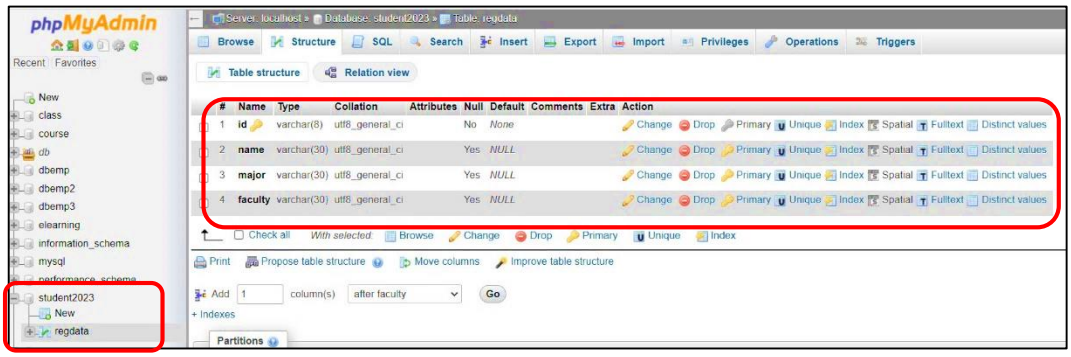
ภายใต้โฟลเด้อ www เราสามารถวางไฟล์ได้โดยตรง หรือสร้างโฟลเด้อย่อยอีก 1 ชั้น เพื่อการจัดเก็บไฟล์งานเป็นหมวดหมู่ ดังกรณีตัวอย่างภายใต้โฟลเด้อ www มีการสร้างโฟลเด้อ test สำหรับเก็บไฟล์ ดังนั้น ไฟล์โปรแกรมจะถูกจัดเก็บที่ (C > AppServ > www > test)

1.3 เมื่อบันทึกไฟล์เรียบร้อย ให้เปิด web browser และ เรียก URL: 127.0.01/test/student2023_create.php หรือ localhost/test/student2023_create.php กรณีที่เขียนโค้ดโปรแกรมถูกต้องหน้าเว็บจะแสดงข้อความดังนี้



ภาพที่ 8.16 แสดงผลการสร้างฐานข้อมูลและตาราง

1.4 การตรวจสอบว่า โปรแกรมทำการสร้างฐานข้อมูลได้ตามที่ออกแบบหรือไม่ ในเบื้องต้นให้เปิดโปรแกรม phpMyAdmin ตรวจสอบมีชื่อฐานข้อมูลที่สร้างในรายการฐานข้อมูลหรือไม่ และตรวจสอบตารางภายใต้ฐานข้อมูล ดังนี้



ภาพที่ 8.17 แสดงฐานข้อมูลและตาราง

2. การเขียนโปรแกรม PHP เพื่อเข้าถึงฐานข้อมูลและตาราง

เมื่อทำการสร้างฐานข้อมูลเสร็จสมบูรณ์ เราสามารถเขียนโปรแกรมด้วยภาษา PHP เพื่อเชื่อมโยงกับฐานข้อมูลเพื่อดำเนินการตามวัตถุประสงค์ที่ต้องการ เช่น คำสั่งนำข้อมูลเข้าตาราง คำสั่งแสดงรายการข้อมูล คำสั่งค้นหาข้อมูล คำสั่งแก้ไขข้อมูล คำสั่งลบข้อมูลในตาราง เป็นต้น

การเขียนโปรแกรมภาษา PHP เชื่อมโยงกับฐานข้อมูล ควรสร้างหน้าเว็บ 1 หน้าเพื่อเป็นหน้าหลักของเว็บไซต์ระบบฐานข้อมูล การสร้างหน้าเว็บหลักสามารถเขียนด้วยภาษา HTML ดังนี้

2.1 เขียนหน้าเว็บหลักด้วยภาษา HTML

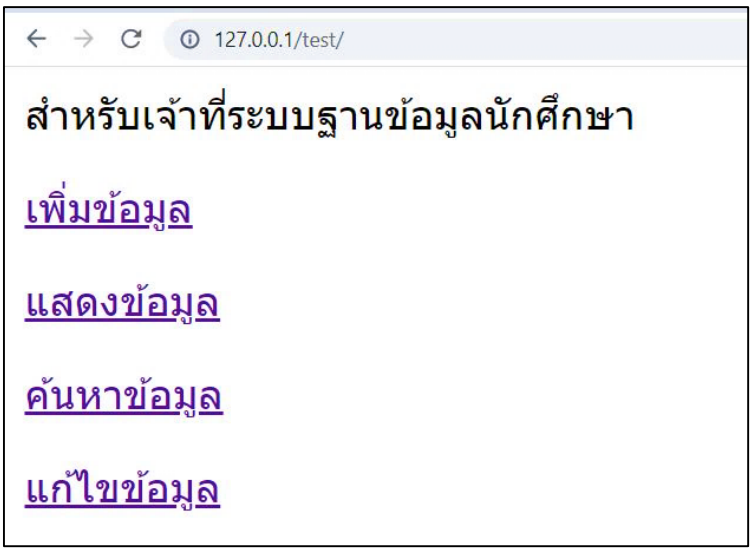
ตัวอย่างการสร้างหน้าเว็บหลักโดยการเขียนโค้ดด้วยภาษา HTML เพื่อเป็นหน้ารวมเมนูเข้าถึงฐานข้อมูล ดังนี้

index.html

```
1  <html>
2  <body>
3  สำหรับเจ้าที่ระบบฐานข้อมูลนักศึกษา<p>
4  <a href="insert.php">เพิ่มข้อมูล</a><p>
5  <a href="report.php">แสดงข้อมูล</a><p>
6  <a href="search.php">ค้นหาข้อมูล</a><p>
7  <a href="editdata.php">แก้ไขข้อมูล</a><p>
8  </body>
9  </html>
10
```

ภาพที่ 8.18 โค้ดภาษา HTML สร้างหน้าเว็บหลัก

เมื่อเขียนโค้ดด้วยภาษา HTML เสร็จสมบูรณ์ ให้บันทึกไฟล์เป็นนามสกุล .html ในกรณีตัวอย่างตั้งชื่อไฟล์เป็น index.html เราสามารถตั้งชื่อเว็บหน้าหลักเป็นชื่ออื่นได้ แต่การเปิดเว็บไซต์ต้องระบุชื่อไฟล์ต่อจากชื่อเวบไซต์ด้วย ดังนั้น ตามมาตรฐานควรตั้งชื่อไฟล์หน้าแรกของเว็บไซต์ชื่อ index และการเรียกดูผ่าน we browser เมื่อป้อนชื่อเว็บไซต์ระบบจะแสดงหน้าเว็บ index.html โดยอัตโนมัติ เช่น กรณีนี้บันทึกไฟล์ index.html ที่ (C > AppServ > www > test) web browser และเรียกดูด้วย URL: 127.0.01/test/ หรือ localhost/test/ กรณีที่เขียนโค้ด HTML ถูกต้องหน้าเว็บจะแสดงข้อความดังนี้

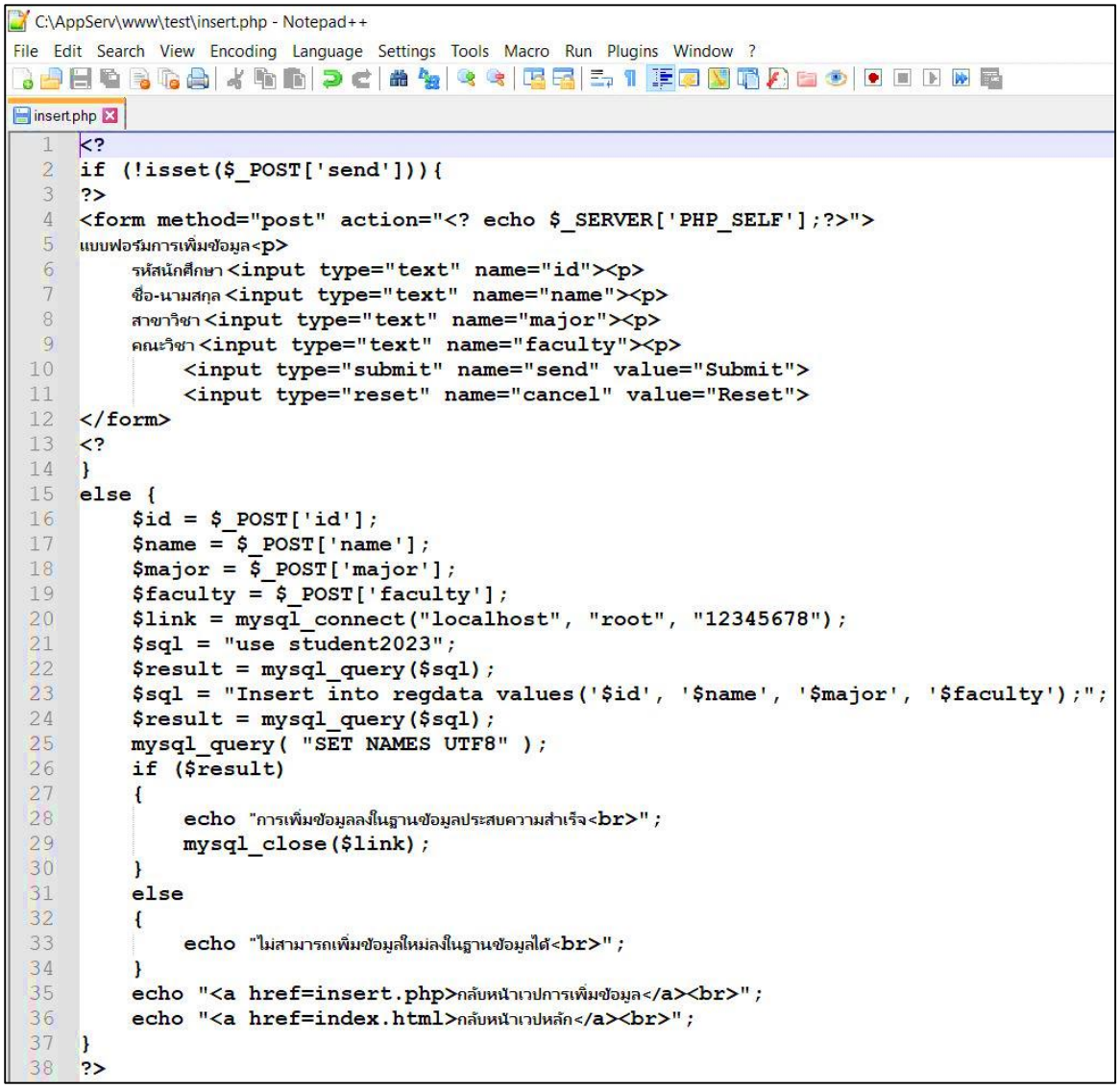


ภาพที่ 8.19 แสดงหน้าเว็บหลัก

2.2 การเขียนโปรแกรมด้วยภาษา PHP สำหรับเพิ่มข้อมูลเข้าฐานข้อมูล

ตัวอย่างการเขียนโปรแกรมเพื่อนำข้อมูลเข้าฐานข้อมูลชื่อ student2023 โดยเพิ่มข้อมูลเข้าตารางชื่อ regdata ซึ่งเป็นฐานข้อมูลที่ได้สร้างขึ้นตามขั้นตอนในข้อที่ 1 เรื่อง การสร้างฐานข้อมูลและตารางด้วยโปรแกรมภาษา PHP

การเพิ่มข้อมูลเข้าฐานข้อมูลผ่านหน้าเว็บ เราต้องเขียนโค้ดเพื่อสร้างหน้าเว็บแบบฟอร์มสำหรับป้อนข้อมูล และโค้ดสำหรับคำสั่งเชื่อมโยงฐานข้อมูลเพื่อการนำข้อมูลจากการป้อนผ่านเว็บบันทึกลงฐานข้อมูล กรณีนี้ตั้งชื่อไฟล์ insert.php โดยประยุกต์โค้ดโปรแกรมจาก (ชาญชัย ศุภอรรถกร, 2561) โค้ดโปรแกรม PHP ดังนี้


C:\AppServ\www\test\insert.php - Notepad++

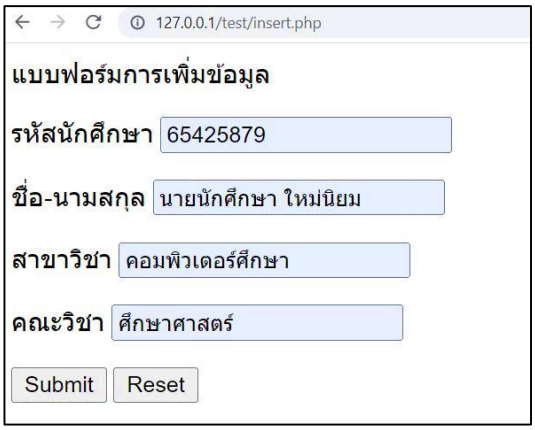
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

insert.php

```
<?
if (!isset($_POST['send'])){
?>
<form method="post" action="<? echo $_SERVER['PHP_SELF'];?>">
แบบฟอร์มการเพิ่มข้อมูล<p>
    รหัสนักศึกษา <input type="text" name="id"><p>
    ชื่อ-นามสกุล <input type="text" name="name"><p>
    สาขาวิชา <input type="text" name="major"><p>
    คณะวิชา <input type="text" name="faculty"><p>
        <input type="submit" name="send" value="Submit">
        <input type="reset" name="cancel" value="Reset">
</form>
<?
}
else {
    $id = $_POST['id'];
    $name = $_POST['name'];
    $major = $_POST['major'];
    $faculty = $_POST['faculty'];
    $link = mysql_connect("localhost", "root", "12345678");
    $sql = "use student2023";
    $result = mysql_query($sql);
    $sql = "Insert into regdata values('$id', '$name', '$major', '$faculty');";
    $result = mysql_query($sql);
    mysql_query( "SET NAMES UTF8" );
    if ($result)
    {
        echo "การเพิ่มข้อมูลลงในฐานข้อมูลประสบความสำเร็จ<br>";
        mysql_close($link);
    }
    else
    {
        echo "ไม่สามารถเพิ่มข้อมูลใหม่ลงในฐานข้อมูลได้<br>";
    }
    echo "<a href=insert.php>กลับหน้าเวปการเพิ่มข้อมูล</a><br>";
    echo "<a href=index.html>กลับหน้าเวปหลัก</a><br>";
}
?>
```

ภาพที่ 8.20 โค้ดภาษา PHP สร้างโปรแกรมเพิ่มข้อมูลเข้าฐานข้อมูล

เมื่อเขียนโปรแกรมเสร็จสมบูรณ์ และบันทึกไฟล์ insert.php ใน server เรียบร้อย ให้เรียกดูหน้าเว็บผ่าน web browser ด้วยURL: 127.0.01/test/insert.php หรือ localhost/test/insert.php จะปรากฏหน้าเว็บดังภาพที่ 8.21 นี้

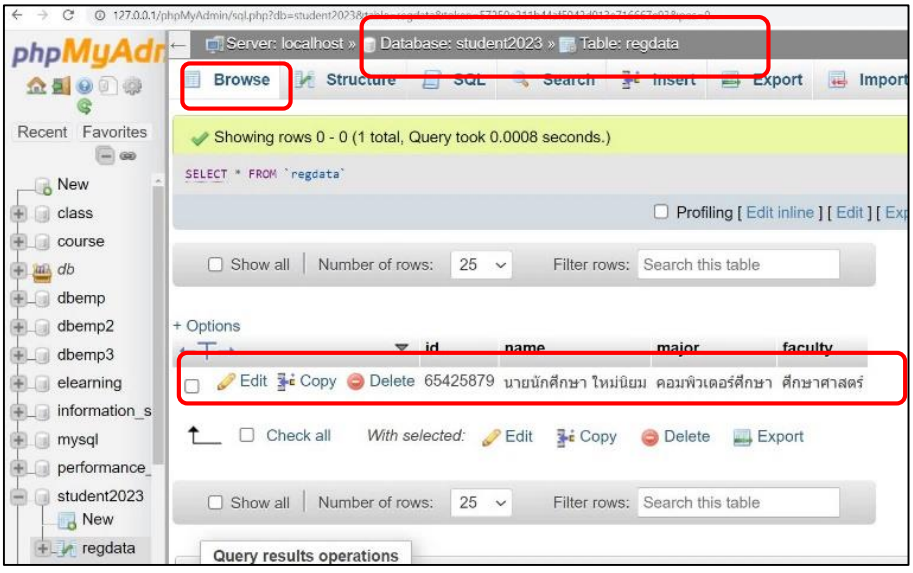


ภาพที่ 8.21 หน้าเว็บโปรแกรมเพิ่มข้อมูลเข้าฐานข้อมูล

ทดสอบโดยการป้อนข้อมูลตามแบบฟอร์มการเพิ่มข้อมูล เมื่อป้อนข้อมูลแล้วกดปุ่ม Submit เพื่อส่งข้อมูลเข้าฐานข้อมูล หรือปุ่ม Reset เพื่อล้างข้อมูลในแบบฟอร์ม (ไม่เกี่ยวข้องกับข้อมูลที่มีอยู่ในฐานข้อมูล) ถ้าเขียนโค้ดโปรแกรมถูกต้อง ทำให้โปรแกรมสามารถรับคำสั่งได้สมบูรณ์ จะแสดงข้อความหน้าเว็บดังนี้

ภาพที่ 8.22 หน้าเว็บแสดงผลการเพิ่มข้อมูลเข้าฐานข้อมูล

การเข้าตรวจสอบข้อมูลเบื้องต้น สามรถเข้าดูผ่านโปรแกรม phpMyAdmin โดยตรง ซึ่งภายในตาราง regdata จะมีข้อมูลเพิ่มเข้ามาตามการส่งจากแบบฟอร์มหน้าเว็บ



ภาพที่ 8.23 แสดงข้อมูลในโปรแกรม phpMyAdmin

2.3 การเขียนโปรแกรมด้วยภาษา PHP เพื่อแสดงข้อมูล

การแสดงรายการข้อมูลในฐานข้อมูล MySQL สามารถเขียนโปรแกรมภาษา PHP เพื่อแสดงบนหน้าเว็บได้ ซึ่งจะทำให้ผู้ใช้งานมีความสะดวกในการดูข้อมูล ตัวอย่างการเขียนโปรแกรมเพื่อแสดงข้อมูลโดยตั้งชื่อไฟล์ dataview.php โค้ดโปรแกรม ดังนี้

C:\AppServ\www\test\dataview.php - Notepad++

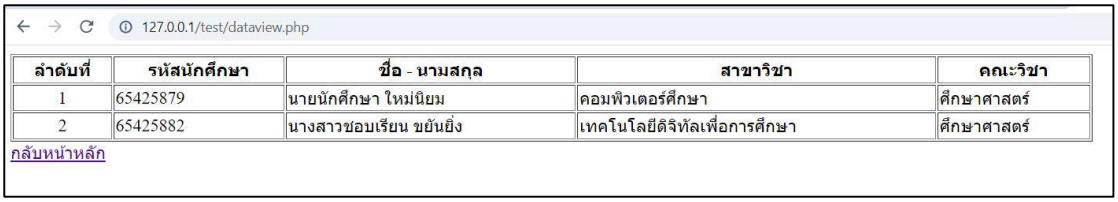
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

dataview.php

```
<?php
    header("Content-Type: text/html; charset=UTF-8");
     $host="localhost";
    $user="root"; // MySql ชื่อ user พิมพ์ลงแทน xxxx
    $pass="12345678"; // MySql Password พิมพ์ลงแทน xxxx
    $dbname="student2023"; // Database Name พิมพ์ลงแทน xxxx
    $conn=mysql_connect($host,$user,$pass) or die("ไม่สามารถเชื่อมต่อฐานข้อมูลได้"); // เชื่อมต่อ ฐานข้อมูล
    mysql_select_db($dbname,$conn); // เลือกฐานข้อมูล
    mysql_query("SET NAMES utf8"); // กำหนด charset ให้ฐานข้อมูล เพื่ออ่านภาษาไทย
 ?>
<html>
<head>
    <title>studentdata</title>
</head>
<body>
<table border="1" width="1000">
    <tbody>
        <tr>
            <th>ลำดับที่</th>
            <th>รหัสนักศึกษา</th>
            <th>ชื่อ - นามสกุล</th>
            <th>สาขาวิชา</th>
            <th>คณะวิชา</th>
           </tr>
        <?php
            $sql="Select * From regdata"; // คำสั่ง sql อ่านข้อมูลจากตาราง tbl_name พิมพ์ลงแทน xxxx
            $result=mysql_query($sql); // คิวรี่คำสั่ง sql
            $num=mysql_num_rows($result); // ตรวจสอบจำนวน record ที่คิวรี่ออกมา
            if($num>0){ // ถ้าจำนวน record มากกว่า 0
                $count=1; // กำหนดตัวแปร count เพื่อระบุตำแหน่ง record
                while($recordset=mysql_fetch_assoc($result)){ // วน loop ดึงข้อมูลออกมา ทีละ record
        ?>
        <tr>
            <td align="center"><?php echo $count; ?></td>
            <td><?php echo $recordset['id']; ?></td>
            <td><?php echo $recordset['name']; ?></td>
            <td><?php echo $recordset['major']; ?></td>
            <td><?php echo $recordset['faculty']; ?></td>
          </tr>
        <?
                    $count+=1; // เพิ่ม count ทีละ 1
                }
            }
        ?>
    </tbody>
</table>
<a href="index.html">กลับหน้าหลัก</a><p>
</body>
</html>
```

ภาพที่ 8.24 โค้ดภาษา PHP สร้างโปรแกรมแสดงข้อมูล

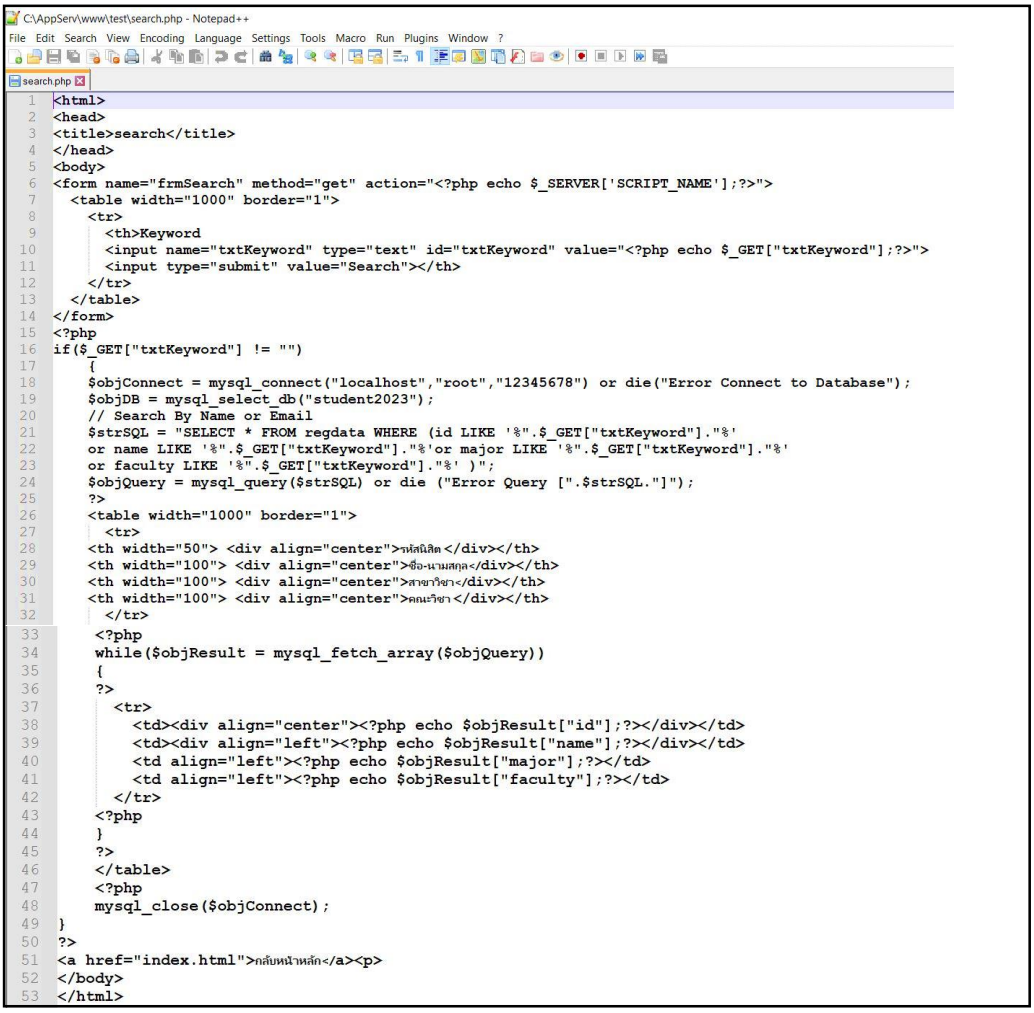
เมื่อเขียนโปรแกรมเสร็จสมบูรณ์ และบันทึกไฟล์ dataview.php ใน server เรียบร้อย ให้เรียกดูหน้าเว็บผ่าน Web browser ด้วยURL: 127.0.01/test/dataview.php หรือ localhost/test/dataview.php จะปรากฏหน้าเว็บดังนี้



ภาพที่ 8.25 หน้าเว็บผลการสร้างโปรแกรมแสดงข้อมูล

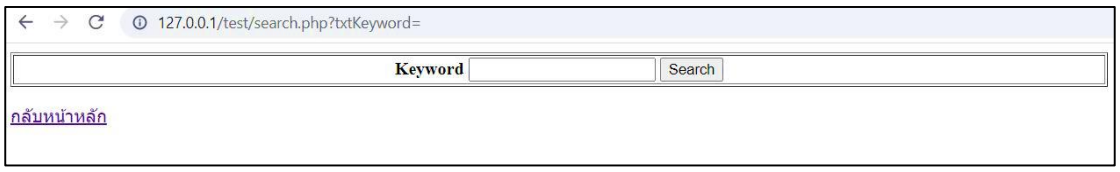
2.3 การเขียนโปรแกรมด้วยภาษา PHP เพื่อค้นหาข้อมูล

เมื่อฐานข้อมูลมีการจัดเก็บข้อมูลจำนวนมากขึ้น จึงเป็นเรื่องยากที่ค้นหาข้อมูลจากหน้าเว็บที่นำเสนอแบบปกติ เพื่อแก้ปัญหานี้เราสามารถเขียนโปรแกรมด้วยภาษา PHP เพื่อค้นหาข้อมูลที่เราต้องการ ตัวอย่างการเขียนโปรแกรมค้นหาข้อมูลโดยตั้งชื่อโปรแกรม search.php ในโปรแกรมจะประกอบด้วยการสร้างหน้าเว็บแสดงแบบฟอร์มสำหรับป้อนข้อความที่ต้องการค้นหา ภายในโปรแกรมกำหนดให้สามารถค้นหาจากรหัสนักศึกษา หรือ ชื่อ-นามสกุล หรือ สาขาวิชา หรือ คณะวิชา โดยป้อนตัวอักษร หรือตัวเลขที่ต้องการ แล้วกดปุ่ม Search ถ้ามีข้อมูลสัมพันธ์กับคำค้นหา โปรแกรมจะแสดงข้อมูลออกมาทั้งหมด กรณีที่คำค้นไม่สัมพันธ์กับข้อมูลใดในฐานข้อมูลจะไม่มีรายการแดงบนหน้าเว็บ โค้ดโปรแกรม php ดังนี้



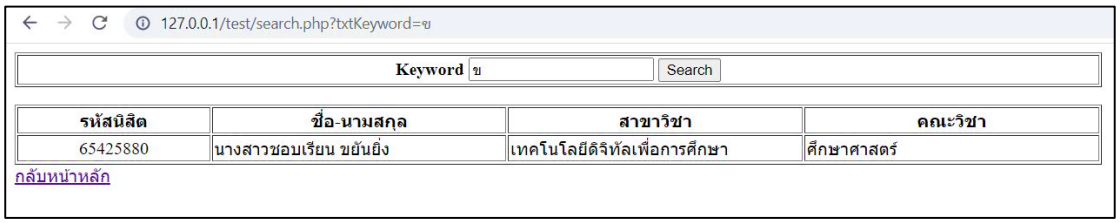
ภาพที่ 8.26 โค้ดภาษา PHP สร้างโปรแกรมค้นหาข้อมูล

เมื่อเขียนโปรแกรมเสร็จสมบูรณ์ และบันทึกไฟล์ searcht.php ใน server เรียบร้อย ให้เรียกดูหน้าเว็บผ่าน web browser ด้วยURL: 127.0.01/test/search.php หรือ localhost/test/search.php จะปรากฏหน้าเว็บดังนี้



ภาพที่ 8.27 หน้าเว็บโปรแกรมค้นหาข้อมูล

ทดสอบการค้นหาข้อมูล สมมติค้นด้วยตัวอักษร ข และ ข สัมพันธ์กับข้อมูลในฐานข้อมูล 1 รายการจึงแสดงผลการค้นหา ดังปรากฏข้อความบนหน้าเว็บดังนี้



ภาพที่ 8.28 หน้าเว็บแสดงผลการค้นหาข้อมูล

2.3 การเขียนโปรแกรมด้วยภาษา PHP เพื่อแก้ไขข้อมูลหรือลบข้อมูล

เมื่อต้องการแก้ไขรายละเอียดของข้อมูล หรือต้องการลบข้อมูลบางรายการที่เคยบันทึกไว้ในฐานข้อมูล สามารถเขียนโปรแกรมด้วยภาษา PHP เพื่อดำเนินการตามวัตถุประสงค์ดังกล่าว ในกระบวนการทำงานของโปรแกรม เมื่อทำการเปิดข้อมูลที่ต้องการแก้ไขและดำเนินการแก้ไขข้อมูลแล้ว ระบบจะนำข้อความใหม่จัดเก็บแทนที่ข้อมูลเดิม ส่วนการลบข้อมูลที่ไม่ต้องการออก เมื่อลบข้อมูลเรียบร้อยโปรแกรมจะทำการจัดเรียงข้อมูลใหม่โดยอัตโนมัติ รูปแบบการเขียนโค้ดโปรแกรมเพื่อแก้ไขหรือลบข้อมูลมีหลายรูปแบบ ตัวอย่างนี้เป็นวิธีการเขียนโค้ดโปรแกรมแบบผสมทั้ง 2 อย่าง เข้าในไฟล์เดียวกัน ดังนี้

C:\AppServ\www\test\editdata.php - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

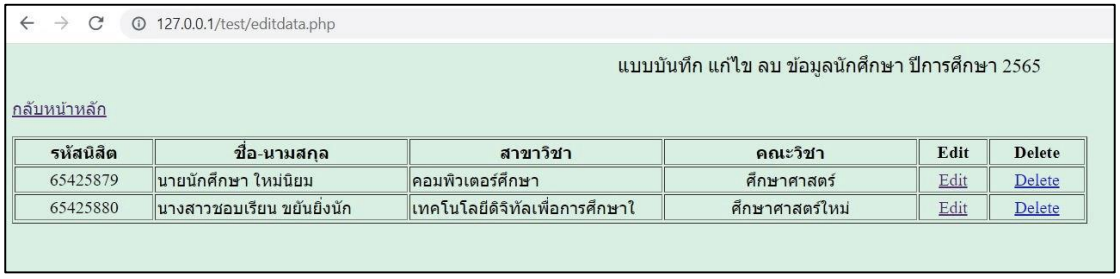
editdata.php

```php
<html>
<head>
<title>editdata</title>
</head>
<body>
<body bgcolor="#D5F5E3">
 <center><font size = "4"> โปรแกรมแก้ไขมูลนักศึกษา หรือ ลบข้อมูลนักศึกษา </font></center><br>
<?php
$objConnect = mysql_connect("localhost","root","12345678") or die("Error Connect to Database");
$objDB = mysql_select_db("student2023");

//*** Update Condition ***//
if($_POST["hdnCmd"] == "Update")
{
    $strSQL = "UPDATE regdata SET ";
    $strSQL .="id = '".$_POST["txtEditid"]."' ";
    $strSQL .=",name = '".$_POST["txtEditname"]."' ";
    $strSQL .=",major = '".$_POST["txtEditmajor"]."' ";
    $strSQL .=",faculty = '".$_POST["txtEditfaculty"]."' ";
    $strSQL .="WHERE id = '".$_POST["hdnEditid"]."' ";
    $objQuery = mysql_query($strSQL);
    if(!$objQuery)
    {
        echo "Error Update [".mysql_error()."]";
    }
    //header("location:$_SERVER[PHP_SELF]");
    //exit();
}

//*** Delete Condition ***//
if($_GET["Action"] == "Del")
{
    $strSQL = "DELETE FROM regdata ";
    $strSQL .="WHERE id = '".$_GET["id"]."' ";
    $objQuery = mysql_query($strSQL);
    if(!$objQuery)
    {
        echo "Error Delete [".mysql_error()."]";
    }
    //header("location:$_SERVER[PHP_SELF]");
    //exit();
}

$strSQL = "SELECT * FROM regdata";
$objQuery = mysql_query($strSQL) or die ("Error Query [".$strSQL."]");
?>
<form name="frmMain" method="post" action="<?php echo $_SERVER["PHP_SELF"];?>">
<input type="hidden" name="hdnCmd" value="">
<table width="1000" border="1">
  <tr>
   <th width="50">รหัสนิสิต</th>
            <th width="120">ชื่อ-นามสกุล</th>
            <th width="120">สาขาวิชา</th>
            <th width="120">คณะวิชา</th>
            <th width="30"> <div align="center">Edit </div></th>
            <th width="30"> <div align="center">Delete </div></th>
  </tr>
<?php
while($objResult = mysql_fetch_array($objQuery))
{
?>

  <?php
    if($objResult["id"] == $_GET["id"] and $_GET["Action"] == "Edit")
    {
  ?>
  <tr>
      <td><div align="center">
        <input type="text" name="txtEditid" size="10" value="<?php echo $objResult["id"];?>">
        <input type="hidden" name="hdnEditid" size="20" value="<?php echo $objResult["id"];?>">
    </div></td>
    <td><input type="text" name="txtEditname" size="30" value="<?php echo $objResult["name"];?>"></td>
    <td><input type="text" name="txtEditmajor" size="30" value="<?php echo $objResult["major"];?>"></td>
    <td><input type="text" name="txtEditfaculty" size="30" value="<?php echo $objResult["faculty"];?>"></div></td
    <td colspan="2" align="right"><div align="center">
      <input name="btnAdd" type="button" id="btnUpdate" value="Update"
      OnClick="frmMain.hdnCmd.value='Update';frmMain.submit();">
      <input name="btnAdd" type="button" id="btnCancel" value="Cancel"
      OnClick="window.location='<?php echo $_SERVER["PHP_SELF"];?>';">
    </div></td>
  </tr>
  <?php
    }
```

```
  else
    {
  ?>
  <tr>
     <td><div align="center"><?php echo $objResult["id"];?></div></td>
    <td><?php echo $objResult["name"];?></td>
    <td><?php echo $objResult["major"];?></td>
    <td><div align="center"><?php echo $objResult["faculty"];?></div></td>
    <td align="center"><a href="<?php echo $_SERVER["PHP_SELF"];?>?Action=Edit&id=
    <?php echo $objResult["id"];?>">Edit</a></td>
    <td align="center"><a href="JavaScript:if(confirm('Confirm Delete?')==true){window.location='
    <?php echo $_SERVER["PHP_SELF"];?>?Action=Del&id=<?php echo $objResult["id"];?>';}">Delete</a></td>
  </tr>
  <?php
    }
  ?>
<?php
}
mysql_close($objConnect);
?>
<a href="index.html">กลับหน้าหลัก</a><p>
</body>
</html>
```

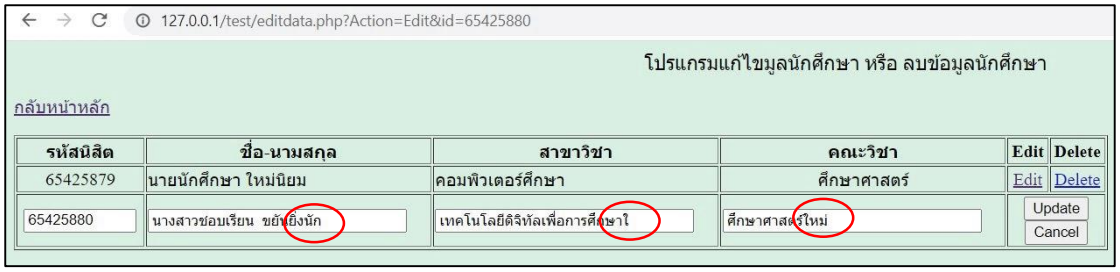
ภาพที่ 8.29 โค้ดภาษา PHP สร้างโปรแกรมแก้ไขข้อมูล และลบข้อมูล

เมื่อเขียนโปรแกรมเสร็จสมบูรณ์ และบันทึกไฟล์ editdata.php ใน server เรียบร้อย ให้เรียกดูหน้าเว็บผ่าน web browser ด้วยURL: 127.0.01/test/editdata.php
หรือ localhost/test/editdata.php จะปรากฏหน้าเว็บดังนี้



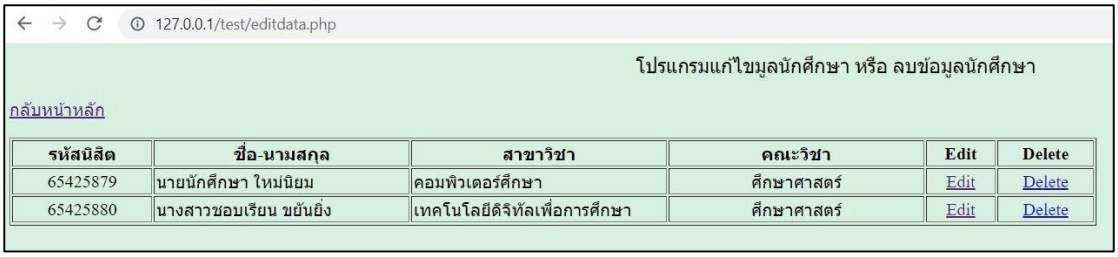
ภาพที่ 8.30 หน้าเว็บโปรแกรมแก้ไขข้อมูล และลบข้อมูล

ทดสอบการแก้ไขข้อมูลโดยคลิกที่ Edit ของรายการข้อมูลที่ต้องการแก้ไข จะปรากฏช่องสำหรับแก้ไขข้อมูล กรณีตัวอย่าง ต้องการแก้ไขนามสกุล สาขาวิชา และคณะวิชาดังนี้



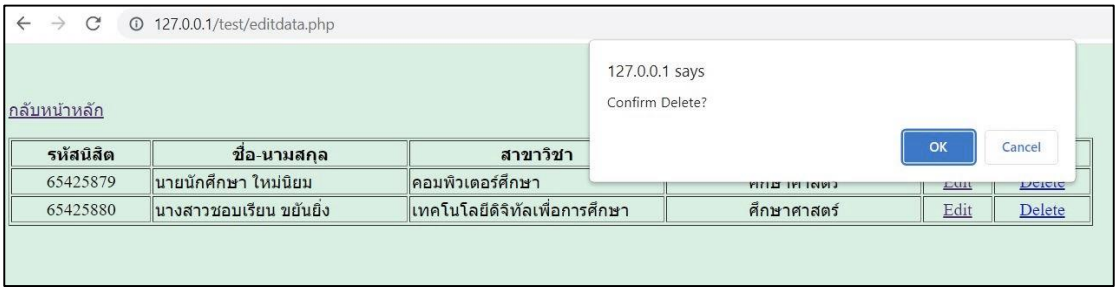
ภาพที่ 8.31 หน้าเว็บโปรแกรมแสดงการแก้ไขข้อมูล

เมื่อแก้ไขข้อมูลเสร็จแล้วคลิกปุ่ม Update เพื่อส่งข้อมูลที่แก้ไขแล้วเข้าฐานข้อมูล และระบบจะปรับเปลี่ยนข้อมูลใหม่ ดังนี้



ภาพที่ 8.32 หน้าเว็บแสดงผลการแก้ไขข้อมูล

กรณีที่ต้องการลบข้อมูลให้คลิก Delete ตรงแถวข้อมูลที่ต้องการลบ เมื่อคลิก Delete โปรแกรมให้ยืนยันการลบข้อมูลอีกครั้ง ดังภาพที่ 8.33 ดังนี้



ภาพที่ 8.33 หน้าเว็บแสดงการยืนยันการลบข้อมูล

เมื่อคลิกปุ่ม OK ข้อมูลแถวที่เลือกจะถูกลบออกจากฐานข้อมูลทันที ดังแสดงในภาพที่ 8.34 จะเห็นว่ารายการข้อมูลถูกลบ 1 รายการทำให้เหลือ 1 รายการ จากเดิมมีข้อมูลอยู่ 2 รายการ

ภาพที่ 8.34 หน้าเว็บแสดงผลการลบข้อมูล

จากตัวอย่างการเขียนโปรแกรมด้วยภาษา PHP เพื่อเข้าถึงระบบฐานข้อมูล MySQL จะเห็นว่าสามารถเขียนโปรแกรมเพื่อการใช้ฐานข้อมูลได้อย่างหลากหลาย ทำให้เกิดความสะดวกต่อผู้ดูแลฐานข้อมูลในระดับต่าง ๆ รวมถึงผู้ใช้งานข้อมูลทั่วไป เนื่องจากระบบฐานข้อมูล MySQL และโปรแกรมที่เขียนด้วยภาษา PHP เมื่อนำขึ้น Web hosting จะทำให้ผู้ใช้งานระดับต่าง ๆ เข้าถึงฐานข้อมูลผ่านหน้าเว็บในระบบอินเตอร์เน็ตได้ ส่งผลทีมพัฒนาฐานข้อมูลสามารถทำงานร่วมกันผ่านด้วยระบบออนไลน์ได้สะดวกเมื่อต้องการเพิ่มข้อมูล แก้ไขข้อมูล หรือลบข้อมูล ทำได้เร็วขึ้น ทำให้ฐานข้อมูลมีความถูกต้องทันสมัยต่อเหตุการณ์ต่าง ๆ นอกจากนั้นผู้ใช้งานข้อมูลสามารถเข้าถึงข้อมูล

ได้สะดวกเช่นเดียวกัน เนื่องจากการเขียนโปรแกรมด้วยภาษา PHP มีความยืดหยุ่น และสามารถกำหนดตัวแปรต่าง ๆ ให้สอดคล้องกับความต้องการของผู้ใช้งานได้เป็นอย่างดี เนื่องจากระบบฐานข้อมูลมีความสำคัญต่อองค์กร ดังนั้น การรักษาความปลอดภัยของฐานข้อมูลถือเป็นภารกิจที่สำคัญเช่นเดียวกัน โดยความปลอดภัยของฐานข้อมูลจะนำเสนอในบทที่ 9 ลำดับถัดไป

สรุป

ภาษาพีเอชพี ในชื่อภาษาอังกฤษว่า PHP ซึ่งใช้เป็นคำย่อจากคำว่า PHP Hypertext Preprocessor กระบวนการประมวลผลภาษา PHP โดยเริ่มต้นเมื่อเว็บไคลเอ็นท์ร้องขอไฟล์ หรือเอกสารที่ใช้ภาษา PHP มายังเว็บเซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์จะเรียกใช้ Engine ของภาษา PHP หรือตัวแปลคำสั่งของภาษา PHP กรณีที่มีการใช้คำสั่งในการเชื่อมต่อฐานข้อมูลระบบก็จะติดต่อไปยังฐานข้อมูลตามการร้องขอ มีการตรวจสอบสิทธิ์การเข้าถึงฐานข้อมูลและอื่น ๆ ตามลำดับขั้นตอนวิธีลักษณะการแปลคำสั่งของภาษา PHP นั้นจะแปลและเปลี่ยนคำสั่งจากภาษา PHP เป็นภาษา HTML แล้วส่งกลับไปแสดงผลลัพธ์ที่เว็บไคลเอ็นท์ที่ร้องขอ ปัจจุบันสามารถดาวโหลดชุดโปรแกรมสำหรับสร้างเว็บเซิร์ฟเวอร์บนเครื่องคอมพิวเตอร์ตั้งโต๊ะหรือโน๊ทบุค และรองรองรับการเขียนสคริปต์ PHP, การสร้างฐานข้อมูล MySQL ด้วยโปรแกรม phpMyAdmin เช่น AppServ หรือ XAMPP

MySQL เป็นระบบจัดการฐานข้อมูล มีลักษณะเป็นโครงสร้างของการเก็บรวบรวมข้อมูลในการเพิ่มข้อมูล เข้าถึงฐานข้อมูล หรือประมวลผลข้อมูลที่เก็บในฐานข้อมูลจำเป็นจะต้องอาศัยระบบจัดการฐานข้อมูล MySQL ซึ่ง MySQL จะทำหน้าที่เป็นตัวกลางในการจัดการกับข้อมูลในฐานข้อมูลทั้งสำหรับการใช้งานเฉพาะ และรองรับการทำงานของแอพลิเคชันอื่น ๆ ที่ต้องการใช้งานข้อมูลในฐานข้อมูล เพื่อให้ได้รับความสะดวกในการจัดการกับข้อมูลจำนวนมาก การเขียนโปรแกรมเพื่อเข้าถึงฐานข้อมูล MySQL ด้วยภาษา PHP มีความสำคัญต่อผู้ดูแลระบบ และผู้นำข้อมูลไปใช้ เพราะมีความสะดวก และทำให้ฐานข้อมูลมีความปลอดภัยขึ้น เนื่องจากผู้ใช้งานระดับต่าง ๆ สามารถเข้าถึงฐานข้อมูลผ่านเว็บไซต์บนอุปกรณ์ต่าง ๆ ได้ การเขียนโปรแกรมด้วยภาษา PHP เพื่อเชื่อมโยงกับฐานข้อมูล เช่น คำสั่งนำข้อมูลเข้าตาราง คำสั่งแสดงรายการข้อมูล คำสั่งค้นหาข้อมูล คำสั่งแก้ไขข้อมูล คำสั่งลบข้อมูลในตาราง เป็นต้น

บทที่ 9

ความปลอดภัยของฐานข้อมูล

ในปัจจุบันเรื่องของการรักษาความปลอดภัยของข้อมูลถือเป็นส่วนสำคัญของการนำระบบสารสนเทศเข้ามาใช้ในองค์กร เนื่องจากระบบสารสนเทศใช้คอมพิวเตอร์เป็นหลักในการเก็บรักษาข้อมูล และใช้ระบบเครือข่ายเป็นกลางในการติดต่อสื่อสาร จึงเป็นเรื่องง่ายต่อการคุกคามข้อมูลจากผู้ไม่ประสงค์ สำหรับภัยคุกคามต่อระบบคอมพิวเตอร์จะหมายความครอบคลุมทั้งการคุกคามทางฮาร์ดแวร์ซอฟต์แวร์และข้อมูล โดยสาเหตุของภัยคุกคามอาจจะมาจากทางกายภาพ เช่น อัคคีภัย ปัญหาวงจรไฟฟ้า ระบบสื่อสาร ความผิดพลาดของฮาร์ดแวร์ความผิดพลาดของซอฟต์แวร์ หรือภัยคุกคามที่เกิดจากคนหรือผู้ใช้ระบบ เช่น การบุกรุกจากผู้ที่ไม่ได้รับอนุญาต หรือผู้ใช้ไม่เข้าใจระบบทำให้ระบบเกิดความเสียหาย ภัยคุกคามเหล่านี้เป็นสาเหตุให้ข้อมูลในระบบเสียหาย สูญหาย ถูกขโมยหรือแก้ไขบิดเบือน ในบทนี้จะกล่าวถึงความปลอดภัยของฐานข้อมูล โดยนำเสนอเนื้อหาสำหรับการศึกษาดังต่อไปนี้

- ความหมายความสำคัญความปลอดภัยของฐานข้อมูล
- การสร้างความเสียหาย (Threats)
- ประเภทของภัยระบบฐานข้อมูล
- วัตถุประสงค์การรักษาความปลอดภัย
- การควบคุมความปลอดภัยของฐานข้อมูลด้วยวิว (VIEW)
- ข้อควรคำนึงในการรักษาความปลอดภัยฐานข้อมูล

ความหมายความปลอดภัยของฐานข้อมูล

ฐานข้อมูลถือเป็นระบบพื้นฐานสำหรับใช้เก็บข้อมูลสำคัญต่าง ๆ ขององค์กรในยุคดิจิทัล ส่งผลให้ฐานข้อมูลมักตกเป็นเป้าหมายของแฮกเกอร์บ่อยครั้ง ในกรณีที่ฐานข้อมูลถูกเจาะ นอกจากข้อมูลความลับขององค์กรจะถูกขโมยแล้ว ยังส่งผลกระทบต่อภาพลักษณ์ขององค์กร และสร้างภาระในการชดเชยความเสียหายในเชิงกฎหมายอีกด้วย

พนิดา พานิชกุล และณัฐพงษ์ วารีประเสริฐ (2552) ได้ให้ความหมายของปลอดภัยของฐานข้อมูล (Database Security) ว่า หมายถึง การป้องกันฐานข้อมูลจากภัยคุกคามทั้งที่ตั้งใจ และไม่ตั้งใจโดยการควบคุมด้วยระบบคอมพิวเตอร์และการควบคุมอื่น ๆ

ความมั่นคงของฐานข้อมูล ไม่ได้หมายถึงเพียงความมั่นคงของข้อมูลในฐานข้อมูลเท่านั้น แต่ยังหมายถึงส่วนอื่นของระบบฐานข้อมูลด้วย ได้แก่ ฮาร์ดแวร์ ซอฟต์แวร์ และบุคลากร การที่จะรักษาความมั่นคงให้กับฐานข้อมูลได้ จะต้องอาศัยมาตรการควบคุมที่เหมาะสมกับภัยคุกคามแต่ละชนิด ซึ่งถูกกำหนดไว้ในนโยบายด้านความมั่นคงของระบบเป้าหมายของการรักษาความมั่นคงของฐานข้อมูลมี 3 ประการ ดังนี้

1. ปฏิเสธการเข้าถึงฐานข้อมูลโดยผู้ใช้ที่ไม่มีสิทธิ์
2. รับประกันว่าผู้ใช้ที่มีสิทธิ์จะสามารถเข้าใช้ข้อมูลได้เสมอ (ความพร้อมใช้ของข้อมูล)
3. รักษาข้อมูลที่เป็นความลับ

เพื่อให้บรรลุเป้าหมายทั้ง 3 ประการข้างต้น ระบบฐานข้อมูลจำเป็นต้องมีระบบรักษาความมั่นคงของฐานข้อมูลที่มีความสามารถดังต่อไปนี้

1. ป้องกันความเสียหายของข้อมูลจากภัยพิบัติต่าง ๆ ได้ เช่น ไฟไหม้ น้ำท่วม เป็นต้น
2. ป้องกันการลักลอบขโมยข้อมูลจากบุคคลผู้ไม่หวังดีได้
3. ป้องกันการทำลายฐานข้อมูลโดยบุคคลที่ไม่หวังดี
4. ป้องกันการนำข้อมูลไปใช้เพื่อการโกง
5. รักษาข้อมูลส่วนตัวของบุคคลหรือกลุ่มบุคคลได้
6. สามารถระบุตัวตน (Identification) ของผู้ใช้แต่ละรายได้
7. สามารถกำหนดสิทธิ์ในการเข้าใช้ฐานข้อมูลให้กับผู้ใช้แต่ละรายได้
8. สามารถกำหนดขอบเขตของข้อมูลที่ผู้ใช้แต่ละคนสามารถใช้งานได้
9. สามารถกำหนดระดับของสิทธิ์ในการเข้าใช้ฐานข้อมูลของผู้ใช้ได้
10. ติดตามการใช้งานฐานข้อมูลของผู้ใช้แต่ละราย โดยบันทึกรายละเอียดการใช้งานนั้นไว้เรียกบันทึกดังกล่าวว่า “Audit trail” เพื่อประโยชน์ในการตรวจสอบการใช้งานของผู้ใช้ต่อไป

ความมั่นคงปลอดภัยของระบบสารสนเทศ (Information Security) มีจุดเริ่มต้นมาจากความต้องการความปลอดภัยของคอมพิวเตอร์ (Computer Security) เนื่องจากในยุคแรกภัยคุกคามส่วนใหญ่เป็นภัยคุกคามทางกายภาพ เช่น การลักขโมยอุปกรณ์ การก่อวินาศกรรมและการโจรกรรมผลผลิตที่ได้จากระบบ เป็นต้น ต่อมาเมื่อเทคโนโลยีคอมพิวเตอร์และการสื่อสารมีความก้าวหน้ามากขึ้น จึงเกิดภัยคุกคามหลายรูปแบบ เช่น การโจรกรรมข้อมูลที่เป็นความลับ การลักลอบเข้าสู่ระบบโดยไม่ได้รับอนุญาตตลอดจนการทำลายระบบด้วยวิธีต่าง ๆ ซึ่งก่อให้เกิดความเสียหายทั้งต่อบุคคลและทรัพย์สินอย่างมาก จึงต้องมีการกำหนดขอบเขตของความมั่นคงปลอดภัยคอมพิวเตอร์จึงมีเพิ่มขึ้น โดยให้ครอบคลุมถึงความปลอดภัยของข้อมูลการจำกัดการเข้าถึงข้อมูล และการจำกัด

ระดับความเกี่ยวข้องกับข้อมูลของคนในองค์กรด้วย ซึ่งขอบเขตที่เพิ่มเติมเข้ามาเหล่านี้ เพื่อเสริม “ความมั่นคงปลอดภัยของสารสนเทศ” (ศิริพร อ่องรุ่งเรือง, 2560)

การสร้างความเสียหาย (Threats)

การสร้างความเสียหาย หมายถึง การกระทำหรือเหตุการณ์ใด ๆ ที่เกิดขึ้นด้วยความตั้งใจหรือด้วยเหตุสุดวิสัยที่ทำให้ข้อมูลถูกเปิดเผย (Data Confidentiality) หรือมีการเปลี่ยนแปลงข้อมูล (Data Integrity) หรือทำให้ระบบหยุดบริการ (system Availability) ทำให้เกิดความเสียหายขึ้นกับข้อมูลที่บรรจุในฐานข้อมูลหรือทำให้ข้อมูลในฐานข้อมูลไม่ปลอดภัย ส่งผลต่อการทำงานของระบบองค์กรหน่วยงานต่าง ๆ (สุจิตรา อดุลย์เกษม, 2553)

ตัวอย่างการสร้างความเสียหายให้กับฐานข้อมูลระบบ

1. ข้อมูลสูญหายหรือเสียหายโดยเหตุสุดวิสัย (Accidental Losses) หมายถึง ความเสียหายของข้อมูลที่เกิดขึ้นเนื่องจากอุบัติเหตุ หรือปรากฏการณ์ธรรมชาติ เช่น ไฟไหม้ แผ่นดินไหว น้ำท่วม ฯลฯ หรือเป็นการทำงานที่ผิดพลาดของฮาร์ดแวร์ ซอฟต์แวร์ เช่น ฮาร์ดดิสก็เสียหาย (Disk crash) รวมทั้งข้อผิดพลาดอันเนื่องมาจากมนุษย์ จะเห็นได้ว่าเราไม่สามารถกำจัดสาเหตุที่ทำให้เกิดความเสียหายเหล่านี้ลงได้ แต่เราสามารถกำหนดนโยบาย หรือกำหนดขั้นตอนการทำงานที่ช่วยลดปริมาณและโอกาสที่จะทำให้เกิดความเสียหายดังกล่าวได้

2. การโกงและการหลอกลวง (Theft and Fraud) หมายถึง กลุ่มคนที่ไม่มีสิทธิ์และไม่ได้รับอนุญาตให้เข้าถึงฐานข้อมูล ทำการลักลอบเข้าถึงฐานข้อมูลของระบบ โดยมีจุดประสงค์ที่แตกต่างกัน ซึ่งอาจจะเป็นกลุ่มคนที่พยายามหารายได้ให้ตนเอง เช่น พนักงานของบริษัทเขียนโปรแกรมเพื่อโอนย้ายเงินของบริษัทเข้าไปไว้ในบัญชีของตนเอง หรืออาจจะเป็นกลุ่มคนที่ต้องการโจรกรรมข้อมูลทางทหารหรือทางธุรกิจ เช่น บริษัท A ลักลอบเข้าไปดูข้อมูลด้านการตลาด และข้อมูลลูกค้าของบริษัทคู่แข่ง เพื่อจะได้วางแผนการตลาดที่เหนือกว่าและตึงลูกค้าจากบริษัทคู่แข่ง จะเห็นว่าการกระทำของกลุ่มคนเหล่านี้สร้างความเสียหายให้กับข้อมูลของระบบ ดังนั้น เราจำเป็นต้องมีมาตรการและวิธีการป้องกันไม่ให้บุคคลที่ไม่มีสิทธิ์และไม่ได้รับอนุญาตเข้าถึงฐานข้อมูลของระบบได้

3. การสูญเสียความเป็นส่วนตัวหรือความลับของข้อมูล (Loss of privacy or confidentiality) หมายถึง ภัยความเสียหายของข้อมูลในระบบ ที่เกิดขึ้นเนื่องจากข้อมูลของระบบถูกเปิดเผยให้กับบุคคลที่ไม่มีสิทธิ์หรือไม่ได้รับอนุญาต แต่มีความอยากรู้อยากเห็นและชอบบุกรุกเข้าไปในบริเวณที่ไม่ได้รับอนุญาต เช่น ระบบงานทะเบียนนักศึกษา อาจจะมีนักศึกษาบางคนที่เกิดความอยากรู้อยากเห็นต้องการรู้ผลการเรียนหรือข้อมูลส่วนตัวของเพื่อนนักศึกษาและพยายามลักลอบเข้าไปดูข้อมูลเหล่านั้น ไม่ได้ต้องการทำให้ข้อมูลเกิดความเสียหาย แต่เจ้าของข้อมูลได้รับความเสียหายในส่วนที่ขาดความเป็นส่วนตัวหรือบริษัท A เก็บข้อมูลลูกค้าไว้ในฐานข้อมูลของบริษัทอาจจะมีบุคคลภายนอกลักลอบเปิดแฟ้มข้อมูลประวัติส่วนตัวลูกค้าที่บริษัทเก็บไว้ในฐานข้อมูล หรือขโมยรหัสบัตรเครดิตของลูกค้า และนำข้อมูลส่วนตัวของลูกค้าเปิดเผยต่อสาธารณะทำให้ลูกค้าสูญเสียความเป็นส่วนตัว และเกิดความเสียหายจากการที่รหัสบัตรเครดิตไม่ได้ถูกเก็บเป็นความลับของลูกค้าอีกต่อไป

4. การสูญเสียความเชื่อถือได้ของข้อมูล (Loss of integrity) หมายถึง ความเสียหายของข้อมูลในระบบที่เกิดขึ้น เนื่องจากมีบุคคลที่ไม่มีสิทธิ์หรือไม่ได้รับอนุญาตเข้าไปเปลี่ยนแปลงแก้ไขข้อมูลในฐานข้อมูลของระบบ ทำให้ข้อมูลของระบบเปลี่ยนไปจากเดิม เช่น ระบบงานทะเบียน

นักศึกษามีการเก็บข้อมูลเกรดของนักศึกษา ในการทำงานของระบบนั้น ระบบจะอนุญาตให้นักศึกษาเข้าไปดูเกรดของตัวเองได้ แต่นักศึกษาไม่สามารถแก้ไขเกรดได้ บุคคลที่สามารถแก้ไขเกรดของนักศึกษาแต่ละคนในระบบนี้ได้มีเฉพาะอาจารย์ประจำรายวิชาเท่านั้นข้อมูลเกรดในระบบจึงมีความถูกต้อง สามารถเชื่อถือได้ แต่ถ้าระบบนักศึกษาแก้ไขเกรดได้ จะเกิดความเสียหายขึ้นกับข้อมูล ทำให้ข้อมูลเกิดผิดพลาดและไม่น่าเชื่อถืออีกต่อไป

5. การสูญเสียความพร้อมใช้ (Loss of availability) หมายถึง ความเสียหายของระบบที่เกิดขึ้นเนื่องจากมีบุคคลที่ไม่ได้รับอนุญาตเข้าไปกระทำการใด ๆ กับระบบ ทำให้ระบบหยุดการทำงาน ไม่สามารถให้บริการกับผู้ใช้งานระบได้ เป็นการสร้างภัยคุกคาม และเกิดความเสียหายกับระบบคอมพิวเตอร์

ประเภทของภัยระบบฐานข้อมูล

ภัยคุกคามอาจแบ่งประเภทตามส่วนประกอบของระบบคอมพิวเตอร์ได้ คือ ฮาร์ดแวร์ ซอฟต์แวร์ ข้อมูลและเครือข่าย ภัยต่อฮาร์ดแวร์เป็นภัยที่เห็นได้ชัดเจนที่สุด เช่น การขโมยตัวเครื่องหรือชิ้นส่วนไป มาตรการในระดับนโยบายและบุคคลมีบทบาทสำคัญในการป้องกันภัยทางฮาร์ซอฟต์แวร์ที่สำคัญ คือ การโจมตีต่อความพร้อมใช้ เช่น เปลี่ยนแปลงซอฟต์แวร์หรือข้อมูลบางส่วนซึ่งทำให้ระบบหยุดการทำงานไป และยังรวมถึงการทำสำเนาซอฟต์แวร์ไปใช้อย่างไม่ได้รับอนุญาตหรือหนอนทางคอมพิวเตอร์ ซึ่งแพร่หลายและสร้างความเสียหายอย่างมากนับตั้งแต่กลางทศวรรษ 1990

ปัจจุบันนี้ภัยด้านข้อมูลมีความสำคัญและแพร่หลายขึ้นมาก ที่สำคัญ ได้แก่ การเข้าถึงไฟล์ข้อมูลโดยไม่ได้รับอนุญาต อันอาจสร้างความเสียหายได้อย่างมาก รวมทั้งการพยายามบันทึกสถิติจากข้อมูลที่รวบรวมมา เป็นการละเมิดความเป็นส่วนตัวของบุคคล

ภัยทางเครือข่าย ได้แก่ การดักจับข้อมูลบนสื่อ (eavesdropping) ที่ส่งข้อมูล โดยอาศัยการดักจับ และวิเคราะห์เนื้อความของข้อความและรูปแบบการส่งข้อมูล หรือการพราง (Masquerade) เพื่อกระทำการตามประสงค์ร้ายบางอย่าง เช่น พรางตัวเป็นคู่สนทนารับข้อมูล หรือเปลี่ยนแปลงข้อมูลเป็นอย่างอื่นด้วย นอกจากนี้ การโจมตีบนเครือข่ายยังรวมถึงการขัดขวางการให้บริการ (denial of service) ตามปกติของระบบ

นอกจากการแบ่งประเภทภัยคุกคามตามส่วนประกอบของระบบคอมพิวเตอร์แล้ว การใช้งานข้อมูลมักจะมีการเคลื่อนย้ายจากต้นทางไปยังปลายทาง เช่น จากแฟ้มข้อมูลในเครื่องหนึ่งไปยังอีกเครื่องหนึ่ง เมื่อพิจารณาในแง่การใช้งานนี้ ภัยหรือการโจมตีข้อมูลของคอมพิวเตอร์ อาจแบ่งออกเป็นประเภทกว้าง ๆ ได้ 4 รูปแบบ คือ

1. การขัดจังหวะ (interruption) หมายถึง การทำให้ส่วนของระบบคอมพิวเตอร์ใช้งานไม่ได้เป็นการโจมตีความพร้อมใช้ของระบบ เช่น ทำลายสายส่งข้อมูล

2. การสอดแทรก (interception) หมายถึง การที่บุคคลที่ไม่ได้รับอนุญาตมาเข้าใช้ทรัพย์สินของระบบคอมพิวเตอร์ เป็นการโจมตีต่อการรักษาความลับของระบบ เช่น การดักจับข้อมูลหรือการทำสำเนาแฟ้มข้อมูลโดยไม่ได้รับอนุญาต

3. การเปลี่ยนแปลง (modification) หมายถึง การที่ผู้ไม่ได้รับอนุญาตนอกจากจะเข้าถึงแล้วยังทำให้ทรัพย์สินของระบบคอมพิวเตอร์เปลี่ยนแปลงไป เป็นการโจมตีบูรณภาพของระบบ เช่น การเปลี่ยนข้อมูลในแฟ้มข้อมูล หรือข้อมูลที่กำลังส่งไปผ่านสื่อบนเครือข่าย

4. การปลอมขึ้น (fabrication) หมายถึง การที่ผู้ไม่ได้รับอนุญาตทำการเพิ่มเติมสิ่งแปลกปลอมเข้าไปในระบบ เป็นการโจมตีการพิสูจน์ตัวจริง เช่น การแทรกข้อความเข้าไปในเครือข่าย หรือในแฟ้มข้อมูล (ชูเกียรติ วรสุชีพ, 2558)

วัตถุประสงค์การรักษาความปลอดภัย

วัตถุประสงค์ของการรักษาความปลอดภัยของระบบฐานข้อมูลก็เพื่อลดปัจจัยเสี่ยงที่เกี่ยวกับความเสียหายกับฐานข้อมูลเนื่องจากความผิดพลาดในการทำงานของผู้ใช้ระบบฐานข้อมูล แฟ้มข้อมูลเสียหายความผิดพลาดในการทำงานของเครื่อง การปฏิบัติงานที่ไม่เหมาะสม การทุจริต และการเปิดเผยข้อมูลที่เป็นความลับ โดยสามารถแยกวัตถุประสงค์โดยรวมของการรักษาความปลอดภัยในระบบฐานข้อมูลได้ 4 ประการ (วรกฤต แสนโภชน์, 2560)

1. ความลับของข้อมูล (Confidentiality) หมายถึงข้อมูลถูกเก็บเป็นความลับ (Data Confidentiality) การปกป้องข้อมูลไม่ให้ผู้ไม่มีสิทธิ์ในการใช้ข้อมูลเข้ามาใช้ข้อมูลได้ เช่น กำหนดสิทธิ์ให้ผู้ใช้งานแต่ละคนสามารถใช้งานได้ตามสิทธิ์ที่กำหนดเท่านั้น มีการรักษาความปลอดภัยโดยใช้บัตรผ่าน มีความปลอดภัยในการใช้งานในระบบเครือข่าย และมีระบบสำรองกู้คืนข้อมูลที่ดี สามารถใช้งานได้สะดวก

2 ความถูกต้องสมบูรณ์ (Integrity) หมายถึงข้อมูลมีความถูกต้องน่าเชื่อถือ การรักษาข้อมูลให้มีความถูกต้องตามเงื่อนไข กฎเกณฑ์ที่กำหนดไว้ตอนสร้างฐานข้อมูล รวมทั้งความถูกต้องของข้อมูลในการประมวลผลข้อมูลพร้อมกันด้วย

3. ความพร้อมใช้ (Availability) หมายถึงระบบมีความเสถียร (System Availability) และทำงานไม่ผิดพลาด สามารถทำงานได้ตามปกติและเต็มประสิทธิภาพตามจุดมุ่งหมายในการใช้ และมีขีดความสามารถปฏิบัติงานได้ตามที่ต้องการเนื่องถ้าการใช้งานระบบฐานข้อมูลมักจะมีข้อขัดข้องอยู่เสมอ เช่น เครื่องเสีย หรือไฟดับ หรือข้อมูลสูญหายถ้ามีการรักษาความปลอดภัยที่ดีจะทำให้ผู้ใช้งานมีความเชื่อถือในระบบฐานข้อมูลนั้น

4. ความเสี่ยง (Risk Assessment) การรักษาความปลอดภัยที่ดีจะช่วยลดความเสี่ยงในค่าใช้จ่ายที่จะเกิดขึ้นจากการเสียหายของข้อมูล การวางแผนด้านการรักษาความปลอดภัยได้อย่างเหมาะสม จะช่วยลดความเสี่ยงในการเกิดความเสียหายของข้อมูลค่าใช้จ่าย มีการประเมินความสมดุลระหว่างค่าใช้จ่ายหรือต้นทุนคุ้มค่ากับประโยชน์ที่จะได้รับจากการรักษาความปลอดภัย

การควบคุมความปลอดภัยฐานข้อมูล

ปัจจุบันภัยคุกคามฐานข้อมูลมาจากหลายทาง และยังส่งผลต่อระบบคอมพิวเตอร์ร้ายแรงขึ้น การควบคุมความปลอดภัยจึงถูกนำมาพิจารณาตั้งแต่ช่วงแรกของการพัฒนาระบบ เช่น การควบคุมการเข้าถึง (Access control) การยืนยันตัวบุคคล (Authentication) และการให้อำนาจหน้าที่ (Authorization) เพื่อระบุตัวบุคคลที่ติดต่อหรือทำธุรกรรมร่วมด้วยการตรวจสอบ (Auditing) การสร้างข้อมูลให้เป็นความลับหรือการเข้ารหัสข้อมูล (Encryption) การควบคุมความถูกต้องสมบูรณ์ (Integrity controls) การสำรองข้อมูล (Backups) และความปลอดภัยแอพลิเคชัน (Application security) เป็นต้น

1. การควบคุมการเข้าถึง (Access Control) ข้อมูล

สิ่งสำคัญในการสร้างระบบรักษาความปลอดภัยในระบบฐานข้อมูลคือการควบคุมการเข้าถึงข้อมูล เป็นการกำหนดสิทธิ์ในการเข้าถึงฐานข้อมูล การกำหนดการใช้ข้อมูล โดยกำหนดสิทธิ์หรือการยืนยันตัวบุคคล ในการเข้าถึงฐานข้อมูลเฉพาะผู้ที่เกี่ยวข้อง หรือ ทำการเข้ารหัสฐานข้อมูลโดยใช้อัลกอริธึมที่มีความปลอดภัยสูง กำหนดว่าใครบ้างที่สามารถเข้าไปใช้ได้ ผู้ที่จะเข้ามาใช้ระบบฐานข้อมูลได้จะต้องได้รับการอนุญาตก่อน และเมื่อเข้าระบบได้แล้ว ผู้ใช้งานนั้นจะสามารถทำอะไรกับข้อมูลได้บ้างก็จะขึ้นอยู่กับการให้สิทธิ (Authorization) ของผู้บริหารฐานข้อมูล

2. การยืนยันตัวบุคคลและการให้อำนาจหน้าที่ เป็นระบบรักษาความปลอดภัยขั้นแรกที่นิยมใช้กันมากที่สุดในปัจจุบัน เช่น

2.1 การกำหนดรหัสผ่านและรหัสการเข้าใช้ (User name and password) การเข้ารหัสข้อมูลเป็นพื้นฐานสำคัญในการรักษาความปลอดภัยฐานข้อมูล ผู้ใช้จะมีรหัสเฉพาะของตน ควรกำหนดรหัสที่ยากต่อการถอดรหัส หลักการพื้นฐานควรกำหนดรหัสให้มีความยาวไม่น้อยกว่า 8 ตัวอักษร และควรให้มีการผสมระหว่างตัวอักขระพิเศษและตัวเลข ไม่ควรนำเอาคำศัพท์ในพจนานุกรม หรือใช้ชื่อ วันเดือนปีเกิด หมายเลขโทรศัพท์ เพราะรหัสเหล่านี้ง่ายต่อการถอดรหัส หรือทำการเข้ารหัสข้อมูลโดยใช้อัลกอริธึมที่มีความปลอดภัยสูง และควรเปลี่ยนรหัสเมื่อใช้ไปได้ระยะเวลาหนึ่ง

2.2 การใช้บัตรสมาร์ทการ์ด (Smart Card) ผู้ใช้จะต้องมีบัตรสำหรับเข้าระบบคอมพิวเตอร์ บัตรนี้จะคล้ายกับบัตร ATM (Automatic Teller Machine) และต้องป้อนรหัสส่วนตัว (Personnel identification number หรือ PIN) หรือการใช้บัตรกุญแจ (Key card) หรือบัตรผ่านทาง (Badge) ซึ่งเป็นวัตถุครอบครอง (Possessed object) เพื่อผ่านทางเข้าไปใช้ระบบหรือข้อมูลที่เก็บในคอมพิวเตอร์ เป็นรูปแบบที่นิยมกันมากในปัจจุบัน

2.3 การใช้การตรวจสอบจากร่างกายมนุษย์ (Biometric) เช่น ม่านตา หรือเรตินา (Retina) เสียง หรือ ลายนิ้วมือ ตรวจสอบผู้มีสิทธิ์ก่อนเข้าสู่ระบบการตรวจสอบในลักษณะนี้จะต้องนำลักษณะของผู้ที่ต้องการเข้าไปใช้ฐานข้อมูล ไปเปรียบเทียบกับลักษณะข้อมูลของผู้ใช้ที่มีอยู่ในเครื่องคอมพิวเตอร์ ถ้าตรงกันจึงจะมีสิทธิ์เข้าไปใช้ข้อมูลได้

3. การตรวจสอบ (Auditing)

เป็นการตรวจสอบผู้ที่เข้ามาติดต่อกับระบบโดยใช้ซอฟต์แวร์ในการตรวจสอบ โดยบันทึกข้อมูลของการเข้ามาใช้งานทุกครั้งไว้ใน Log files โปรแกรมจะทำการบันทึกทั้งวันที่ เวลา บุคคลที่เข้ามาใช้งาน สามารถทำการตรวจสอบข้อมูลย้อยหลังได้ ปัจจุบันมีซอฟท์แวร์ที่ใช้เป็นเครื่องมือรักษาความปลอดภัยในระบบที่กำลังเริ่มใช้อย่างแพร่หลาย ได้แก่ระบบไฟร์วอลล์ (Firewall) ซึ่งเป็นซอฟท์แวร์ทำหน้าที่เสมือนกับกำแพงกันไฟไม่ให้ลุกลามขยายตัวหากมีไฟไหม้เกิดขึ้น

การติดตามตรวจสอบและสรุปการใช้งานฐานข้อมูลอยู่เสมออย่างน้อยเดือนละครั้ง โดยเน้นส่วนข้อมูลสำคัญ เช่น ผู้ใช้เป็นใคร หมายเลข IP เข้าบ่อยแค่ไหน และทำอะไรไปบ้างทำสำเร็จหรือไม่ เพื่อตรวจสอบพฤติกรรมที่อาจจะผิดปกติไปจากเดิม

ไม่ว่าระบบเครือข่ายจะมีฮาร์ดแวร์หรือซอฟท์แวร์ที่ดีเพียงใดในการปกป้องระบบเครือข่าย สิ่งที่สำคัญอย่างยิ่งก็คือผู้ใช้งานในระบบจะต้องคอยช่วยสอดส่องดูแลและป้องกันไม่ให้ตนเองเป็นช่องทางผ่านของแครกเกอร์ พึงระลึกไว้เสมอว่าไม่มีระบบเครือข่ายใดที่ปลอดภัยร้อยเปอร์เซ็นต์จากแครกเกอร์

4. การสร้างข้อมูลให้เป็นความลับ (Encryption)

เป็นการนำเทคนิคทางด้านการเข้ารหัสข้อมูล (Encode) โดยอาศัยขบวนการทางคณิตศาสตร์ ซึ่งมีหลายวิธีด้วยกัน เช่น

4.1 การเข้ารหัส (Coding) เป็นวิธีการแปลงรูปแบบของข้อมูลให้อยู่ในรูปแบบที่บุคคลอื่น ๆ ไม่สามารถรู้เนื้อหาของข้อมูล ยกเว้นบุคคลที่เป็นผู้รับซึ่งจะต้องมีตัวถอดรหัสทำเป็นการแปลงข้อมูลที่เข้ารหัสนั้นกลับมาเป็นข้อมูลต้นฉบับได้

4.2 การบีบอัด (Compression) วิธีนี้มักจะใช้กับข้อมูลประเภทตัวเลข หรือข้อมูลที่แปลงเป็นเลขฐาน 2 แล้ว วิธีนี้มักนำไปประยุกต์ใช้กับการบีบอัดข้อมูลเพื่อประหยัดพื้นที่ในการเก็บและเวลาในการส่งข้อมูลด้วย

4.3 การแทนค่า (Substitution) วิธีนี้ มีหลักการทำงานคล้ายกับการเข้ารหัส โดยมีการกำหนดค่าที่จะแทนไว้ล่วงหน้า ส่วนการเข้ารหัสจะเป็นการกำหนดหลักการเข้ารหัสไว้

4.4 การสลับตำแหน่งข้อมูล (Transposition) เป็นวิธีการสลับตำแหน่งของข้อมูลโดยไม่ได้เปลี่ยนข้อมูลแต่อย่างใด

5. การทำสำเนาข้อมูล (Data Copy Setting)

กรณีที่ข้อมูลอยู่ในแผ่นบันทึกอาจทำสำเนาข้อมูลทั้งแผ่นโดยใช้คำสั่ง Copy แต่ถ้าข้อมูลอยู่ในจานแม่เหล็กชนิดแข็งหรือกรณีที่มีข้อมูลเป็นจำนวนมากจะทำสำเนาโดยการใช้คำสั่ง Backup ลงบนแผ่นบันทึกหรือในเทปแม่เหล็ก

ในการใช้งานจริงในการรักษาความปลอดภัยของฐานข้อมูลมักจะเป็นการนำเทคนิคต่าง ๆ หลายเทคนิคมาประยุกต์ใช้งานร่วมกัน เพื่อให้ระบบความปลอดภัยนั้นมั่นคงและเชื่อถือได้

6. การควบคุมความถูกต้องหรือ ความคงสภาพของข้อมูล (Integrity Controls)

หมายถึง การที่ระบบจัดการฐานข้อมูลจะจัดการกับข้อมูล (เมื่อมีการใช้คำสั่งเพิ่ม ลบ หรือแก้ไขข้อมูล) เพื่อให้แน่ใจว่าข้อมูลในฐานข้อมูลมีความถูกต้องน่าเชื่อถือตามกฎ เงื่อนไขและข้อตกลง (Integrity rules) ที่ได้กำหนดไว้ตั้งแต่ต้น กฎดังกล่าว เป็นกฎระเบียบที่กำหนดขึ้นในขั้นตอนการออกแบบฐานข้อมูลเพื่อรักษาให้ข้อมูลในฐานข้อมูลมีความถูกต้อง ส่วนการควบคุมความถูกต้องของข้อมูลจะเป็นขั้นตอนที่เกิดขึ้นเมื่อเมื่อนำฐานข้อมูลไปใช้งานแล้ว

7. การสำรองข้อมูล (Backups)

เป็นการคัดลอกแฟ้มข้อมูลเพื่อทำสำเนา เพื่อหลีกเลี่ยงความเสียหายที่อาจจะเกิดขึ้นถ้าข้อมูลเกิดการเสียหายหรือสูญหาย โดยสามารถนำข้อมูลที่สำรองไว้มาใช้งานได้ทันที การสำรองข้อมูลทำได้หลายวิธี เช่น

7.1 ใช้โปรแกรม System Restore ซึ่งเป็นโปรแกรมหนึ่งในการสำรองและเรียกข้อมูลกลับคืน

7.2 สำรองข้อมูลด้วยอุปกรณ์ฮาดแวร์ เช่น ฮาร์ดดิสก์แบบติดตั้ง ภายนอกผ่านพอร์ต USB เทปแบ็กอัพ (นิยมใช้กับการสำรองข้อมูลขนาดใหญ่) ซิปไดรฟ์ (Zip drive) และเครื่องบันทึก DVD/ CD เป็นต้น

7.3 ใช้โปรแกรมสำรองข้อมูล (Backup Program) เช่น โปรแกรม Symantec NetBackup, Norton Ghost, Microsoft DPM เป็นต้น

7.4 การสำรองข้อมูลบนระบบเครือข่ายแบบคลาวด์ (Cloud) เช่น Apple iCloud, Google Drive, One Drive, Dropbox เป็นต้น

ระบบที่มีผู้ใช้เป็นจำนวนมาก จำเป็นอย่างยิ่งที่จะต้องมีการควบคุมการเรียกใช้ข้อมูล เพื่อป้องกันผู้ที่ไม่มีอำนาจในการเรียกใช้ข้อมูลนำข้อมูลจากฐานข้อมูลมาใช้ อันอาจจะเกิดผลเสียกับระบบฐานข้อมูลได้ ระบบบริหารจัดการข้อมูลจะกำหนดสิทธิ์ในการเข้าถึงข้อมูล และมอบอำนาจการเข้าถึงข้อมูลตลอดจนเรียกคืนอำนาจจากผู้ใช้ในระบบได้ด้วยการใช้คำสั่งภาษา SQL

การยืนยันตัวบุคคล เป็นระบบรักษาความปลอดภัยขั้นแรกที่นิยมใช้กันมากที่สุดในปัจจุบัน เพื่อให้มั่นใจได้ว่าผู้ที่จะเข้าระบบได้นั้นจะต้องเป็นผู้ที่มีสิทธิ์จริง ๆ ผู้ใช้งานแต่ละคนจะต้องป้อนรหัสผ่านจึงจะมีสิทธิ์เข้าถึงข้อมูลได้ การยืนยันตัวบุคคลโดยคำสั่ง SQL เป็นการกำหนดรหัสผ่านให้แก่ผู้ใช้แต่ละคนในการจัดการข้อมูลในตารางหรือวิว สามารถใช้คำสั่ง SQL ได้ดังนี้

```
CREATE (รหัสประจำตัวผู้ใช้ IDENTIFIED BY <รหัสผ่าน? ;
```

ตัวอย่างที่ 1: กำหนดข้อมูลผู้ใช้ชื่อ “Somchai” และรหัสผ่าน “S23122560”

```
CREATE Somchai IDENTIFIED BY S23122560 ;
```

|

การให้สิทธิ์ เมื่อผู้ใช้งานเข้าระบบได้แล้วจะสามารถทำอะไรกับข้อมูลได้บ้าง จะขึ้นอยู่กับการให้สิทธิ์ของผู้บริหารฐานข้อมูลหรือผู้ดูแลระบบ ๆ สามารถกำหนดสิทธิ์การใช้ข้อมูลของผู้ใช้งานแต่ละคนหรือแต่ละกลุ่มได้ตามต้องการ ทำให้ป้องกันผู้ที่ไม่เกี่ยวข้องเข้าไปดูข้อมูลที่เป็นความลับ เช่น

1) สิทธิ์ในการอ่าน (Read) การเพิ่ม (Insert) การลบ (Delete) การเปลี่ยนแปลงข้อมูล (Update) เป็นต้น

2) สิทธิ์ในการสร้างดัชนี (Index) การสร้างตารางหรือวิว การลบ ตารางหรือวิว (Drop) การเปลี่ยนแปลงโครงสร้างข้อมูล (Alteration)

3) สิทธิ์ในการพิมพ์เอกสาร สามารถกำหนดได้ว่า ต้องการให้ใครพิมพ์ ดูได้เฉพาะหน้าจอ หรืออนุญาตให้ใครพิมพ์ออกเครื่องพิมพ์ได้

4) กำหนดสิทธิ์ให้กับผู้ใช้แต่ละคนว่าใครมีอำนาจในการอนุมัติ หรือ การรับรองเอกสารได้บ้าง เช่น พิมพ์ก่อนรับรองทีหลัง หรือ รับรองแล้วพิมพ์เป็นต้น

การกำหนดสิทธิ์ในการเข้าถึงข้อมูล และการยกเลิกสิทธิ์โดยใช้คำสั่ง SQL มีดังนี้

- คำสั่ง SQL ในการกำหนดสิทธิ์การเข้าถึงข้อมูล เป็นการกำหนดสิทธิ์ให้แก่ผู้ใช้ในการจัดการข้อมูลในตารางหรือวิว

  GRANT หมายถึง การอนุญาตผู้ใช้ให้ได้รับสิทธิ์ที่กำหนด

  TO ID_User หมายถึง กำหนดสิทธิ์ให้กับรหัสประจำตัวผู้ใช้ หรือหากต้องการกำหนดสิทธิ์ให้กับผู้ใช้งานทุกคนสามารถทำได้โดยใช้คำสั่ง PUBLIC

  ALL PRIVILEGE หมายถึง การกำหนดสิทธิ์ในการจัดการข้อมูลให้ผู้ใช้ เช่น SELECT, INSERT, UPDATE และ DELETE หรือหากต้องการกำหนดทุกสิทธิ์สามารถทำได้โดยใช้คำสั่ง ALL PRIVILEGES

```
GRANT (SELECT, INSERT, UPDATE, DELETE or ALL PRIVILEGE)
    ON <ชื่อตารางหรือวิว> To <รหัสประจำตัวผู้ใช้>;
```

5. การควบคุมความปลอดภัยของฐานข้อมูลด้วยวิว (VIEW)

การควบคุมความปลอดภัยให้กับข้อมูลสามารถสร้างโครงสร้างข้อมูลใหม่ ที่ทำให้ผู้ใช้เข้าถึงโครงสร้างเพียงบ้างส่วนของฐานข้อมูลที่กำหนดให้เท่านั้น เป็นการป้องกันไม่ให้ผู้ใช้ได้เข้าถึงข้อมูลทั้งหมดของฐานข้อมูล เรียกตารางข้อมูลประเภทนี้ว่า “ตารางเสมือน” หรือ “วิว” บางครั้งการออกแบบฐานข้อมูลในระดับกายภาพก็ต้องกำหนดตารางเป็นลักษณะวิว เพื่อเป็นการป้องกันรักษาความปลอดภัยของข้อมูลและการควบคุมการใช้งานในระบบฐานข้อมูล เพราะหากสร้างเป็นตารางข้อมูลจริงอาจจะเกิดความซ้ำซ้อนของข้อมูลมาก ยากต่อการควบคุมการใช้งาน การปรับปรุงแก้ไขวิว ไม่ว่าจะเป็นการเพิ่ม การลบ การเปลี่ยนแปลงข้อมูล จะมีผลกระทบต่อค่าในตารางที่เป็นฐานข้อมูลของวิว วิวไม่สามารถปรับปรุงแก้ไขได้ทุกวิว มีเพียงบางวิวเท่านั้นที่สามารถปรับปรุงแก้ไขได้

การสร้างวิว มีวัตถุประสงค์เพื่อให้ผู้ใช้สามารถเรียกใช้ข้อมูลจากวิวแทนการเรียกจากรีเลชันโดยตรงเพื่อความปลอดภัยของฐานข้อมูล โดยผู้ใช้จะได้รับสิทธิ์ในการเข้าถึงข้อมูลในแต่ละวิวที่แตกต่างกัน ช่วยให้เกิดความเป็นอิสระของข้อมูลและลดความซ้ำซ้อนของฐานข้อมูล

5.1 โครงสร้างของวิว

วิวเป็นตารางข้อมูลที่มีรายละเอียด หรือได้รายละเอียดมาจากตารางหลัก ตารางที่สร้างขึ้นมานี้จะสอดคล้องกับการใช้งานของผู้ใช้ และทำงานเช่นเดียวกับตารางธรรมดา แต่ไม่มีข้อมูลเป็นของตนเอง เป็นกลไกรักษาความปลอดภัยในการปกปิดส่วนต่าง ๆ ของตารางที่เป็นความลับ หรือเกินความจำเป็นสำหรับผู้ใช้

ดังนั้น ในการป้องกันและรักษาความปลอดภัยของข้อมูลในการเรียกใช้ข้อมูลจากผู้ใช้งานที่ได้รับสิทธิ์ในการเข้าถึงข้อมูลที่แตกต่างกัน สามารถสร้างความปลอดภัยให้กับฐานข้อมูลได้โดยการสร้างเป็นวิวให้กับผู้ใช้

5.1.1 คุณสมบัติของวิว

1) ค่าที่ปรากฏบนวิวเป็นค่าจริงในตารางข้อมูล

2) วิวเป็นตารางข้อมูลที่มีรายละเอียดหรือได้รายละเอียดมาจากตารางข้อมูลจริงมีได้มากกว่า 1 ตาราง

3) วิวเป็นกลไกรักษาความปลอดภัย ในการปกปิดส่วนต่าง ๆ ของตารางที่เป็นความลับ หรือเกินความจำเป็นสำหรับผู้ใช้

4) วิวเป็นตารางที่ถูกสร้างขึ้นจากรีเลชันหรือวิวอื่น ๆ วิวไม่มีข้อมูลของตนเองแต่จะเสมือนหน้าต่างที่แสดงข้อมูลจากรีเลชัน

5) สามารถปรับปรุงข้อมูลในวิวได้ ได้แก่ Insert/ Update/ Delete ถ้าได้รับการอนุญาตจากระบบจัดการฐานข้อมูล

6) เพื่อจำกัดการเข้าถึงข้อมูลของผู้ใช้ ผู้บริหารฐานข้อมูลสามารถกำหนดสิทธิ์การใช้งานวิวให้กับผู้ใช้งานได้

5.1.2 การทำงานของวิว เมื่อมีการเรียกใช้วิวระบบจัดการฐานข้อมูลจะทำหน้าที่โดยเริ่มค้นหาข้อกำหนดของวิวในคำสั่ง SQL ที่เก็บไว้ในฐานข้อมูล แล้วแปลคำสั่งของวิวเพื่อไปนำข้อมูลมาจากตารางข้อมูลจริง ทำให้วิวรักษาความถูกต้องของโครงสร้างข้อมูลไว้ได้

สำหรับวิวแบบง่าย ๆ ระบบจัดการฐานข้อมูลอาจสร้างแต่ละแถวของวิวขึ้นมาจากตารางข้อมูลจริงเลย ส่วนวิวที่ซับซ้อนนั้นระบบจัดการฐานข้อมูลจะเก็บแถวของวิวไว้ในตารางชั่วคราว แล้วจึงแสดงผลจากตารางชั่วคราว และเลิกใช้ตารางชั่วคราวนั้นเมื่อหมดความต้องการใช้งานอีกต่อไป

คือ เมื่อสิ้นสุดคำสั่งของ SQL อย่างไรก็ตามไม่ว่าระบบจัดการฐานข้อมูลจะจัดการกับวิวอย่างไร ผลที่ได้ที่ผู้ใช้ได้รับก็ไม่แตกต่างกัน นั่นคือผู้ใช้งานฐานข้อมูลผ่านวิวสามารถอ้างอิงได้จากคำสั่ง SQL เสมือนหนึ่งว่าอ้างอิงไปยังตารางข้อมูลจริง โดยไม่จำเป็นต้องเข้าถึงข้อมูลทั้งหมดของตารางข้อมูลจริง

5.1.3 การสร้างวิว การสร้างวิวได้จากการ query ในคำสั่ง SQL ทำให้เห็นข้อมูลในแถวและคอลัมน์ตามต้องการ การสร้างวิวอาจสร้างมาจากตารางข้อมูลเดียว หรือมากกว่า 1 ตารางได้ ภาษา SQL จะมีการสร้างวิวโดยใช้คำสั่ง CREATE VIEW ซึ่งมีรูปแบบทั่วไปดังนี้

```
CREATE VIEW <ชื่อวิว>
AS SELECT * หรือ ชื่อคอลัมน์ที่ 1>, <ชื่อคอลัมน์ที่ 2>;.....<ชื่อคอลัมน์ที่ n >
FROM <ชื่อตาราง >;
```

5.1.4 การลบวิว (Drop View)

คำสั่งลบวิว “DROP VIEW” เป็นคำสั่งที่ใช้เมื่อต้องการลบโครงสร้างวิวที่สร้างขึ้น

```
DROP VIEW <ชื่อวิว> [RESTRICT|CASCADE];
```

ชื่อวิว = ชื่อวิวที่ต้องการลบ

RESTRICT = สามารถลบวิวนี้ได้ต่อเมื่อไม่ถูกเรียกใช้ หรือไม่ถูกอ้างอิงข้อมูลจากวิวอื่น ๆ

CASCADE = สามารถลบวิวนี้ได้โดยส่งผลกระทบไปยังวิวอื่น ๆ ที่อ้างอิง ข้อมูลมาจากวิวนี้

5.1.5 ลักษณะของวิว

1) วิวทางระดับ (Horizontal View) เป็นการสร้างวิวจากการเลือกข้อเฉพาะบางแถว โดยแสดงข้อมูลในทุกคอลัมน์

2) วิวทางดิ่ง (Vertical View) เป็นการสร้างวิวจากการเลือกข้อมูลบางคอลัมน์ โดยแสดงข้อมูลในทุกแถว

3) วิวทางระดับและวิวทางดิ่ง (Horizontal and Vertical View) เป็นการสร้างจากการเลือกข้อมูลบางคอลัมน์และบางแถว

4) วิวกลุ่ม (Group View) เป็นการสร้างวิวจากการเลือกข้อมูลโดยจัดให้เป็นกลุ่ม

5) วิวร่วม (Join View) เป็นการสร้างวิวจากการเลือกข้อมูลจากหลายตาราง

วิวทางระดับวิวทางดิ่ง และวิวทางระดับและทางดิ่ง สามารถที่จะปรับปรุงข้อมูลผ่านวิวทั้ง 3 แบบนี้ได้ ส่วนวิวกลุ่มไม่สามารถที่จะปรับปรุงข้อมูลผ่านวิวได้และวิวร่วมจะปรับปรุงข้อมูลผ่านวิวนี้ได้ ถ้ามีการเชื่อมวิวกันโดยคีย์หลักของตาราง

5.1.6 ประโยชน์ของวิว

1) สนับสนุนการรักษาความปลอดภัยของข้อมูล โดยจำกัดผู้ใช้ไม่ให้เข้าถึงข้อมูลจริง โดยสามารถกำหนดโครงสร้างข้อมูลตามความต้องการของผู้ใช้งานได้อย่างมีประสิทธิภาพ

2) ควบคุมความสอดคล้องของข้อมูล (Data Consistency) ในการใช้งานระหว่างตารางข้อมูลจริงกับวิว

3) ควบคุมความถูกต้องของข้อมูล (Data Integrity) จากการตรวจสอบโดยระบบจัดการฐานข้อมูล

4) รักษาความปลอดภัยในการปรับปรุงข้อมูล โดยปรับปรุงได้เฉพาะบางส่วนที่กำหนดให้ แสดงในวิวเท่านั้น

5) การกระทำกับวิว ไม่ส่งผลกระทบต่อการเปลี่ยนแปลงโครงสร้างตารางข้อมูลจริง

5.1.7 ข้อจำกัดของวิว

1) ข้อจำกัดในด้านการจัดการข้อมูล ในกรณีที่ผู้ใช้ต้องการจัดการข้อมูลบางคอลัมน์หรือแถวจากวิววิวที่มีซับซ้อนน้อย ระบบจัดการฐานข้อมูลจะรับคำสั่งไปดำเนินการจัดการข้อมูลไปยังคอลัมน์หรือแถวที่ต้องการในตารางข้อมูลจริง แต่สำหรับวิวที่มีซับซ้อนมากจะจำกัดให้ดูข้อมูลได้อย่างเดียว

2) ข้อจำกัดด้านประสิทธิภาพ ในกรณีของวิวที่สร้างมาจากตารางข้อมูลที่เชื่อมกันหรือ Join ตารางข้อมูลหลาย ๆ ตาราง จะส่งผลให้การทำงานของวิวใช้เวลานานในการประมวล

1.1 คำศัพท์ในเรื่องของความปลอดภัยของข้อมูล

Hacker หมายถึง คือผู้ที่มีความชำนาญในการใช้คอมพิวเตอร์ แอบเข้าไปใช้งานระบบคอมพิวเตอร์ของหน่วยงานอื่นโดยไม่ได้รับอนุญาต เพราะรู้ว่าระบบมีจุดโหว่ตรงไหน และวิธีการที่จะเข้าถึงจุดโหว่นั้นได้ ไม่มีเจตนาที่จะทำลายระบบข้อมูล แต่อาจจะเพื่อความท้าทาย ความมีชื่อเสียง หรือต้องการทดสอบความรู้ความสามารถของตนเอง อย่างไรก็ตามก็ถือว่าขาดจริยธรรมด้านคอมพิวเตอร์

Hacking หมายถึง การพยายามที่จะใช้อุบายผ่านระบบความปลอดภัยเพื่อเข้าสู่ระบบฐานข้อมูลและเครือข่าย เป็นการกระทำของ hacker โดยมีเจตนาที่จะลอกเลียนหรือเปลี่ยนแปลงข้อมูลต้นฉบับ การ hacking จะไม่สร้างความเสียหายกับข้อมูลตัวหลัก

Cracker หมายถึง ผู้ที่ใช้ทักษะด้านคอมพิวเตอร์ สร้างโปรแกรมเพื่อเจาะระบบข้อมูล บุกรุกทำลายระบบข้อมูล หรือก่อให้เกิดความเสียหาย รวมทั้งการลักลอบขโมยข้อมูลของบุคคลอื่น

Cyber Crime หมายถึง เป็นการกระทำที่ผิดกฎหมายโดยใช้วิธีการทางอิเล็กทรอนิกส์เพื่อโจมตีระบบคอมพิวเตอร์และข้อมูลที่อยู่บนระบบ

Cracking หมายถึง การเจาะระบบข้อมูล แต่มีเจตนาที่จะสร้างความเสียหายแก่ข้อมูลต้นฉบับโดยตรง

Cookie หมายถึง กลุ่มของข้อมูลที่ถูกส่งจากเว็บเซิร์ฟเวอร์มายังเว็บบราวเซอร์ และถูกส่งกลับมายังเว็บเซิร์ฟเวอร์ทุก ๆ ครั้งที่เว็บบราวเซอร์ร้องขอข้อมูล โดยปกติแล้วคุกกี้จะถูกใช้เพื่อจัดเก็บข้อมูลขนาดเล็ก ๆ ไว้ที่เว็บราวเซอร์ เพื่อให้เว็บเซิร์ฟเวอร์สามารถจดจำสถานการณ์ใช้งานของเว็บบราวเซอร์ที่มีต่อเว็บเซิร์ฟเวอร์

Ethical hacker หมายถึง ผู้ที่มีความเชี่ยวชาญด้านความปลอดภัยข้อมูลใช้ทักษะนี้ในการ hacking ข้อมูลเพื่อจุดประสงค์ในการป้องกันระบบ

Threat หมายถึง ภัยคุกคามหรือ สิ่งที่ละเมิดระบบรักษาความปลอดภัยและอาจก่อให้เกิดผลกระทบซึ่งเป็นอันตรายต่อระบบ

Virus คือ โปรแกรมคอมพิวเตอร์ที่เขียนขึ้นโดยความตั้งใจของโปรแกรมเมอร์ ถูกออกแบบมาให้แพร่กระจายตัวเองจากไฟล์หนึ่งไปยังไฟล์อื่น ๆ ภายในเครื่องคอมพิวเตอร์ เป็นสาเหตุหลักที่ทำลายระบบคอมพิวเตอร์ และสามารถส่งต่อให้กับผู้อื่นได้อีกด้วย

Worms หรือหนอนอินเตอร์เนต เป็นโปรแกรมคอมพิวเตอร์ที่ถูกออกแบบมาให้สามารถแพร่กระจายตัวเองจากคอมพิวเตอร์เครื่องหนึ่งไปยังอีกเครื่องหนึ่ง โดยอาศัยระบบ Network ผ่านสาย Cable การแพร่กระจายสามารถทำได้ด้วยตัวเองอย่างรวดเร็วและรุนแรงกว่าไวรัส

Spam mail คือ การส่งข้อความที่ไม่เป็นที่ต้องการให้กับคน จำนวนมาก ๆ จากแหล่งที่ผู้รับไม่เคยรู้จักหรือติดต่อมาก่อน ส่วนมากมักอยู่ในรูปของ E-mail นอกจากจะทำให้ผู้รับเสียเวลาในการลบข้อความเหล่านั้นแล้ว ยังทำให้ลดประสิทธิภาพการขนส่งข้อมูลบนอินเทอร์เน็ทด้วย

Vulnerability หมายถึง ช่องโหว่ หรือ จุดอ่อน จุดบกพร่องในระบบ

Attack หมายถึง การโจมตีหรือความพยายามที่จะข้ามผ่านระบบ การรักษาความปลอดภัยของเครื่องคอมพิวเตอร์ ซึ่งอาจจะทำให้เกิดการเปลี่ยนแปลงข้อมูล หรืออาจจะทำให้ระบบ หรือเครื่องคอมพิวเตอร์นั้นไม่สามารถให้บริการได้ การโจมตีจะสำเร็จหรือไม่ขึ้นอยู่กับช่องโหว่ของระบบคอมพิวเตอร์และประสิทธิผลของมาตรการรักษาความปลอดภัยของระบบนั้น ๆ

Social Engineering หมายถึง ปฏิบัติการจิตวิทยา ซึ่งเป็นวิธีที่ง่ายที่สุดในการโจมตี เนื่องจากไม่จำเป็นต้องใช้ความรู้ความชำนาญเกี่ยวกับคอมพิวเตอร์มากนัก และส่วนใหญ่จะใช้ได้ผลดี การโจมตีแบบวิศวกรรมสังคม จะเกี่ยวกับการหลอกให้บางคนหลงกลเพื่อเข้าระบบ เช่น การหลอกถามรหัสผ่าน

Spoof หมายถึง การโจมตีโดยใช้ช่องโหว่ของโปรโตคอล ARP เพื่อหลอกให้เหยื่อหลงกล เช่น ทำให้เหยื่อเข้าใจผิดว่าเครื่องของแฮกเกอร์คือ Gateway เพื่อที่จะบังคับให้ข้อมูลที่เหยื่อกำลังส่งไปยังเซิร์ฟเวอร์ (ที่อยู่บนอินเตอร์เน็ต) วิ่งผ่านเครื่องของแฮกเกอร์แล้วแฮกเกอร์ก็สามารถดักจับข้อมูลที่สำคัญของเหยื่อได้ เช่น ข้อมูลรหัสผ่านหรือ Cookie/ Session ID

3. ข้อควรคำนึงในการรักษาความปลอดภัยฐานข้อมูล

ข้อมูลที่นำมาจัดเก็บไว้ในระบบฐานข้อมูลนั้น อาจมีระดับของความสำคัญแตกต่างกันไป กลุ่มข้อมูลบางกลุ่มอาจเป็นความลับสุดยอดห้ามเผยแพร่เด็ดขาด แต่ข้อมูลบางกลุ่มก็เป็นความรู้ทั่วไปสามารถเปิดเผยได้ ดังนั้น จึงมีการพัฒนาเทคนิคที่นำมาใช้ในการให้ความความปลอดภัยแก่ข้อมูลในระบบฐานข้อมูลที่มีผู้ใช้หลาย ๆ คน ระบบจัดการฐานข้อมูลต้องทำหน้าที่ดูแลว่ากลุ่มผู้ใช้กลุ่มใดได้รับอนุญาตให้เข้าใช้ข้อมูลส่วนใดได้บ้าง

การรักษาความปลอดภัยฐานข้อมูลไม่ใช่เพียงแค่การติดตั้งระบบรักษาความปลอดภัยเท่านั้น แต่จะรวมถึงการวิเคราะห์และบริหารความเสี่ยง ที่ประกอบด้วยภัยคุกคาม (Threat) ทั้งจากคนภายในองค์กร และคนภายนอกองค์กร และช่องโหว่หรือ จุดอ่อน (Vulnerability) การกำหนดการบังคับใช้นโยบาย (Policy) และการเฝ้าระวังเหตุการณ์อยู่ตลอดเวลา (Monitoring) นั่นคือ ต้องมีมาตรการหรือการควบคุมความปลอดภัยที่มีประสิทธิภาพ ประกอบด้วย นโยบาย วิธีปฏิบัติ และกระบวนการขององค์กรในการรักษาข้อมูลให้มีความถูกต้องและน่าเชื่อถือ

3.1 นโยบาย

นโยบายขององค์กรมีผลสำคัญยิ่งต่อการรักษาความปลอดภัยของข้อมูล นโยบายขององค์กรจะต้องมุ่งเน้นที่วัตถุประสงค์และการทำงานที่ดี องค์กรจำเป็นต้องมีการกำหนดนโยบายด้านความปลอดภัยให้ชัดเจน โดยมีกฎระเบียบ ข้อบังคับ และหน้าที่ความรับผิดชอบ และวิธีปฏิบัติให้พนักงานใช้เป็นหลักในการทำงาน รวมทั้งการติดตามตรวจสอบการปฏิบัติตามกฎระเบียบที่วางไว้อย่างเคร่งครัดและสม่ำเสมอ เช่น กำหนดให้แน่นอนว่าระบบรักษาความปลอดภัยใครเป็นผู้ปฏิบัติ

ใช้กับส่วนใดบ้างในระบบมีวิธีการปฏิบัติอย่างไร ใครสามารถเข้าถึงข้อมูลส่วนใดได้บ้าง ใครมีสิทธิที่จะเปลี่ยนแปลงแก้ไขข้อมูลนั้น

3.2 สถานภาพของระบบการรักษาความปลอดภัย

จะต้องมีการตรวจสอบสถานภาพของระบบการรักษาความปลอดภัยในปัจจุบันอยู่ในระดับใดและต้องการปรับปรุงหรือเปลี่ยนแปลงอย่างไรบ้าง ความต้องการในการใช้ข้อมูลที่ปลอดภัย และคำแนะนำจากส่วนต่าง ๆ ที่ใช้งานภายในระบบ การแจกงานไปสู่ผู้ที่รับผิดชอบมีตารางเวลาที่กำหนดว่าส่วนใดของระบบจะต้องปรับปรุงอะไรบ้าง ณ เวลาใด มีการจัดทำแผนฉุกเฉินเพื่อให้องค์กรสามารถดำเนินการต่อไปได้เมื่อมีวิกฤตการณ์เกิดขึ้น บุคลากรที่เกี่ยวข้องควรจะคุ้นเคยกับแผนเหล่านี้ และมีการทดสอบให้มั่นใจว่า สามารถใช้งานได้

ตัวอย่างนโยบายและแนวปฏิบัติการรักษาความมั่นคงปลอดภัยด้านสารสนเทศมหาวิทยาลัยบูรพา (มหาวิทยาลัยบูรพา, 2560)

(สำเนา)

ประกาศมหาวิทยาลัยบูรพา
ที่ ๐๗๕๔ /๒๕๖๐
เรื่อง นโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ พ.ศ. ๒๕๖๐

เพื่อให้ระบบสารสนเทศของมหาวิทยาลัยบูรพา มีความมั่นคง ปลอดภัย สามารถดำเนินงานได้อย่างมีประสิทธิภาพ และมิให้มีผู้กระทำด้วยประการใด ๆ ให้ระบบสารสนเทศไม่สามารถทำงานตามคำสั่งหรือผิดพลาดไปจากคำสั่งที่กำหนดไว้ หรือใช้วิธีการใด ๆ เข้าล่วงรู้ แก้ไข หรือทำลายข้อมูลของบุคคลอื่นในระบบสารสนเทศโดยมิชอบ หรือใช้ระบบสารสนเทศเพื่อการเผยแพร่ข้อมูลอันเป็นเท็จ หรือมีลักษณะอันลามกอนาจาร ซึ่งอาจก่อให้เกิดความเสียหายแก่มหาวิทยาลัยบูรพา และเป็นความผิดตามพระราชบัญญัติว่าด้วยการกระทำความผิดเกี่ยวกับคอมพิวเตอร์ พ.ศ. ๒๕๕๐ และที่แก้ไขเพิ่มเติม หรือไม่สอดคล้องตามมาตรา ๗ ในพระราชกฤษฎีกากำหนดหลักเกณฑ์และวิธีการในการทำธุรกรรมทางอิเล็กทรอนิกส์ภาครัฐ พ.ศ. ๒๕๔๙

อาศัยอำนาจตามความในมาตรา ๒๖ และมาตรา ๓๒ (๓) แห่งพระราชบัญญัติมหาวิทยาลัยบูรพา พ.ศ. ๒๕๕๐ จึงออกประกาศไว้ ดังต่อไปนี้

ข้อ ๑ ประกาศนี้เรียกว่า "ประกาศมหาวิทยาลัยบูรพา เรื่อง นโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ พ.ศ. ๒๕๖๐"

ข้อ ๒ ประกาศนี้ให้ใช้บังคับตั้งแต่วันถัดจากวันประกาศเป็นต้นไป

ข้อ ๓ บรรดาประกาศ ระเบียบ คำสั่งหรือแนวปฏิบัติอื่นใดที่ได้กำหนดไว้แล้ว ซึ่งขัดหรือแย้งกับประกาศนี้ ให้ใช้ประกาศนี้แทน

ข้อ ๔ ในประกาศนี้

๔.๑ "มหาวิทยาลัย" หมายความว่า มหาวิทยาลัยบูรพา

๔.๒ "ส่วนงาน" หมายความว่า ส่วนงานตามมาตรา ๙ ของพระราชบัญญัติมหาวิทยาลัยบูรพา พ.ศ. ๒๕๕๐

๔.๓ "สำนักคอมพิวเตอร์" หมายความว่า สำนักคอมพิวเตอร์ มหาวิทยาลัยบูรพา

๔.๔ "ผู้บริหารระดับสูงสุด" หมายความว่า อธิการบดี มหาวิทยาลัยบูรพา

๔.๕ "ผู้บริหารระดับสูง" หมายความว่า รองอธิการบดี และหัวหน้าส่วนงาน

๔.๖ "ผู้อำนวยการสำนักคอมพิวเตอร์" หมายความว่า ผู้อำนวยการสำนักคอมพิวเตอร์ มหาวิทยาลัยบูรพา

หน้า ๒

๔.๗ "ผู้ใช้งาน" หมายความว่า ผู้ปฏิบัติงานในมหาวิทยาลัยบูรพา และให้หมายความรวมถึงนิสิตมหาวิทยาลัยบูรพา นักเรียนโรงเรียนสาธิตอาชีวศึกษา มหาวิทยาลัยบูรพา นักเรียนโรงเรียนสาธิต "พิบูลบำเพ็ญ" มหาวิทยาลัยบูรพา บุคคลอื่นที่มหาวิทยาลัยบูรพามอบหมายให้ปฏิบัติงานตามสัญญา และผู้ใช้งานทั่วไป

๔.๘ "ผู้ดูแลระบบ" หมายความว่า ผู้ที่ได้รับมอบหมายจากหัวหน้าส่วนงาน ให้มีหน้าที่รับผิดชอบในการดูแลรักษาระบบคอมพิวเตอร์ และระบบเครือข่ายให้ทำงานได้อย่างมีประสิทธิภาพ

๔.๙ "สิทธิของผู้ใช้งาน" หมายความว่า สิทธิทั่วไป สิทธิจำเพาะ สิทธิพิเศษ และสิทธิอื่นใดที่เกี่ยวข้องกับระบบสารสนเทศของมหาวิทยาลัยบูรพา

๔.๑๐ "สินทรัพย์" หมายความว่า ฮาร์ดแวร์ ซอฟต์แวร์ ข้อมูลและสารสนเทศของมหาวิทยาลัยบูรพา

๔.๑๑ "ระบบเครือข่าย" หมายความว่า เครือข่ายคอมพิวเตอร์ของมหาวิทยาลัยบูรพา

๔.๑๒ "การเข้าถึงหรือควบคุมการใช้งานสารสนเทศ" หมายความว่า การอนุญาต การกำหนดสิทธิ หรือการมอบอำนาจให้ผู้ใช้งาน เข้าถึงหรือใช้งานระบบเครือข่าย ระบบสารสนเทศและอุปกรณ์ในการประมวลผลข้อมูล ทั้งทางอิเล็กทรอนิกส์และทางกายภาพ รวมทั้งการอนุญาตสำหรับบุคคลภายนอก ตลอดจนกำหนดข้อปฏิบัติเกี่ยวกับการเข้าถึงโดยมิชอบ

๔.๑๓ "ความมั่นคงปลอดภัยด้านสารสนเทศ" หมายความว่า การธำรงไว้ซึ่งความลับ (confidentiality) ความถูกต้องครบถ้วน (integrity) และสภาพพร้อมใช้งาน (availability) ของสารสนเทศ รวมทั้งคุณสมบัติอื่น ได้แก่ ความถูกต้องแท้จริง (authenticity) ความรับผิด (accountability) การห้ามปฏิเสธความรับผิด (non-repudiation) และความเชื่อถือ (reliability)

๔.๑๔ "เหตุการณ์ด้านความมั่นคงปลอดภัย (information security incident)" หมายความว่า กรณีที่ระบุการเกิดเหตุการณ์ สภาพของบริการหรือระบบเครือข่ายที่แสดงให้เห็นความเป็นไปได้ที่จะเกิดการฝ่าฝืนนโยบายด้านความมั่นคงปลอดภัยหรือมาตรการป้องกันที่ล้มเหลว หรือเหตุการณ์อันไม่อาจรู้ได้ว่าอาจเกี่ยวข้องกับความมั่นคงปลอดภัย

๔.๑๕ "สถานการณ์ด้านความมั่นคงปลอดภัยที่ไม่พึงประสงค์หรือไม่อาจคาดคิด" หมายความว่า สถานการณ์ด้านความมั่นคงปลอดภัยที่ไม่พึงประสงค์หรือไม่อาจคาดคิด (unwanted or unexpected) ซึ่งอาจทำให้ระบบของมหาวิทยาลัยถูกบุกรุกหรือโจมตี และความมั่นคงปลอดภัยถูกคุกคาม

ข้อ ๕ กรณีระบบคอมพิวเตอร์หรือข้อมูลสารสนเทศเกิดความเสียหาย หรืออันตรายใด ๆ แก่มหาวิทยาลัย หรือผู้หนึ่งผู้ใดอันเนื่องมาจากความบกพร่อง ละเลย หรือฝ่าฝืนการปฏิบัติตามนโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ กำหนดให้ผู้บริหารระดับสูงสุดเป็นผู้รับผิดชอบต่อความเสี่ยง ความเสียหาย หรืออันตรายที่เกิดขึ้น

ข้อ ๖ ให้อธิการบดีรักษาการให้เป็นไปตามประกาศนี้

ส่วนที่ ๑
นโยบายในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ พ.ศ. ๒๕๖๐

ข้อ ๗ นโยบายหลักในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศของมหาวิทยาลัย (information security event) กำหนดประเด็นสำคัญ ดังนี้

หมวดที่ ๑ การเข้าถึงหรือควบคุมการใช้งานสารสนเทศ

(ส่วนที่ ๒ แนวปฏิบัติฯ หมวดที่ ๑)

(๑) การเข้าถึงระบบสารสนเทศและระบบเครือข่าย เพื่อควบคุมการเข้าถึงข้อมูลและอุปกรณ์ในการประมวลผลข้อมูล ให้คำนึงถึงความมั่นคงปลอดภัยในการใช้งาน โดยกำหนดกฎเกณฑ์ที่เกี่ยวกับการอนุญาตให้เข้าถึง กำหนดสิทธิ และการปรับปรุงสิทธิ เพื่อให้ผู้ใช้งานทุกระดับได้เข้าถึงข้อมูลและใช้งานได้ตามสิทธิที่กำหนดให้

(๒) การบริหารจัดการการเข้าถึงของผู้ใช้งาน เพื่อควบคุมการเข้าถึงระบบสารสนเทศเฉพาะผู้ที่ได้รับอนุญาตแล้ว และป้องกันการเข้าถึงจากผู้ซึ่งไม่ได้รับอนุญาต

(๓) การควบคุมการเข้าถึงระบบปฏิบัติการ เพื่อป้องกันการเข้าถึงระบบปฏิบัติการโดยไม่ได้รับอนุญาต

(๔) การควบคุมการเข้าถึงโปรแกรมประยุกต์ และสารสนเทศ เพื่อป้องกันการเข้าถึงหรือเข้าใช้งานของผู้ใช้งานระบบสารสนเทศของมหาวิทยาลัย และป้องกันความเสียหายของการใช้อุปกรณ์คอมพิวเตอร์และสื่อสารเคลื่อนที่

หมวดที่ ๒ การจัดให้มีระบบสารสนเทศและระบบสำรองของสารสนเทศ

(ส่วนที่ ๒ แนวปฏิบัติฯ หมวดที่ ๒)

ระบบสารสนเทศต้องจัดทำระบบสำรองของสารสนเทศให้อยู่ในสภาพพร้อมใช้งาน พร้อมทั้งจัดทำแผนเตรียมความพร้อมฉุกเฉิน เพื่อให้สามารถใช้งานสารสนเทศได้ตามปกติอย่างต่อเนื่อง

หมวดที่ ๓ การตรวจสอบและประเมินความเสี่ยงด้านสารสนเทศ

(ส่วนที่ ๒ แนวปฏิบัติฯ หมวดที่ ๓)

กำหนดให้ผู้ดูแลระบบตรวจสอบและประเมินความเสี่ยงที่อาจเกิดขึ้นกับระบบสารสนเทศ อย่างน้อยปีละ ๑ ครั้ง โดยการตรวจสอบและประเมินความเสี่ยงจะต้องดำเนินการโดยผู้ตรวจสอบภายในส่วนงานของแต่ละส่วนงาน (internal auditor) หรือผู้ตรวจสอบด้านความมั่นคงปลอดภัยจากหน่วยงานภายนอกมหาวิทยาลัย (external auditor)

หมวดที่ ๔ การทบทวน ปรับปรุงนโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ

ให้ทำการทบทวน ปรับปรุงนโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศให้เป็นปัจจุบัน อย่างน้อยปีละ ๑ ครั้ง

หน้า ๔

ส่วนที่ ๒
แนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ พ.ศ. ๒๕๖๐

หมวดที่ ๑ การเข้าถึงหรือควบคุมการใช้งานสารสนเทศ

ตอนที่ ๑ การเข้าถึงหรือควบคุมการใช้งานสารสนเทศ (access control)

ข้อ ๘ ผู้ดูแลระบบกำหนดสิทธิการเข้าถึงหรือควบคุม (๑) การใช้งานระบบเครือข่าย ระบบสารสนเทศ และอุปกรณ์ในการประมวลผลข้อมูล โดยแบ่งกลุ่มผู้ใช้งานและกำหนดสิทธิของผู้ใช้งานให้เหมาะสมกับบทบาทและหน้าที่ของผู้ใช้งานแต่ละกลุ่ม (๒) กรณีของผู้รับเหมาดำเนินการ (outsource) ในเรื่องต่าง ๆ ให้ขออนุญาตเป็นลายลักษณ์อักษร ระบุระยะเวลาการใช้งาน ลงนามรักษาความลับ และได้รับการพิจารณาอนุญาตจากหัวหน้าส่วนงาน (๓) นอกจากนี้ให้ผู้ดูแลระบบตรวจสอบและปรับปรุงความถูกต้องของการให้สิทธิ ระงับสิทธิและยกเลิกสิทธิอย่างสม่ำเสมอ

ข้อ ๙ การยืนยันตัวบุคคลสำหรับผู้ใช้งานที่อยู่ภายนอกองค์กร ให้เชื่อมต่อเข้าใช้งานระบบเครือข่ายโดยใช้บัญชีผู้ใช้งานที่ได้รับจากสำนักคอมพิวเตอร์ ผ่านระบบ VPN (Virtual Private Network) ของมหาวิทยาลัย

ข้อ ๑๐ ผู้ดูแลระบบจัดแบ่งประเภทของข้อมูล ลำดับความสำคัญหรือลำดับชั้นความลับของข้อมูล เวลาและช่องทางการเข้าถึงข้อมูล ดังนี้

(๑) ประเภทของข้อมูล จัดเรียงตามลำดับความสำคัญ ดังนี้
- (ก) ข้อมูลด้านการบริหาร
- (ข) ข้อมูลด้านการเรียนการสอน
- (ค) ข้อมูลด้านการวิจัย
- (ง) ข้อมูลสำหรับประชาชนทั่วไป

(๒) ระดับชั้นการเข้าถึง

(ก) ระดับผู้บริหารระดับสูง เข้าถึงข้อมูลภาพรวมด้านการบริหาร ด้านการเรียนการสอน และด้านการวิจัย ตามอำนาจหน้าที่และลำดับชั้นการบังคับบัญชาในส่วนงาน

(ข) ระดับผู้ดูแลระบบ หรือผู้ที่ได้รับมอบหมาย มีสิทธิในการบริหารจัดการระบบและเข้าถึงข้อมูลตามที่ได้รับมอบหมาย

(ค) ระดับผู้ใช้งานภายในส่วนงาน เข้าถึงข้อมูลได้ตามอำนาจหน้าที่ที่ได้รับมอบหมาย

(ง) ประชาชนทั่วไป เข้าถึงข้อมูลได้เฉพาะข้อมูลสาธารณะ ข้อมูลสำหรับประชาชนทั่วไป

(๓) ช่องทางการเข้าถึงข้อมูล

เข้าถึงได้โดยตรงหรือผ่านระบบงานได้ตลอดเวลา ตามสิทธิของผู้ใช้งาน

ข้อ ๑๑ ผู้ดูแลระบบ ทำการติดตั้งระบบบันทึกและติดตามการใช้งานระบบสารสนเทศของส่วนงาน และตรวจตราการละเมิดความปลอดภัยที่มีต่อระบบสารสนเทศ

ข้อ ๑๒ ผู้ดูแลระบบ ทำการบันทึกรายละเอียดการเข้าถึงระบบสารสนเทศและการแก้ไขเปลี่ยนแปลงสิทธิของผู้ใช้งาน เพื่อเป็นหลักฐานในการตรวจสอบ

ข้อ ๑๓ ผู้ดูแลระบบ ทำการบันทึกการผ่านเข้าออกสถานที่ตั้งของระบบสารสนเทศ

หน้า ๕

ตอนที่ ๒ การบริหารจัดการการเข้าถึงของผู้ใช้งาน (user access management)

ข้อ ๑๔ ผู้ดูแลระบบ ดำเนินการดังนี้

(๑) จัดทำแบบฟอร์มการลงทะเบียนของผู้ใช้งานประเภทบุคลากร เพื่อเข้าใช้งานสารสนเทศ กรณีนิสิต ระบบจะสร้างบัญชีผู้ใช้งานให้อัตโนมัติเมื่อรายงานตัวเข้าเป็นนิสิต

(๒) ตรวจสอบบัญชีผู้ใช้งาน เพื่อไม่ให้มีการลงทะเบียนซ้ำซ้อน

(๓) ตรวจสอบ ให้สิทธิของผู้ใช้งานและยกเลิกสิทธิที่เหมาะสมต่อหน้าที่ความรับผิดชอบ

(๔) กำหนดให้มีการแจกเอกสารหรือสิ่งที่แสดงเป็นลายลักษณ์อักษรให้แก่ผู้ใช้งาน เพื่อแสดงถึงสิทธิของผู้ใช้งานและหน้าที่ความรับผิดชอบในการเข้าถึงระบบสารสนเทศ

ข้อ ๑๕ สิทธิของผู้ใช้งาน มีการบริหารจัดการในแต่ละระดับดังนี้

(๑) ผู้บริหาร และผู้ดูแลระบบ มีสิทธิเข้าถึงข้อมูลทั้งหมด

(๒) ผู้ใช้งานภายในส่วนงาน มีสิทธิเข้าถึงข้อมูลตามหน้าที่ความรับผิดชอบ

ข้อ ๑๖ ผู้ดูแลระบบ ทบทวนบัญชีผู้ใช้งาน สิทธิการใช้งาน อย่างสม่ำเสมอ อย่างน้อยปีละ ๑ ครั้ง เพื่อป้องกันการเข้าถึงระบบโดยไม่ได้รับอนุญาต โดยปฏิบัติตามแนวทาง ดังนี้

(๑) พิมพ์รายชื่อของผู้ที่ยังมีสิทธิในระบบแยกตามส่วนงาน

(๒) จัดส่งรายชื่อให้แก่หัวหน้าส่วนงานเพื่อทบทวนรายชื่อและสิทธิการใช้งานว่าถูกต้องหรือไม่

(๓) ดำเนินการแก้ไขข้อมูลสิทธิต่าง ๆ ให้ถูกต้องตามที่ได้รับแจ้งกลับจากส่วนงาน

(๔) ขั้นตอนปฏิบัติสำหรับการยกเลิกสิทธิการใช้งาน เมื่อลาออกต้องดำเนินการภายใน ๓ วัน หรือเมื่อเปลี่ยนตำแหน่งงานภายในต้องดำเนินการภายใน ๗ วัน สำหรับประเภทบุคลากร และยกเลิกสิทธิโดยระบบสำหรับนิสิตเมื่อสำเร็จการศึกษา

ข้อ ๑๗ ผู้ดูแลระบบ บริหารจัดการรหัสผ่าน ดังนี้

(๑) กำหนดการเปลี่ยนแปลงและการยกเลิกรหัสผ่าน เมื่อผู้ใช้งานลาออกหรือพ้นจากตำแหน่ง หรือยกเลิกการใช้งาน

(๒) กำหนดชื่อผู้ใช้งานหรือรหัสผู้ใช้งานต้องไม่ซ้ำกัน

(๓) ส่งมอบรหัสผ่านชั่วคราวให้กับผู้ใช้งานด้วยวิธีการที่ปลอดภัย หลีกเลี่ยงการให้บุคคลอื่น หรือการส่งจดหมายอิเล็กทรอนิกส์ที่ไม่มีการป้องกันในการส่งรหัสผ่าน

(๔) ในกรณีมีความจำเป็นต้องให้สิทธิพิเศษกับผู้ใช้งานที่มีสิทธิสูงสุด ผู้ใช้งานนั้นจะต้องได้รับความเห็นชอบและอนุมัติจากหัวหน้าส่วนงาน โดยมีการกำหนดระยะเวลาการใช้งานและระงับการใช้งานทันทีเมื่อพ้นระยะเวลาดังกล่าวหรือพ้นจากตำแหน่งและมีการกำหนดสิทธิพิเศษที่ได้รับ ว่าสามารถเข้าถึงได้ถึงระดับใดได้บ้าง และต้องกำหนดให้รหัสผู้ใช้งานต่างจากรหัสผู้ใช้งานตามปกติ

ข้อ ๑๘ ผู้ดูแลระบบ บริหารจัดการการเข้าถึงข้อมูลตามประเภทชั้นความลับ ในการควบคุมการเข้าถึงข้อมูลแต่ละประเภทชั้นความลับทั้งการเข้าถึงโดยตรงและการเข้าถึงผ่านระบบงาน รวมถึงวิธีการทำลายข้อมูลแต่ละประเภทชั้นความลับ ดังนี้

(๑) ควบคุมการเข้าถึงข้อมูลแต่ละประเภทชั้นความลับทั้งการเข้าถึงโดยตรงและการเข้าถึงผ่านระบบงาน

(๒) กำหนดรายชื่อผู้ใช้งาน (username) และรหัสผ่าน เพื่อใช้ในการตรวจสอบตัวตนจริงของผู้ใช้งานข้อมูลในแต่ละชั้นความลับของข้อมูล

หน้า ๖

(๓) กำหนดระยะเวลาการใช้งานและระงับการใช้งานทันทีเมื่อพ้นระยะเวลาดังกล่าว

(๔) ในการรับส่งข้อมูลสำคัญผ่านระบบเครือข่ายสาธารณะ ให้ใช้วิธีการเข้ารหัส (encryption) ที่เป็นมาตรฐานสากล

(๕) กำหนดการเปลี่ยนรหัสผ่าน ตามระยะเวลาที่กำหนดของระดับความสำคัญของข้อมูล

(๖) กำหนดมาตรการรักษาความมั่นคงปลอดภัยของข้อมูลในกรณีที่นำสินทรัพย์ออกนอกส่วนงาน เช่น บำรุงรักษา ตรวจซ่อม ให้ดำเนินการสำรองและลบข้อมูลที่เก็บอยู่ในสื่อบันทึกก่อน เป็นต้น

ข้อ ๑๙ หัวหน้าส่วนงานพิจารณาประเด็นต่าง ๆ ทางด้านความมั่นคงปลอดภัย และจุดอ่อนต่าง ๆ ก่อนตัดสินใจใช้ข้อมูลร่วมกันในระบบงานสารสนเทศทางธุรกิจที่เชื่อมโยงกัน (business information systems) หรือระบบสารสนเทศที่จะเชื่อมโยง ดังนี้

(๑) กำหนดนโยบายและมาตรการเพื่อควบคุม ป้องกัน และบริหารจัดการการใช้ข้อมูลร่วมกัน

(๒) พิจารณาจำกัดหรือไม่อนุญาตการเข้าถึงข้อมูลส่วนบุคคล

(๓) พิจารณาว่ามีบุคลากรใดบ้างที่มีสิทธิหรือได้รับอนุญาตให้เข้าใช้งาน

(๔) พิจารณาเรื่องการลงทะเบียนผู้ใช้งาน

(๕) ไม่อนุญาตให้มีการใช้งานข้อมูลสำคัญหรือข้อมูลลับร่วมกัน ในกรณีที่ระบบไม่มีมาตรการป้องกันเพียงพอ

หน้า ๗

ตอนที่ ๓ การกำหนดหน้าที่ความรับผิดชอบของผู้ใช้งาน (user responsibilities)

ข้อ ๒๐ การใช้งานรหัสผ่าน กำหนดให้ผู้ใช้งานปฏิบัติ ดังนี้

(๑) ป้องกัน ดูแล รักษาข้อมูลบัญชีชื่อผู้ใช้งาน และรหัสผ่าน ห้ามใช้ร่วมกับผู้อื่น รวมทั้งห้ามทำการเผยแพร่ แจกจ่าย ทำให้ผู้อื่นล่วงรู้รหัสผ่าน

(๒) กำหนดรหัสผ่านด้วยตัวอักษรไม่น้อยกว่า ๘ ตัวอักษร ประกอบด้วย ตัวเลข (numerical character) ตัวอักษร (alphabet) และตัวอักษรพิเศษ (special character)

(๓) ไม่ใช้โปรแกรมคอมพิวเตอร์ช่วยในการจำรหัสผ่านส่วนบุคคลอัตโนมัติ (save password) สำหรับเครื่องคอมพิวเตอร์ส่วนบุคคลที่ผู้ใช้งานครอบครองอยู่

(๔) ไม่จดหรือบันทึกรหัสผ่านส่วนบุคคลไว้ในสถานที่ ที่ง่ายต่อการสังเกตเห็นของบุคคลอื่น

(๕) เปลี่ยนรหัสผ่านภายใน ๑๘๐ วันหรือเมื่อมีการแจ้งเตือนให้เปลี่ยนรหัสผ่าน

ข้อ ๒๑ การนำการเข้ารหัสมาใช้กับข้อมูลที่เป็นความลับ ผู้ใช้งานต้องใช้ใบรับรองอิเล็กทรอนิกส์สำหรับบุคคลธรรมดาหรือนิติบุคคล (digital signature) มาใช้สำหรับการเข้ารหัสข้อมูล

ข้อ ๒๒ การกระทำใด ๆ ที่เกิดจากการใช้บัญชีของผู้ใช้งาน อันมีกฎหมายกำหนดให้เป็นความผิด ไม่ว่าการกระทำนั้นจะเกิดจากผู้ใช้งานหรือไม่ก็ตาม ให้ถือว่าเป็นความรับผิดชอบส่วนบุคคล ซึ่งผู้ใช้งานจะต้องรับผิดชอบต่อความผิดที่เกิดขึ้นเอง

ข้อ ๒๓ ผู้ใช้งานทำการพิสูจน์ตัวตนทุกครั้งก่อนที่จะใช้สินทรัพย์หรือระบบสารสนเทศของส่วนงาน และหากการพิสูจน์ตัวตนนั้นมีปัญหา ไม่ว่าจะเกิดจากรหัสผ่านหมดอายุ หรือเกิดจากความผิดพลาดใด ๆ ผู้ใช้งานจะต้องแจ้งให้ผู้ดูแลระบบทราบทันที

ข้อ ๒๔ ผู้ใช้งานต้องตั้งเวลาล็อคหน้าจอคอมพิวเตอร์ เมื่อไม่มีการใช้งานนานเกิน ๑๕ นาที

ข้อ ๒๕ ข้อมูลที่เป็นความลับหรือมีระดับความสำคัญ ที่อยู่ในการครอบครองหรือดูแลของส่วนงาน ห้ามเผยแพร่ เปลี่ยนแปลง ทำซ้ำ หรือทำลาย โดยไม่ได้รับอนุญาตจากหัวหน้าส่วนงาน

ข้อ ๒๖ ผู้ใช้งานมีส่วนร่วมในการดูแลรักษาและรับผิดชอบต่อข้อมูลของมหาวิทยาลัย และข้อมูลของผู้รับบริการ หากเกิดการสูญหาย การนำไปใช้ในทางที่ผิด การเผยแพร่โดยไม่ได้รับอนุญาต ผู้ใช้งานจะต้องร่วมรับผิดชอบต่อความเสียหายนั้นด้วย

ข้อ ๒๗ ผู้ใช้งานต้องป้องกัน ดูแล รักษาไว้ซึ่งความลับ ความถูกต้อง และความพร้อมใช้ของข้อมูล ตลอดจนเอกสาร สื่อบันทึกข้อมูลคอมพิวเตอร์ หรือสารสนเทศต่าง ๆ ที่เสี่ยงต่อการเข้าถึงโดยผู้ซึ่งไม่มีสิทธิ

ข้อ ๒๘ ผู้ใช้งานมีสิทธิโดยชอบธรรมที่จะเก็บรักษา ใช้งาน และป้องกันข้อมูลส่วนบุคคลตามเห็นสมควร มหาวิทยาลัยจะให้การสนับสนุนและเคารพต่อสิทธิส่วนบุคคล และไม่อนุญาตให้บุคคลหนึ่งบุคคลใดทำการละเมิดต่อข้อมูลส่วนบุคคลโดยไม่ได้รับอนุญาตจากผู้ใช้งานที่ครอบครองข้อมูลนั้น ยกเว้นในกรณีที่มหาวิทยาลัยต้องการตรวจสอบข้อมูล หรือคาดว่าข้อมูลนั้นเกี่ยวข้องกับมหาวิทยาลัย ซึ่งมหาวิทยาลัยอาจแต่งตั้งให้ผู้ทำหน้าที่ตรวจสอบ ทำการตรวจสอบข้อมูลเหล่านั้นได้ตลอดเวลา โดยไม่ต้องแจ้งให้ผู้ใช้งานทราบ

ข้อ ๒๙ ห้ามผู้ใช้งานเปิดหรือใช้งานโปรแกรมประเภท peer-to-peer (วิธีการจัดระบบเครือข่ายคอมพิวเตอร์แบบหนึ่ง ที่กำหนดให้คอมพิวเตอร์ในระบบเครือข่ายทุกเครื่องเหมือนกันหรือเท่าเทียมกัน หมายความว่า แต่ละเครื่องต่างมีโปรแกรมหรือมีแฟ้มข้อมูลเก็บไว้เอง การจัดแบบนี้ทำให้สามารถใช้โปรแกรมหรือแฟ้มข้อมูลของคอมพิวเตอร์เครื่องใดก็ได้ แทนที่จะต้องใช้จากเครื่องบริการแฟ้ม (file server) เท่านั้น)

หน้า ๘

ข้อ ๓๐ ห้ามผู้ใช้งานใช้สินทรัพย์ของส่วนงาน ที่จัดเตรียมให้ เพื่อการเผยแพร่ ข้อมูล ข้อความ รูปภาพหรือสิ่งอื่นใด ที่มีลักษณะขัดต่อศีลธรรม ความมั่นคงของประเทศ กฎหมาย หรือกระทบต่อภารกิจของมหาวิทยาลัย

ข้อ ๓๑ ห้ามผู้ใช้งานใช้สินทรัพย์ของส่วนงาน เพื่อการรบกวน ก่อให้เกิดความเสียหาย หรือใช้ในการโจรกรรมข้อมูล หรือสิ่งอื่นใดอันเป็นการขัดต่อกฎหมายและศีลธรรม หรือกระทบต่อภารกิจของมหาวิทยาลัย

ข้อ ๓๒ ห้ามผู้ใช้งานใช้สินทรัพย์ของมหาวิทยาลัยเพื่อประโยชน์ทางการค้า

ข้อ ๓๓ ห้ามผู้ใช้งานกระทำการใด ๆ เพื่อการดักข้อมูล ไม่ว่าจะเป็นข้อความ ภาพ เสียง หรือสิ่งอื่นใดในระบบเครือข่าย ระบบสารสนเทศของมหาวิทยาลัยโดยเด็ดขาด ไม่ว่าจะด้วยวิธีการใด ๆ ก็ตาม

ข้อ ๓๔ ห้ามผู้ใช้งานกระทำการรบกวน ทำลาย หรือทำให้ระบบสารสนเทศของส่วนงาน ต้องหยุดชะงัก

ข้อ ๓๕ ห้ามผู้ใช้งานใช้ระบบสารสนเทศของมหาวิทยาลัย เพื่อการควบคุมคอมพิวเตอร์ หรือระบบสารสนเทศภายนอก โดยไม่ได้รับอนุญาตจากหัวหน้าส่วนงานหรือผู้ดูแลระบบที่ได้รับมอบหมาย

ข้อ ๓๖ ห้ามผู้ใช้งานกระทำการใด ๆ อันมีลักษณะเป็นการลักลอบใช้งานหรือรับรู้รหัสผ่านส่วนบุคคลของผู้อื่นไม่ว่าจะเป็นกรณีใด ๆ เพื่อประโยชน์ในการเข้าถึงข้อมูล หรือเพื่อการใช้ทรัพยากรก็ตาม

ข้อ ๓๗ ห้ามผู้ใช้งานติดตั้งอุปกรณ์หรือกระทำการใด ๆ เพื่อเข้าถึงระบบสารสนเทศของมหาวิทยาลัย โดยไม่ได้รับอนุญาตจากหัวหน้าส่วนงานหรือผู้ดูแลระบบ

ตอนที่ ๔ การควบคุมการเข้าถึงระบบเครือข่าย (network access control)

ข้อ ๓๘ ในการใช้บริการระบบเครือข่าย ผู้ดูแลระบบกำหนดสิทธิการใช้งานโดยให้สิทธิเฉพาะการปฏิบัติงานในหน้าที่และต้องได้รับความเห็นชอบจากผู้บังคับบัญชาของส่วนงานเจ้าของระบบเป็นลายลักษณ์อักษร รวมทั้งต้องทบทวนสิทธิดังกล่าวปีละ ๑ ครั้ง

ข้อ ๓๙ ผู้ดูแลระบบกำหนดให้มีการยืนยันตัวบุคคลก่อนที่จะอนุญาตให้เข้าใช้งานระบบเครือข่ายและระบบสารสนเทศของมหาวิทยาลัยได้ โดยให้ผู้ใช้งานที่จะเข้าใช้งานระบบ แสดงตัวตน (identification) ด้วยชื่อผู้ใช้งานทุกครั้ง

ข้อ ๔๐ การควบคุมอุปกรณ์บนเครือข่าย (equipment control in network) ดำเนินการดังนี้

(๑) ผู้ดูแลระบบใช้ซอฟต์แวร์ควบคุมสำหรับการบริหารจัดการอุปกรณ์บนระบบเครือข่ายและอุปกรณ์สื่อสารเคลื่อนที่ ซึ่งสามารถตรวจสอบสถานะของการทำงานของอุปกรณ์ โดยระบุจุดเชื่อมต่อ และ mac address ของอุปกรณ์บนเครือข่ายที่ต่อพ่วงกับระบบเครือข่ายได้

(๒) ผู้ใช้งานที่ต้องการนำอุปกรณ์คอมพิวเตอร์และอุปกรณ์โมบายมาใช้ภายในมหาวิทยาลัย จะต้องนำอุปกรณ์คอมพิวเตอร์และอุปกรณ์โมบาย มาลงทะเบียน mac address ผ่านระบบให้บริการ ก่อนการใช้งานในมหาวิทยาลัย

(๓) ผู้ใช้งานภายในมหาวิทยาลัยแสดงตัวตน ด้วยชื่อผู้ใช้งานทุกครั้งผ่านซอฟต์แวร์ควบคุม เพื่อระบุตัวตน จุดเชื่อมต่อ ของอุปกรณ์ที่เชื่อมต่อกับระบบเครือข่าย

(๔) ผู้ดูแลระบบมีสิทธิในการจำกัดอุปกรณ์ต่อพ่วงที่ไม่ได้รับอนุญาตได้

หน้า ๙

ข้อ ๔๑ การป้องกันพอร์ตที่ใช้สำหรับตรวจสอบและปรับแต่งระบบ (remote diagnostic and configuration port protection) ให้ควบคุมการเข้าถึงพอร์ตที่ใช้สำหรับตรวจสอบและปรับแต่งระบบทั้งการเข้าถึงทางกายภาพและทางระบบเครือข่าย ดังนี้

(๑) ผู้ดูแลระบบปิดพอร์ตที่ไม่จำเป็นทุกพอร์ตเพื่อจำกัดและควบคุมการเข้าถึงพอร์ตโดยไม่ได้รับอนุญาต

(๒) ผู้ดูแลระบบกำหนดพอร์ตสำหรับตรวจสอบและปรับแต่งระบบทั้งการเข้าถึงทางกายภาพและทางระบบเครือข่าย และแจ้งให้ผู้ดูแลระบบที่มีสิทธิในการตรวจสอบและปรับแต่งระบบทราบ

(๓) หากผู้ดูแลระบบตรวจสอบพบการใช้งานพอร์ตโดยผู้ใช้ที่ไม่ได้รับอนุญาต ผู้ดูแลระบบสามารถปิดการใช้งานพอร์ตที่ไม่ได้รับอนุญาตได้ทันที

ข้อ ๔๒ การแบ่งแยกเครือข่าย (segregation in networks) ผู้ดูแลระบบจะทำการแบ่งแยกเครือข่ายสำหรับกลุ่มผู้ใช้งาน โดยแบ่งออกเป็น ๒ เครือข่าย คือ เครือข่ายสำหรับผู้ใช้งานภายใน และเครือข่ายสำหรับผู้ใช้งานภายนอก

ข้อ ๔๓ การควบคุมการเชื่อมต่อทางเครือข่าย (network connection control) ผู้ดูแลระบบจะควบคุมการเข้าถึงหรือใช้งานเครือข่ายที่มีการใช้ร่วมกันหรือเชื่อมต่อ ดังนี้

(๑) มีการตรวจสอบการเชื่อมต่อเครือข่าย

(๒) จำกัดสิทธิ ความสามารถของผู้ใช้งานในการเชื่อมต่อเข้าสู่เครือข่าย

(๓) ใช้อุปกรณ์ Firewall สำหรับควบคุมการเชื่อมต่อ

(๔) มีระบบการตรวจจับผู้บุกรุกทั้งในระดับเครือข่าย และระดับเครื่องคอมพิวเตอร์แม่ข่าย

(๕) ควบคุมไม่ให้มีการเปิดให้บริการบนเครือข่าย โดยไม่ได้รับอนุญาต

ข้อ ๔๔ การควบคุมการจัดเส้นทางบนเครือข่าย (network routing control) ผู้ดูแลระบบจะควบคุมการจัดเส้นทางบนเครือข่าย เพื่อให้การเชื่อมต่อของคอมพิวเตอร์และการส่งผ่านหรือไหลเวียนของข้อมูลหรือสารสนเทศสอดคล้องกับแนวปฏิบัติการควบคุมการเข้าถึงหรือการประยุกต์ใช้งานตามภารกิจ ดังนี้

(๑) ควบคุมไม่ให้มีการเปิดเผยแผนการใช้หมายเลขเครือข่าย (ip address plan)

(๒) กำหนดให้มีการแปลงหมายเลขเครือข่าย เพื่อแยกเครือข่ายย่อย

(๓) กำหนดมาตรการการบังคับใช้เส้นทางเครือข่าย กล่าวคือ สามารถเชื่อมเครือข่ายปลายทางผ่านช่องทางที่กำหนดไว้ หรือจำกัดสิทธิในการใช้บริการเครือข่าย

ข้อ ๔๕ การควบคุมการเข้าใช้งานระบบจากภายนอก ให้ปฏิบัติดังนี้

(๑) การเข้าสู่ระบบจากระยะไกล (remote access) ผู้ดูแลระบบต้องมีการตรวจสอบเพื่อพิสูจน์ตัวตนของผู้ใช้งาน

(๒) การเข้าสู่ระบบจากระยะไกลสู่ระบบสารสนเทศและเครือข่ายของมหาวิทยาลัย ต้องมีการกำหนดมาตรการรักษาความปลอดภัยที่เพิ่มขึ้นจากมาตรฐานการเข้าสู่ระบบภายใน

(๓) วิธีการใด ๆ ก็ตามที่สามารถเข้าสู่ระบบสารสนเทศและเครือข่ายได้จากระยะไกลต้องได้รับการอนุมัติจากผู้อำนวยการสำนักคอมพิวเตอร์ก่อนและมีการควบคุมอย่างเข้มงวดก่อนนำมาใช้และผู้ใช้งานต้องปฏิบัติตามข้อกำหนดของการเข้าสู่ระบบสารสนเทศอย่างเคร่งครัด

ตอนที่ ๕ การควบคุมการเข้าถึงระบบปฏิบัติการ

หน้า ๑๐

ข้อ ๔๖ ติดตั้งโปรแกรมช่วยบริหารจัดการและกำหนดชื่อผู้ใช้งาน และรหัสผ่าน ให้กับผู้ใช้งาน

ข้อ ๔๗ กำหนดขั้นตอนปฏิบัติการเข้าถึงระบบปฏิบัติการมีแนวปฏิบัติ ดังนี้

(๑) ไม่ให้ระบบแสดงรายละเอียดสำคัญหรือความผิดพลาดต่าง ๆ ของระบบ ก่อนที่การเข้าสู่ระบบจะเสร็จสมบูรณ์

(๒) ระบบสามารถยุติการเชื่อมต่อจากเครื่องปลายทางได้ เมื่อพบว่ามีการพยายามคาดเดารหัสผ่านจากเครื่องปลายทาง

ข้อ ๔๘ การยืนยันตัวตนของผู้ใช้งาน (user identification and authentication) ให้นำชื่อผู้ใช้งาน และรหัสผ่าน มาตรวจสอบสิทธิของผู้ใช้งานบนระบบ AD (Active Directory) เพื่อเข้าสู่ระบบ

ข้อ ๔๙ ส่วนงานต้องมีระบบบริหารจัดการรหัสผ่านที่สามารถทำงานเชิงโต้ตอบ (interactive) หรือมีการทำงานในลักษณะอัตโนมัติ ซึ่งเอื้อต่อการกำหนดรหัสผ่านที่มีคุณภาพ โดยมีแนวปฏิบัติดังนี้

(๑) มีระบบบริหารจัดการรหัสผ่าน ผ่านระบบเครือข่ายสารสนเทศของมหาวิทยาลัย

(๒) ผู้ใช้งานต้องเปลี่ยนรหัสผ่านของตนเองในครั้งแรกที่มีการเข้าสู่ระบบ

(๓) มีระบบแจ้งระดับความปลอดภัยของรหัสผ่าน

ข้อ ๕๐ การใช้งานโปรแกรมอรรถประโยชน์ ให้จำกัดและควบคุมการใช้งานโปรแกรมอรรถประโยชน์สำหรับโปรแกรมคอมพิวเตอร์ที่สำคัญ เนื่องจากการใช้งานโปรแกรมอรรถประโยชน์บางชนิดสามารถทำให้ผู้ใช้งานหลีกเลี่ยงมาตรการป้องกันทางด้านความมั่นคงปลอดภัยของระบบได้ เพื่อป้องกันการละเมิดหรือหลีกเลี่ยงมาตรการความมั่นคงปลอดภัยที่ได้กำหนดไว้หรือที่มีอยู่แล้ว ให้ผู้ดูแลระบบดำเนินการดังนี้

(๑) จำกัดสิทธิการเข้าถึง และกำหนดสิทธิอย่างรัดกุมในการอนุญาตให้ใช้โปรแกรมอรรถประโยชน์

(๒) กำหนดให้อนุญาตใช้งานโปรแกรมอรรถประโยชน์เป็นรายครั้งไป

(๓) จัดเก็บโปรแกรมอรรถประโยชน์ไว้ในสื่อภายนอก ถ้าไม่ต้องใช้งานเป็นประจำ

(๔) การเก็บบันทึกการเรียกใช้งานโปรแกรมเหล่านี้

(๕) กำหนดให้มีการถอดถอนโปรแกรมอรรถประโยชน์ที่ไม่จำเป็นออกจากระบบ

(๖) ตรวจสอบการละเมิดลิขสิทธิ์และจัดเก็บหลักฐานการใช้งาน

ตอนที่ ๖ การควบคุมการเข้าถึงโปรแกรมประยุกต์และสารสนเทศ (application and information access control)

ข้อ ๕๑ ผู้ดูแลระบบกำหนดการลงทะเบียนผู้ใช้งานใหม่ ในการใช้งานตามความจำเป็น รวมทั้งขั้นตอนปฏิบัติสำหรับการยกเลิกสิทธิการใช้งาน เช่น การลาออก หรือการเปลี่ยนตำแหน่งงานภายในส่วนงานเป็นต้น

ข้อ ๕๒ ผู้ดูแลระบบกำหนดสิทธิการใช้งานระบบสารสนเทศที่สำคัญ ได้แก่ ระบบคอมพิวเตอร์ โปรแกรมประยุกต์ จดหมายอิเล็กทรอนิกส์ ระบบเครือข่ายไร้สาย ระบบอินเทอร์เน็ต เป็นต้น โดยให้สิทธิเฉพาะการปฏิบัติงานในหน้าที่และต้องได้รับความเห็นชอบจากหัวหน้าส่วนงานเป็นลายลักษณ์อักษร รวมทั้งทำการทบทวนสิทธิดังกล่าวอย่างสม่ำเสมอ

ข้อ ๕๓ การกำหนดระยะเวลาในการเชื่อมต่อระบบสารสนเทศและแอพพลิเคชั่น

(๑) เมื่อผู้ใช้งานไม่มีการใช้งานระบบสารสนเทศ เกิน ๑๕ นาที (session/idle time out) ให้ตัดการเชื่อมต่อ

(๒) ระบบสารสนเทศหรือแอพพลิเคชั่นที่มีความเสี่ยงหรือมีความสำคัญสูง ให้ใช้งานต่อเนื่องได้ไม่เกิน ๖๐ นาที (limitation of connection time) แล้วตัดการเชื่อมต่อ

ข้อ ๕๔ ผู้ดูแลระบบบริหารจัดการสิทธิการใช้งานระบบและรหัสผ่านของบุคลากรสำหรับเข้าใช้โปรแกรมประยุกต์ดังนี้

(๑) กำหนดการเปลี่ยนแปลงและการยกเลิกรหัสผ่าน เมื่อผู้ใช้งานระบบลาออก หรือพ้นจากตำแหน่ง หรือยกเลิกการใช้งาน

(๒) กำหนดให้ผู้ใช้งานไม่บันทึกหรือเก็บรหัสผ่าน ไว้ในระบบคอมพิวเตอร์ในรูปแบบที่ไม่ได้ป้องกันการเข้าถึง

(๓) กำหนดชื่อผู้ใช้งาน หรือรหัสผู้ใช้งานต้องไม่ซ้ำกัน

(๔) ในกรณีมีความจำเป็นต้องให้สิทธิพิเศษกับผู้ใช้งานที่มีสิทธิสูงสุด ผู้ใช้งานนั้นจะต้องได้รับความเห็นชอบและอนุมัติจากหัวหน้าส่วนงาน โดยมีการกำหนดระยะเวลาการใช้งานและระงับการใช้งานทันที เมื่อพ้นระยะเวลาดังกล่าวหรือพ้นจากตำแหน่ง และมีการกำหนดสิทธิพิเศษที่ได้รับว่าเข้าถึงได้ถึงระดับใดได้บ้าง และต้องกำหนดให้รหัสผู้ใช้งานต่างจากรหัสผู้ใช้งานตามปกติ

ตอนที่ ๗ การเข้าถึงเครื่องคอมพิวเตอร์แม่ข่าย (server access control)

ข้อ ๕๕ ผู้ดูแลระบบควบคุมการติดตั้งซอฟต์แวร์ลงไปยังระบบเครื่องคอมพิวเตอร์แม่ข่ายที่ให้บริการ ให้ปฏิบัติดังนี้

(๑) การติดตั้งหรือปรับปรุงซอฟต์แวร์ของระบบสารสนเทศ ต้องได้รับอนุมัติจากหัวหน้าส่วนงานก่อนดำเนินการ

(๒) ให้ผู้ดูแลระบบเป็นผู้ทำหน้าที่ดำเนินการเปลี่ยนแปลงต่อระบบสารสนเทศของมหาวิทยาลัย

(๓) ควบคุมการเปลี่ยนแปลงและบันทึกการปฏิบัติงานสำหรับการเปลี่ยนแปลงต่อระบบสารสนเทศของมหาวิทยาลัย

(๔) ไม่ควรติดตั้งรหัสต้นฉบับ (source code) ของระบบสารสนเทศในเครื่องคอมพิวเตอร์แม่ข่ายที่ให้บริการนั้น ๆ

(๕) ให้จัดเก็บรหัสต้นฉบับและคลังโปรแกรม (library) สำหรับซอฟต์แวร์ของระบบสารสนเทศไว้ในสถานที่ที่มีความมั่นคงปลอดภัย

(๖) ให้ผู้ที่เกี่ยวข้องต้องทำการทดสอบด้านความมั่นคงปลอดภัยของระบบสารสนเทศอย่างครบถ้วน ก่อนดำเนินการติดตั้งบนเครื่องให้บริการระบบสารสนเทศ

(๗) ให้จัดเก็บซอฟต์แวร์เวอร์ชั่นเก่า ข้อมูลที่เกี่ยวข้องกับระบบสารสนเทศเดิมและขั้นตอนปฏิบัติที่เกี่ยวข้องของระบบสารสนเทศในกรณีที่จำเป็นต้องกลับไปใช้เวอร์ชั่นเก่าเหล่านั้น ตามระยะเวลาที่เหมาะสม

หน้า ๑๒

ข้อ ๕๖ ให้ทบทวนการทำงานของระบบสารสนเทศภายหลังจากที่เปลี่ยนแปลงระบบปฏิบัติการ ดังนี้

(๑) แจ้งให้ผู้ที่เกี่ยวข้องกับระบบสารสนเทศได้รับทราบเกี่ยวกับการเปลี่ยนแปลงระบบปฏิบัติการ เพื่อให้บุคคลเหล่านั้นมีเวลาเพียงพอในการดำเนินการทดสอบ และทบทวนก่อนที่จะดำเนินการเปลี่ยนแปลงระบบปฏิบัติการ

(๒) พิจารณาวางแผนดำเนินการเปลี่ยนแปลงระบบปฏิบัติการของระบบสารสนเทศ รวมทั้งวางแผนด้านงบประมาณที่จำเป็นต้องใช้ ในกรณีที่มหาวิทยาลัยต้องเปลี่ยนไปใช้ระบบปฏิบัติการใหม่

ข้อ ๕๗ การพัฒนาซอฟต์แวร์โดยหน่วยงานภายนอก

(๑) ควรจัดให้มีการควบคุมการพัฒนาซอฟต์แวร์ที่จัดจ้างจากบุคคลหรือหน่วยงานภายนอก

(๒) ให้ระบุว่าใครจะเป็นผู้มีสิทธิในสินทรัพย์ทางปัญญาสำหรับรหัสต้นฉบับในการพัฒนาซอฟต์แวร์ โดยผู้รับจ้างให้บริการจากภายนอก

(๓) ให้กำหนดเรื่องการสงวนสิทธิที่จะตรวจสอบด้านคุณภาพและความถูกต้องของซอฟต์แวร์ที่จะมีการพัฒนาโดยผู้ให้บริการภายนอก โดยระบุไว้ในสัญญาจ้างที่ทำกับผู้ให้บริการภายนอกนั้น

(๔) ให้มีการตรวจสอบโปรแกรมไม่ประสงค์ดี (malware) ในซอฟต์แวร์ต่าง ๆ ที่จะทำการติดตั้งก่อนดำเนินการติดตั้ง

ข้อ ๕๘ ส่วนงานต้องปฏิบัติตามมาตรการควบคุมช่องโหว่ทางเทคนิค ประกอบด้วย

(๑) กำหนดให้มีการจัดทำบัญชีของระบบสารสนเทศเพื่อใช้สำหรับกระบวนการบริหารจัดการ

(๒) ช่องโหว่ของระบบเหล่านั้น ควรมีการบันทึกดังนี้

(ก) ชื่อซอฟต์แวร์และเวอร์ชั่นที่ใช้งาน

(ข) สถานที่ที่ติดตั้ง

(ค) เครื่องที่ติดตั้ง

(ง) ผู้ผลิตซอฟต์แวร์

(จ) ข้อมูลสำหรับติดต่อผู้ผลิตหรือผู้พัฒนาซอฟต์แวร์นั้น ๆ

(๓) กำหนดให้มีการจัดการกับช่องโหว่สำคัญของระบบสารสนเทศอย่างเหมาะสม โดยทันที

(๔) กระบวนการบริหารจัดการช่องโหว่ของระบบสารสนเทศให้ผู้ดูแลระบบ ดำเนินการดังนี้

(ก) มีการเฝ้าระวังและติดตาม ประเมินความเสี่ยงสำหรับช่องโหว่ของระบบสารสนเทศรวมทั้งการประสานงานเพื่อให้ผู้ที่เกี่ยวข้องดำเนินการแก้ไขช่องโหว่ตามความเหมาะสม

(ข) กำหนดให้ผู้ที่เกี่ยวข้องดำเนินการประเมินความเสี่ยงเมื่อได้รับแจ้ง หรือทราบเกี่ยวกับช่องโหว่นั้น

(๕) ปิดการใช้งานหรือควบคุมการเข้าถึงพอร์ตที่ใช้สำหรับตรวจสอบ และปรับแต่งระบบให้ใช้งานได้อย่างจำกัดระยะเวลาเท่าที่จำเป็น โดยต้องได้รับการอนุญาตจากผู้รับผิดชอบเป็นลายลักษณ์อักษร

หน้า ๑๓

ข้อ ๕๙ การบันทึกเหตุการณ์ที่เกี่ยวข้องกับการใช้งานระบบสารสนเทศ (audit logging) ให้ทำการบันทึกพฤติกรรมการใช้งาน (log) การเข้าถึงระบบสารสนเทศ ดังนี้

(๑) ข้อมูลชื่อบัญชีผู้ใช้งาน
(๒) ข้อมูลวันเวลาที่เข้าถึงระบบ
(๓) ข้อมูลวันเวลาที่ออกจากระบบ
(๔) ข้อมูลเหตุการณ์สำคัญที่เกิดขึ้น
(๕) ข้อมูลการล็อกอิน ทั้งที่สำเร็จและไม่สำเร็จ
(๖) ข้อมูลความพยายามในการเข้าถึงทรัพยากรทั้งที่สำเร็จและไม่สำเร็จ
(๗) ข้อมูลการเปลี่ยนแปลงการตั้งค่า (configuration) ของระบบ
(๘) ข้อมูลแสดงการใช้งานซอฟต์แวร์
(๙) ข้อมูลแสดงการเข้าถึงไฟล์และการกระทำกับไฟล์ เช่น เปิด ปิด เขียน หรืออ่านไฟล์ ฯลฯ
(๑๐) ข้อมูลเลขที่อยู่ไอพีที่เข้าถึง
(๑๑) ข้อมูลโพรโทคอลเครือข่ายที่ใช้
(๑๒) ข้อมูลแสดงการหยุดการทำงานของระบบป้องกันไวรัสคอมพิวเตอร์
(๑๓) ข้อมูลแสดงการสำรองข้อมูลไม่สำเร็จ

ตอนที่ ๘ การใช้งานเครื่องคอมพิวเตอร์ส่วนบุคคลและเครื่องคอมพิวเตอร์แบบพกพา

ข้อ ๖๐ ในการใช้งานทั่วไป ให้ผู้ใช้งานปฏิบัติดังนี้

(๑) เครื่องคอมพิวเตอร์ที่มหาวิทยาลัยอนุญาตให้ผู้ใช้งานใช้งานเป็นสินทรัพย์ของมหาวิทยาลัย ดังนั้น ผู้ใช้งานจึงควรใช้งานเครื่องคอมพิวเตอร์อย่างมีประสิทธิภาพเพื่องานของมหาวิทยาลัย

(๒) โปรแกรมที่ได้ถูกติดตั้งลงบนเครื่องคอมพิวเตอร์ของมหาวิทยาลัย ที่เป็นโปรแกรมที่มหาวิทยาลัยได้ซื้อลิขสิทธิ์มาอย่างถูกต้องตามกฎหมาย ดังนั้นห้ามผู้ใช้งานคัดลอกโปรแกรมต่าง ๆ และนำไปติดตั้งบนเครื่องคอมพิวเตอร์ส่วนตัวหรือแก้ไข หรือนำไปให้ผู้อื่นใช้งานโดยผิดกฎหมาย

(๓) ไม่อนุญาตให้ผู้ใช้งานทำการติดตั้งและแก้ไขเปลี่ยนแปลงโปรแกรมในเครื่องคอมพิวเตอร์ของมหาวิทยาลัย

(๔) การเคลื่อนย้ายหรือส่งเครื่องคอมพิวเตอร์ไปตรวจซ่อมจะต้องดำเนินการโดยเจ้าหน้าที่ของส่วนงานหรือผู้รับจ้างในการบำรุงรักษาเครื่องคอมพิวเตอร์และอุปกรณ์ที่ได้ทำสัญญากับมหาวิทยาลัยเท่านั้น

(๕) ก่อนการใช้งานสื่อบันทึกพกพาต่าง ๆ ต้องมีการตรวจสอบเพื่อหาไวรัสคอมพิวเตอร์โดยโปรแกรมป้องกันไวรัส

(๖) ผู้ใช้งานมีหน้าที่รับผิดชอบในการป้องกันการสูญหาย เช่น ควรล็อกเครื่องขณะที่ไม่ได้ใช้งาน ไม่วางเครื่องทิ้งไว้ในที่สาธารณะ หรือในบริเวณที่มีความเสี่ยงต่อการสูญหาย ฯลฯ

(๗) ห้ามผู้ใช้งานทำการเปลี่ยนแปลงแก้ไขส่วนประกอบย่อย (sub component) ที่ติดตั้งอยู่ภายในรวมถึงแบตเตอรี่

หน้า ๑๔

ตอนที่ ๙ การบริหารจัดการสินทรัพย์ (ip, webhost, storage, network equipment, data)

ข้อ ๖๑ ผู้ใช้งานต้องไม่เข้าไปในห้องปฏิบัติการเครือข่ายคอมพิวเตอร์ ที่เป็นเขตหวงห้าม โดยเด็ดขาด เว้นแต่ได้รับอนุญาตจากผู้ดูแลระบบ

ข้อ ๖๒ ผู้ใช้งานต้องไม่นำอุปกรณ์หรือชิ้นส่วนใดออกจากห้องปฏิบัติการเครือข่ายคอมพิวเตอร์ เว้นแต่ได้รับอนุญาตจากผู้ดูแลระบบ

ข้อ ๖๓ ผู้ใช้งานต้องไม่นำเครื่องมือ หรืออุปกรณ์อื่นใด เชื่อมเข้าเครือข่ายเพื่อการประกอบธุรกิจส่วนบุคคล

ข้อ ๖๔ ผู้ใช้งานต้องไม่คัดลอกหรือทำสำเนาแฟ้มข้อมูลที่มีลิขสิทธิ์กำกับการใช้งาน ก่อนได้รับอนุญาต และผู้ใช้งานต้องไม่ใช้ หรือลบแฟ้มข้อมูลของผู้อื่น ไม่ว่ากรณีใด ๆ

ข้อ ๖๕ ผู้ใช้งานต้องทำลายข้อมูลสำคัญในอุปกรณ์สื่อบันทึกข้อมูล แฟ้มข้อมูล ก่อนที่จะทำลายหรือจำหน่ายอุปกรณ์ดังกล่าว เพื่อป้องกันไม่ให้มีการเข้าถึงข้อมูลสำคัญนั้นได้

ข้อ ๖๖ ผู้ใช้งานมีหน้าที่ต้องรับผิดชอบต่อสินทรัพย์ที่ส่วนงานมอบไว้ให้ใช้งานเสมือนหนึ่งเป็นสินทรัพย์ของผู้ใช้งานเอง โดยบรรดารายการสินทรัพย์ ที่ผู้ใช้งานต้องรับผิดชอบ การรับหรือคืนสินทรัพย์จะถูกบันทึกและตรวจสอบทุกครั้งโดยเจ้าหน้าที่ที่ส่วนงานมอบหมาย กรณีทำงานนอกสถานที่ผู้ใช้งานต้องดูแลและรับผิดชอบสินทรัพย์ของส่วนงานที่ได้รับมอบหมาย

ข้อ ๖๗ ผู้ใช้งานต้องจัดเก็บเอกสาร สื่อบันทึกข้อมูล และเครื่องคอมพิวเตอร์ ภายหลังจากการใช้งานแล้ว ในสถานที่ที่มีการป้องกันการเข้าถึงโดยผู้ซึ่งไม่มีสิทธิ

ข้อ ๖๘ ผู้ใช้งานมีหน้าที่ต้องชดใช้ค่าเสียหายไม่ว่าสินทรัพย์นั้นจะชำรุด หรือสูญหายตามมูลค่าสินทรัพย์ หากความเสียหายนั้นเกิดจากความประมาทของผู้ใช้งาน

ข้อ ๖๙ ผู้ใช้งานต้องไม่ให้ผู้อื่นยืมเครื่องคอมพิวเตอร์พกพา ไม่ว่าในกรณีใด ๆ เว้นแต่การยืมนั้นได้รับการอนุมัติเป็นลายลักษณ์อักษรจากหัวหน้าส่วนงาน

ข้อ ๗๐ ผู้ใช้งานมีสิทธิใช้สินทรัพย์และระบบสารสนเทศต่าง ๆ ที่ส่วนงานจัดเตรียมไว้ให้ใช้งานโดยมีวัตถุประสงค์เพื่อการใช้งานของส่วนงานเท่านั้น ห้ามผู้ใช้งานนำสินทรัพย์และระบบสารสนเทศต่าง ๆ ไปใช้ในกิจกรรมที่ส่วนงานไม่ได้กำหนด หรือทำให้เกิดความเสียหายต่อมหาวิทยาลัย

ข้อ ๗๑ ความเสียหายใด ๆ ที่เกิดจากการละเมิดตามข้อปฏิบัติข้างต้น ให้ถือเป็นความผิดส่วนบุคคลโดยผู้ใช้งานต้องรับผิดชอบต่อความเสียหายที่เกิดขึ้น

ตอนที่ ๑๐ การควบคุมการใช้อินเทอร์เน็ต

ข้อ ๗๒ ผู้ดูแลระบบกำหนดเส้นทางการเชื่อมต่อระบบคอมพิวเตอร์เพื่อการเข้าใช้งานระบบอินเทอร์เน็ตที่เชื่อมต่อผ่านระบบรักษาความปลอดภัยที่มหาวิทยาลัยจัดสรรไว้เท่านั้น ห้ามผู้ใช้งานทำการเชื่อมต่อระบบคอมพิวเตอร์ผ่านช่องทางอื่นที่ไม่ได้รับการอนุมัติจากผู้อำนวยการสำนักคอมพิวเตอร์

ข้อ ๗๓ การใช้งานเครื่องคอมพิวเตอร์จะต้องมีการติดตั้งโปรแกรมป้องกันไวรัสคอมพิวเตอร์และทำการอุดช่องโหว่ก่อนที่จะทำการเชื่อมต่อระบบอินเทอร์เน็ตผ่านเว็บบราวเซอร์

หน้า ๑๕

ข้อ ๗๔ ผู้ใช้งานต้องเข้าถึงแหล่งข้อมูลตามสิทธิที่ได้รับตามหน้าที่ความรับผิดชอบ เพื่อประสิทธิภาพของระบบเครือข่ายและความปลอดภัยทางข้อมูลของมหาวิทยาลัย และต้องไม่ใช้ระบบอินเทอร์เน็ตของมหาวิทยาลัยเพื่อหาประโยชน์ในเชิงพาณิชย์เป็นการส่วนบุคคล และทำการเข้าสู่เว็บไซต์ที่ไม่เหมาะสม เช่น เว็บไซต์ที่ขัดต่อศีลธรรม เว็บไซต์ที่มีเนื้อหาอันอาจกระทบกระเทือน หรือเป็นภัยต่อความมั่นคงแห่งชาติ ศาสนา พระมหากษัตริย์ หรือเว็บไซต์ที่เป็นภัยต่อสังคมหรือละเมิดสิทธิของผู้อื่น หรือข้อมูลที่อาจก่อความเสียหายให้กับมหาวิทยาลัย

ข้อ ๗๕ ห้ามผู้ใช้งานเปิดเผยข้อมูลสำคัญที่เป็นความลับเกี่ยวกับงานของมหาวิทยาลัย ที่ยังไม่ได้ประกาศอย่างเป็นทางการผ่านระบบอินเทอร์เน็ต

ข้อ ๗๖ ผู้ใช้งานต้องระมัดระวังการดาวน์โหลดโปรแกรมใช้งานจากระบบอินเทอร์เน็ต ซึ่งรวมถึงการดาวน์โหลดการปรับปรุงโปรแกรมต่าง ๆ ต้องเป็นไปโดยไม่ละเมิดลิขสิทธิ์หรือสินทรัพย์ทางปัญญา

ข้อ ๗๗ ในการใช้งานกระดานสนทนาอิเล็กทรอนิกส์ ผู้ใช้งานต้องไม่เปิดเผยข้อมูลที่สำคัญและเป็นความลับของมหาวิทยาลัย ไม่เสนอความคิดเห็น หรือใช้ข้อความที่ยั่วยุ ให้ร้าย ที่จะทำให้เกิดความเสื่อมเสียต่อชื่อเสียงของมหาวิทยาลัย การทำลายความสัมพันธ์กับบุคลากรของส่วนงานอื่น ๆ

ข้อ ๗๘ หลังจากใช้งานระบบอินเทอร์เน็ตเสร็จแล้ว ให้ผู้ใช้งานทำการปิดเว็บบราวเซอร์ เพื่อป้องกันการเข้าใช้งานโดยบุคคลอื่น ๆ

ตอนที่ ๑๑ การใช้งานเครือข่ายสังคมออนไลน์ (social network)

ข้อ ๗๙ อนุญาตให้ใช้งานเครือข่ายสังคมออนไลน์ในรูปแบบและลักษณะตามที่มหาวิทยาลัยได้กำหนดไว้เท่านั้น

ข้อ ๘๐ ผู้ใช้งานที่ใช้งานเครือข่ายสังคมออนไลน์ ที่อาจมีผลกระทบกับมหาวิทยาลัย ผู้ใช้งานจะต้องแจ้งต่อผู้ดูแลระบบโดยเร็วที่สุด เพื่อดำเนินการตามความเหมาะสม

ตอนที่ ๑๒ การควบคุมการใช้งานระบบจดหมายอิเล็กทรอนิกส์

ข้อ ๘๑ การใช้งานจดหมายอิเล็กทรอนิกส์ของมหาวิทยาลัย ให้ผู้ใช้งานปฏิบัติดังนี้

(๑) ใช้จดหมายอิเล็กทรอนิกส์ ของมหาวิทยาลัยเพื่อติดต่องานของมหาวิทยาลัยเท่านั้น

(๒) ไม่ควรใช้บัญชีจดหมายอิเล็กทรอนิกส์ของผู้อื่นเพื่ออ่าน รับส่งข้อความ ยกเว้นจะได้รับการยินยอมจากเจ้าของจดหมายอิเล็กทรอนิกส์ และให้ถือว่าเจ้าของจดหมายอิเล็กทรอนิกส์เป็นผู้รับผิดชอบต่อการใช้งานต่าง ๆ ในจดหมายอิเล็กทรอนิกส์ของตน

(๓) หลังจากการใช้งาน ควรลงชื่อออกจากระบบทุกครั้ง เพื่อป้องกันบุคคลอื่นเข้าใช้งานระบบ

(๔) ในกรณีที่ต้องการส่งข้อมูลที่เป็นความลับ ผู้ใช้งานไม่ควรระบุความสำคัญของข้อมูลลงในหัวข้อจดหมายอิเล็กทรอนิกส์

(๕) ควรตรวจสอบและลบจดหมายอิเล็กทรอนิกส์ของตนเองทุกวัน เพื่อลดปริมาณการใช้พื้นที่ของระบบจดหมายอิเล็กทรอนิกส์ให้เหลือจำนวนน้อยที่สุด

หน้า ๑๖

(๖) ผู้ใช้งานมีหน้าที่จะต้องรักษาชื่อผู้ใช้งาน และรหัสผ่าน เป็นความลับไม่ให้รั่วไหลไปถึงบุคคลที่ไม่เกี่ยวข้อง

(๗) ปฏิบัติตามวิธีการใช้งานรหัสผ่านที่ได้กำหนดไว้อย่างเคร่งครัด

ข้อ ๘๒ แนวทางการควบคุมการใช้งานระบบจดหมายอิเล็กทรอนิกส์สำหรับผู้ดูแลระบบ มีดังนี้

(๑) กำหนดสิทธิเข้าถึงระบบจดหมายอิเล็กทรอนิกส์ของมหาวิทยาลัยให้เหมาะสมกับการเข้าใช้บริการของผู้ใช้ระบบและหน้าที่ความรับผิดชอบของผู้ใช้งาน

(๒) กำหนดจำนวนครั้งที่ยอมให้ผู้ใช้งานใส่รหัสผ่าน ผิดพลาดได้ไม่เกิน ๕ ครั้ง

(๓) ทบทวนสิทธิการเข้าใช้งานและปรับปรุงบัญชีผู้ใช้งาน ปีละ ๑ ครั้ง หรือเมื่อมีการเปลี่ยนแปลง เช่น มีการลาออกหรือเปลี่ยนแปลงตำแหน่ง โอน ย้าย หรือสิ้นสุดการจ้าง ฯลฯ

(๔) ควบคุมการเข้าถึงระบบตามแนวทางการบริหารจัดการเข้าถึงผู้ใช้งานที่ได้กำหนดไว้อย่างเคร่งครัด

ตอนที่ ๑๓ การบริหารจัดการซอฟต์แวร์และลิขสิทธิ์ และการป้องกันโปรแกรมไม่ประสงค์ดี (software licensing and intellectual property and preventing malware)

ข้อ ๘๓ มหาวิทยาลัยให้ความสำคัญต่อเรื่องสินทรัพย์ทางปัญญา ดังนั้นซอฟต์แวร์ที่ส่วนงานอนุญาตให้ใช้งานหรือที่ส่วนงานมีลิขสิทธิ์ ผู้ใช้งานสามารถขอใช้งานได้ตามหน้าที่ความจำเป็น และห้ามผู้ใช้งานทำการติดตั้งหรือใช้งานซอฟต์แวร์อื่นใดที่ไม่มีลิขสิทธิ์ หากมีการตรวจสอบพบความผิดฐานละเมิดลิขสิทธิ์ ให้ถือว่าเป็นความผิดส่วนบุคคล ผู้ใช้งานจะต้องรับผิดชอบแต่เพียงผู้เดียว

ข้อ ๘๔ ซอฟต์แวร์ที่ส่วนงานได้จัดเตรียมไว้ให้ผู้ใช้งาน ถือเป็นสิ่งจำเป็นต่อการทำงาน ห้ามผู้ใช้งานทำการถอดถอน เปลี่ยนแปลง แก้ไข หรือทำสำเนาเพื่อนำไปใช้งานที่อื่น ๆ ยกเว้นได้รับการอนุญาตจากหัวหน้าส่วนงานหรือผู้ที่ได้รับมอบหมายที่มีสิทธิในลิขสิทธิ์

ข้อ ๘๕ เครื่องคอมพิวเตอร์ของผู้ใช้งานต้องติดตั้งโปรแกรมป้องกันไวรัสคอมพิวเตอร์ เว้นแต่เครื่องคอมพิวเตอร์นั้นเป็นเครื่องเพื่อการศึกษา โดยต้องได้รับอนุญาตจากหัวหน้าส่วนงาน

ข้อ ๘๖ บรรดาข้อมูล ไฟล์ ซอฟต์แวร์ หรือสิ่งอื่นใด ที่ได้รับจากผู้ใช้งานอื่นต้องได้รับการตรวจสอบไวรัสคอมพิวเตอร์และโปรแกรมไม่ประสงค์ดีก่อนนำมาใช้งานหรือเก็บบันทึกทุกครั้ง

ข้อ ๘๗ ผู้ใช้งานต้องทำการปรับปรุงข้อมูล สำหรับตรวจสอบและปรับปรุงระบบปฏิบัติการ (update patch) ให้ใหม่เสมอ เพื่อเป็นการป้องกันความเสียหายที่อาจเกิดขึ้น

ข้อ ๘๘ ผู้ใช้งานต้องพึงระวังไวรัสคอมพิวเตอร์และโปรแกรมไม่ประสงค์ดีตลอดเวลา รวมทั้งเมื่อพบสิ่งผิดปกติผู้ใช้งานต้องแจ้งเหตุแก่ผู้ดูแลระบบ

ข้อ ๘๙ เมื่อผู้ใช้งานพบว่าเครื่องคอมพิวเตอร์ติดไวรัส ผู้ใช้งานต้องไม่เชื่อมต่อเครื่องคอมพิวเตอร์เข้าสู่เครือข่าย และต้องแจ้งแก่ผู้ดูแลระบบ

ข้อ ๙๐ ห้ามลักลอบทำสำเนา เปลี่ยนแปลง ซึ่งข้อมูล ข้อความ เอกสาร หรือสิ่งใด ๆ ที่เป็นสินทรัพย์ทางปัญญาของมหาวิทยาลัย เว้นแต่จะได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้มีอำนาจลงนาม

ข้อ ๙๑ การบริหารจัดการซอฟต์แวร์ที่พัฒนาโดยส่วนงานภายนอก (outsourced software development) ส่วนงานต้องปฏิบัติดังนี้

(๑) จัดให้มีการควบคุมโครงการพัฒนาซอฟต์แวร์โดยผู้รับจ้างให้บริการจากภายนอก

(๒) พิจารณาระบุว่าใครจะเป็นผู้มีสิทธิในสินทรัพย์ทางปัญญาสำหรับรหัสต้นฉบับในการพัฒนาซอฟต์แวร์โดยผู้รับจ้างให้บริการจากภายนอก

(๓) พิจารณากำหนดเรื่องการสงวนสิทธิที่จะตรวจสอบด้านคุณภาพและความถูกต้องของซอฟต์แวร์ที่จะมีการพัฒนาโดยผู้ให้บริการภายนอก โดยระบุไว้ในสัญญาจ้างที่ทำกับผู้ให้บริการภายนอกนั้น

(๔) ให้มีการตรวจสอบโปรแกรมไม่ประสงค์ดี ในซอฟต์แวร์ต่าง ๆ ที่จะทำการติดตั้งก่อนดำเนินการติดตั้ง

(๕) หลังจากการส่งมอบการพัฒนาซอฟต์แวร์จากส่วนงานภายนอก ส่วนงานต้องดำเนินการเปลี่ยนรหัสผ่านต่าง ๆ

หน้า ๑๘

ตอนที่ ๑๔ การตรวจจับการบุกรุกและการป้องกันโปรแกรมไม่ประสงค์ดี (intrusion detection system / intrusion prevention system policy : ids/ips)

ข้อ ๙๒ ระบบทั้งหมดที่สามารถเข้าถึงได้จากอินเทอร์เน็ตหรือที่สาธารณะจะต้องผ่านการตรวจสอบจากระบบ ids/ips (คือระบบตรวจสอบการบุกรุก และตรวจสอบความปลอดภัยของเครือข่าย เพื่อป้องกันทรัพยากร ระบบสารสนเทศ และข้อมูลบนเครือข่ายภายในส่วนงานให้มีความมั่นคงปลอดภัย เป็นแนวทางการปฏิบัติเกี่ยวกับการตรวจสอบการบุกรุกเครือข่าย พร้อมกับบทบาทและความรับผิดชอบที่เกี่ยวข้อง)

ข้อ ๙๓ ผู้ดูแลระบบกำหนด policy ของ ids/ips ให้ครอบคลุมทุกโฮสต์ (host) ในเครือข่ายของส่วนงานและเครือข่ายข้อมูลทั้งหมด รวมถึงเส้นทางที่ข้อมูลอาจเดินทาง ซึ่งไม่อยู่ในเครือข่ายอินเทอร์เน็ตทุกเส้นทาง และบันทึกข้อมูลจราจร (log) ของการส่งผ่านข้อมูล

ข้อ ๙๔ ผู้ดูแลระบบตรวจสอบรูปแบบการให้บริการก่อนการติดตั้งและเปิดให้บริการระบบทั้งหมดใน dmz (demilitarized zone) โดยแยกระบบที่ไวต่อการรบกวน คือ Video Conference ไว้ในโซน dmz และไม่อนุญาตให้ใช้งานจากภายนอกองค์กร

ข้อ ๙๕ ผู้ดูแลระบบทำการ update patch/signature ของระบบ ids/ips เป็นประจำ

ข้อ ๙๖ ผู้ดูแลระบบตรวจสอบเหตุการณ์ ข้อมูลจราจร พฤติกรรมการใช้งาน กิจกรรม และบันทึกปริมาณข้อมูลเข้าใช้งานเครือข่ายเป็นประจำทุกวัน

ข้อ ๙๗ ผู้ดูแลระบบตรวจสอบข้อมูลประจำวันของเครื่องแม่ข่ายที่มีการติดตั้ง host-based ids หากพบพฤติกรรมการใช้งาน กิจกรรม หรือเหตุการณ์ทั้งหมด ที่มีความเสี่ยงต่อการบุกรุกการโจมตีระบบ พฤติกรรมที่น่าสงสัย หรือการพยายามเข้าระบบ ทั้งที่ประสบความสำเร็จและไม่ประสบความสำเร็จ ให้ผู้ดูแลระบบรายงานให้หัวหน้าส่วนงานทราบ และเก็บบันทึกข้อมูลจราจรไว้ไม่น้อยกว่า ๙๐ วัน

ข้อ ๙๘ ส่วนงานมีสิทธิในการยุติการเชื่อมต่อเครือข่ายของเครื่องคอมพิวเตอร์ที่มีพฤติกรรมเสี่ยงต่อการบุกรุกระบบ โดยไม่ต้องมีการแจ้งแก่ผู้ใช้งานล่วงหน้า หากการกระทำดังกล่าวเป็นการกระทำความผิดที่สอดคล้องกับกฎหมายว่าด้วยการกระทำความผิดเกี่ยวกับคอมพิวเตอร์ หรือเป็นการกระทำที่ส่งผลให้เกิดความเสียหายต่อข้อมูล และทรัพยากรระบบของส่วนงาน จะต้องถูกดำเนินคดีตามขั้นตอนของกฎหมาย

ตอนที่ ๑๕ การจัดเก็บข้อมูลจราจรคอมพิวเตอร์

ข้อ ๙๙ จัดเก็บข้อมูลจราจรทางคอมพิวเตอร์ ไว้ในสื่อเก็บข้อมูลที่สามารถรักษาความครบถ้วน ถูกต้อง และจะต้องระบุตัวบุคคลที่เข้าถึงสื่อดังกล่าวได้

ข้อ ๑๐๐ ห้ามผู้ดูแลระบบแก้ไขข้อมูลที่เก็บรักษาไว้ ยกเว้นผู้ตรวจสอบระบบสารสนเทศของมหาวิทยาลัย (it auditor) หรือบุคคลที่มหาวิทยาลัยมอบหมาย

ข้อ ๑๐๑ บันทึกการทำงานของระบบบันทึกการปฏิบัติงานของผู้ใช้งาน (application logs) และบันทึกรายละเอียดของระบบป้องกันการบุกรุก เช่น บันทึกการเข้า-ออกระบบ บันทึกการพยายามเข้าสู่ระบบ ฯลฯ เพื่อประโยชน์ในการใช้ตรวจสอบและต้องเก็บบันทึกไว้ ๙๐ วันนับตั้งแต่การใช้งานสิ้นสุดลง

ตอนที่ ๑๖ การควบคุมการเข้าถึงระบบเครือข่ายไร้สาย (wireless lan access control)

ข้อ ๑๐๒ ผู้ใช้งานที่ต้องการเข้าถึงระบบเครือข่ายไร้สายของมหาวิทยาลัย จะต้องทำการลงทะเบียนกับผู้ดูแลระบบ โดยจะต้องขออนุญาตเป็นลายลักษณ์อักษรและได้รับการพิจารณาอนุญาตจากผู้อำนวยการสำนักคอมพิวเตอร์

ข้อ ๑๐๓ ผู้ดูแลระบบจัดการควบคุมการเข้าถึงระบบเครือข่ายไร้สายโดย

(๑) ทำการลงทะเบียนกำหนดสิทธิของผู้ใช้งานการเข้าถึงระบบเครือข่ายไร้สายให้เหมาะสมกับหน้าที่ความรับผิดชอบในการปฏิบัติงานก่อนเข้าใช้ระบบเครือข่ายไร้สาย รวมทั้งมีการทบทวนสิทธิการเข้าถึงอย่างสม่ำเสมอ ทั้งนี้ระบบจะต้องได้รับอนุญาตจากผู้ดูแลระบบตามความจำเป็นในการใช้งาน

(๒) ทำการลงทะเบียนอุปกรณ์ทุกตัวที่ใช้ติดต่อระบบเครือข่ายไร้สาย

(๓) ควบคุมสัญญาณของอุปกรณ์กระจายสัญญาณ เพื่อป้องกันไม่ให้สัญญาณของอุปกรณ์รั่วไหลออกนอกพื้นที่ใช้งานระบบเครือข่ายไร้สาย และป้องกันไม่ให้ผู้โจมตีสามารถรับส่งสัญญาณจากภายนอกอาคารหรือบริเวณขอบเขตที่ควบคุมได้

(๔) ทำการเปลี่ยนค่า ssid ที่ถูกกำหนดเป็นค่าโดยปริยายมาจากผู้ผลิตทันทีที่นำอุปกรณ์กระจายสัญญาณมาใช้งาน

(๕) เปลี่ยนค่าชื่อบัญชีรายชื่อ และรหัสผ่านในการเข้าสู่ระบบสำหรับการตั้งค่าการทำงานของอุปกรณ์ไร้สาย และควรจะเลือกใช้ชื่อบัญชีรายชื่อและรหัสผ่านที่คาดเดาได้ยาก เพื่อป้องกันผู้โจมตีไม่ให้สามารถเดาหรือเจาะรหัสได้โดยง่าย

(๖) เข้ารหัสข้อมูลระหว่าง wireless lan client และอุปกรณ์กระจายสัญญาณเพื่อให้ยากต่อการดักจับและทำให้ปลอดภัยมากขึ้น

(๗) ติดตั้งอุปกรณ์ป้องกันการบุกรุก (firewall) ระหว่างเครือข่ายไร้สายกับเครือข่ายภายในมหาวิทยาลัย

(๘) ตรวจสอบความมั่นคงปลอดภัยของระบบเครือข่ายอย่างสม่ำเสมอเพื่อคอยตรวจสอบและบันทึกเหตุการณ์ที่น่าสงสัยที่เกิดขึ้นในระบบเครือข่ายไร้สาย และเมื่อพบการใช้งานระบบเครือข่ายไร้สายที่ผิดปกติให้รายงานต่อหัวหน้าส่วนงาน และ/หรือ ผู้อำนวยการสำนักคอมพิวเตอร์ทราบโดยทันที

หน้า ๒๐

หมวดที่ ๒ การจัดทำระบบสำรองของสารสนเทศ

ข้อ ๑๐๔ ผู้ดูแลระบบจัดทำแนวทางปฏิบัติในการสำรองและกู้คืนข้อมูล โดยจัดทำระบบสำรองที่เหมาะสมให้อยู่ในสภาพพร้อมใช้งาน ตามแนวทางต่อไปนี้

(๑) จัดทำบัญชีระบบสารสนเทศทั้งหมดของส่วนงาน พร้อมจัดทำระบบสำรองและจัดทำระบบแผนเตรียมพร้อมกรณีฉุกเฉินอย่างน้อยปีละ ๑ ครั้ง

(๒) สำรองข้อมูลของระบบสารสนเทศแต่ละระบบและกำหนดความถี่ในการสำรองข้อมูล หากระบบใดที่มีการเปลี่ยนแปลงบ่อยควรกำหนดให้มีความถี่ในการสำรองข้อมูลมากขึ้น โดยให้มีวิธีการสำรองข้อมูลดังนี้

(ก) กำหนดประเภทของข้อมูลที่ต้องทำการสำรองและความถี่ในการสำรอง

(ข) กำหนดรูปแบบการสำรองข้อมูลให้เหมาะสมกับข้อมูลที่จะทำการสำรอง

(ค) บันทึกข้อมูลที่เกี่ยวข้องกับกิจกรรมการสำรองข้อมูล ได้แก่ ผู้ดำเนินการวัน/เวลาชื่อข้อมูลที่สำรองสำเร็จ/ไม่สำเร็จ เป็นต้น

(ง) ตรวจสอบข้อมูลทั้งหมดของระบบว่ามีการสำรองข้อมูลไว้อย่างครบถ้วน

(จ) จัดเก็บข้อมูลที่สำรองนั้นในสื่อเก็บข้อมูล โดยพิมพ์ชื่อบนสื่อเก็บข้อมูลนั้นให้สามารถแสดงถึงระบบซอฟต์แวร์ วันที่เวลาที่สำรองข้อมูล และผู้รับผิดชอบในการสำรองข้อมูลไว้อย่างชัดเจน

(ฉ) จัดเก็บข้อมูลที่สำรองไว้นอกสถานที่ ระยะทางระหว่างสถานที่ที่จัดเก็บข้อมูลสำรองกับส่วนงานควรห่างกันเพียงพอ เพื่อไม่ให้ส่งผลกระทบต่อข้อมูลที่จัดเก็บไว้นอกสถานที่นั้นในกรณีที่เกิดภัยพิบัติ

(ช) ดำเนินการป้องกันทางกายภาพอย่างเพียงพอต่อสถานที่สำรองที่ใช้จัดเก็บข้อมูลนอกสถานที่

(ซ) ทดสอบบันทึกข้อมูลสำรองอย่างสม่ำเสมอ เพื่อตรวจสอบว่ายังคงสามารถเข้าถึงข้อมูลได้ตามปกติ

(ฌ) จัดทำขั้นตอนปฏิบัติสำหรับการกู้คืนข้อมูลที่เสียหายจากข้อมูลที่ได้สำรองเก็บไว้

(ญ) ตรวจสอบและทดสอบขั้นตอนปฏิบัติในการกู้คืนข้อมูลอย่างสม่ำเสมอ

(ฎ) กำหนดให้มีการเข้ารหัสข้อมูลกับข้อมูลที่ได้สำรองเก็บไว้

ข้อ ๑๐๕ ให้ส่วนงานจัดทำแผนเตรียมความพร้อมกรณีฉุกเฉินในกรณีที่ไม่สามารถดำเนินการด้วยวิธีการทางอิเล็กทรอนิกส์ เพื่อให้สามารถใช้งานสารสนเทศได้ตามปกติอย่างต่อเนื่อง ตามแนวทางต่อไปนี้

(๑) จัดทำแผนเตรียมความพร้อมกรณีฉุกเฉินในกรณีที่ไม่สามารถดำเนินการด้วยวิธีการทางอิเล็กทรอนิกส์ โดยมีรายละเอียดอย่างน้อยดังนี้

(ก) กำหนดหน้าที่และความรับผิดชอบของผู้ที่เกี่ยวข้องทั้งหมด

(ข) ประเมินความเสี่ยงสำหรับระบบที่มีความสำคัญเหล่านั้นและกำหนดมาตรการเพื่อลดความเสี่ยงเหล่านั้น

(ค) กำหนดขั้นตอนปฏิบัติในการกู้คืนระบบสารสนเทศ

(ง) กำหนดขั้นตอนปฏิบัติในการสำรองข้อมูลและทดสอบกู้คืนข้อมูลที่สำรองไว้

(จ) กำหนดช่องทางในการติดต่อกับผู้ให้บริการภายนอกเมื่อเกิดเหตุจำเป็นที่จะต้องติดต่อ

(ฉ) สร้างความตระหนักหรือให้ความรู้แก่เจ้าหน้าที่ผู้ที่เกี่ยวข้องกับขั้นตอนการปฏิบัติหรือสิ่งที่ต้องทำเมื่อเกิดเหตุเร่งด่วน

(๒) ทบทวนเพื่อปรับปรุงแผนเตรียมความพร้อมกรณีฉุกเฉินดังกล่าวให้สามารถปรับใช้ได้อย่างเหมาะสมและสอดคล้องกับการใช้งานตามภารกิจอย่างน้อยปีละ ๑ ครั้ง

ข้อ ๑๐๖ ส่วนงานกำหนดหน้าที่และความรับผิดชอบของบุคลากรที่ดูแลรับผิดชอบระบบสารสนเทศ ระบบสำรอง และการจัดทำแผนเตรียมพร้อมกรณีฉุกเฉินในกรณีที่ไม่สามารถดำเนินการด้วยวิธีการทางอิเล็กทรอนิกส์

ข้อ ๑๐๗ ส่วนงานทำการทดสอบสภาพพร้อมใช้งานของระบบสารสนเทศ ระบบสำรอง และระบบแผนเตรียมพร้อมกรณีฉุกเฉินอย่างน้อยปีละ ๑ ครั้ง

ข้อ ๑๐๘ ส่วนงานทบทวนระบบสารสนเทศ ระบบสำรอง และระบบแผนเตรียมพร้อมกรณีฉุกเฉิน ที่เพียงพอต่อสภาพความเสี่ยงที่ยอมรับได้ของแต่ละส่วนงาน อย่างน้อยปีละ ๑ ครั้ง

หน้า ๒๒

หมวดที่ ๓ การตรวจสอบและประเมินความเสี่ยงด้านสารสนเทศ

ข้อ ๑๐๙ การตรวจสอบและประเมินความเสี่ยงด้านสารสนเทศ ให้ผู้ดูแลระบบดำเนินการดังนี้

(๑) แต่งตั้งคณะกรรมการตรวจสอบและประเมินความเสี่ยงด้านสารสนเทศจากผู้เชี่ยวชาญทั้งภายในและภายนอกมหาวิทยาลัย

(๒) มีการตรวจสอบและประเมินความเสี่ยงด้านสารสนเทศที่อาจเกิดขึ้นกับระบบสารสนเทศ อย่างน้อยปีละ ๑ ครั้ง

ข้อ ๑๑๐ แนวทางในการตรวจสอบและประเมินความเสี่ยงที่ต้องดำเนินการ มีอย่างน้อยดังนี้

(๑) ระบุความเสี่ยงและผลกระทบให้สอดคล้องตามแผนบริหารความเสี่ยงของส่วนงาน

(๒) ทบทวนแผนแก้ไขปัญหาจากสถานการณ์ความไม่แน่นอน และภัยพิบัติที่อาจจะเกิดขึ้นกับระบบสารสนเทศอย่างน้อยปีละ ๑ ครั้ง

(๓) ประเมินความเสี่ยงให้คำนึงถึงองค์ประกอบดังนี้

(ก) ความรุนแรงของผลกระทบที่เกิดจากความเสี่ยงที่ระบุ

(ข) ภัยคุกคามหรือสิ่งที่อาจก่อให้เกิดเหตุการณ์ที่ระบุ รวมถึงความเป็นไปได้ที่จะเกิดขึ้น

(ค) จุดอ่อนหรือช่องโหว่ที่อาจถูกใช้ในการก่อให้เกิดเหตุการณ์ที่ระบุ

(๔) กำหนดมาตรการจัดการความเสี่ยงด้านสารสนเทศ อย่างน้อยดังนี้

(ก) กำหนดให้ผู้ตรวจสอบเข้าถึงข้อมูลที่จำเป็นต้องตรวจสอบแบบอ่านได้อย่างเดียว

(ข) ในกรณีที่จำเป็นต้องเข้าถึงข้อมูลในแบบอื่น ๆ ให้สร้างสำเนาสำหรับข้อมูลนั้นเพื่อให้คณะกรรมการตรวจสอบฯ ใช้งาน รวมทั้งทำลายหรือลบโดยทันทีที่ตรวจสอบเสร็จ หรือต้องจัดเก็บไว้โดยมีการป้องกันเป็นอย่างดี

(ค) กำหนดให้มีการระบุและจัดสรรทรัพยากรที่จำเป็นต้องใช้ในการตรวจสอบระบบบริหารจัดการความมั่นคงปลอดภัย

(ง) กำหนดให้มีการเฝ้าระวังการเข้าถึงระบบโดยผู้ตรวจสอบ รวมทั้งบันทึกข้อมูล log แสดงการเข้าถึงนั้น ซึ่งรวมถึงวันและเวลาที่เข้าถึงระบบงานที่สำคัญ ๆ

(จ) กำหนดหน้าที่และความรับผิดชอบของบุคลากรซึ่งดูแลรับผิดชอบระบบสารสนเทศ ระบบสำรอง และการจัดทำแผนเตรียมพร้อมกรณีฉุกเฉินในกรณีที่ไม่สามารถดำเนินการด้วยวิธีการทางอิเล็กทรอนิกส์

(๕) ผู้ดูแลระบบจัดทำแผนเตรียมความพร้อมกรณีฉุกเฉินในกรณีที่ไม่สามารถดำเนินการด้วยวิธีการทางอิเล็กทรอนิกส์ เพื่อให้สามารถใช้งานสารสนเทศได้ตามปกติอย่างต่อเนื่อง โดยระบุผู้รับผิดชอบและหน้าที่ความรับผิดชอบอย่างชัดเจน

(๖) ผู้ดูแลระบบทดสอบและปรับปรุงแผนเตรียมความพร้อมฉุกเฉินอยู่เสมอ เพื่อให้แผนมีความทันสมัยและสามารถใช้งานได้หากเกิดเหตุการณ์ขึ้นจริง

หน้า ๒๓

หมวดที่ ๔ การรักษาความปลอดภัยด้านกายภาพ สถานที่และสิ่งแวดล้อม

ข้อ ๑๑๑ ห้องควบคุมระบบเครือข่ายคอมพิวเตอร์ ส่วนงานต้องกำหนดให้ห้องมีลักษณะดังนี้

(๑) กำหนดเป็นเขตหวงห้ามเด็ดขาด หรือเขตหวงห้ามเฉพาะโดยพิจารณาตามความสำคัญแล้วแต่กรณี

(๒) ต้องเป็นพื้นที่ที่ไม่ตั้งอยู่ในบริเวณที่มีการผ่านเข้าออกของบุคคลเป็นจำนวนมาก

(๓) จะต้องไม่มีป้ายหรือสัญลักษณ์ที่บ่งบอกถึงการมีระบบสำคัญอยู่ภายในสถานที่ดังกล่าว

(๔) จะต้องปิดล็อกห้อง หรือใส่กุญแจประตูหน้าต่างเสมอ เมื่อไม่มีเจ้าหน้าที่ประจำอยู่

(๕) หากจำเป็นต้องใช้เครื่องโทรสารหรือเครื่องถ่ายเอกสาร ให้ติดตั้งแยกออกมาจากบริเวณดังกล่าว

(๖) ไม่อนุญาตให้ถ่ายรูปหรือบันทึกภาพเคลื่อนไหวในบริเวณดังกล่าว เป็นอันขาด

(๗) จัดพื้นที่สำหรับการส่งมอบผลิตภัณฑ์ โดยแยกจากบริเวณที่มีทรัพยากรสารสนเทศจัดตั้งไว้ เพื่อป้องกันการเข้าถึงระบบจากผู้ไม่ได้รับอนุญาต

ข้อ ๑๑๒ การกำหนดบริเวณที่ต้องมีการรักษาความมั่นคงปลอดภัย ส่วนงานต้องดำเนินการดังนี้

(๑) มีการจำแนกและกำหนดพื้นที่ของระบบสารสนเทศต่าง ๆ อย่างเหมาะสม เพื่อเฝ้าระวัง ควบคุม การรักษาความมั่นคงปลอดภัย จากผู้ที่ไม่ได้รับอนุญาต รวมทั้งป้องกันความเสียหายอื่น ๆ ที่อาจเกิดขึ้น

(๒) กำหนดและแบ่งแยกบริเวณพื้นที่ใช้งานระบบสารสนเทศให้ชัดเจน รวมทั้งจัดทำแผนผังแสดงตำแหน่งของพื้นที่ใช้งานและประกาศให้รับทราบทั่วกัน โดยการกำหนดพื้นที่ดังกล่าวอาจแบ่งออกได้เป็นพื้นที่ทำงานทั่วไป (general working area) พื้นที่ทำงานของผู้ดูแลระบบ (system administrator area) พื้นที่ติดตั้งอุปกรณ์ระบบสารสนเทศ (it equipment area) พื้นที่จัดเก็บข้อมูลคอมพิวเตอร์ (data storage area) และพื้นที่ใช้งานเครือข่ายไร้สาย (wireless lan coverage area) เป็นต้น

ข้อ ๑๑๓ การควบคุมการเข้าออก อาคารสถานที่ ส่วนงานต้องดำเนินการดังนี้

(๑) กำหนดสิทธิของผู้ใช้งาน ที่มีสิทธิผ่านเข้า-ออก และช่วงเวลาที่มีสิทธิในการผ่านเข้าออกในแต่ละพื้นที่ใช้งานระบบอย่างชัดเจน

(๒) การเข้าถึงอาคารของส่วนงาน ของบุคคลภายนอกหรือผู้มาติดต่อ เจ้าหน้าที่รักษาความปลอดภัย จะต้องให้มีการแลกบัตรที่ใช้ระบุตัวตนของบุคคลนั้น ๆ เช่น บัตรประจำตัวประชาชน ใบอนุญาตขับขี่ เป็นต้น แล้วทำการลงบันทึกข้อมูลบัตรในสมุดบันทึกและรับแบบฟอร์มการเข้าออกพร้อมกับบัตรผู้ติดต่อ

(๓) ให้มีการบันทึกวันและเวลาการเข้า-ออกพื้นที่สำคัญของผู้ที่มาติดต่อ

(๔) ผู้มาติดต่อต้องติดบัตรให้เห็นเด่นชัดตลอดระยะเวลาที่อยู่ภายในส่วนงาน

(๕) บริษัทผู้ได้รับการว่าจ้างต้องติดบัตรให้เห็นเด่นชัดตลอดระยะเวลาการทำงาน

(๖) จัดเก็บบันทึกการเข้า-ออกสำหรับพื้นที่หรือบริเวณที่มีความสำคัญ เช่น data center เป็นต้น เพื่อใช้ในการตรวจสอบในภายหลังเมื่อมีความจำเป็น

(๗) ดูแลผู้ที่มาติดต่อในพื้นที่หรือบริเวณที่มีความสำคัญจนกระทั่งเสร็จสิ้นภารกิจและจากไป เพื่อป้องกันการสูญหายของสินทรัพย์หรือป้องกันการเข้าถึงทางกายภาพโดยไม่ได้รับอนุญาต

หน้า ๒๔

(๘) มีกลไกการอนุญาตการเข้าถึงพื้นที่หรือบริเวณที่มีความสำคัญของบุคคลภายนอก และต้องมีเหตุผลที่เพียงพอในการเข้าถึงบริเวณดังกล่าว

(๙) สร้างความตระหนักให้ผู้ที่มาติดต่อจากภายนอกเข้าใจในกฎเกณฑ์หรือข้อกำหนดต่าง ๆ ที่ต้องปฏิบัติระหว่างที่อยู่ในพื้นที่หรือบริเวณที่มีความสำคัญ

(๑๐) มีการควบคุมการเข้าถึงพื้นที่ที่มีข้อมูลสำคัญจัดเก็บหรือประมวลผลอยู่

(๑๑) ไม่อนุญาตให้ผู้ไม่มีกิจเข้าไปในพื้นที่หรือบริเวณที่มีความสำคัญ เว้นแต่ได้รับการอนุญาต

(๑๒) มีการพิสูจน์ตัวตน เช่น การใช้บัตรรูด การใช้รหัสผ่าน เป็นต้น เพื่อควบคุมการเข้าออกในพื้นที่หรือบริเวณที่มีความสำคัญ ได้แก่ data center

(๑๓) จัดให้มีการดูแลและเฝ้าระวังการปฏิบัติงานของบุคคลภายนอกในขณะที่ปฏิบัติงานในพื้นที่หรือบริเวณที่มีความสำคัญ

(๑๔) จัดให้มีการทบทวน หรือยกเลิกสิทธิการเข้าถึงพื้นที่หรือบริเวณที่มีความสำคัญอย่างน้อยปีละ ๑ ครั้ง

ข้อ ๑๑๔ ระบบและอุปกรณ์สนับสนุนการทำงาน (supporting utilities) ส่วนงานต้องดำเนินการดังนี้

(๑) มีระบบสนับสนุนการทำงานของระบบสารสนเทศของส่วนงานที่เพียงพอต่อความต้องการใช้งาน ประกอบด้วย

(ก) ระบบสำรองกระแสไฟฟ้า (ups)

(ข) เครื่องกำเนิดกระแสไฟฟ้าสำรอง (generator)

(ค) ระบบระบายอากาศ

(ง) ระบบปรับอากาศ และควบคุมความชื้น

(๒) ให้มีการตรวจสอบหรือทดสอบระบบสนับสนุนเหล่านั้นอย่างน้อยปีละ ๑ ครั้ง เพื่อให้มั่นใจได้ว่าระบบทำงานตามปกติ และลดความเสี่ยงจากการล้มเหลวในการทำงานของระบบ

(๓) ติดตั้งระบบแจ้งเตือน เพื่อแจ้งเตือนกรณีที่ระบบสนับสนุนทำงานผิดปกติหรือหยุดการทำงาน

ข้อ ๑๑๕ การเดินสายไฟ สายสื่อสาร และสายเคเบิ้ลอื่น ๆ (cabling security) ส่วนงานต้องดำเนินการดังนี้

(๑) หลีกเลี่ยงการเดินสายสัญญาณเครือข่ายของส่วนงานในลักษณะที่ต้องผ่านเข้าไปในบริเวณที่มีบุคคลภายนอกเข้าถึงได้

(๒) ให้มีการร้อยท่อสายสัญญาณต่าง ๆ เพื่อป้องกันการดักจับสัญญาณ หรือการตัดสายสัญญาณ

(๓) ให้เดินสายสัญญาณสื่อสารและสายไฟฟ้าแยกออกจากกัน เพื่อป้องกันการแทรกแซงรบกวนของสัญญาณซึ่งกันและกัน

(๔) ทำป้ายชื่อสำหรับสายสัญญาณและบนอุปกรณ์ เพื่อป้องกันการตัดต่อสายสัญญาณผิดเส้น

(๕) จัดทำผังสายสัญญาณสื่อสารต่าง ๆ ให้ครบถ้วนและถูกต้อง

(๖) ห้องที่มีสายสัญญาณสื่อสารต่าง ๆ ให้ปิดใส่สลักให้สนิท เพื่อป้องกันการเข้าถึงของบุคคลภายนอก

หน้า ๒๕

(๗) พิจารณาใช้งานสายใยแก้วนำแสงแทนสายสัญญาณสื่อสารแบบเดิม เช่น สายสัญญาณแบบ coaxial cable เป็นต้น สำหรับระบบสารสนเทศที่สำคัญ

(๘) ดำเนินการสำรวจระบบสายสัญญาณสื่อสารทั้งหมด เพื่อตรวจหาการติดตั้งอุปกรณ์ดักจับสัญญาณโดยผู้ไม่ประสงค์ดี

ข้อ ๑๑๖ การบำรุงรักษาอุปกรณ์ (equipment maintenance) ส่วนงานต้องดำเนินการดังนี้

(๑) ให้มีกำหนดการบำรุงรักษาอุปกรณ์ตามรอบระยะเวลาที่แนะนำโดยผู้ผลิต

(๒) ปฏิบัติตามคำแนะนำในการบำรุงรักษาตามที่ผู้ผลิตแนะนำ

(๓) จัดเก็บบันทึกกิจกรรมการบำรุงรักษาอุปกรณ์สำหรับการให้บริการทุกครั้ง เพื่อใช้ในการตรวจสอบหรือประเมินในภายหลัง

(๔) จัดเก็บบันทึกปัญหาและข้อบกพร่องของอุปกรณ์ที่พบ เพื่อใช้ในการประเมินและปรับปรุงอุปกรณ์ดังกล่าว

(๕) ควบคุมและสอดส่องดูแลการปฏิบัติงานของผู้ให้บริการภายนอกที่มาทำการบำรุงรักษาอุปกรณ์ภายในส่วนงาน

(๖) จัดให้มีการอนุมัติสิทธิการเข้าถึงอุปกรณ์ที่มีข้อมูลสำคัญโดยผู้รับจ้างให้บริการจากภายนอก (ที่มาทำการบำรุงรักษาอุปกรณ์) เพื่อป้องกันการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต

ข้อ ๑๑๗ การนำสินทรัพย์ของส่วนงานออกนอกส่วนงาน (removal of property) ส่วนงานต้องดำเนินการดังนี้

(๑) ให้มีการขออนุญาตก่อนนำอุปกรณ์หรือสินทรัพย์นั้นออกไปใช้งานนอกส่วนงาน

(๒) กำหนดผู้รับผิดชอบในการเคลื่อนย้ายหรือนำอุปกรณ์ออกนอกส่วนงาน

(๓) กำหนดระยะเวลาของการนำอุปกรณ์ออกไปใช้งานนอกส่วนงาน

(๔) เมื่อมีการนำอุปกรณ์ส่งคืน ให้ตรวจสอบว่าสอดคล้องกับระยะเวลาที่อนุญาต และตรวจสอบการชำรุดเสียหายของอุปกรณ์ด้วย

(๕) บันทึกข้อมูลการนำอุปกรณ์ของส่วนงานออกไปใช้งานนอกส่วนงาน เพื่อเก็บไว้เป็นหลักฐานป้องกันการสูญหาย รวมทั้งบันทึกข้อมูลเพิ่มเติมเมื่อนำอุปกรณ์ส่งคืน

ข้อ ๑๑๘ การป้องกันอุปกรณ์ที่ใช้งานอยู่นอกส่วนงาน (security of equipment off-premises) ส่วนงานต้องดำเนินการดังนี้

(๑) กำหนดมาตรการความปลอดภัย เพื่อป้องกันความเสี่ยงจากการนำอุปกรณ์หรือสินทรัพย์ของส่วนงานออกไปใช้งาน เช่น การขนส่ง การเกิดอุบัติเหตุกับอุปกรณ์ เป็นต้น

(๒) ไม่ทิ้งอุปกรณ์หรือสินทรัพย์ของส่วนงานไว้โดยลำพังในที่สาธารณะ

(๓) เจ้าหน้าที่มีความรับผิดชอบดูแลอุปกรณ์หรือสินทรัพย์เสมือนเป็นสินทรัพย์ของตนเอง

ข้อ ๑๑๙ การกำจัดอุปกรณ์หรือการนำอุปกรณ์กลับมาใช้งานอีกครั้ง (secure disposal or re-use of equipment) ส่วนงานต้องดำเนินการดังนี้

(๑) ให้ทำลายข้อมูลสำคัญในอุปกรณ์ก่อนที่จะกำจัดอุปกรณ์ดังกล่าว

(๒) มีมาตรการหรือเทคนิคในการลบหรือเขียนข้อมูลทับบนข้อมูลที่มีความสำคัญ

หน้า ๒๖

ในอุปกรณ์สำหรับจัดเก็บข้อมูลก่อนที่จะอนุญาตให้ผู้อื่นนำอุปกรณ์นั้นไปใช้งานต่อ เพื่อป้องกันไม่ให้มีการเข้าถึงข้อมูลสำคัญนั้นได้

หมวดที่ ๕ การดำเนินการตอบสนองเหตุการณ์ด้านความมั่นคงปลอดภัย

ข้อ ๑๒๐ การวิเคราะห์รูปแบบการโจมตีของระบบตรวจหาการบุกรุก ผู้ดูแลระบบปฏิบัติดังนี้

(๑) ตรวจติดตามข้อมูลการใช้งานเครือข่าย เพื่อตรวจสอบความผิดปกติเป็นประจำทุกวัน และตอบสนองต่อการถูกบุกรุก

(๒) ติดตามข่าวสารใหม่ ๆ เรื่องการรูปแบบการโจมตีและภัยคุกคามของสารสนเทศ

ข้อ ๑๒๑ การตรวจหาช่องโหว่ของระบบเครือข่ายและระบบสารสนเทศ ผู้ดูแลระบบปฏิบัติดังนี้

(๑) ตรวจหาช่องโหว่ของระบบปฏิบัติการ และซอฟต์แวร์ประยุกต์ที่ให้บริการ

(๒) อุดช่องโหว่ (patch) ระบบปฏิบัติการ และซอฟต์แวร์ประยุกต์จากผู้พัฒนาผลิตภัณฑ์

ข้อ ๑๒๒ การกำหนดมาตรการในการป้องกันการบุกรุกและการโจมตี ผู้ดูแลระบบปฏิบัติดังนี้

(๑) ป้องกันทางด้านกายภาพ โดยกำหนดให้ห้องที่ใช้เป็นศูนย์ข้อมูลเป็นบริเวณที่ต้องรักษาความปลอดภัย โดยจัดให้มีการควบคุมและการเข้า-ออกสามารถทำได้เฉพาะผู้ที่ได้รับอนุญาตเท่านั้น

(๒) ปรับปรุงแนวทางในการตอบสนองต่อภัยคุกคามให้เป็นปัจจุบัน

(๓) ปรับปรุงนโยบายและกฎ (policy and rules) ของอุปกรณ์หรือซอฟต์แวร์ที่เกี่ยวข้องกับการตรวจจับและป้องกันภัยคุกคามให้เป็นปัจจุบันอยู่เสมอ

(๔) ประชาสัมพันธ์ผ่านสื่อทุกช่องทาง เช่น หนังสือเวียนแจ้ง จดหมายอิเล็กทรอนิกส์ เว็บไซต์ และสื่อทางสังคม (social media) เป็นต้น เพื่อให้ผู้ใช้งานทราบและตระหนักถึงภัยคุกคามด้านสารสนเทศใหม่ ๆ และปฏิบัติตามนโยบายฯ อย่างเคร่งครัด

หมวดที่ ๖ การสร้างความตระหนักในเรื่องการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ

ข้อ ๑๒๓ ผู้อำนวยการสำนักคอมพิวเตอร์นำเสนอ "นโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ" ในวาระการประชุมกรรมการบริหารมหาวิทยาลัย เพื่อสร้างความรู้ความเข้าใจและความตระหนักให้แก่ผู้บริหารระดับสูงถึงความสำคัญของความมั่นคงปลอดภัยด้านสารสนเทศ

ข้อ ๑๒๔ การสร้างความรู้ ความเข้าใจและความตระหนัก ดำเนินการดังนี้

(๑) จัดฝึกอบรมแนวปฏิบัติตามแนวนโยบายฯ ให้แก่ผู้ใช้งานสารสนเทศของมหาวิทยาลัยอย่างสม่ำเสมอ

(๒) จัดทำสื่อสำหรับฝึกอบรมในรูปแบบอิเล็กทรอนิกส์ (e-training) เรื่อง "นโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ" และเผยแพร่ผ่านทางเว็บไซต์

(๓) ประชาสัมพันธ์ให้ความรู้เรื่อง "นโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ" โดยผ่านสื่อต่าง ๆ เช่น ป้ายประกาศ เว็บไซต์ และสื่อทางสังคม เป็นต้น เพื่อให้ผู้ใช้งานตระหนักและปฏิบัติตามกฎหมายใด ๆ ที่ได้ประกาศใช้และนโยบายฯของมหาวิทยาลัยอย่างเคร่งครัด

หน้า ๒๘

หมวดที่ ๗ หน้าที่และความรับผิดชอบ

ข้อ ๑๒๕ ผู้บริหารระดับสูงสุด มีหน้าที่และความรับผิดชอบเชิงนโยบาย ดังนี้

(๑) กำหนดนโยบายด้านความมั่นคงปลอดภัยด้านสารสนเทศ

(๒) ให้ข้อเสนอแนะ คำปรึกษา กำกับ และติดตามให้ผู้ปฏิบัติงานดำเนินงานตามนโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ

(๓) รับผิดชอบต่อความเสี่ยง ความเสียหาย หรือ อันตรายที่เกิดขึ้นกับระบบสารสนเทศของมหาวิทยาลัย หรืออันตรายใด ๆ ที่เกิดขึ้นกับองค์กรใดหรือผู้หนึ่งผู้ใด อันเนื่องมาจากความบกพร่อง ละเลยการปฏิบัติตามนโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ

ข้อ ๑๒๖ ผู้ดูแลระบบ มีหน้าที่และความรับผิดชอบดังนี้

(๑) ปฏิบัติตามนโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ

(๒) แก้ปัญหาและประสานงานกับผู้ที่เกี่ยวข้อง เพื่อให้สารสนเทศมีความมั่นคงปลอดภัย

(๓) รับผิดชอบระบบสารสนเทศ ระบบสำรองข้อมูล และจัดทำแผนเตรียมความพร้อมฉุกเฉิน ในกรณีที่ไม่สามารถดำเนินการด้วยวิธีการทางอิเล็กทรอนิกส์

(๔) ตรวจสอบและทดสอบสภาพความพร้อมในการใช้งานของระบบสารสนเทศ

(๕) ตรวจสอบและประเมินความเสี่ยงของระบบสารสนเทศ

(๖) รับผิดชอบในการควบคุม ดูแล รักษาความปลอดภัย และบำรุงรักษาระบบสารสนเทศและการสื่อสารของมหาวิทยาลัย

(๗) รับผิดชอบต่อความเสี่ยง ความเสียหาย หรือ อันตรายที่เกิดขึ้นกับระบบสารสนเทศของมหาวิทยาลัย หรืออันตรายใด ๆ ที่เกิดขึ้นกับองค์กรใดหรือผู้หนึ่งผู้ใดอันเนื่องมาจากความบกพร่อง ละเลยการปฏิบัติตามนโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ

ข้อ ๑๒๗ ผู้พัฒนาระบบ มีหน้าที่และความรับผิดชอบดังนี้

(๑) ปฏิบัติตามนโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ

(๒) พัฒนาระบบสารสนเทศโดยให้มีความปลอดภัย และไม่เปิดเผยข้อมูลของมหาวิทยาลัย

(๓) ตรวจสอบและประเมินความเสี่ยงของระบบสารสนเทศอย่างสม่ำเสมอ เพื่อให้สามารถใช้งานระบบสารสนเทศได้ตามปกติและอย่างต่อเนื่อง

(๔) รับผิดชอบต่อความเสี่ยง ความเสียหาย หรือ อันตรายที่เกิดขึ้นกับระบบสารสนเทศของมหาวิทยาลัย หรืออันตรายใด ๆ ที่เกิดขึ้นกับองค์กรใดหรือผู้หนึ่งผู้ใดอันเนื่องมาจากความบกพร่อง ละเลยการปฏิบัติตามนโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ

หน้า ๒๙

ข้อ ๑๒๘ ผู้ใช้งาน มีหน้าที่และความรับผิดชอบดังนี้

(๑) ปฏิบัติตามนโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศของมหาวิทยาลัยบูรพา

(๒) รับผิดชอบต่อความเสี่ยง ความเสียหาย หรือ อันตรายที่เกิดขึ้นกับระบบสารสนเทศของมหาวิทยาลัย หรืออันตรายใด ๆ ที่เกิดขึ้นกับองค์กรใดหรือผู้หนึ่งผู้ใดอันเนื่องมาจากความบกพร่อง ละเลยการปฏิบัติตามนโยบายและแนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ

(๓) ปฏิบัติตามพระราชบัญญัติว่าด้วยการกระทำความผิดเกี่ยวกับคอมพิวเตอร์ พ.ศ. ๒๕๕๐ และที่แก้ไขเพิ่มเติม

ประกาศ ณ วันที่ ๒ พฤศจิกายน พ.ศ. ๒๕๖๐

(ลงชื่อ) สมนึก ธีระกุลพิศุทธิ์

(รองศาสตราจารย์สมนึก ธีระกุลพิศุทธิ์)

ผู้ปฏิบัติหน้าที่อธิการบดีมหาวิทยาลัยบูรพา

สำเนาถูกต้อง

(นายสยาม ศรีพั่ว)

นักวิชาการคอมพิวเตอร์

สรุป

ความปลอดภัยของระบบฐานข้อมูล เป็นการป้องกันฐานข้อมูลจากภัยคุกคามทั้งที่ตั้งใจและไม่ตั้งใจโดยการควบคุมด้วยระบบคอมพิวเตอร์และการควบคุมอื่น ๆ การสร้างความเสียหายให้กับฐานข้อมูลระบบ ได้แก่ 1) ข้อมูลสูญหายหรือเสียหายโดยเหตุสุดวิสัย 2) การโกงและการหลอกลวง 3) การสูญเสียความเป็นส่วนตัวหรือความลับของข้อมูล 4) การสูญเสียความเชื่อถือได้ของข้อมูล 5) การสูญเสียความพร้อมใช้ ซึ่งวัตถุประสงค์การรักษาความปลอดภัยข้อมูล ได้แก่ 1) รักษาความลับของข้อมูล 2) คงความถูกต้องสมบูรณ์ 3) ความพร้อมใช้ และ 4) ลดความเสี่ยงที่จะทำให้ข้อมูลเสียหาย

การควบคุมความปลอดภัยฐานข้อมูล ต้องดำเนินการตั้งแต่เริ่มต้นการพัฒนาระบบ เช่น การควบคุมการเข้าถึง (Access control) การยืนยันตัวบุคคล (Authentication) และการให้อำนาจหน้าที่ (Authorization) เพื่อระบุตัวบุคคลที่ติดต่อหรือทำธุรกรรมร่วมด้วยการตรวจสอบ (Auditing) การสร้างข้อมูลให้เป็นความลับหรือการเข้ารหัสข้อมูล (Encryption) การควบคุมความถูกต้องสมบูรณ์ (Integrity controls) การสำรองข้อมูล (Backups) และความปลอดภัยแอพลิเคชัน (Application security)

การควบคุมความปลอดภัยของฐานข้อมูลด้วยวิว (View) ทำให้ผู้ใช้เข้าถึงโครงสร้างเพียงบ้างส่วนของฐานข้อมูลที่กำหนดให้เท่านั้น เป็นการป้องกันไม่ให้ผู้ใช้ได้เข้าถึงข้อมูลทั้งหมดของฐานข้อมูล การสร้างวิว มีวัตถุประสงค์เพื่อให้ผู้ใช้สามารถเรียกใช้ข้อมูลจากวิวแทนการเรียกจาก Relation โดยตรงเพื่อความปลอดภัยของฐานข้อมูล โดยผู้ใช้จะได้รับสิทธิ์ในการเข้าถึงข้อมูลในแต่ละวิวที่แตกต่างกัน ช่วยให้เกิดความเป็นอิสระของข้อมูลและลดความซ้ำซ้อนของฐานข้อมูล

การรักษาความปลอดภัยฐานข้อมูล รวมถึงการวิเคราะห์และบริหารความเสี่ยงที่ประกอบด้วยภัยคุกคาม (Threat) ทั้งจากคนภายในองค์กร และคนภายนอกองค์กร และช่องโหว่หรือ จุดอ่อน (Vulnerability) การกำหนดการบังคับใช้นโยบาย (Policy) และการเฝ้าระวังเหตุการณ์อยู่ตลอดเวลา (Monitoring) ต้องมีมาตรการหรือการควบคุมความปลอดภัยที่มีประสิทธิภาพประกอบ ด้วยนโยบายวิธีปฏิบัติ และกระบวนการขององค์กรในการรักษาข้อมูลให้มีความถูกต้องและน่าเชื่อถือ

บรรณานุกรม

กิตติ ภักดีวัฒนะกุล และจันทรขจร แซอุน. (2552). *PHP* (พิมพครั้งที่ 12). กรุงเทพฯ: เคทีพี คอมพ แอนดคอนซัลท.

กิตติ ภักดีวัฒนะกุล และจำลอง ครูอุตสาหะ. (2550). *ระบบฐานข้อมูล = Database systems.* กรุงเทพฯ: เคทีพี คอมพ์ แอนด์ คอนซัลท์.

โกเมศ อัมพวัน. (2564). *เอกสารประกอบการสอน 886301-ฐานข้อมูล,* วันที่ค้นข้อมูล 20 มกราคม 2564, เข้าถึงได้จาก https://staff.informatics.buu.ac.th/~komate/886301/DB-Chapter-2.pdf

เจริญศักดิ์ รัตนวราห์ และฐิสันต ทิพยศุภธนนท. (2554). *PHP & MySQL*. กรุงเทพฯ: เน็ตดีไซน พับลิชชิ่ง

ชนวัฒน์ ศรีสอ้าน. (2550). *ฐานข้อมูล คลังข้อมูลและเหมืองข้อมูล.* ปทุมธานี: มหาวิทยาลัยรังสิต.

ชาญชัย ศุภอรรถกร. (2554). *จัดการฐานข้อมูลด้วย MySQL*. กรุงเทพฯ: ซิมพลิฟาย.

ชาญชัย ศุภอรรถกร. (2561). *สร้างเว็บแอพพลิเคชั่น PHP+MySQL สำหรับผู้เริ่มต้น* (พิมพ์ครั้งที่ 2). กรุงเทพฯ: รีไวว่า.

ณัฐพงษ์ วารีประเสริฐ. (2553). *การออกแบบฐานข้อมูลเชิงสัมพันธ์ (Relational database design).* กรุงเทพฯ: เคทีพี คอมพ์ แอนด์ คอนซัลท์.

เทพฤทธิ์ บัณฑิตวัฒนาวงศ์. (2554). *คู่มือเรียน วิเคราะห์และออกแบบฐานข้อมูล.* กรุงเทพฯ: โปรวิชั่น.

บัญชา ปะสีละเตสัง. (2553). *พัฒนาเว็บแอปพลิเคชั่นดวย PHP รวมกับ MySQL และ Dreamweaver*. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.

ปริญญา นอยดอนไพร. (2556). *การเขียนโปรแกรมบนเว็บดวย PHP รวมกับฐานขอมูล MySQL*. สุราษฎรธานี: มหาวิทยาลัยราชภัฏสุราษฎรธานี.

ปิยะดนัย วิเคียน. (2564). *ความรู้เบื้องต้นเกี่ยวกับระบบฐานข้อมูล.*วันที่ค้นข้อมูล 15 มีนาคม 2564 เข้าถึงได้จาก ttps://krupiyadanai.wordpress.com/tag/ระบบฐานข้อมูล

พนิดา พานิชกุล และณัฐพงษ์ วารีประเสริฐ. (2552). *การออกแบบ พัฒนา และดูแลระบบฐานข้อมูล (Database Systems-Design, Development and Management).* กรุงเทพฯ: เคทีพี คอมพ์ แอนด์ คอนซัลท์.

พจนานุกรม ฉบับราชบัณฑิตยสถาน. (2564). *ความหมาย "ข้อมูล"*, วันที่ค้นข้อมูล 15 มกราคม 2564, เข้าถึงได้จาก https://dictionary.orst.go.th/

พิมรินทร คีรินทร และอุไรวรรณ รักผกาวงศ. (2564). *เอกสารประกอบการสอนวิชา COMP233 Database System*, วันที่ค้นข้อมูล 15 มกราคม 2564, เข้าถึงได้จาก http://elearning.psru.ac.th/courses/256/%e0%b8%9a%e0%b8%97%e0%b8%97%e0%b8%b5%e0%b9%88%205.pdf

มหาวิทยาลัยบูรพา. (2560). *ประกาศนโยบายและแนวปฏิบัติการรักษาความมั่นคงปลอดภัยด้านสารสนเทศ ปี ๒๕๖๐*. วันที่ค้นข้อมูล 15 เมษายน 2564, เข้าถึงได้จาก https://www.buu.ac.th/download/policy2560.pdf

มณีโชติ สมานไทย. (2546). *คู่มือการออกแบบฐานข้อมูลและภาษา SQL ฉบับผู้เริ่มต้น.* นนทบุรี: อินโฟเพรส.

รัฐสิทธิ์ สุขะหุต. (2555). *ฐานข้อมูลเบื้องต้น (Database fundamentals).* เชียงใหม่: ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่.

วรกฤต แสนโภชน์. (2551). *235031 Database System*. คณะเทคโนโลยีสารสนเทศและการสื่อสาร มหาวิทยาลัยพะเยา, วันที่ค้นข้อมูล 15 มกราคม 2564, เข้าถึงได้จาก https://www.ict.up.ac.th/worrakits/Database.htm

วิกิพีเดีย สารานุกรมเสรี. (2021). *มายเอสคิวเอล.* วันที่ค้นข้อมูล 30 มกราคม 2564, เข้าถึงได้จาก https://th.wikipedia.org/wiki/มายเอสคิวเอล.

วิเชียร เปรมชัยสวัสดิ์. (2555). *ระบบฐานข้อมูล (ฉบับปรับปรุง)* (พิมพ์ครั้งที่ 18). กรุงเทพฯ: สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).

ศิริพร อ่องรุ่งเรือง. (2560). *ความมั่นคงปลอดภัยของสารสนเทศ.* วันที่ค้นข้อมูล 11 กุมภาพันธ์ 2564 เข้าถึงได้จาก ttp://www.cpe.ku.ac.th/~srp/603352/Chapter2_InfoSecurity.pptx

ศิริลักษณ์ โรจนกิจอำนวย. (2552). *การออกแบบและบริหารฐานข้อมูล.* กรุงเทพฯ: ดวงกมลสมัย.

สมลักษณ์ ละอองศรี. (2544). *เอกสารการสอนชุดวิชาการจัดการระบบฐานข้อมูล: เล่ม 1 หน่วยที่ 1-7, หน่วยที่ 2 ความรู้พื้นฐานเกี่ยวกับระบบฐานข้อมูล*. นนทบุรี: สำนักพิมพ์มหาวิทยาลัยสุโขทัยธรรมาธิราช.

สุจิตรา อดุลย์เกษม. (2553). *ระบบฐานข้อมูล (Database systems).* กรุงเทพฯ: ท้อป.

สำนักทดสอบทางการศึกษา. (2555) *การจัดระบบบริหารและสารสนเทศในสถานศึกษาตามกฎกระทรวงว่าด้วยระบบ หลักเกณฑ์ และวิธีการประกันคุณภาพการศึกษา พ.ศ. 2553*. กรุงเทพฯ: ชุมนุมสหกรณ์การเกษตรแห่งประเทศไทย.

โอภาส เอี่ยมสิริวงศ์. (2546). *การออกแบบและจัดการฐานข้อมูล.* กรุงเทพฯ: ซีเอ็ดยูเคชั่น.

โอภาส เอี่ยมสิริวงศ์. (2548). *การวิเคราะห์และออกแบบระบบ.* กรุงเทพฯ: ซีเอ็ดยูเคชั่น.

โอภาส เอี่ยมสิริวงศ์. (2558). *ระบบฐานข้อมูล (Database Systems).* กรุงเทพฯ: ซีเอ็ดยูเคชั่น.

AmplySoft. (2563). *DML (Data Manipulation Language) คืออะไร มีความสำคัญอย่างไร.* วันที่ค้นข้อมูล 25 มกราคม 2564, เข้าถึงได้จาก https://www.amplysoft.com/knowledge/DML

Chen, L. (2012). COMP5331: Knowledge Discovery and Data Mining. Retrieved May 3, 2022, from https://slideplayer.com/slide/9028096/

Compaq COBOL. (2022). User Manual. Retrieved May 3, 2022, from http://odl.sysworks.biz/disk$axpdocmar022/progtool/cobol27/cobum_017.htm

Coronel, C., Morris, S., & Rob, P. (2011). *Database Systems: Design, Implementation, and Management, Ninth Edition.* USA: Boston, MA.

Eso. (2021). *The servers and the client computer relationship*. Retrieved June 9, 2021, from https://www.firehousesoftware.com/webhelp/FHWeb/Content/FHWebInstallGuide/01_ClientServerRelationship.htm

Javatpoint. (2021). *Three schema Architecture*. Retrieved May 20, 2001, from https://www.javatpoint.com/dbms-three-schema-architecture

Mantra & WordPress. (2021). *AppServ: Apache + PHP + MYSQL*. Retrieved June 25, 2021, from https://www.appserv.org/th/

Thakur, S. (2016). *E-R Diagrams in DBMS: Components, Symbols, And Notations*. Retrieved June 25, 2021, from https://whatisdbms.com/wp-content/uploads/2016/06/Er-Symbols-and-Notations.jpg

Software School of Hunan University. (2006). Relation Example. Retrieved May 3, 2022, from https://images.slideplayer.com/16/5204315/slides/slide_7.jpg

Stair, R. & Reynolds, G. (2001). *Principles of Information Systems A Managerial Approach.* Retrieved June 15, 2021, from https://drive.uqu.edu.sa/_/fbshareef/files/principles%20of%20information%20systems%209th%20-tair,%20reynolds.pdf

Index

A

Accessible, 23
Accuracy, 2, 23
Application Programmer, 18
Application System, 21
AppServ, 127, 128, 129, 132, 141, 142, 143, 154, 156, 165
Attribute, 34, 35, 45, 46, 47, 52, 56, 57, 58, 61, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 77, 80, 82, 83, 85, 86, 87, 100, 101, 105, 106, 109, 111, 116, 134, 153
Auditing, 171, 172, 210
Availability, 168, 171

B

Backups, 171, 173, 210
Binary, 8, 20, 92
Binary Relationship, 92
Bit, 8, 20
Byte, 8, 20
Cardinality, 57, 64, 65, 66
Clarity, 23
Codd's 12 Rules, 62, 74
Column, 65, 67, 75
Complete, 23
Conceptual design, 33, 34
Conciseness, 23
Constraints, 36, 45, 47
Counting, 3, 19
Counting Data, 19
Cross-sectional, 6, 19

D

Data, 1, 2, 3, 4, 5, 6, 11, 15, 16, 17, 18, 19, 20, 28, 29, 51, 57, 60, 61, 62, 73, 75, 76, 78, 79, 102, 104, 108, 109, 110, 116, 117, 119, 120, 121, 122, 123, 124, 125, 168, 171, 173, 177
Data Backup and Recovery, 15
Data Redundancy, 11, 20, 51, 124
Data Security, 15, 123
Database, 1, 12, 13, 14, 15, 16, 19, 20, 22, 25, 27, 28, 30, 31, 33, 38, 39, 42, 43, 44, 51, 54, 56, 58, 59, 61, 62, 63, 64, 96, 97, 98, 99, 105, 117, 118, 120, 121, 122, 125, 126, 130, 131, 132, 133, 142, 167
Database Management System, 1, 12, 15, 16, 20, 27, 117, 118, 121
Database System, 1, 12, 20
Database Systems, 12
DBLC, 30, 31, 42
DBMS, 12, 15, 16, 17, 19, 20, 27, 33, 40, 45, 103, 104, 107, 117, 118, 121, 126, 136
DCL, 102, 104, 110, 116
DDL, 102, 104, 108, 116, 120
DDLC, 22, 25
Degree, 57, 64, 66, 92
DELETE, 40, 108, 110, 111, 175
Deletion Anomalies, 11
Discrete, 4
DML, 79, 102, 104, 108, 109, 110, 111, 116, 120
Domain, 57, 58, 64, 66, 68

E

Encryption, 171, 172, 210
Entity, 34, 35, 45, 46, 47, 48, 49, 50, 51, 52, 58, 61, 69, 77, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 100, 101, 111, 121

Entity Relationship, 45, 47, 81, 82, 87, 100
E-R Model, 35
F
Fabrication, 170
File, 1, 6, 7, 8, 9, 10, 11, 12, 20, 57, 65, 110
Flexible, 24
H
Hardware, 17, 121
HTML, 137, 138, 139, 140, 141, 144, 145, 155, 165
I
Incompatible, 11, 1
Information, 14, 21, 22, 23, 24, 34, 75, 168
Information System, 14, 21, 22, 23
Insert, 40, 108, 109, 111, 175
Insertion Anomalies, 11
Integrity, 2, 15, 16, 20, 57, 62, 73, 76, 77, 119, 123, 168, 169, 171, 173, 177, 210
Interception, 170
Interruption, 170
Interval Scale, 5, 19
Interview, 4, 19, 27
Interview Data, 19
K
Key, 9, 56, 57, 62, 66, 67, 68, 69, 70, 71, 72, 75, 87, 98, 100, 172
L
localhost, 132, 142, 143, 154, 156, 157, 159, 160, 163
Logical design, 33, 35
Loss of Availability, 169
Loss of Privacy or Confidentiality, 169
M
Maintenance, 27, 41, 125
Many to Many Relationships, 50, 91
Many to One Relationships, 50
Measurement, 3, 19
MIS, 14
Modification, 170
Modification Anomalies, 12
Multidimensional, 44, 59, 60, 61, 62
MySQL, 117
N
N-ary Relationship, 93
Network, 18, 44, 51, 54, 61, 62
Network Database, 62
Nominal Scale, 4, 19
Non-numeric Data, 13
Normal Forms, 82, 96, 101
Numeric Data, 13, 20
O
Observation, 3, 19, 27
Observation Data, 19
One to Many Relationships, 49, 90
One to One Relationships, 48, 88
Ordinal Scale, 5, 19
P
Persistency, 16
Personnel, 14, 17, 18, 172
PHP, 18, 127, 130, 131, 136, 137, 138, 139, 141, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 164, 165
phpMyAdmin, 117, 118, 127, 130, 132, 133, 134, 135, 136, 137, 141, 142, 143, 153, 154, 158, 165
Physical design, 33, 37
Primary Key, 66
Procedure, 17, 18, 123
Q
Qualitative, 4, 19

Quantitative, 4, 19

R

Ratio Scale, 5, 19
Record, 8, 20, 52, 53, 54, 55, 65
Relation, 57, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 104, 105, 106, 109, 134, 210
Relevant, 23
Reliability, 16, 20, 29
Reliable, 2, 23
Risk Assessment, 171
Row, 65, 67

S

Scalability, 16, 17
SDLC, 22, 25, 26, 35
Secure, 24
Segment, 10
Select, 40, 108, 109, 111, 175, 176
Software, 17, 18, 121
SQL, 16, 56, 63, 74, 80, 102, 103, 104, 111, 112, 113, 116, 117, 118, 121, 126, 127, 137, 153, 174, 176
System Administrator, 18
System Development Life Cycle, 22, 25, 26
Systems Analysis, 27
Systems Design, 27
Systems Implementation, 27
Systems Planning, 27
Text editor, 153
Theft and Fraud, 168
Threats, 168
Timely, 23
Time-series, 5, 19
Tuple, 56, 64, 65, 66, 67, 68, 69, 70, 71, 73, 77, 108, 109

U

Unary Relationship, 92
Up to Date, 23
UPDATE, 40, 108, 110, 111, 175

V

Verifiable, 23
View, 74, 75, 76, 78, 79, 81, 122, 176, 177, 210

W

VIEW, 105, 107, 108, 166, 175, 176
Web Server, 130, 131, 141

ประวัติผู้เขียน

ชื่อ-สกุล ดร.วีระพันธ์ พานิชย์

ตำแหน่ง อาจารย์

ประวัติการทำงาน 2540-2558 นักวิชาการโสตทัศนศึกษา
ศูนย์วิทยพัฒนา มหาวิทยาลัยสุโขทัยธรรมาธิราช จันทบุรี
2558-ปัจจุบัน อาจารย์ประจำ
ภาควิชานวัตกรรมและเทคโนโลยีการศึกษา
คณะศึกษาศาสตร์ มหาวิทยาลัยบูรพา

E-mail Address weerapun@go.buu.ac.th
weerapun@buu.ac.th

ประวัติการศึกษา

พ.ศ. 2539	ครุศาสตรบัณฑิต (เทคโนโลยีและนวัตกรรมทางการศึกษา) สถาบันราชภัฏจันทรเกษม
พ.ศ. 2548	การศึกษามหาบัณฑิต (เทคโนโลยีทางการศึกษา) มหาวิทยาลัยบูรพา
พ.ศ. 2554	ปรัชญาดุษฎีบัณฑิต (เทคโนโลยีการศึกษา) มหาวิทยาลัยบูรพา
พ.ศ. 2564	วิศวกรรมศาสตรมหาบัณฑิต (คอมพิวเตอร์และโทรคมนาคม) มหาวิทยาลัยธุรกิจบัณฑิตย์