

บทที่ 2

หลักการและทฤษฎี

2.1 ฐานข้อมูล

2.1.1 **ฐานข้อมูล** คือ กลุ่มข้อมูล (data) ที่เป็นข้อมูลข้อเท็จจริง (real fact) ที่ถูกนำมาเก็บรวบรวมไว้ในที่เดียวกันอย่างเป็นระบบเพื่อนำไปใช้ในวัตถุประสงค์อย่างใดอย่างหนึ่งโดยกลุ่มผู้ใช้งานตั้งแต่หนึ่งกลุ่มขึ้นไป ข้อมูลเหล่านี้อาจเป็นข้อเท็จจริงที่เกี่ยวกับบุคคล สิ่งของ สถานที่ หรือเหตุการณ์ใดๆ ซึ่งเป็นได้ทั้งตัวเลข ข้อความ รูปภาพ หรืออื่นๆ

จากคำจำกัดความข้างต้น ลักษณะของฐานข้อมูลจะประกอบด้วย

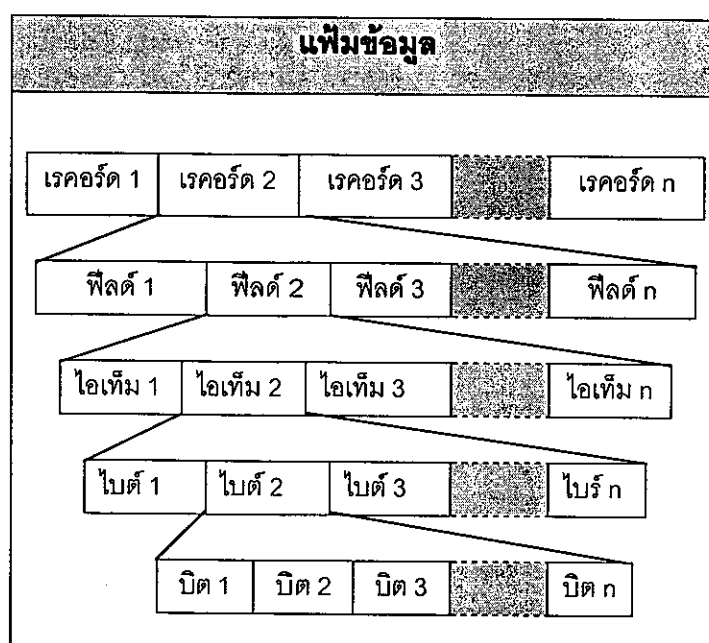
- ข้อมูลทั้งหมดจะต้องถูกจัดเก็บรวบรวมไว้ด้วยกัน
- จะต้องมีการจัดการข้อมูลอย่างเป็นระบบ
- ต้องสามารถนำข้อมูลนั้นไปใช้ตามต้องการ

ฐานข้อมูลเข้ามามีบทบาทเกี่ยวข้องกับเราตลอดเวลา นับตั้งแต่สิ่งที่อยู่ใกล้ตัวเราซึ่งเกี่ยวข้องกับชีวิตประจำวัน การทำงาน การศึกษาตลอดจนถึงข้อมูลในระดับประเทศ เช่น

- สมุดโทรศัพท์ ซึ่งมีข้อมูลชื่อ ที่อยู่ และเบอร์โทรศัพท์ของบุคคลทั่วไป บริษัท ห้างร้าน องค์กรต่างๆ ทั้งภาครัฐและเอกชน
- ข้อมูลของบริษัท ประกอบด้วยข้อมูลพนักงาน ข้อมูลสินค้า ข้อมูลลูกค้า ข้อมูลการสั่งซื้อของลูกค้าแต่ละราย
- ข้อมูลทะเบียนนักศึกษา ประกอบด้วยรหัสประจำตัว ชื่อ - นามสกุล ที่อยู่ รหัสคณะ รหัสสาขาวิชา และอื่นๆที่เกี่ยวข้อง
- ข้อมูลทะเบียนสำมะโนประชากรของประเทศ

ตัวอย่างฐานข้อมูลที่เกี่ยวข้องกับชีวิตประจำวันซึ่งเห็นได้ชัดคือ สมุดโทรศัพท์ซึ่งแสดงรายชื่อผู้ใช้โทรศัพท์โดยเรียงตามลำดับชื่อและนามสกุล จาก ก - ฮ วิธีค้นหาเบอร์โทรศัพท์ก็คือ ต้องเริ่มต้นที่ตัวอักษรแรกของชื่อที่ต้องการแล้วค้นหาแล้วหาไปจนพบชื่อนั้น ถ้ามีชื่อซ้ำกันก็ต้องไปหาที่นามสกุลต่อไปอีกจนกว่าจะได้ชื่อและนามสกุลที่ต้องการ ในทางกลับกันถ้าต้องการค้นหาชื่อจากเบอร์โทรศัพท์ หรือจากนามสกุล หรือจากที่อยู่ อาจต้องหาจากสมุดโทรศัพท์ทั้งเล่มซึ่งต้องใช้เวลาอย่างมากแต่ถ้านำไปรวมการจัดการฐานข้อมูลเป็นเครื่องมือในการจัดเก็บข้อมูลในกระดาษเหล่านี้ลงในคอมพิวเตอร์ คุณจะสามารค้นหาข้อมูลในลักษณะต่างๆ ที่กล่าวมาข้างต้นได้อย่าง

รวดเร็ว นอกจากนี้ยังสามารถบำรุงรักษาฐานข้อมูลได้ง่ายขึ้นเมื่อมีการเปลี่ยนแปลง ไม่ว่าจะเป็นการเพิ่ม ลบ หรือแก้ไขข้อมูล



รูปที่ 2.1 แสดงโครงสร้างข้อมูลที่น่ามาประเมินผลด้วยคอมพิวเตอร์

- บิต (Bit) ย่อมาจาก Binary digit เป็นหน่วยข้อมูลที่เล็กที่สุด ที่แทนด้วยเลขฐานสอง (0 หรือ 1)
- ไบต์ (Byte) คือ กลุ่มของบิตแทนตัวอักษร ตัวเลข หรือสัญลักษณ์พิเศษมาเรียงต่อกันและมีความหมาย เช่น Smith แทนชื่อคน, Bangkok แทนชื่อจังหวัด, มกราคม แทนชื่อเดือน, 100/1 แทนบ้านเลขที่
- ฟิลด์ (Field) คือ ข้อมูลที่ประกอบด้วยไอเท็มตั้งแต่ 1 ไอเท็มขึ้นไป เช่น ไอเท็มวัน เดือน และปี รวมกันเป็นฟิลด์วันเกิด หรือไอเท็มจะเป็นฟิลด์แบบ Group item ถ้าประกอบด้วยหนึ่งไอเท็มจะเป็นฟิลด์แบบ Elementary item ตัวอย่างเช่น ฟิลด์วันเกิดเป็น Group item เนื่องจากสามารถแยกข้อมูลลงไปเป็นวัน เดือน ปีเกิดได้ ในขณะที่วัน เดือน ปีเกิดเป็น Elementary item เนื่องจากแยกย่อยลงไปอีกไม่ได้ ถ้าแยกย่อยลงไปได้ไม่มีข้อมูลที่สื่อความหมาย
- เรคคอร์ด (Record) คือ กลุ่มของฟิลด์ที่มีความสัมพันธ์กัน เช่น เรคคอร์ดพนักงานประกอบด้วยฟิลด์รหัสพนักงาน ชื่อ นามสกุล แผนก ตำแหน่ง

สถานภาพสมรส วันเข้าทำงาน ที่อยู่ และอื่นๆ เป็นต้น หนึ่งเรคอร์ดจะเก็บข้อมูลพนักงาน 1 คน โดยฟิลด์เดียวกันในทุกๆ เรคอร์ดจะต้องเก็บข้อมูลชนิดเดียวกัน เช่น ฟิลด์ชื่อและนามสกุล จะเก็บข้อมูลแบบตัวอักษรเท่านั้น

- ไฟล์ (File) หรือ แฟ้มข้อมูล คือ กลุ่มเรคอร์ดที่ข้อมูลเป็นเรื่องเดียวกัน เช่น แฟ้มข้อมูลพนักงาน แฟ้มข้อมูลลูกค้า แฟ้มข้อมูลการสั่งซื้อสินค้า เป็นต้น

ตัวอย่างแฟ้มข้อมูลชื่อ นามสกุล ที่อยู่และหมายเลขโทรศัพท์

ชื่อ-นามสกุล	ที่อยู่	โทรศัพท์	ชื่อฟิลด์ (Field Name)
กุลสตรี ศิริพงษ์	130/11 ถนนพระราม1 วังใหม่ ปทุมวัน	248-1147	เรคอร์ด (Record)
จันทนา ปวงรัตน์	53 ถนนลาดหญ้า สมเด็จฯ คลองสาน	465-4444	
อัญชลี ไชยรัตน์	36 ซอยสะพานใหม่ พหลโยธิน บางเขน	525-6300	
ศรธรรม ไทยแท้	1399 ถนนพิบูลสงคราม นนทบุรี	526-4900	
ปัทมา เพียรกิจ	183 ถนนศรีนครินทร์ อ.เมือง สมุทรปราการ	753-0000	

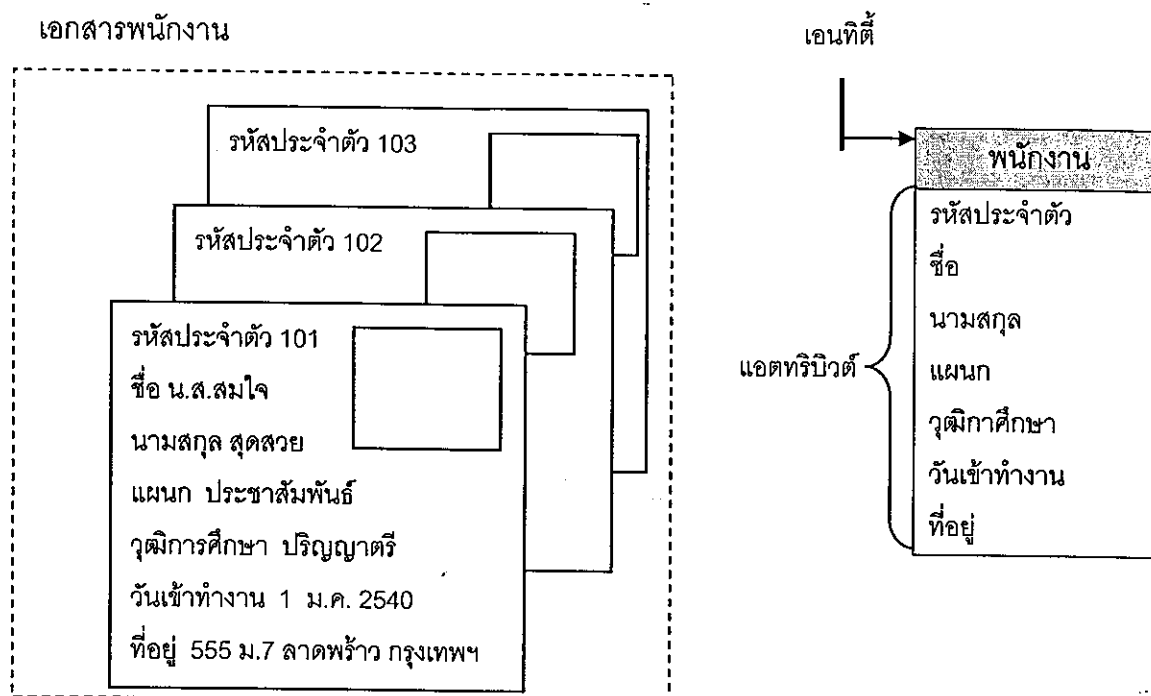
ฟิลด์ (Field)

รูปที่ 2.2 ตัวอย่างแฟ้มข้อมูล

2.1.2 ศัพท์ที่ควรรู้เกี่ยวกับฐานข้อมูล

ศัพท์ในวิชา Database ที่ตำราทุกเล่มจะต้องกล่าวเนื่องจากเกี่ยวข้องกับการออกแบบฐานข้อมูลโดยตรง คือ

2.1.2.1 เอนทิตี หมายถึง สิ่งต่างๆ ที่อ้างอิงในฐานข้อมูล ซึ่งเป็นได้ทั้งรูปธรรม คือ คน สัตว์ สิ่งของ เช่น เอนทิตีพนักงาน เอนทิตีนักศึกษา เอนทิตีสินค้า หรือเป็นนามธรรม เช่น เอนทิตีความชำนาญ เป็นต้น ดังนั้นเอนทิตีจึงประกอบด้วยกลุ่มข้อมูลประเภทเดียวกันที่เป็นบิวด์ต่างๆ คือ รหัสประจำตัว ชื่อ นามสกุล แผนก วุฒิการศึกษา วันเข้าทำงาน ที่อยู่ เป็นต้น



เทเบิล (แอนทิตี) : พนักงาน

เรคอร์ด

รหัสพนักงาน	ชื่อ	นามสกุล	แผนก	วุฒิการศึกษา	วันเข้าทำงาน	ที่อยู่
101	น.ส.สมใจ	สุดสวย	ประชาสัมพันธ์	ปริญญาตรี	1ม.ค. 2540	555 ม.7 ลาดพร้าว กรุงเทพฯ
102	น.ส.สมศรี	ใจดี	บุคคล	ปริญญาตรี	1 ก.พ. 2540	1000 ม.1 แขวงสามเสน เขตห้วยขวาง กรุงเทพฯ
103	นายสำลี	รักไทย	สินเชื่อ	ปวส.	15 พ.ค. 2541	1234 ม.5 แขวงลาดยาว เขตจตุจักร กรุงเทพฯ

ฟิลด์ (แอตทริบิวต์)

รูปที่ 2.3 ตัวอย่างแอนทิตี

2.1.2.2 ความสัมพันธ์ (Relationship)

หัวใจสำคัญในการออกแบบฐานข้อมูลที่มีโครงสร้างเชิงสัมพันธ์หรือ Relational database ก็คือการออกแบบเทเบิลเพื่อเก็บข้อมูลกลุ่มต่างๆ โดยจะต้องสามารถกำหนดความสัมพันธ์ระหว่างกลุ่มข้อมูลเหล่านั้นได้ ดังนั้นคุณจะต้องวิเคราะห์ให้ได้ว่าข้อมูลเหล่านั้นมีความสัมพันธ์กันอย่างไร ซึ่งความสัมพันธ์ระหว่างเทเบิลจะมีด้วยกันทั้งหมด 3 ลักษณะ คือ

ก. ความสัมพันธ์แบบ 1: 1 (One-to-One)

เป็นความสัมพันธ์ที่เรคอร์ดหนึ่งเรคอร์ดในเทเบิลใดๆ สามารถจับคู่กับเรคอร์ดในอีกเทเบิลหนึ่งเพียงเรคอร์ดเดียวเท่านั้น หรือเป็นการจับคู่กันตัวต่อตัว ความสัมพันธ์ลักษณะนี้จะพบเห็นได้น้อย เช่น กรณีที่จำนวนฟิลด์ในเทเบิลนั้นมีมากเกินไป จนทำให้เทเบิลมีขนาดใหญ่ไม่สะดวกในการประมวลผลและบำรุงรักษา จึงต้องแยกข้อมูลบางส่วนที่สำคัญออกมาเก็บไว้ในอีกเทเบิลหนึ่ง โดยเลือกฟิลด์ที่มีคุณสมบัติเหมาะสมเป็นตัวเชื่อมความสัมพันธ์ระหว่างเทเบิลทั้งสอง ในตัวอย่างจะใช้รหัสพนักงานเป็นตัวเชื่อมเนื่องจากเป็นฟิลด์ร่วม (Common field) ของทั้ง 2 เทเบิล

พนักงาน

รหัสพนักงาน	ชื่อ	นามสกุล	ที่อยู่	โทรศัพท์	
1001	Michael	Suyama	Coventry House, London	(71)555-7773	
1002	Laura	Callahan	4726-11 th Ave.N.E., Seattle	(71)555-4848	
1003	Robert	King	Edgeham Hollow, London	(206)555-1189	
1004	Anna	Dodsworth	7 Houndstodd Rd., London	(71)555-4444	

ข้อมูลส่วนตัว

รหัสพนักงาน	เงินเดือน	เลขที่บัตรประกันสังคม
1001	25,000	324-45-6400
1002	20,000	245-37-5600
1003	34,000	459-65-8522
1004	30,000	895-56-1200

รูปที่ 2.4 ตัวอย่างความสัมพันธ์ 1:1

ข. ความสัมพันธ์แบบ 1:N (One-to-Many)

เป็นความสัมพันธ์ที่เรคอร์ดในเทเบิลใดๆ สามารถจับคู่กับเรคอร์ดในอีกเทเบิลหนึ่งได้หลายเรคอร์ด ตัวอย่างเช่น ความสัมพันธ์ระหว่างเทเบิลลูกค้า และเทเบิลการสั่งซื้อ ลูกค้าหนึ่งคนจะมีใบสั่งซื้อสินค้าได้หลายใบ ในขณะที่ใบสั่งซื้อแต่ละใบจะต้องมาจากลูกค้าเพียงคนเดียวเท่านั้น หรือตัวอย่างของอาจารย์ที่ปรึกษาที่มีนักศึกษาในความรับผิดชอบได้หลายคน แต่นักศึกษาแต่ละคนมีอาจารย์ที่ปรึกษาคนเดียวเท่านั้น

ลูกค้า

รหัสลูกค้า	ชื่อบริษัท	ที่อยู่	...
1001	Let's Stop N shop	87 Polk St., San Francisco	...
1002	B's Beverage	Fauntleroy Circus, London	...
1003	Chop-suey Chinese	Hauptstr. 29, Bern	...

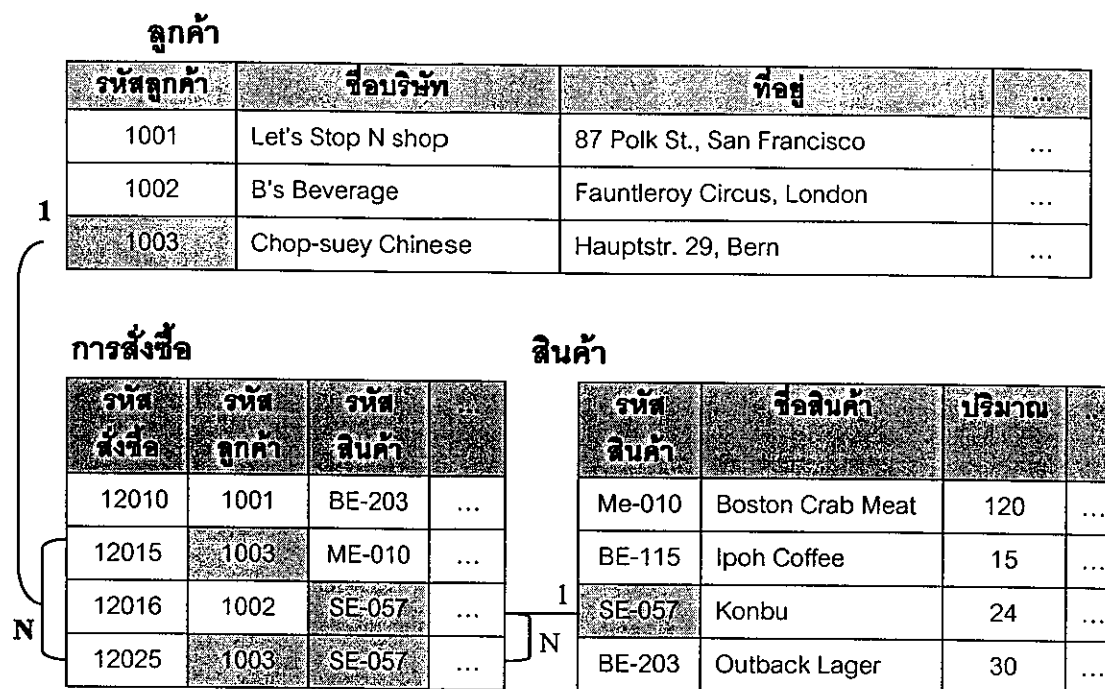
การสั่งซื้อ

รหัสสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า	...
12010	1001	BE-203	...
12015	1003	ME-010	...
12016	1002	SE-057	...
12025	1003	SE-057	...

รูปที่ 2.5 ตัวอย่างความสัมพันธ์ 1 : N

ค. ความสัมพันธ์แบบ M:N (Many-to-Many)

คือลักษณะที่เรคอร์ดหลายๆ เรคอร์ดในเทเบิลหนึ่ง มีความสัมพันธ์กับอีกหลายๆ เรคอร์ดในอีกเทเบิลหนึ่งพร้อมกัน เช่น ความสัมพันธ์ระหว่างเทเบิลลูกค้า และเทเบิลสินค้า ลูกค้าหนึ่งคนสามารถซื้อสินค้าได้หลายชนิด ในขณะที่สินค้าแต่ละชนิดก็จะถูกเลือกซื้อโดยลูกค้าหลายๆ คนได้ ด้วย ถ้าเรานำเทเบิลทั้งสองนี้มาเชื่อมโยงกันโดยตรงจะเห็นว่าทำไม่ได้ เนื่องจากไม่มีฟิลด์ที่เป็นฟิลด์ที่เป็นฟิลด์ร่วมของทั้ง 2 เทเบิล การสร้างความสัมพันธ์แบบ M:N จึงต้องใช้เทเบิลอื่นมาช่วยเป็นสะพานในการเชื่อมโยง ซึ่งในที่นี้จะใช้เทเบิลการสั่งซื้อมาช่วย



รูปที่ 2.6 ตัวอย่างความสัมพันธ์ M : N

จากภาพจะเกิดความสัมพันธ์แบบ 1:N สองความสัมพันธ์ ความสัมพันธ์แรกคือ ความสัมพันธ์ระหว่างเทเบิลลูกค้าและเทเบิลการสั่งซื้อ ความสัมพันธ์ที่สองคือความสัมพันธ์ระหว่างเทเบิลสินค้า และเทเบิลการสั่งซื้อ ฟิลด์รหัสลูกค้าและฟิลด์รหัสสินค้าจะถูกนำมาใช้เป็น ฟิลด์ร่วมเพื่อสร้างความสัมพันธ์ระหว่างเทเบิลลูกค้า และเทเบิลสินค้า ให้เป็นแบบ M:N จาก ตัวอย่างลูกค้ารหัส 1003 ซื้อสินค้ารหัส ME-010 และ SE-057 ในขณะที่สินค้า SE-057 ก็ถูกซื้อ โดยลูกค้ารหัส 1002 และ 1003

2.1.2.3 คีย์ (Key)

เพื่อเพิ่มประสิทธิภาพในการจัดการฐานข้อมูล เช่น การอ้างอิง การค้นหา การแก้ไข เปลี่ยนแปลงข้อมูลในเรคคอร์ดหรือการกำหนดความสัมพันธ์ระหว่างเทเบิล เราต้องกำหนด "คีย์ (Key)" ให้กับเทเบิลก่อน ถึงแม้ใน Access จะไม่ได้บังคับไว้ว่าจะต้องให้ฟิลด์ใดฟิลด์หนึ่ง เป็นคีย์ แต่คุณควรกำหนดไว้ มิฉะนั้นคุณจะไม่สามารถนำเทเบิลมาเชื่อมโยงความสัมพันธ์กันได้เลย นอกจากนี้การกำหนดคีย์จะทำให้การอ้างอิงและประมวลผลข้อมูลด้วยคอมพิวเตอร์สะดวก รวดเร็วขึ้นรวมทั้งประหยัดเนื้อที่ในการเก็บข้อมูล เช่น กำหนดรหัสประจำตัวให้กับพนักงานทุกคน ในเทเบิลพนักงาน ให้พนักงานคนใดทำงานล่วงเวลาก็จะนำเฉพาะรหัสพนักงานพร้อมทำข้อมูลการทำงานไปใส่ในเทเบิลการทำงาน

ก. Primary Key

หรือ คีย์หลัก เป็นคีย์ที่กำหนดจากฟิลด์ที่ไม่มีข้อมูลซ้ำกัน (unique) โดยเด็ดขาดในเทเบิลนั้น เช่น ฟิลด์รหัสพนักงาน หรือฟิลด์เลขประจำตัวประชาชนในเทเบิลพนักงาน หรือฟิลด์รหัสสินค้าในเทเบิลสินค้า เนื่องจากเป็นคีย์ที่มีข้อมูลซ้ำกันไม่ได้จึงสามารถนำไปใช้ในการจัดเรียงและแยกแยะข้อมูลแต่ละเรคคอร์ดออกจากกันได้ดี ทั้งนี้ฟิลด์ที่กำหนดให้เป็นคีย์จะต้องมีค่าเสมอ จะเป็นฟิลด์ว่าง (Null) ไม่ได้ ข้อดีของการกำหนด Primary Key อีกข้อหนึ่งคือ จะช่วยไม่ให้เกิดการผิดพลาดเนื่องจากป้อนข้อมูลที่ซ้ำกันลงในฟิลด์ที่ไม่ได้รับอนุญาตให้มีข้อมูลซ้ำ ถ้าใส่ข้อมูลซ้ำตัว DBMS เช่น Access จะแสดงข้อความเตือนและไม่ทำงานต่อจนกว่าจะแก้ไขข้อมูลให้ถูกต้อง

พนักงาน

รหัสพนักงาน	ชื่อ	นามสกุล	ที่อยู่	โทรศัพท์	
1001	Michael	Suyama	Coventry House, London	(71)555-7773	
1002	Laura	Callahan	4726-11 th Ave.N.E., Seattle	(71)555-4848	
1003	Robert	King	Edgeham Hollow, London	(206)555-1189	
1004	Anna	Dodsworth	7 Houndstod Rd., London	(71)555-4444	

↑ ใช้ฟิลด์เป็น Primary key

รูปที่ 2.7 ตัวอย่าง Primary Key

ข. Secondary Key

หรือ คีย์ลำดับรอง Access จะเรียกคีย์ชนิดนี้ว่า Index หรือ ดัชนี ซึ่งนอกจากนี้กำหนด Primary Key แล้วเรายังสามารถใช้ Index ช่วยในการค้นหาหรือจัดเรียงกลุ่มเรคคอร์ดที่มีจำนวนมากได้อย่างรวดเร็ว เช่น การค้นหาชื่อและนามสกุลของพนักงานในเทเบิลพนักงาน ถ้าไม่ได้กำหนดฟิลด์ชื่อและนามสกุลเป็น Index ไว้ก่อน DBMS จะค้นหาตั้งแต่เรคคอร์ดแรกไล่ไปจนถึงเรคคอร์ดที่ต้องการ ถ้ามีข้อมูลเป็นพันๆ หมื่น เรคคอร์ด และเรคคอร์ดที่ต้องการอยู่ในตำแหน่งท้ายๆ จะเสียเวลาในการค้นหามาก แต่ถ้ากำหนดให้ฟิลด์ชื่อและนามสกุลเป็น Index จะทำให้ชื่อและนามสกุลของพนักงานพร้อมตำแหน่งของเรคคอร์ดถูกเก็บไว้ในแฟ้ม Index ของ DBMS โดยเรียงลำดับชื่อและนามสกุลตามตัวอักษร เมื่อมีคำสั่งค้นหาข้อมูล DBMS จะไปค้นชื่อและนามสกุลในแฟ้ม Index ก่อน เมื่อพบก็จะทราบตำแหน่งของเรคคอร์ดนั้น ทำให้สามารถเข้าถึงเรคคอร์ดที่ต้องการในเทเบิลได้ทันที ฟิลด์ที่เป็น Index นี้อาจยอมให้มีข้อมูลซ้ำกันไม่ได้ (ต่างกับ

Primary Key ซึ่งจะมีข้อมูลซ้ำกันไม่ได้ Primary Key ทุกตัวจะมีคุณสมบัติเป็น Index อยู่แล้ว (โดยอัตโนมัติ) ข้อควรระวังในการสร้าง Index คือ เมื่อมีการเพิ่มหรือลบเรคอร์ดในเทเบิลจะทำให้ข้อมูลในแฟ้ม Index ต้องถูกแก้ไขปรับปรุงตามไปด้วยทุกครั้ง ทำให้ DBMS ต้องทำงานมากขึ้น ซึ่งส่งผลให้ประสิทธิภาพและความเร็วในการทำงานโดยรวมของระบบลดลงได้ อีกประเด็นหนึ่งคือ การสร้าง Index จากฟิลด์ที่มีข้อมูลซ้ำกันมากๆ อาจไม่ให้ผลดีมากนัก

Candidate Key

ถ้าในเทเบิลหนึ่งมีฟิลด์หลายฟิลด์ที่มีคุณสมบัติครบถ้วนจนนำมาใช้ Primary Key แทนกันได้ จะเรียกฟิลด์เหล่านั้นแต่ละฟิลด์ว่าเป็น Candidate Key หรือ คีย์คู่แข่ง เช่น ในเทเบิลพนักงาน ถ้าสามารถประกันได้ว่าชื่อพนักงานจะไม่ซ้ำกันเลย กรณีนี้เราสามารถนำฟิลด์รหัสพนักงานหรือฟิลด์ชื่อมาเป็น Primary Key ก็ได้ ทำให้ทั้งสองฟิลด์นี้กลายเป็น Candidate Key ซึ่งถ้าผู้ออกแบบฐานข้อมูลตัดสินใจให้รหัสพนักงานเป็น Primary Key ฟิลด์ชื่อก็จะกลายเป็น Alternate Key หรือ คีย์สำรอง ไปโดยอัตโนมัติ

พนักงาน

รหัสพนักงาน	ชื่อ	นามสกุล	ที่อยู่	โทรศัพท์	
1001	Michael	Suyama	Coventry House, London	(71)555-7773	
1002	Laura	Callahan	4726-11 th Ave.N.E., Seattle	(71)555-4848	
1003	Robert	King	Edgeham Hollow, London	(206)555-1189	
1004	Anna	Dodsworth	7 Houndstodd Rd., London	(71)555-4444	



ถ้ารับประกันได้ว่าชื่อพนักงานจะไม่ซ้ำกันเลย เทเบิลนี้จะมี Candidate Key 2ฟิลด์ คือ รหัสพนักงานและชื่อ

รูปที่ 2.8 ตัวอย่าง Primary Key

ค. Compound Key

หรือ คีย์รวม (บางครั้งเรียก Composite Key) เป็นคีย์ที่เกิดจากการนำฟิลด์หลายๆฟิลด์มารวมกันเพื่อให้มีคุณสมบัติเป็น Primary Key คือไม่มีข้อมูลซ้ำกันและไม่มีค่าว่าง (Null) เนื่องจากในบางครั้งการสร้าง Primary Key จากฟิลด์เดียวกันอาจมีโอกาที่จะเกิดข้อมูลซ้ำกันได้ ตัวอย่างเช่น ในเทเบิลพนักงาน หากไม่มีการกำหนดรหัสใดๆ ให้กับพนักงานแต่ละคน เราอาจใช้ชื่อและนามสกุลของพนักงานประกอบกันเข้าเป็น Primary Key ของเทเบิลก็ได้ ด้วยสมมติฐานที่ว่าไม่มีใคร (ตั้งแต่ 2 คนขึ้นไป) ใช้ชื่อและนามสกุลเดียวกัน คีย์ที่สร้างจากฟิลด์มากกว่า 1 ฟิลด์แบบนี้เรียกว่า Compound Key

พนักงาน

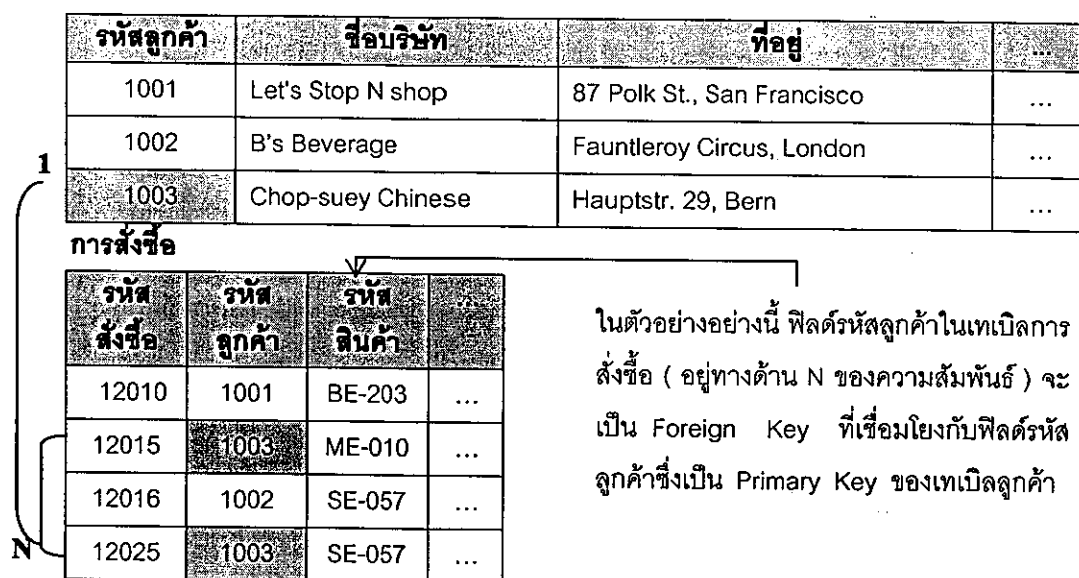
ชื่อ	นามสกุล	ที่อยู่	โทรศัพท์	...
Michael	Suyama	Coventry House, London	(71)555-7773	...
Laura	Callahan	4726-11 th Ave.N.E., Seattle	(71)555-4848	...
Robert	King	Edgeham Hollow, London	(206)555-1189	...
Anna	Dodsworth	7 Houndstod Rd., London	(71)555-4444	...

(หากไม่มีฟิลด์รหัสพนักงาน) เราอาจใช้ฟิลด์ชื่อและนามสกุล รวมกันเป็น Primary Key ได้ด้วยสมมติฐานที่ว่าไม่มีใครจะใช้ชื่อและนามสกุลเดียวกัน คีย์แบบนี้เรียกว่า Compound Key

รูปที่ 2.9 ตัวอย่างCompound Key

ง. Foreign Key

หรือ คีย์นอก เป็นคีย์ที่เชื่อมเทเบิลที่เกี่ยวข้องเข้าด้วยกัน เช่น ในเทเบิลลูกค้าจะมีฟิลด์รหัสลูกค้าเป็น Primary Key เราจะให้รหัสลูกค้าในเทเบิลลูกค้าเชื่อมโยงรหัสลูกค้าในเทเบิลการสั่งซื้อ เพื่อที่จะได้ทราบชื่อและที่อยู่ของลูกค้าที่สั่งซื้อสินค้านั้น ในกรณีนี้ฟิลด์รหัสลูกค้าในเทเบิลการสั่งซื้อจะมีคุณสมบัติเป็น Foreign Key (ในขณะที่ฟิลด์รหัสลูกค้าเมื่ออยู่ในเทเบิลลูกค้า จะคุณสมบัติเป็น Primary Key) จากตัวอย่างนี้ความสัมพันธ์ระหว่างเทเบิลทั้งสองเป็นแบบ 1:N เนื่องจากลูกค้า 1 คน สามารถสั่งซื้อสินค้าได้มากกว่า 1 รายการ รหัสลูกค้าในเทเบิลการสั่งซื้อจึงมีค่าซ้ำกันได้ ซึ่งต่างจาก Primary Key ที่มีค่าซ้ำกันไม่มาก



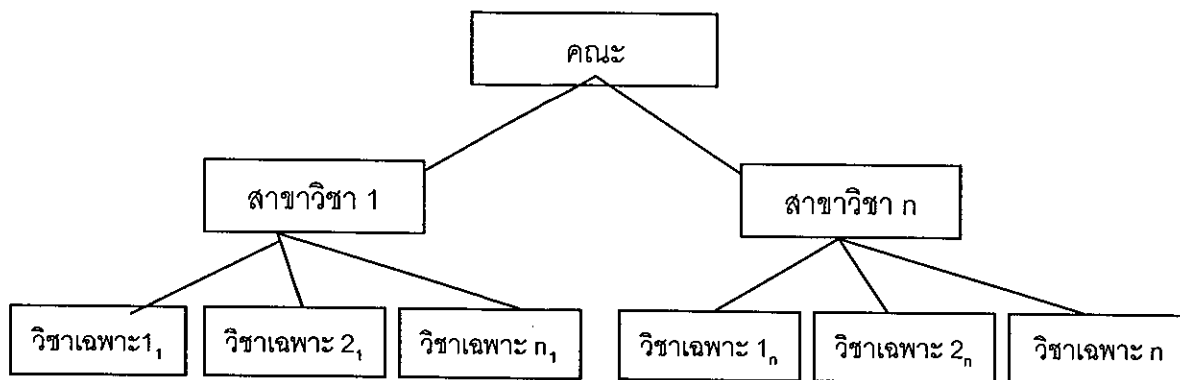
รูปที่ 2.10 ตัวอย่าง Foreign Key

2.1.3 ประเภทของฐานข้อมูล

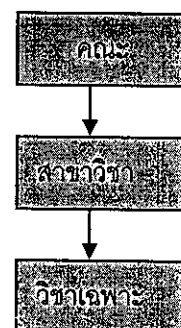
ฐานข้อมูลแบ่งได้เป็น 2 ประเภท ตามลักษณะโครงสร้างที่มองเห็นภายนอก

2.1.3.1 โครงสร้างแบบลำดับชั้น (Hierarchical structure) หรือเรียกอีกอย่างว่า โครงสร้างแบบต้นไม้ (Tree structure) เนื่องจากมีลักษณะคล้ายต้นไม้ที่คว่ำหัวลง เรคอร์ดที่อยู่แถวบนจะเป็นเรคอร์ดพ่อแม่ (Parent record) ของเรคอร์ดถัดมาซึ่งเป็นเรคอร์ดลูก (Child record) เรคอร์ดพ่อแม่แต่ละเรคอร์ดสามารถมีเรคอร์ดลูกได้หลายๆ เรคอร์ด ในขณะที่เรคอร์ดลูกแต่ละเรคอร์ดจะมีเรคอร์ดพ่อแม่ได้เพียงหนึ่งเรคอร์ดเท่านั้น ความสัมพันธ์ของข้อมูลในโครงสร้างแบบนี้อาจเป็นแบบ 1:1 หรือ 1:N

ตัวอย่าง ในภาพแสดงโครงสร้างแบบลำดับชั้นของหลักสูตรการศึกษา แต่ละคณะในมหาวิทยาลัย ซึ่งประกอบด้วยสาขาวิชาต่างๆ และวิชาเฉพาะของแต่ละสาขา แต่ละคณะจะมีสาขาวิชาต่างๆ ที่ไม่ซ้ำกันกับคณะอื่นให้นักศึกษาเลือกเรียน และในแต่ละสาขาวิชาที่จะมีวิชาเฉพาะสำหรับสาขานั้น เรคอร์ดคณะจะเป็นเรคอร์ดพ่อแม่ของเรคอร์ดสาขาวิชา ในขณะที่เรคอร์ดเฉพาะก็จะเป็นเรคอร์ดลูกของเรคอร์ดสาขาวิชา



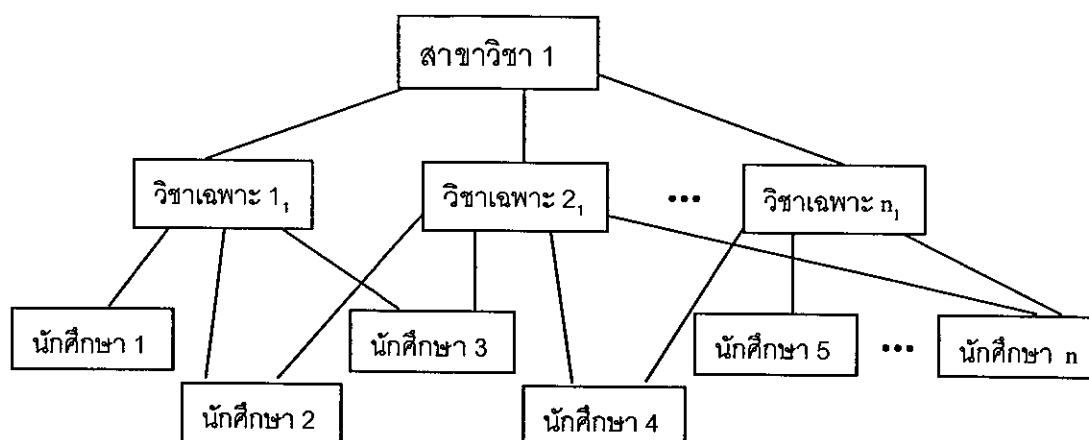
ความสัมพันธ์ในตัวอย่างนี้เป็นแบบ 1:N เราจะใช้ลูกศรแสดงความสัมพันธ์ระหว่างข้อมูล โดยให้หัวลูกศรออกจากส่วนที่เป็นเรคอร์ดพ่อแม่ (ส่วนที่เป็น “1”) ไปยังเรคอร์ดลูก (ส่วนที่เป็น “N”) และกรอบสี่เหลี่ยมแทนกลุ่มของเรคอร์ด จะเห็นว่าโครงสร้างแบบลำดับชั้น จะมีหัวลูกศรไปยังข้อมูลแต่ละกลุ่มได้ไม่เกินหนึ่งหัว เนื่องจากความสัมพันธ์จะเป็นลักษณะจากบนลงล่าง



รูปที่ 2.11 ตัวอย่างประเภทของฐานข้อมูล

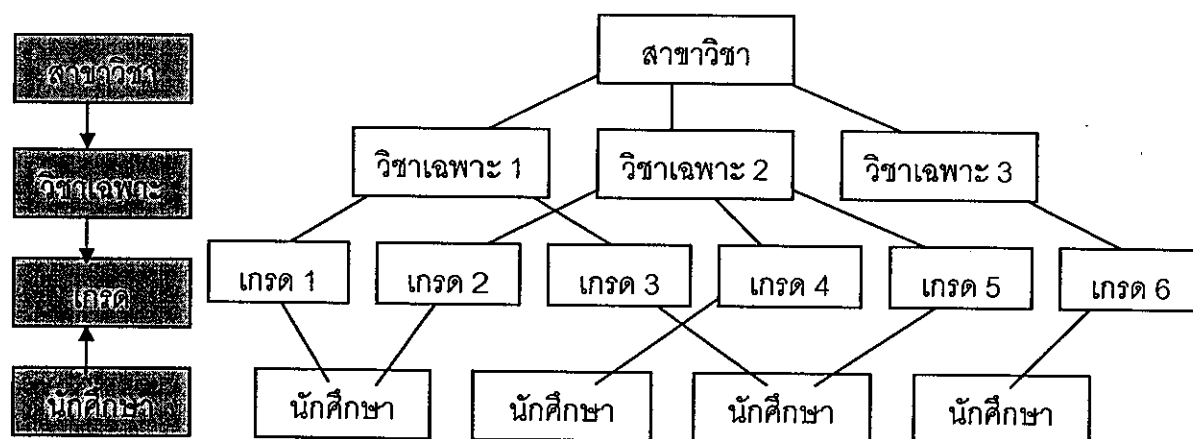
2.1.3.2 โครงสร้างแบบเครือข่าย (Network structure) มีลักษณะคล้ายกับโครงสร้างแบบลำดับชั้น ต่างกันตรงความสัมพันธ์ระหว่างข้อมูลเป็นแบบ M:N ได้ ในขณะที่โครงสร้างแบบลำดับชั้นจะไม่มีความสัมพันธ์ในลักษณะนี้

ตัวอย่างในภาพแสดงโครงสร้างการลงทะเบียนเรียนวิชาต่างๆในแต่ละสาขา โดยแต่ละวิชาจะมีนักศึกษาลงทะเบียนเรียนได้หลายคน ขณะเดียวกันนักศึกษาคนหนึ่งสามารถลงทะเบียนเรียนได้มากกว่าหนึ่งวิชา



รูปที่ 2.12 ตัวอย่างโครงสร้างแบบเครือข่าย

ความสัมพันธ์ระหว่างสาขาวิชา และวิชาเฉพาะจะเป็นแบบ 1 : N ในขณะที่วิชาเฉพาะจะมีความสัมพันธ์กับนักศึกษาแบบ M:N การใช้หัวลูกศรแสดงความสัมพันธ์ระหว่างข้อมูล M:N นี้จะต้องใช้กลุ่มข้อมูลซึ่งมีความสัมพันธ์กับข้อมูลนักศึกษาและข้อมูลวิชาเฉพาะเป็นตัวเชื่อมโยง ในที่นี่จะใช้ใบแสดงผลการศึกษา (Transcript) ซึ่งประกอบด้วยข้อมูลทั้งรหัสนักศึกษา รหัสวิชา ชื่อวิชาที่ลงทะเบียนเรียน และเกรดของแต่ละวิชา



รูปที่ 2.13 ตัวอย่างความสัมพันธ์ระหว่างข้อมูล

จากภาพจะมีหัวลูกศรสองหัวชี้จากข้อมูลนักศึกษาและวิชาเฉพาะมายังข้อมูลเกรด ซึ่งจะแตกต่างจากโครงสร้างแบบลำดับชั้น ที่มีหัวลูกศรชี้ไปยังกลุ่มข้อมูลใดๆ ไม่ได้เกินหนึ่งหัว

โครงสร้างแบบเชิงสัมพันธ์ (Relational structure) เป็นโครงสร้างที่ได้รับการพัฒนาขึ้นภายหลังทั้งสองแบบที่กล่าวมาข้างต้น โครงสร้างแบบนี้จะทำให้ผู้ใช้งานเห็นข้อมูลถูกเก็บใน

ลักษณะของเทเบิลแบบสองมิติซึ่งประกอบด้วยแถวและคอลัมน์ โดยไม่ต้องสนใจว่าข้อมูลจะถูกเก็บจริงในลักษณะใด และสามารถกำหนดความสัมพันธ์ระหว่างเทเบิลได้ ความสัมพันธ์ระหว่างข้อมูลจะเป็นทั้งแบบ 1:1 และ M:N

ลูกค้า

รหัสลูกค้า	ชื่อบริษัท	ที่อยู่	...
1001	Let's Stop N shop	87 Polk St., San Francisco	...
1002	B's Beverage	Fauntleroy Circus, London	...
1003	Chop-suey Chinese	Hauptstr. 29, Bern	...

การสั่งซื้อ

รหัส สั่งซื้อ	รหัส ลูกค้า	รหัส สินค้า	...
12010	1001	BE-203	...
12015	1003	ME-010	...
12016	1002	SE-057	...

สินค้า

รหัส สินค้า	ชื่อสินค้า	ปริมาณ	...
Me-010	Boston Crab Meat	120	...
BE-115	Ipoh Coffee	15	...
SE-057	Konbu	24	...

รูปที่ 2.14 ตัวอย่างโครงสร้างแบบเชิงสัมพันธ์

จากเทเบิลลูกค้าและเทเบิลสั่งซื้อ จะมีฟิลด์รหัสลูกค้าเป็นตัวเชื่อมความสัมพันธ์ระหว่าง 2 เทเบิลนี้ ถ้าต้องการทราบชื่อและที่อยู่ของลูกค้ารหัส 1002 นี้ไปค้นหาข้อมูลในเทเบิลลูกค้าที่มีรหัสตรงกัน ในทางกลับกันถ้าต้องการทราบว่ารหัสสินค้า SE-057 ในเทเบิลการสั่งซื้อคือสินค้าอะไร มีจำนวนคงเหลือในคลังเท่าไร รวมทั้งราคาต่อหน่วย เราจะใช้รหัส SE-057 ไปค้นหาข้อมูลที่มีรหัสสินค้าตรงกันในเทเบิลสินค้าได้ (นันทนิ แวงโสภา, 2544. หน้า 63-73)

2.2 ระบบฐานข้อมูล

ฐานข้อมูล (Database) คือ การนำเอาข้อมูลต่างๆที่มีความสัมพันธ์กันมาเก็บรวบรวมกันไว้ภายในฐานข้อมูลเดียว และสามารถใช้อย่างพร้อมกันได้

ระบบฐานข้อมูล (Database system) คือ การจัดข้อมูลอย่างเป็นระบบซึ่งผู้ใช้สามารถเรียกใช้ข้อมูลดังกล่าวได้ในลักษณะต่างๆเช่น การเพิ่มข้อมูล (Add Data) การเรียกใช้ข้อมูล

(Retrieve Data) การแก้ไขและลบข้อมูล (Update & Delete) ตลอดจนการเคลื่อนย้ายข้อมูล (move Data)

ไปตามกำหนด

หรือระบบฐานข้อมูล อาจหมายถึง การจัดทำฐานข้อมูลขึ้นเพื่อสนับสนุนการดำเนินงานหรือกิจกรรมอย่างใดอย่างหนึ่ง

2.3 องค์ประกอบของระบบฐานข้อมูล

ระบบฐานข้อมูลโดยทั่วไป จะเกี่ยวข้องกับ 4 ส่วนหลักๆ ดังนี้

2.3.1 ข้อมูล (Data)

ข้อมูลที่เกิดขึ้นอยู่ในระบบฐานข้อมูล ไม่ว่าจะเป็นบนเครื่องคอมพิวเตอร์ส่วนบุคคล ไปจนถึงเครื่องคอมพิวเตอร์ขนาดใหญ่อย่างเช่นเครื่อง Mainframe ข้อมูลในแต่ละส่วนสามารถนำมาใช้ประกอบกันได้ (Data Integrated) ..

2.3.2 ฮาร์ดแวร์ (Hardware)

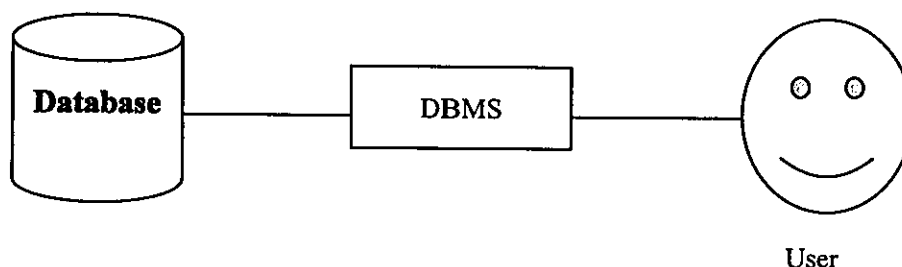
อุปกรณ์ทางคอมพิวเตอร์ที่มีส่วนเกี่ยวข้องกับระบบฐานข้อมูล จะประกอบด้วย 2 ส่วนหลักๆ ดังนี้

2.3.2.1 หน่วยความจำสำรอง

2.3.2.2 หน่วยประมวลผล และหน่วยความจำหลัก

2.3.3 ซอฟต์แวร์ (Software)

ในการติดต่อกับข้อมูลในฐานข้อมูลของผู้ใช้ จะต้องกระทำผ่านโปรแกรมที่มีชื่อว่าโปรแกรม Database Management System (DBMS) ดังรูป



รูปที่ 2.15 ข้อมูลในฐานข้อมูลของผู้ใช้

2.3.4 ผู้ใช้ระบบฐานข้อมูล (User)

ผู้ที่เรียกใช้ข้อมูลจากระบบฐานข้อมูลมาใช้งาน สามารถแบ่งออกเป็น 2 กลุ่มได้ดังนี้

2.3.4.1 Application Programmer ได้แก่ ผู้ทำหน้าที่พัฒนาโปรแกรม Application Program

2.3.4.2 End User ได้แก่ผู้ที่นำข้อมูลจากฐานข้อมูลไปใช้งาน ซึ่งแบ่งเป็น 2 กลุ่ม ดังนี้

- Naive User
- Sophisticated User

2.3.5 Database Administrator (DBA) ได้แก่ ผู้บริหารที่ทำหน้าที่ควบคุมและตัดสินใจในการกำหนดโครงสร้างของฐานข้อมูล ชนิดของข้อมูล วิธีการจัดเก็บข้อมูล รูปแบบในการเรียกใช้ข้อมูล (กิตติ-จำลอง ,2544. หน้า 3-12)

2.4 Data Independence

ในการเรียกใช้ข้อมูลที่จัดเก็บอยู่ในระบบแฟ้มข้อมูล จะต้องอาศัยโปรแกรมที่เขียนขึ้น เพื่อเรียกใช้ข้อมูลในแฟ้มข้อมูลนั้นโดยเฉพาะ ผู้ใช้จะต้องให้ โปรแกรมเมอร์จัดทำโปรแกรม เพื่ออ่านข้อมูลจากแฟ้มข้อมูล และพิมพ์รายงานที่แสดงเฉพาะข้อมูลที่ตรงตามเงื่อนไขที่กำหนด ดังนั้นเมื่อมีการเปลี่ยนแปลงโครงสร้างทางกายภาพของแฟ้มข้อมูลใดข้อมูลหนึ่ง จึงส่งผลให้โปรแกรมต่างๆ ที่เรียกใช้ข้อมูลในแฟ้มนั้น ต้องมีการเปลี่ยนแปลงตามไปด้วย

ภายในระบบฐานข้อมูล ไม่สามารถที่จะยอมรับให้ความอิสระ ระหว่างข้อมูล และโปรแกรม เกิดขึ้นได้เนื่องจาก 2 สาเหตุหลักๆดังนี้

1. เนื่องจากในฐานข้อมูล จะต้องไม่ปรากฏข้อมูลที่ซ้ำซ้อนกันเกิดขึ้น แต่ในแง่ความเป็นจริงแล้ว ข้อมูลที่ผู้ใช้ต้องการถึงแม้จะเป็นข้อมูลเดียวกัน ก็ต้องการรูปแบบข้อมูลที่ต่างกัน
2. เนื่องจาก DBA มีสิทธิ์ที่จะเปลี่ยนแปลงโครงสร้างของข้อมูลภายในฐานข้อมูลได้ ดังนั้นจึงจำเป็นที่จะต้องทำให้ข้อมูลภายในฐานข้อมูล เป็นอิสระจากโปรแกรมต่างๆที่เรียกใช้ เพื่อที่จะทำให้ DBA สามารถเปลี่ยนแปลงโครงสร้างของข้อมูลภายในฐานข้อมูลได้ โดยไม่ส่งผลกระทบกับโปรแกรม

ในการกำหนดให้ข้อมูลเป็นอิสระจากโปรแกรมที่เรียกใช้จะแบ่งออกเป็น 2 ระดับ ดังนี้

1. ระดับ Physical

เป็นระดับที่โครงสร้างทางกายภาพของข้อมูลเป็นอิสระจากโปรแกรมที่ใช้ เช่น สามารถเปลี่ยนแปลง โครงสร้างของ Index File ได้โดยไม่ต้องแก้ไขโปรแกรมที่เรียกใช้ข้อมูลนั้น

2. ระดับ Logical

เป็นระดับความสัมพันธ์ระหว่างข้อมูลในส่วนต่างๆ ภายในฐานข้อมูลเป็นอิสระจากโปรแกรมที่เรียกใช้ เช่น สามารถแยกบาง Field ออกไปเป็นไปเพิ่มข้อมูลใหม่ โดยไม่ต้องเรียก แก้ไข โปรแกรมที่เรียกใช้ข้อมูลนั้น

เมื่อพิจารณาเป็นความอิสระของข้อมูลที่มีต่อ โปรแกรมที่เรียกใช้ในทุก 2 ระดับจะสังเกตเห็นได้ว่า การกำหนดให้เป็นอิสระจากโปรแกรมที่เรียกใช้ในระดับ Logical จะกระทำได้ยากกว่าระดับ Physical เนื่องจากความสัมพันธ์ระหว่างข้อมูลในส่วนต่างๆ ภายในเพิ่มข้อมูลเดียวกัน จะมีความเกี่ยวข้องกับโปรแกรมมากกว่าโครงสร้างทางกายภาพของข้อมูล

2.5 Database Management System (DBMS)

เป็นโปรแกรมที่ทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างผู้ใช้กับฐานข้อมูล เพื่อการจัดการและควบคุมความถูกต้อง ความซ้ำซ้อน และความสัมพันธ์ระหว่างข้อมูลต่างๆ ภายในฐานข้อมูล ซึ่งต่างจากระบบเพิ่มข้อมูลที่ทำหน้าที่เหล่านี้จะเป็นหน้าที่ของโปรแกรมเมอร์ ในการติดต่อหับข้อมูลในฐานข้อมูลไม่ว่าจะด้วยการใช้คำสั่งในกลุ่มคำสั่ง DML หรือ DDL หรือจะด้วยโปรแกรมต่างๆ ทุกคำสั่งที่ใช้กระทำกับข้อมูล จะถูกโปรแกรม DBMS มาแปล (Compile) เป็นการกระทำ (Operation) ต่างๆ ภายใต้คำสั่งนั้นๆ เพื่อจะนำไปกระทำกับตัวข้อมูลภายในฐานข้อมูลต่อไป สำหรับส่วนการทำงานต่างๆ ภายในระบบ DBMS ที่ทำหน้าที่ในการแปลคำสั่งไปเป็นการกระทำต่างๆ ที่จะกระทำกับข้อมูลนั้นๆ ประกอบด้วยส่วนการทำงานต่างๆ ดังนี้

2.5.1 Database Manager

เป็นส่วนที่ทำหน้าที่กำหนดการกระทำต่างๆ ให้กับส่วน File Manager เพื่อไปกระทำกับข้อมูลที่เก็บอยู่ในฐานข้อมูล (File Manager เป็นส่วนที่ทำหน้าที่บริหารและจัดการกับข้อมูลที่เก็บอยู่ในฐานข้อมูลในระดับกายภาพ) Query Processor

เป็นส่วนที่ทำหน้าที่แปลงประโยคคำสั่งของ Query Language ให้อยู่ในรูปแบบของคำสั่ง Database Manager

2.5.2 Data Manipulation Language Precompiled

เป็นส่วนที่ทำหน้าที่แปล (Compile) ประโยคคำสั่งของกลุ่มคำสั่ง DML ให้อยู่ในรูปแบบที่ส่วน Application Programs Object Code จะนำไปเข้ารหัสเพื่อส่งต่อไปยังส่วน Database Manager ในการแปลประโยคคำสั่งของกลุ่มคำสั่ง DML ของส่วน Data Manipulation Language Precompiled นี้จะต้องทำงานร่วมกับส่วนของ Query Manager

2.5.3 Data Definition Language Precompiled

เป็นส่วนที่ทำหน้าที่แปล (Compiler) ประโยคคำสั่งของคำสั่ง DDL ให้อยู่ในรูปแบบของ Metadata ที่เก็บอยู่ในส่วน Data Dictionary ของฐานข้อมูล (Metadata ได้แก่ รายละเอียดที่บอกถึงโครงสร้างต่างๆของข้อมูล

2.5.4 Application Programs Object Code

เป็นส่วนที่ทำหน้าที่แปลงคำสั่งต่างๆของโปรแกรมรวมทั้งคำสั่งในกลุ่มของคำสั่ง DML ที่ส่งต่อมาจากส่วน Data Manipulation Language Precompiled ให้อยู่ในรูปของ Object Code ที่จะส่งต่อไปให้ Database Manager เพื่อกระทำกับข้อมูลในฐานข้อมูลทั้ง 5 ส่วนของโปรแกรม DBMS

โปรแกรม DBMS นี้ได้ถูกพัฒนาเพื่อแก้ปัญหาทางด้าน Database Independence ที่ไม่มีในระบบแฟ้มข้อมูล ดังนั้นจึงมีความเป็นอิสระจากทั้งตัว Hardware และตัวข้อมูลภายในฐานข้อมูลกล่าวคือ โปรแกรม DBMS จะมีการทำงานที่ไม่ขึ้นกับรูปแบบ (Platform) ของตัว Hardware ที่มาใช้กับระบบฐานข้อมูล รวมทั้งมีรูปแบบในการอ้างถึงข้อมูลที่ไม่ขึ้นอยู่กับโครงสร้างทางกายภาพของข้อมูลด้วยการใช้ Query Language ในการติดต่อกับข้อมูลในฐานข้อมูลแทนคำสั่งของภาษาคอมพิวเตอร์ในรุ่นที่ 3 ส่งให้ผู้ที่ใช้เรียกใช้ข้อมูลจากฐานข้อมูล โดยไม่จำเป็นต้องทราบประเภทของฐานข้อมูลหรือขนาดของข้อมูลนั้น หรือสามารถกำหนดลำดับที่ของ Field ในการแสดงผลโดยไม่คำนึงถึงลำดับที่แท้จริงของ Field นั้น (กิตติ-จำลอง ,2544. หน้า 12-14)

2.6 การออกแบบฐานข้อมูล

การจะนำฐานข้อมูลไปใช้งานอย่างมีประสิทธิภาพนั้น หัวใจสำคัญอยู่ที่การออกแบบฐานข้อมูลโดยผู้ออกแบบจะต้องสามารถวิเคราะห์หาเอนทิตีดีและแอตทริบิวต์ ทั้งหมดที่จะต้องนำมาใช้ในฐานข้อมูลได้อย่างครบถ้วน รวมทั้งกำหนดความสัมพันธ์ระหว่างเอนทิตีได้อย่างถูกต้องและเหมาะสม การออกแบบฐานข้อมูลใน Access จะมีลำดับขั้นตอนดังนี้

ขั้นที่ 1 เก็บรวบรวมข้อมูลรายละเอียดทั้งหมด

เป็นขั้นตอนของการเก็บรวบรวมข้อมูลและรายละเอียดต่างๆ ของงาน รวมทั้งความต้องการของผู้ใช้ เช่น

- ข้อมูลใดบ้างที่เป็นเรื่องเดียวกัน ให้จัดกลุ่มข้อมูลนั้นเป็นเอนทิตีดี
- ชนิดของข้อมูลเป็นแบบใด ตัวอักษรหรือตัวเลข มีเงื่อนไขหรือข้อกำหนดอย่างไร เช่น รหัสพนักงานจะต้องเป็นเลข 4 หลัก อายุพนักงานต้องไม่เกิน 55 ปี วุฒิการศึกษาของพนักงานต้องไม่ต่ำกว่าระดับ ปวส.

- มีข้อมูลอะไรบ้างที่จะต้องนำมาค้นหาหรือประมวลผล ผลที่ได้ต้องส่งออก ระบบภายนอกหรือไม่
- ใครบ้างที่เป็นผู้ใช้งานข้อมูลนี้ ให้อย่างใด ให้ความสำคัญอย่างไร
- ลักษณะของรายงาน ประกอบด้วยรายงานอะไรบ้าง ระยะเวลาในการออกรายงาน
- ข้อมูลอื่นๆ ที่สามารถรวบรวมได้ โดยพยายามเก็บรายละเอียดให้มากที่สุด

ขั้นที่ 2 กำหนดโครงสร้างของเทเบิล

จากกลุ่มข้อมูลแต่ละเอนทิตีที่รวบรวมได้จากเอกสารต่างๆ ในขั้นที่ 1 นำมากำหนดแอตทริบิวต์หรือเนื้อหาของข้อมูล เพื่อจะได้ทราบว่าในเอนทิตีนั้นจะต้องนำข้อมูลอะไรมาใช้บ้าง หลังจากนั้นนำเอนทิตีและแอตทริบิวต์นี้มากำหนดโครงสร้างเบื้องต้นของเทเบิล โดยแปลงแอตทริบิวต์เป็นฟิลด์ของแต่ละเรคคอร์ดในเทเบิลนั้น พร้อมกำหนดชนิดและขนาดของข้อมูลในแต่ละฟิลด์ รวมทั้งเงื่อนไขหรือกฎเกณฑ์ที่ใช้กำหนดลักษณะข้อมูล

ขั้นที่ 3 กำหนดคีย์

ขั้นตอนนี้จะพิจารณาว่าฟิลด์ใดบ้างในเทเบิลมีคุณสมบัติเหมาะสมที่จะนำมาเป็นคีย์ต่างๆ เช่น Primary Key , Foreign Key หรือ Compound Key ถ้าไม่มีฟิลด์ใดเลยที่เหมาะสม จะต้องกำหนดฟิลด์ใหม่เพื่อใช้เป็นคีย์โดยเฉพาะ เช่น ในเทเบิลลูกค้าจะกำหนดฟิลด์รหัสลูกค้าขึ้นใหม่เพื่อเป็น Primary Key แทนฟิลด์ชื่อลูกค้า เนื่องจากชื่อลูกค้าอาจซ้ำกันได้

ขั้นที่ 4 การทำ Normalization

ถ้าเทเบิลที่สร้างในขั้นที่ 2 ยังมีความซ้ำซ้อนกันของข้อมูล หรือข้อมูลบางฟิลด์ไม่เกี่ยวข้องโดยตรงกับเนื้อหาในเทเบิลนั้น จะต้องนำมาปรับแก้ให้มีโครงสร้างหรือรูปแบบที่เหมาะสมก่อนนำไปประมวลผล ถ้านำโครงสร้างที่ออกแบบไว้มาใช้โดยไม่ทำ Normalization ก่อนอาจเกิดปัญหาได้ เช่น ปัญหาการสิ้นเปลืองเนื้อที่เนื่องจากเก็บข้อมูลซ้ำซ้อน ปัญหาความผิดพลาดของข้อมูลเมื่อมีการแก้ไข , เพิ่ม หรือลบเรคคอร์ด รวมทั้งปัญหาในการกำหนดความสัมพันธ์ในขั้นที่ 5 จะทำได้ยาก

ขั้นที่ 5 กำหนดความสัมพันธ์

นำเทเบิลทั้งหมดที่ได้หลังจากทำ Normalization มาสร้างความสัมพันธ์โดยใช้คีย์ที่กำหนดในขั้นที่ 3 หรือคีย์ที่เกิดขึ้นใหม่จากการทำ Normalization เป็นตัวเชื่อม ซึ่งอาจจะเป็นแบบ One-to-One , One-to-Many หรือ Many-to-Many ขึ้นกับลักษณะของข้อมูล การกำหนด

ความสัมพันธ์ระหว่างเทเบิลนี้มีความสำคัญมาก ผู้ออกแบบจะต้องวิเคราะห์ให้ได้ว่าข้อมูลในเทเบิลต่างๆ นั้นมีความสัมพันธ์กันในลักษณะใด (นันทน์ แขวงโสภา ,2544. หน้า 75-76)

2.7 เบื้องต้นกับ PHP

แต่เดิม PHP คือ Professional Home Page แต่ในปัจจุบัน PHP Hypertext Preprocessor ซึ่งเป็นภาษาสคริปต์แบบหนึ่งที่เราเรียกว่า Server Side Script ที่ประมวลผลฝั่งเซิร์ฟเวอร์ แล้วส่งผลลัพธ์ไปฝั่งไคลเอนต์ผ่านเว็บเบราว์เซอร์เช่นเดียวกับ ASP(Active Server Pages) ปัจจุบันได้รับความนิยมเนื่องจากมีจุดเด่นหลายประการ รูปแบบของภาษา PHP มีเค้าโครงมาจากภาษา C และ Perl ที่นำมาปรับปรุงทำให้มีประสิทธิภาพสูง และทำงานได้เร็วขึ้น ในปัจจุบันมีโปรแกรมเมอร์ใช้งานมากกว่า 1 ล้านคนแล้ว

2.7.1 ความเป็นมาของ PHP

PHP เกิดขึ้นในปี 1994 โดยโปรแกรมเมอร์ชาวสหรัฐอเมริกาชื่อ Rasmus Lerdorf ได้พัฒนาเครื่องมือที่ใช้สำหรับการสร้างเว็บเพจข้อมูลส่วนตัวของเขา โดยตอนแรกใช้ภาษา Perl แต่มีอุปสรรคในเรื่องความเร็ว เขาจึงพัฒนาเครื่องมือใหม่นี้ขึ้นมาโดยใช้ไวยากรณ์ภาษา C และเรียกว่า Personal Home Page ในขณะเดียวกันก็พัฒนาส่วนที่ใช้ติดต่อกับฐานข้อมูลที่เราเรียกว่า Form Interpreter(FI) ทั้งสองรวมกันเป็น PHP/FI ตรงนี้เองที่เป็นจุดเริ่มต้นของ PHP เนื่องจากเมื่อมีผู้เข้าชมเว็บเพจของเขาต่างนิยมชมชอบจึงติดต่อขอโค้ดเพื่อไปพัฒนาต่อในลักษณะที่เรียกว่า Open Source ด้วยเหตุนี้ในปี 1997 มีเว็บไซต์มากกว่า 50,000 แห่งที่ใช้ PHP/FI เพื่องานในด้านต่างๆ ทั้งการติดต่อฐานข้อมูล การแสดงข้อมูลแบบไดนามิก และอื่นๆอีกมากมาย

เมื่อมีผู้ใช้งานมากขึ้นก็มีความต้องการให้พัฒนาประสิทธิภาพของ PHP/FI ให้สูงขึ้น การพัฒนาด้วยตัวเองของ Rasmus Lerdorf ไม่เพียงพอ โชคดีที่มีผู้ช่วยเหลือ 2 คนที่ชื่อ Zeev Suraski กับ Andi Gutmans ชาวอิสราเอล เข้ามาปรับปรุงโค้ดเดิมของ Lerdorf โดยใช้ C++ และมีทีมงานเพิ่มเติมอีก 3 คนคือ Stig Bakken, Shane Caraveo และ Jim Winstead โดยนาย Stig Bakken รับผิดชอบเกี่ยวกับความสามารถในการสนับสนุน Oracle, Shane Caraveo ดูแล PHP บน Windows 9x/NT และ Jim Winstead คอยตรวจสอบข้อบกพร่องต่างๆอีกครั้ง และได้ชื่อเป็น Professional Home Pages สำหรับ PHP3 ที่ออกสู่สายตาโปรแกรมเมอร์เมื่อ มิ.ย.1998 ที่ผ่านมามีคุณสมบัติที่เด่นคือการสนับสนุนหลายแพลตฟอร์มของระบบปฏิบัติการ (ทั้ง Windows 95/98/ME/NT และ Linux) และเว็บเซิร์ฟเวอร์ (เช่น IIS, PWS, Apache, OmniHTTPd เป็นต้น) นอกจากนี้ยังสนับสนุนระบบฐานข้อมูลหลายแบบ(SQL Server, MySQL, mSQL, Oracle,

Informix, ODBC เป็นต้น) สนับสนุน SNMP(Simple Network Management Protocol) และ IMAP(Internet Message Access Protocol)

ปัจจุบัน Zeev และ Andi Gutmas ได้ร่วมพัฒนาต่อเป็น PHP4 (พ.ย. 43 ออกเวอร์ชันล่าสุดคือ 4.02) โดยตั้งชื่อว่า Zend ซึ่งเป้าหมายคือประสิทธิภาพที่เหนือกว่า ASP โดย Zend (www.zend.com) จะเป็น compile script ซึ่งต่างจากเดิมที่เป็น embed script interpreter ซึ่งจะทำให้ทำงานได้เร็วกว่า

ปัจจุบัน PHP จะหมายถึง PHP Hypertext Preprocessor ซึ่งบ่งบอกได้ว่าจะมีประสิทธิภาพในระดับโปรเฟสเซอร์เบื้องต้นสำหรับไฮเปอร์เท็กซ์

2.7.2 จุดเด่นของ PHP

ถึงแม้จะรู้จักและนำมาใช้งานได้ไม่นานนัก แต่PHPกลับได้รับความนิยมในการใช้เป็นเครื่องมือเพื่อพัฒนาเว็บเพจ เนื่องจากPHPมีจุดเด่นดังนี้

2.7.2.1 Free เนื่องจากสิ่งที่ต้องการสูงสุดของโปรแกรมเมอร์ในการพัฒนาเว็บ คือของฟรี PHP ได้ตอบสนองโปรแกรมเมอร์เป็นอย่างดีเพราะเครื่องมือที่ใช้พัฒนาทุกอย่าง สามารถหาได้ฟรีไม่ว่าจะเป็นระบบปฏิบัติการ(Windows, Linux), โปรแกรมเว็บเซิร์ฟเวอร์(IIS, PWS, Apache, OmniHTTPd), โปรแกรมระบบฐานข้อมูล(MySQL, mSQL) และ Server Site Script อย่าง PHP

2.7.2.2 Speed เนื่องจาก PHP นำข้อดีของภาษาสคริปต์ที่เคยมีในภาษา C, Perl และ Java รวมกับความเร็วของ CGI นำมาพัฒนาอยู่ใน PHP

2.7.2.3 Open Source เนื่องจากการพัฒนาของ PHP ไม่ได้ยึดติดกับบุคคลหรือกลุ่มคนเล็กๆแต่เปิดโอกาสให้โปรแกรมเมอร์ทั่วไปได้เข้ามาช่วยพัฒนา ทำให้มีคนใช้งานจำนวนมาก และพัฒนาได้เร็วขึ้น

2.7.2.4 Crossable Platform เนื่องจาก PHP ใช้ได้กับหลายๆระบบปฏิบัติการไม่ว่าบน Windows, Unix, Linux หรืออื่นๆโดยแทบจะไม่ต้องเปลี่ยนแปลงโค้ดคำสั่งเลย

2.7.2.5 Database Access เนื่องจาก PHP สามารถติดต่อกับฐานข้อมูลอย่าง dBASE, Access, SQL Server, Oracle, Sybase, Informix, PostgreSQL, Empress, FilePro, mSQL, PostgreSQL ได้อย่างมีประสิทธิภาพ

2.7.2.6 Protocol Support เนื่องจาก PHP สามารถสนับสนุนโปรโตคอลหลายแบบ ทั้ง IMAP, SNMP, NNTP, POP3, HTTP

2.7.2.7 Library เนื่องจาก PHP มีไลบรารีสำหรับการติดต่อกับแอปพลิเคชันได้มากมาย



2.7.2.8 Flexible ด้วยเหตุที่ PHP มีความยืดหยุ่นตัวสูง ทำให้สามารถนำไปสร้างแอปพลิเคชันได้หลายประเภท

2.7.2.9 Easy เนื่องจาก PHP เป็นภาษาคำสั่งที่เรียบง่าย ทำให้สามารถแทรกตำแหน่งใดก็ได้ในแท็กของ HTML

2.7.3 PHP ทำอะไรได้บ้าง

PHP ทำสิ่งที่คุณต้องการ รวมทั้งการจัดการเกี่ยวกับกราฟฟิกและไดนามิก HTML ด้วย ตามคู่มือของ PHP ที่กล่าวว่า "The goal of the language is to allow Web developers to write dynamically generated pages quickly." นั่นคือเป้าหมายหลักของ PHP โดยเฉพาะเรื่องไดนามิกที่สามารถเขียนได้อย่างรวดเร็ว ต่อไปนี้เป็นงานทั่วไปที่ PHP สามารถทำได้

2.7.3.1 ทำตามฟังก์ชันของระบบ ได้แก่ การสร้าง, การเปิด, อ่าน และปิดไฟล์ในระบบ

2.7.3.2 เอ็กซีคิวต์คำสั่งของระบบ ได้แก่ การสร้างไพลเดอร์ และปรับแต่งสิทธิการใช้งาน

2.7.3.3 จัดการข้อมูลจากฟอร์ม ได้แก่ การจัดการข้อมูลลงไฟล์, การส่งข้อมูลผ่านทางอีเมลล์, ส่งค่าข้อมูลอัปเดตข้อมูลในฐานข้อมูลของคุณ

2.7.3.4 เซ็ตคุกกี้และเอ็กเซสตัวแปรคุกกี้

2.7.3.5 ใช้ PHP เพื่อรักษาความปลอดภัยของเว็บเพจ

2.7.3.6 เข้ารหัสข้อมูล

2.7.4 PHP กับฐานข้อมูล

ความสามารถของ PHP อีกอย่างหนึ่งที่ถือว่าเป็นความสามารถที่โดดเด่นของ PHP คือการติดต่อกับระบบฐานข้อมูลต่างๆ เพื่อแสดงบนเว็บทำได้ง่ายและมีประสิทธิภาพมาก ซึ่งในปัจจุบันในแต่ละหน่วยงานหรือแต่ละบริษัทมีระบบฐานข้อมูลที่แตกต่างกันออกไป ดังนั้นถ้ามีเครื่องมือที่จะนำข้อมูลจากฐานข้อมูลที่แตกต่างกันเหล่านี้ออกมาใช้งานได้ก็ยิ่งเป็นผลดีทั้งในการบำรุงรักษา การซ่อมแซม ตลอดจนการใช้งานได้เต็มประสิทธิภาพ PHP ได้เตรียมฟังก์ชันเพื่อสนับสนุนการติดต่อกับระบบฐานข้อมูลไว้มากมาย โดยระบบฐานข้อมูลที่ PHP สนับสนุนได้มีดังนี้

2.7.4.1 Adabas D

2.7.4.2 dBase

2.7.4.3 FilePro

2.7.4.4 Informix

2.7.4.5 Microsoft SQL Server

2.7.4.6 mSQL

2.7.4.7 MySQL

2.7.4.8 ODBC

2.7.4.9 Oracle

2.7.4.9 PostgreSQL

2.7.4.10 Solid

2.7.4.11 Sybase

2.7.4.12 Unix dbm

2.7.4.13 Velocis

2.7.5 หลักการทำงานของ PHP

เนื่องจาก PHP จะทำงานโดยมีตัวแปลและเอ็กซิคิวต์ที่ฝั่งเซิร์ฟเวอร์ อาจจะเรียกการทำงานว่าเป็นเซิร์ฟเวอร์ไซด์(Server Side) ส่วนการทำงานของบราวเซอร์ของผู้ใช้เรียกว่าไคลเอนต์ไซด์(Client Side) โดยการทำงานจะเริ่มต้นที่ผู้ใช้ส่งความต้องการผ่านเว็บบราวเซอร์ทาง HTTP (HTTP Request) ซึ่งอาจจะเป็นการกรอกแบบฟอร์ม หรือใส่ข้อมูลที่ต้องการ ข้อมูลเหล่านั้นจะเป็นเอกสาร PHP (เอกสารนี้จะเป็นส่วนขยายเป็น PHP หรือ PHP3 แล้วแต่ผู้กำหนด เช่น search.php เป็นต้น) เมื่อเอกสาร PHP เข้ามาถึงเว็บเซิร์ฟเวอร์ก็จะถูกส่งไปให้ PHP เพื่อทำหน้าที่แปลคำสั่งแล้วเอ็กซิคิวต์คำสั่งนั้น หลังจากนั้น PHP จะสร้างผลลัพธ์ในรูปแบบเอกสาร HTML ส่งกลับไปให้เว็บเซิร์ฟเวอร์เพื่อส่งต่อไปให้บราวเซอร์แสดงผลทางฝั่งผู้ใช้ต่อไป (HTTP Response) ซึ่งลักษณะการทำงานแบบนี้จะคล้ายกับการทำงานของ CGI (Common Gateway Interface) หรืออาจกล่าวได้ว่า PHP ก็คือโปรแกรม CGI ประเภทหนึ่งก็ได้จะทำงานคล้ายกับ ASP นั่นเอง

2.7.6 โครงสร้างภาษา PHP

โครงสร้างภาษา PHP ก็คล้ายกับโครงสร้างของภาษาคอมพิวเตอร์ทั่วไป ไม่ว่าจะเป็นภาษา C, ภาษา VBScript หรือภาษาอื่นๆ ที่ประกอบด้วยหลายๆ ส่วนที่ช่วยให้โปรแกรมเมอร์สร้างสรรค์รูปแบบโปรแกรมที่ต้องการ ไม่ว่าจะเป็นชนิดข้อมูล ตัวแปร โอเปอเรเตอร์ คำคงที่ เงื่อนไขเพื่อการตัดสินใจตลอดจนการทำงานวนรอบ สิ่งเหล่านี้เป็นพื้นฐานของการศึกษาเพื่อการใช้งานภาษาที่เรียกว่า Programming Language

2.7.6.1 ชนิดของข้อมูล

ในการพัฒนาโปรแกรมทั่วไป ข้อมูลมีความสำคัญมากที่สุด ชนิดของข้อมูลมีมากมาย เพื่อให้การใช้งานได้ง่ายขึ้น PHP จึงได้แบ่งหมวดหมู่ข้อมูลไว้ 6 แบบ ดังตาราง 2.1

ชนิด	คำอธิบาย
Integer	เก็บข้อมูลตัวเลขที่เป็นจำนวนเต็มทั้งจำนวนเต็มบวก และจำนวนเต็มลบ รวมทั้งสามารถแสดงค่าเป็นเลขฐานสิบ(0-9) ฐานแปด(0-7) และฐานสิบหก(0-9,A-F หรือ a-f)
Floating point numbers	เก็บข้อมูลตัวเลขจำนวนจริงทั้งบวกและลบ ที่มีทศนิยมหรือไม่มีทศนิยมก็ได้
String	เก็บข้อมูลตัวอักษร หรือข้อความ เช่น "Hello", "สวัสดีครับ", "123"
Array	เก็บข้อมูลที่เป็นชุด หรืออาร์เรย์
Object	เก็บข้อมูลในลักษณะออบเจกต์เพื่อการเรียกใช้เป็น Class Object หรือ Function
Type juggling	เก็บข้อมูลในลักษณะเฉพาะหรือผู้ใช้เพิ่มเติมเข้ามา

ตารางที่ 2.1 แสดงชนิดของข้อมูล

2.8 เบื้องต้นกับ MySQL

MySQL เป็นโปรแกรมฐานข้อมูลในลักษณะ Database Server ซึ่งทำงานทั้งบนระบบ Telnet บน Linux Redhat หรือ Unix System และบน Win32 (Windows 95/98/ME) เพื่อใช้กับ Internet & Intranet หมายความว่า คุณสามารถเรียกใช้ MySQL ได้ทั่วโลก กรณีเป็น Internet และยังสามารถเรียกใช้บนเว็บเบราว์เซอร์ได้ในกรณีที่ใช้ภาษาอินเทอร์เฟซเข้ามาใช้งานฐานข้อมูล เช่น PHP , Perl, C, C++ ฯลฯ

2.8.1 คำสั่งเบื้องต้นสำหรับ MySQL

สำหรับคำสั่งเพื่อการใช้งาน MySQL บน Windows มีคำสั่งพื้นฐานในการใช้งานใน ดังนี้

2.8.1.1 help ดูระบบช่วยเหลือ

2.8.1.2 status แสดงสถานะของ MySQL เช่น เวอร์ชัน ฐานข้อมูลที่ใช้อยู่ ผู้ใช้ปัจจุบัน เป็นต้น

2.8.1.3 exit ออกจาก MySQL

2.8.1.4 quit ออกจาก MySQL

2.8.1.5 use ใช้ฐานข้อมูล

2.8.1.6 create database สร้างฐานข้อมูลใหม่

2.8.1.7 create table สร้างตารางใหม่

2.8.1.8 show databases แสดงฐานข้อมูลที่มีอยู่ใน MySQL

- 2.8.1.9 show tables แสดงตารางที่มีอยู่ในฐานข้อมูลปัจจุบันที่ใช้อยู่
- 2.8.1.10 select เลือกฟิลด์ที่จะแสดงผลข้อมูล
- 2.8.1.11 insert into เพิ่มข้อมูลเข้าสู่ตารางที่กำหนด
- 2.8.1.12 delete from ลบข้อมูลออกจากตารางตามเงื่อนไข
- 2.8.1.13 load data local infile โหลดข้อมูลจากเท็กซ์ไฟล์เข้าสู่ตาราง