บทที่ 3

ระบบฐานข้อมูล

ระบบสารสนเทศ

ระบบสารสนเทศคือ ระบบที่เก็บรวบรวมข้อมูล ซึ่งข้อมูลนั้นอาจใช้เป็นข้อมูลพื้นฐานที่ใช้ในการคาดการณ์ การพยากรณ์ หรือเป็นข้อมูลที่ช่วยในการตัดสินใจ หรืออาจเป็นข้อมูลที่ต้องผ่านขั้นตอนต่าง ๆ เพื่อวิเคราะห์ข้อมูลนั้นในรูปแบบต่าง ๆ แล้วจึงนำผลที่ได้มาช่วยในการวางแผน ช่วยในการจัดการ ตลอดจนช่วยในการตัดสินใจ ในปัจจุบันที่เทคโนโลยีทางคอมพิวเตอร์ และการสื่อสารมีความเจริญก้าวหน้าอย่างมาก ระบบสารสนเทศจึงหมายถึงระบบที่นำคอมพิวเตอร์มาช่วยประมวลผล การนำข้อมูลเก็บไว้ในระบบคอมพิวเตอร์ หรือการนำข้อมูลมาใช้งานจึงต้องใช้ชุดคำสั่งหรือโปรแกรมที่เป็นตัวเชื่อมระหว่างผู้ใช้ระบบและข้อมูลที่เก็บไว้ในคอมพิวเตอร์ ฉะนั้นระบบสารสนเทศจึงต้องมีส่วนประกอบที่สำคัญ 2 ส่วนดังรูปที่ 3.1 คือ

1. ระบบฐานข้อมูล (Database system)
2. โปรแกรมประยุกต์ (Application program) ซึ่งทำหน้าที่ประมวลผลข้อมูลตามความต้องการของผู้ใช้ระบบ

ระบบฐานข้อมูล (Whittington, 1988) ประกอบด้วย ฐานข้อมูล (Database) และระบบจัดการฐานข้อมูล (Database management system)

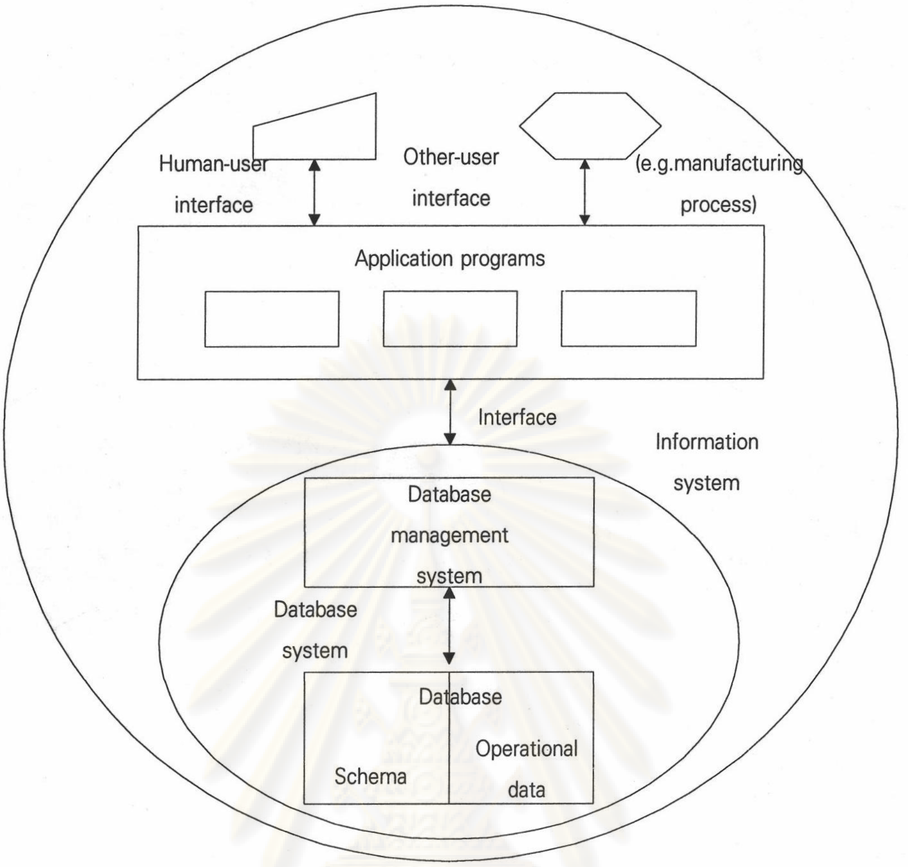
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ฐานข้อมูล (Database)

นิยามของคำว่า ฐานข้อมูล (Database) มีได้หลายรูปแบบ ขึ้นอยู่กับหลักเกณฑ์ที่ใช้ในการพิจารณา ซึ่งพอสรุปได้ดังนี้ (Whittington, 1988)

1. พิจารณาจากภาพรวม ฐานข้อมูลแบ่งเป็น

   1.1 ฐานข้อมูลเชิงกายภาพ (Physically) คือ บิต (Bit) จำนวนมากที่บันทึกไว้บนสื่อที่เป็นหน่วยเก็บ (Storage media)



รูปที่ 3.1 ความสัมพันธ์ระหว่างระบบฐานข้อมูล (Database system) และระบบสารสนเทศ (Information system) (Whittington, 1988)

1.2 ฐานข้อมูลเชิงความหมาย (Semantically) คือ ตัวแทนของการบรรยายถึงสิ่งทั้งมวล (Universe of discourse)

2. พิจารณาจากบทบาทหน้าที่ของผู้ที่เกี่ยวข้องกับขบวนการ การพัฒนาระบบ

2.1 ผู้จัดการองค์กร

ฐานข้อมูลคือ ทรัพยากรที่มีคุณค่า และราคาแพงจึงต้องจัดการอย่างมีประสิทธิภาพ และใช้ประโยชน์ได้อย่างเต็มที่

2.2 นักเขียนโปรแกรม

ฐานข้อมูลคือ ที่ที่รวบรวมประเภทของระเบียน (Record Type) และ เส้นทางชี้นำ (Navigation paths)

2.3 ผู้ใช้ระบบ

ฐานข้อมูลคือ แหล่งของสารสนเทศ (Source of information)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3. พิจารณาจากด้านเครื่องจักร (Machine-eye)

ฐานข้อมูลเป็นที่รวมของแฟ้มข้อมูล บางแฟ้มข้อมูลจะเก็บข้อมูลดำเนินการ (Operational data) บางแฟ้มข้อมูลเก็บสารสนเทศระบบ (System information) รวมถึงเค้าร่างของฐานข้อมูล (Database schema) และดรรชนี เพื่อการเข้าถึงข้อมูลอย่างมีประสิทธิภาพ

ข้อมูลดำเนินการ (Engles. 1974 อ้างถึงใน Date. 1986) แตกต่างจากข้อมูลเข้า (Input data) ข้อมูลออก (Output data) หรือข้อมูลประเภทอื่น เช่น ลำดับงาน (Work sequence) ผลลัพธ์ชั่วคราว หรือ สารสนเทศชั่วครู่ (Transient information) แต่หมายถึง เอนทิตีพื้นฐาน (Basic entities)

ข้อมูลนำเข้า หมายถึงข้อมูลที่บันทึกเข้าระบบครั้งแรก ผ่านทางแผงแป้นอักขระ ตัวกราดตรวจ (Scanner) หรืออุปกรณ์นำเข้าอื่น ๆ และผ่านขั้นตอนการเปลี่ยนแปลงจึงได้ข้อมูลดำเนินการ

ข้อมูลออก หมายถึงข่าวสาร (Message) และผลที่ได้จากการนำข้อมูลดำเนินการผ่านขบวนการต่าง ๆ และนำมาแสดงทางจอภาพหรือรายงาน

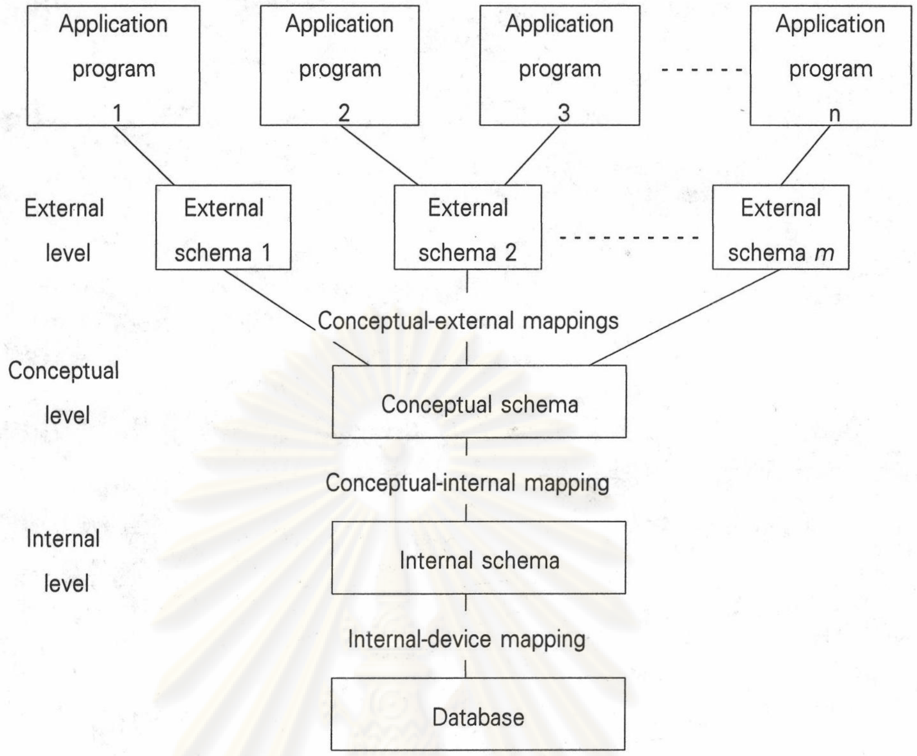
อย่างไรก็ตามนิยามของคำว่า ฐานข้อมูลที่นิยมใช้กันมากคือ คลังข้อมูลที่เบ็ดเสร็จ (Integrated) และอธิบายตัวมันเองได้ (Self-describing)

เบ็ดเสร็จหมายถึง ฐานข้อมูลที่มีความครบถ้วน (Wholeness) สามารถรักษาให้มีความสอดคล้องกัน (Consistent) เมื่อมีการอ้างถึงซึ่งกันและกัน (Cross-reference) และนำมาใช้ประโยชน์ได้โดยการใช้โปรแกรมประยุกต์

สถาปัตยกรรมฐานข้อมูล

The ANSI/SPARC Study Group on Database Management Systems (ANSI, 1975) ได้นำเสนอหลักการของฐานข้อมูล ซึ่งได้ใช้เป็นแนวทางในการกำหนดกรอบงาน (Framework) ของระบบจัดการฐานข้อมูล ในขณะเดียวกันหลักการดังกล่าวก็สามารถใช้เป็นข้อมูลในการเปรียบเทียบ และประเมินคุณสมบัติของระบบจัดการฐานข้อมูลทั่ว ๆ ไปของแต่ละบริษัท หรือแต่ละยี่ห้อว่ามีความแตกต่าง หรือเหมือนกันอย่างไร

ในรายงานดังกล่าวได้บรรยาย และเน้นถึงระบบจัดการฐานข้อมูลในลักษณะของความสัมพันธ์ระหว่างบทบาทของคน หน้าที่การประมวลผล (Processing functions) และการไหลไปของข้อมูล (Information flow) และเสนอลักษณะสำคัญของการย่อข้อมูล (Data abstraction) เป็น 3 ระดับคือ คือ เค้าร่างภายใน (Internal schema) เค้าร่างมโนภาพ (Conceptual schema) และเค้าร่างภายนอก (External schema) ดังแสดงในรูปที่ 3.2



รูปที่ 3.2 ระบบการจัดการฐานข้อมูลแบบ 3 ระดับ (Whittington, 1988)

1. เค้าร่างภายใน (Internal schema) อธิบายการจัดการข้อมูลในระดับต่ำ แต่มิใช่การเก็บข้อมูลระดับกายภาพ เพราะมิได้อธิบายถึงการเก็บข้อมูลเป็นทรงกระบอก (Cylinder) หรือเป็นวง (Track) (Date, 1986) ลักษณะการเก็บข้อมูลกายภาพจะต้องพึ่งพิงโครงสร้างของหน่วยเก็บเพราะหน่วยเก็บแต่ละชนิดจะมีโครงสร้างแตกต่างกันไป เค้าร่างภายในเป็นการอธิบายถึงข้อมูลในระดับที่อยู่เหนือจากกายภาพ และไม่พึ่งพิงโครงสร้างของหน่วยเก็บ โดยจะอธิบายถึงประเภทระเบียน (Record) เขตข้อมูล (Field) มีอะไรบ้าง เก็บอย่างไร ใช้อะไรเป็นดรรชนี มีการเรียงลำดับกายภาพ (Physical sequence) อย่างไร ความยาวของระเบียน เค้าร่างภายในเป็นเพียงมุมมองภายในของลักษณะการเก็บข้อมูลในฐานข้อมูล และเพื่อเป็นการเชื่อมโยงระหว่างเค้าร่างภายในกับการเก็บข้อมูลกายภาพที่ขึ้นอยู่กับโครงสร้างของฮาร์ดแวร์ (Hardware) นั้นจะต้องใช้วิธีแปลงส่ง (Map) ภาษาที่ใช้อธิบายลักษณะข้อมูลในระดับนี้เรียกว่า Internal data definition language หรือ Internal-DDL เป็นภาษาของอุปกรณ์เฉพาะ (Device dependent language)

โดยทั่วไปการเขียนโปรแกรมประยุกต์จะไม่ได้รับอนุญาตให้จัดการข้อมูลในระดับเค้าร่างภายในนี้ นอกจากโปรแกรมอรรถประโยชน์ เพราะโปรแกรมประยุกต์ที่เข้ามาจัดการเค้าร่างภายในนี้จะไม่ผ่านระบบรักษาความปลอดภัยทำให้เกิดความเสี่ยงต่อบูรณภาพของข้อมูล (Data integrity)

แต่บางครั้งเพื่อผลการกระทำ (Performance) ที่ดี หรือเพื่อให้ได้ฟังก์ชันบางอย่างตามต้องการ นักเขียนโปรแกรมจำเป็นต้องเขียนโปรแกรมจัดการถึงระดับเค้าร่างภายในนี้

2. เค้าร่างมโนภาพ (Conceptual schema) หรือการมองภาพรวมของข้อมูลธุรกิจทั้งหมด (Fleming, Halle, 1989) ภาพที่เห็นในระดับนี้จะเป็นตัวแทนของเนื้อหาทั้งหมดของฐานข้อมูล ซึ่งเป็นการมองข้อมูลที่แท้จริงที่ควรจะเป็น โดยไม่สนใจว่าจะใช้ภาษาอะไรในการจัดการ หรือใช้ฮาร์ดแวร์อะไรในการเก็บข้อมูล ผู้บริหารฐานข้อมูลที่เรียกว่า Database administrator หรือ DBA ทำหน้าที่รวบรวม (Integrate) ความต้องการของผู้ใช้ในองค์กรทั้งหมด และกำหนดเนื้อหาของฐานข้อมูลควรจะเก็บข้อมูลอะไรบ้าง อย่างไร และสร้างเป็นเค้าร่างมโนภาพ โดยไม่สนใจโครงสร้างของหน่วยเก็บหรือกลยุทธ์ในการเข้าถึงข้อมูล สิ่งที่จะกำหนดในเค้าร่างมโนภาพมีดังนี้

2.1 กำหนดลักษณะของข้อมูล (Data definition) อธิบายประเภทข้อมูล ความยาวของเขตข้อมูลที่เล็กที่สุดในฐานข้อมูล ในฐานข้อมูลนั้นจะประกอบด้วยกลุ่มข้อมูล ข้อมูลแต่ละกลุ่มเรียกว่า รีเลชัน (Relations) หรือเอนทิตี (Entities) หรือเซทของเอนทิตี (Entity set) และ 1 เอนทิตีประกอบด้วย ชิ้นข้อมูล (Data items) เรียกว่า แอตตริบิว (Attributes) หรือเขตข้อมูล แต่โดยทั่วไปจะใช้คำว่า เอนทิตี (Entity) แทนคำว่า เซทของเอนทิตี เค้าร่างมโนภาพจะกำหนดว่าฐานข้อมูลประกอบด้วย เอนทิตีอะไรบ้าง (Entity type) แต่ละเอนทิตีประกอบด้วยระเบียนที่มีความยาวเท่าใด มีแอตตริบิว (Attribute) อะไรบ้าง ประเภทและความยาวของแต่ละแอตตริบิว เป็นการอธิบายถึงระเบียนเชิงตรรกะ รวมถึงการกำหนดการตรวจสอบความปลอดภัย และบูรณภาพของข้อมูล จะใช้ข้อมูลอย่างไร การไหลของข้อมูล การตรวจสอบและควบคุมข้อมูลแต่ละจุด

2.2 ความสัมพันธ์ของข้อมูล (Data relationship) กำหนดการเชื่อมโยงของข้อมูลระหว่างเอนทิตีที่มีความสัมพันธ์กัน

ข้อกำหนดในเค้าร่างมโนภาพนี้จะเป็นลักษณะของระเบียนเชิงตรรกะ (Logical records) ไม่มีข้อมูลเกี่ยวข้องกับหน่วยเก็บ (Storage) การกำหนดลักษณะเค้าร่างมโนภาพจะใช้ภาษาที่เรียกว่า Conceptual data definition language หรือ Conceptual-DDL ซึ่งจะไม่พึ่งพิงต่อโครงสร้างของหน่วยเก็บเช่นกัน เค้าร่างมโนภาพนี้นอกจากจะทำหน้าที่ควบคุมลักษณะข้อมูลทั้งหมดขององค์กรแล้ว ยังทำหน้าที่เป็นตัวเชื่อมทางอ้อมระหว่างเค้าร่างภายในและเค้าร่างภายนอก และการเชื่อมโยงระหว่างเค้าร่างมโนภาพและเค้าร่างภายในจะใช้วิธีการแปลงส่ง เรียกว่าการแปลงส่งเค้าร่างมโนภาพและเค้าร่างภายใน (Conceptual/internal mapping) เพื่อบอกให้ทราบว่าจะเก็บระเบียนและแอตตริบิวอย่างไรในเค้าร่างภายใน แต่การแปลงส่งนี้จะกระทำโดยระบบซึ่งผู้ใช้ระบบจะมองไม่เห็น หรือไม่จำเป็นต้องเรียนรู้วิธีการ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3. เค้าร่างภายนอก (External schema) เค้าร่างภายนอกนี้เป็นการมองข้อมูลของผู้ใช้แต่ละคนซึ่งอาจเป็นผู้ใช้ขั้นสุดท้าย (End user) ต้องการใช้ข้อมูลในลักษณะเชื่อมตรง (On-line) หรือเป็นนักเขียนโปรแกรมจะใช้ข้อมูลในแต่ละโปรแกรมเพื่อประมวลผลตามต้องการ เค้าร่างภายนอกจะมองภาพข้อมูลจากฐานข้อมูลเป็นส่วน ๆ อธิบายมุมมองของผู้ใช้แต่ละคน ภาพของข้อมูลที่ผู้ใช้แต่ละคนเห็นนั้นอาจไม่เหมือนกัน ผู้ใช้หนึ่งคนอาจกำหนดเค้าร่างภายนอกหนึ่งชุดหรือหลายชุด ทำนองเดียวกันเค้าร่างภายนอกหนึ่งชุดมีผู้ใช้ได้หลายคน และเค้าร่างภายนอกแต่ละชุดอาจเหมือนหรือไม่เหมือนกับข้อมูลที่เก็บไว้จริงในฐานข้อมูล แต่จะมีการบำรุงรักษาให้มีความสอดคล้องกับเค้าร่างมโนภาพ กล่าวคือเค้าร่างภายนอกนี้จะเป็นการบอกลักษณะข้อมูลที่ใช้ในโปรแกรมประยุกต์แต่ละโปรแกรมตามความต้องการของผู้ใช้แต่ละคน โดยอธิบายให้ทราบว่าเค้าร่างภายนอกแต่ละชุดนั้นประกอบด้วยเอนทิตีอะไรบ้างจากฐานข้อมูล ไม่ใช่ทุกเอนทิตี แอตตริบิวอะไรบ้าง หรือทุกแอตตริบิวของระเบียน เค้าร่างแต่ละชุดจึงเป็นเพียงส่วนหนึ่งของเค้าร่างมโนภาพ บางครั้งจึงเรียกเค้าร่างภายนอกแต่ละชุดว่า เค้าร่างย่อย (Subschema) หรือเซทย่อย (Subset) ดังรูปที่ 3.2 ประกอบด้วยเค้าร่างภายนอก m ชุด แต่ละชุดเป็นส่วนหนึ่งของเค้าร่างมโนภาพ

การจัดการเค้าร่างภายนอกจะใช้ภาษาสองลักษณะคือ ภาษาที่ใช้อธิบายถึงฐานข้อมูลในระดับเค้าร่างภายนอกเรียกว่า External data definition language หรือ External-DDL โดยทั่วไปจะเรียกว่า DDL และภาษาที่ใช้จัดการหรือประมวลผลข้อมูลเหล่านั้นเรียกว่า Data manipulation language หรือ DML และเพื่อให้เค้าร่างภายนอกเชื่อมโยงกับเค้าร่างมโนภาพจึงต้องมีการแปลงส่งซึ่งเรียกว่า การแปลงส่งเค้าร่างภายนอกและเค้าร่างมโนภาพ (External/conceptual mapping)

การทำงานของระบบฐานข้อมูลที่ใช้วิธีแปลงส่ง (Mapping) ระหว่างเค้าร่างมโนภาพและเค้าร่างภายใน และระหว่างเค้าร่างภายนอกและเค้าร่างมโนภาพ เมื่อมีการเปลี่ยนแปลงลักษณะการเก็บข้อมูลทางกายภาพ จะต้องมีการปรับปรุงการแปลงส่งเค้าร่างมโนภาพและเค้าร่างภายใน (Conceptual/internal mapping) แต่ไม่ต้องแก้ไขในส่วนของเค้าร่างมโนภาพ ซึ่งเรียกคุณสมบัตินี้ว่า ความไม่พึ่งพิงข้อมูล (Data independence) หรือความไม่พึ่งพิงหน่วยเก็บข้อมูล (Data-storage independence) (Whittinton, 1988)

กรณีการประยุกต์ใช้ข้อมูลของผู้ใช้ระบบมีการเปลี่ยนแปลง เช่น มีการเพิ่มโปรแกรมประยุกต์ การแก้ไขนั้นเพียงเพิ่มเค้าร่างภายนอก และการแปลงส่งเค้าร่างภายนอกและเค้าร่างมโนภาพ หรือต้องการปรับปรุงโปรแกรมประยุกต์ที่มีอยู่ก็เพียงปรับเฉพาะการแปลงส่งเฉพาะส่วนที่ปรับปรุงเท่านั้น ไม่มีผลกระทบต่อเค้าร่างภายนอกอื่น และไม่จำเป็นต้องแก้ไขเค้าร่างมโนภาพ

หรือเค้าร่างภายใน คุณลักษณะเช่นนี้เรียกว่า ความไม่พึ่งพิงโปรแกรมประยุกต์ (Data-application independent) (Whittington, 1988)

ระบบฐานข้อมูลที่มีโครงสร้างตามแนวทางที่ ANSI/SPARC กำหนดไว้ และนิยมใช้เป็นแบบจำลองข้อมูลในการพัฒนาระบบจัดการฐานข้อมูลคือ ฐานข้อมูลแบบสัมพันธ์ (Relational database) ฐานข้อมูลชนิดนี้จะมองข้อมูลในลักษณะของตาราง (Table) 2 มิติ แต่ละตารางจะแยกจากกัน ไม่เป็นลำดับชั้น หรือโครงสร้างที่ซับซ้อน การค้นคืนข้อมูลจึงสามารถเข้าถึงแต่ละตารางได้โดยตรงโดยไม่ต้องกำหนดเส้นทาง (Path) ไว้ล่วงหน้า และสามารถเชื่อมโยงตาราง 2 ตารางด้วยเขตข้อมูลที่มีความสัมพันธ์กัน ในระบบฐานข้อมูลแบบสัมพันธ์มีส่วนประกอบต่าง ๆ และคำศัพท์ (Terminology) ที่ใช้เรียกกันดังนี้ (Tsai, 1988)

1. รีเลชัน (Relation) หมายถึงตาราง ซึ่งเป็นที่ที่เก็บข้อมูลที่มีความสัมพันธ์กัน ตารางประกอบด้วยรายการ (Occurrences) และแต่ละรายการจะประกอบด้วยเขตข้อมูล (Field)

2. แอตตริบิว (Attribute) แต่ละสดมภ์ของรีเลชันจะเรียกว่า แอตตริบิว หรือเขตข้อมูล (Field) นั่นเอง

3. โดเมน (Domain) คือเซทของข้อมูลที่แต่ละแอตตริบิวสามารถมีค่าได้ เช่น โดเมนของแอตตริบิว ฤดู (Season) คือ ฤดูใบไม้ร่วง ฤดูร้อน ฤดูใบไม้ผลิ ฤดูฝน เป็นต้น

4. ทัพเพิล (Tuple) คือแถว (Row) ของตาราง (Table) แต่ละทัพเพิลจะแทนข้อมูล 1 รายการ (Record occurrence) และ 1 รีเลชัน จะประกอบด้วยเซทของทัพเพิล

5. กุญแจหลัก หรือคีย์หลัก (Primary key) และแคนดิเดทคีย์ (Candidate key) เขตข้อมูลที่สามารถกำหนดข้อมูลของทัพเพิลได้ชัดเจน หรือยูนีค (Uniquely identify) เรียกว่า แคนดิเดทคีย์ (Candidate key) แต่ละรีเลชันมีได้หลายแคนดิเดทคีย์ แต่หนึ่งในหลายแคนดิเดทคีย์จะถูกกำหนดให้ทำหน้าที่เป็นคีย์หลักซึ่งอาจประกอบด้วยหนึ่งหรือหลายแอตตริบิว

6. กุญแจภายนอก หรือฟอเรนคีย์ (Foreign key) เขตข้อมูลที่มีค่าตรงกับค่าของคีย์หลักในอีกรีเลชัน

7. แฟ้มระนาบ (Flat file) คือแฟ้มข้อมูลที่แต่ละแถวจะประกอบด้วยแอตตริบิวที่แต่ละแอตตริบิวจะมีค่าเพียง 1 ค่า การสร้างรีเลชันของฐานข้อมูลควรจะมีลักษณะของแฟ้มระนาบ ดังตัวอย่างรูปที่ 3.3 แฟ้ม PROJECT เป็นแฟ้มระนาบ แต่ละ PROJECT ID จะมีแอตตริบิว PROJECT

MANAGER ได้เพียง 1 ค่า และตัวอย่างในรูปที่ 3.4 แฟ้ม EMPLOYEE ไม่เป็นแฟ้มระนาบ เพราะแอตตริบิว CHILDREN มีค่ามากกว่า 1 ค่า

PROJECT file

PROJECT ID	PROJECT NAME	PROJECT MANAGER
P001	LENDING	E.LEE
P002	FUNDING	H.COHEN
P003	SECURITY	G.TURNER

รูปที่ 3.3 ตัวอย่างตารางที่มีลักษณะเป็นแฟ้มระนาบ

EMPLOYEE file

EMP-NO	EMP-NAME	CHILDREN
E105	E.LEE	EDWARD, CINDY
E106	H.COHEN	DAN
E107	G.TURNER	MARY, BOB, GREG
E108	R.CHARON	ALBERT

รูปที่ 3.4 ตัวอย่างตารางที่ไม่ใช่แฟ้มระนาบ เพราะ CHILDREN มีค่าได้มากกว่า 1 ค่า (Tsai, 1988)

ฐานข้อมูลแบบสัมพันธ์ซึ่งมีโครงสร้าง 3 ระดับข้างต้นจึงมีข้อดีดังนี้

1. เป็นโครงสร้างที่เข้าใจง่าย ยืดหยุ่น สามารถปรับปรุงได้ง่าย
2. สามารถเข้าถึงข้อมูลแต่ละตารางได้โดยตรง ไม่ยุ่งยาก
3. มีลักษณะของความไม่พึ่งพิงข้อมูล และความไม่พึ่งพิงโปรแกรมประยุกต์
4. สามารถทำงานแบบข้อคำถามที่มิได้เตรียมล่วงหน้า (Ad-hoc queries) ได้อย่างมีประสิทธิภาพมากกว่างานผลิต (Production)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ระบบจัดการฐานข้อมูล (Database management system - DBMS)

เพื่อให้การทำงานระบบฐานข้อมูลเป็นไปตามสถาปัตยกรรมข้างต้นจะต้องมีตัวจัดการฐานข้อมูลที่สอดคล้องกับโครงสร้างนั้น ระบบฐานข้อมูลจึงมีส่วนประกอบที่สำคัญอีกส่วนคือระบบจัดการฐานข้อมูล (DBMS)

ระบบจัดการฐานข้อมูลประกอบด้วยซอฟต์แวร์รูทีน (Software routine) จำนวนมากที่มีความสัมพันธ์กัน โดยแต่ละรูทีนจะทำงานเฉพาะอย่าง (Tsai. 1988) แต่หน้าที่สำคัญของระบบจัดการฐานข้อมูล (Whittington. 1988) คือ

1. การกำหนดรายละเอียดข้อมูล (Data definition) การสร้าง และการปรับปรุงโครงสร้างของฐานข้อมูล ตลอดจนทำหน้าที่บรรจุข้อมูลดำเนินการเข้าสู่ฐานข้อมูล
2. การจัดการข้อมูล (Data manipulation) ทำการปรับปรุง (Update) หรือค้นคืน (Retrieval) ข้อมูลจากฐานข้อมูลทันทีพร้อม ๆ กันหลายทาง และตอบสนองตามความต้องการของผู้ใช้ โดยมิให้เกิดความขัดแย้ง
3. การควบคุมข้อมูล (Data control) กำหนดและควบคุมการใช้ข้อมูลให้เป็นไปตามเอกสิทธิ์ (Privileges) เพื่อรักษาความปลอดภัย (Security) และบูรณภาพของข้อมูล (Data Integrity)และป้องกันมิให้เกิดความเสียหายต่อข้อมูล รวมถึงการสำรองข้อมูล และสามารถกู้กลับ (Recovery) เมื่อระบบเกิดปัญหาด้วย

บูรณภาพของข้อมูล (Data integrity) (Fleming, Halle, 1989) หมายถึงกฎธุรกิจ (Business rules) ที่เป็นข้อกำหนดเกี่ยวกับค่าของข้อมูล หรือความสัมพันธ์ของข้อมูลควรเป็นอย่างไร เป็นเพียงข้อกำหนดทางธุรกิจซึ่งไม่ขึ้นอยู่กับเทคนิคทางคอมพิวเตอร์ บูรณภาพของข้อมูลสามารถแบ่งเป็น 3 อย่างคือ

3.1 บูรณภาพเอนทิตี (Entity integrity) กฎนี้กล่าวไว้ว่า สดมภ์หรือแอตตริบิวที่เป็นส่วนหนึ่งของคีย์หลัก (Primary key - คีย์หลักอาจประกอบด้วยหนึ่งหรือหลายแอตตริบิว) จะมีค่าว่าง (Null value) หรือค่าที่ไม่รู้จักไม่ได้ ทั้งนี้เพราะคีย์หลักนี้จะเป็นตัวชี้เฉพาะข้อมูลแต่ละแถวหรือระเบียน จึงไม่ควรมีค่าว่างหรือค่าที่ไม่รู้จัก

3.2 บูรณภาพอ้างอิง (Referential integrity) ข้อกำหนดเกี่ยวกับฟอเรนคีย์ (Foreign key) ฟอเรนคีย์หมายถึง แอตตริบิวหรือกลุ่มของแอตตริบิวของตารางหนึ่ง มีค่าตรงกับคีย์หลักของ

อีกตารางหนึ่ง เนื่องจากบูรณภาพอ้างอิงเป็นการเชื่อมโยงกันระหว่าง 2 ตาราง เพื่อใช้อ้างอิงข้อมูลที่มีความสัมพันธ์กัน ดังนั้นบูรณภาพอ้างอิงจึงกล่าวไว้ว่า ตารางใดที่มีฟอเรนคีย์ ค่าทุกค่าของฟอเรนคีย์ไม่ว่าจะเป็นค่าว่างหรือไม่จะต้องมีค่าตรงกับคีย์หลักของอีกตารางหนึ่ง การเพิ่ม การปรับปรุง หรือลบข้อมูลของทั้ง 2 ตารางย่อมมีผลกระทบต่อกัน แต่จะมีผลกระทบอย่างไรขึ้นอยู่กับข้อมูลทางธุรกิจของแต่ละฐานข้อมูล

3.3 บูรณภาพโดเมน (Domain integrity) โดเมนคือค่าที่เป็นไปได้ของแต่ละแอตตริบิว ไม่ว่าแอตตริบิวนั้นจะเป็นคีย์หลัก ฟอเรนคีย์ หรือแอตตริบิวอื่นที่ไม่ใช่คีย์ในตาราง ผู้ใช้สามารถกำหนดค่าที่เป็นไปได้ของแต่ละแอตตริบิวไว้ เพื่อใช้ตรวจสอบข้อมูลที่จะบรรจุในแต่ละแอตตริบิว

4. รวบรวมสถิติการใช้งานของระบบ เพื่อให้สามารถควบคุมการทำงานของระบบ และปรับปรุงผลการดำเนินการ (Performance) ให้ดีขึ้นตามที่คาดหวังไว้

5. ใช้ภาษารุ่นที่ 4 (Fourth-generation languages) เช่น แบบฟอร์ม (Forms) โปรแกรมก่อกำเนิดข้อมูลนำเข้า (Input-program generators) โปรแกรมก่อกำเนิดรายงาน (Report-program generators) เป็นต้น เพื่อเพิ่มความสามารถในการประยุกต์ใช้งาน

6. มีพจนานุกรมข้อมูล (Data dictionary) เพื่อใช้ช่วยพัฒนาระบบฐานข้อมูล

7. สามารถเชื่อมโยงกับระบบประมวลผลคำ (Word processing) แผ่นตารางทำการ (Spreadsheet) และระบบงานอื่น ๆ

หน้าที่ของระบบจัดการฐานข้อมูล 3 ประการหลังนี้ เป็นส่วนที่ระบบในปัจจุบันได้พยายามจัดเตรียมให้มีขึ้น เพื่อให้ระบบสามารถเชื่อมโยงกับระบบงานอื่น ๆ และสามารถทำงานได้สมบูรณ์มากยิ่งขึ้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

การเลือกระบบฐานข้อมูล

แต่เดิมการพัฒนาระบบงานต่าง ๆ จะใช้ภาษายุคที่สามและเป็นการประมวลผลของแฟ้มสัญนิยม (Conventional file) การเขียนโปรแกรมประยุกต์จะยุ่งยาก ใช้เวลานาน และอิงกับโครงสร้างของข้อมูล เมื่อมีการเปลี่ยนแปลงข้อมูล หรือต้องการเพิ่มข้อมูลใหม่ ๆ เข้าไปในระบบจะต้องแก้ไขปรับปรุงโปรแกรม หรือเขียนโปรแกรมใหม่ พร้อมทั้งทดสอบโปรแกรมนั้น ๆ ซึ่งเป็นเรื่องที่ต้องเสียค่าใช้จ่าย และเสียเวลามาก ภาษาที่ใช้มักจะเป็นภาษาที่ใช้เฉพาะแต่ละระบบ ถ้าเปลี่ยนระบบจะต้องเรียนรู้ภาษาใหม่ การเก็บข้อมูลก็ซ้ำซ้อน ทำให้ต้องใช้เนื้อที่ในการเก็บข้อมูลมากกว่าที่ควร ไม่มีระบบรักษาความปลอดภัยที่เหมาะสม และที่สำคัญทำให้ยากต่อการบำรุง

รักษาให้ข้อมูลทั้งหมดมีความสอดคล้องกัน ถูกต้องตรงกัน ประกอบกับในปัจจุบันธุรกิจมีการแข่งขันสูงขึ้น ผู้บริหารแต่ละบริษัทมักมีคำถามหรือความต้องการใช้ข้อมูลในลักษณะต่าง ๆ ที่เปลี่ยนแปลงอยู่เสมอ และแต่ละครั้งมักจะต้องการคำตอบในเวลารวดเร็วหรือทันที เพื่อให้สามารถตัดสินใจได้ฉับไวด้วยข้อมูลที่ถูกต้องรวดเร็วทันต่อเหตุการณ์เสมอ จากปัญหาและลักษณะการใช้ข้อมูลที่เปลี่ยนไป ระบบฐานข้อมูลซึ่งเป็นระบบที่ได้รับการออกแบบมาเพื่อแก้ไขปัญหาดังกล่าว พร้อมทั้งการเพิ่มคุณลักษณะของการใช้ภาษาข้อคำถาม (Query language) ที่เรียนรู้ง่าย จึงได้รับความสนใจมากขึ้น โดยเฉพาะระบบฐานข้อมูลแบบสัมพันธ์ซึ่งมีโครงสร้างที่เข้าใจง่าย ไม่ยุ่งยากซับซ้อน สามารถทำงานแบบข้อคำถาม หรือคิวรี (Query) ได้ดี บริษัทผู้ผลิตระบบฐานข้อมูลจึงให้ความสนใจฐานข้อมูลแบบสัมพันธ์ และพยายามปรับปรุงให้มีประสิทธิภาพมากขึ้น ปัจจุบันจึงมีระบบฐานข้อมูลแบบสัมพันธ์จำหน่ายในท้องตลาดหลายระบบ

จากคุณสมบัติที่ดีของระบบฐานข้อมูลแบบสัมพันธ์ ในการพัฒนาระบบจัดการสารสนเทศสำหรับงานวิจัยของภาควิชาวิศวกรรมคอมพิวเตอร์จึงเลือกใช้ระบบฐานข้อมูลแบบสัมพันธ์ โดยเลือกใช้ระบบฐานข้อมูลชื่อ ไมโครซอฟต์แอคเซส (Microsoft Access - MSACCESS) ซึ่งพัฒนาภายใต้ระบบปฏิบัติการวินโดว์ (MS Windows) ผลิตโดยบริษัทไมโครซอฟต์คอร์ปอเรชัน (Microsoft corporation)

ทั้งนี้เพราะแอคเซสเป็นระบบฐานข้อมูลแบบสัมพันธ์ที่ประกอบด้วยระบบจัดการฐานข้อมูลแบบสัมพันธ์ (Relational database management system หรือ RDBMS) ที่มีคุณสมบัติสำคัญ 3 ประการ (Viescas, 1994)คือ

1. การกำหนดรายละเอียดของข้อมูล (Data definition) คือการกำหนดข้อมูลอะไรบ้างที่ควรจะเก็บในฐานข้อมูล กำหนดประเภทข้อมูล (ตัวเลข อักขระ ฯ) รูปแบบควรเป็นอย่างไร มีความสัมพันธ์แบบไหน จะตรวจสอบข้อมูลนั้น ๆ อย่างไร

2. การจัดการข้อมูล (Data manipulation) สามารถเลือกข้อมูลเฉพาะที่สนใจ เรียงลำดับข้อมูลตามต้องการ หรือเปลี่ยนแปลงข้อมูล

3. การควบคุมข้อมูล (Data control) สามารถกำหนดสิทธิ์ของผู้ใช้ระบบในลักษณะต่าง ๆ เพื่อให้ผู้มีสิทธิ์เท่านั้นที่จะใช้ข้อมูลได้ในขอบเขตได้รับอนุญาต เช่น อ่านได้อย่างเดียว แก้ไขข้อมูลไม่ได้ เป็นต้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

แอคเซสนั้นนอกจากจะเป็ นระบบฐานข้อมูลที่มีคุณสมบัติตามข้อกำหนดที่ดีของระบบฐานข้อมูลแบบสัมพันธ์แล้ว แอคเซสสามารถเชื่อมโยงการทำงานกับโปรมแกรมสำเร็จรูปอื่น ๆ ที่มีประสิทธิภาพ และได้รับความนิยมอยู่ในขณะนี้ เช่น โปรแกรมประมวลผลคำชื่อ Microsoft Word โปรแกรมไมโครซอฟต์เอคเซล (Microsoft Excel) เป็นต้น ซึ่งผลิตโดยบริษัทเดียวกัน นอกจากนี้แอคเซสสามารถทำงานร่วมกับระบบฐานข้อมูลชนิดอื่นที่มีจำหน่ายในท้องตลาดและมีผู้นิยมใช้ เช่น dBase Paradox FoxPro และ Betrieve รวมทั้งภาษาเอสคิวแอล (SQL-Structured query language) เป็นต้น และที่สำคัญแอคเซสสามารถเก็บข้อมูลที่มีเขตข้อมูลขนาดใหญ่ถึง 64 กิโลไบต์ เรียกเขตข้อมูลประเภทนี้ว่า Memo และสามารถสืบค้นข้อมูลได้ด้วยการใช้วายคาร์ด (Wildcard) จึงเหมาะสมที่จะใช้เก็บบทคัดย่อของงานวิจัย

ดังได้กล่าวแล้วข้างต้นว่าไมโครซอฟต์แอคเซสเป็นระบบที่สร้างขึ้นสำหรับการประยุกต์ใช้งานภายใต้ไมโครซอฟต์วินโดว์โดยเฉพาะ ดังนั้นแอคเซสจึงสามารถใช้สิ่งอำนวยความสะดวกของวินโดว์ได้แก่ การแลกเปลี่ยนข้อมูลด้วย Dynamic Data Exchange (DDE) และการเชื่อมโยงข้อมูลด้วย Object Linking and Embedding (OLE) ทั้ง DDE และ OLE ทำให้แอคเซสทำงานและใช้ข้อมูลร่วมกับโปรแกรมประยุกต์อื่น ๆ ภายใต้วินโดว์ที่มีคุณสมบัติทั้ง 2 นี้ได้เป็นอย่างดี ซึ่งทำให้เกิดการใช้ข้อมูลร่วมกันอย่างมีประสิทธิภาพ ลดความซ้ำซ้อนของข้อมูล และทำให้สามารถทำงานหลายระบบได้พร้อม ๆ กัน

สถาปัตยกรรมของแอคเซส

ลักษณะฐานข้อมูลของแอคเซสจะเก็บข้อมูลทุกอย่างไว้ในแฟ้มหนึ่งแฟ้มซึ่งมีนามสกุลเป็น MDB แฟ้มข้อมูลนี้จะประกอบด้วยสิ่งต่าง ๆ ที่เรียกว่า ออบเจค (Object) ซึ่งมีอยู่ 6 ชนิดคือ ตาราง (Tables) ข้อคำถามหรือคิวรี (Queries) ฟอร์ม (Forms) รายงาน (Reports) แมโคร (Macros) และมอดูล (Modules) โดยปกติฐานข้อมูลทั่ว ๆ ไปจะหมายถึงแฟ้มข้อมูลที่เก็บข้อมูลในระบบทั้งหมด สำหรับแอคเซสคำว่าฐานข้อมูลจะหมายถึงออบเจคทุกชนิดที่สัมพันธ์กับข้อมูล ฐานข้อมูลของแอคเซสจึงหมายถึงออบเจคทั้ง 6 ชนิด (Viescas, 1994) ซึ่งมีรายละเอียดดังนี้

1. ตาราง (Tables) คือออบเจคที่เก็บข้อมูล และในฐานข้อมูลหนึ่งจะประกอบด้วยตารางหลายตาราง โดยแต่ละตารางจะเก็บข้อมูลที่มีความสัมพันธ์ในเรื่องเดียวกัน เช่น ตารางนิสิต

ข้อมูลแต่ละระเบียน (Record) หรือข้อมูล 1 คน จะประกอบด้วยเขตข้อมูลชนิดต่าง ๆ (Data type) ซึ่งอาจเป็นตัวเลข อักขระ หรือแม้แต่ออบเจคแบบ OLE (Object linking and embedding) ที่ใช้เก็บข้อมูลที่เป็นรูปภาพ หรือกราฟ ซึ่งอาจเป็นข้อมูลที่เชื่อมโยงมาจากโปรแกรมประยุกต์อื่นภายใต้วินโดว์ หรือเป็นการฝังตัวอยู่ในแอคเซสก็ได้ สำหรับเขตข้อมูลแต่ละเขตสามารถกำหนดวิธีหรือกฎเกณฑ์ต่าง ๆ เพื่อตรวจสอบความถูกต้องของข้อมูล (Validation rule) หรือค่าที่เป็นไปได้ของข้อมูลแต่ละเขต

ตารางแต่ละตารางจะต้องมีคีย์หลัก (Primary key) เพื่อใช้กำหนดรายละเอียดของข้อมูลแต่ละระเบียน และใช้เป็นข้อมูลในการค้นคืนซึ่งจะได้ข้อมูลที่เป็นหนึ่งเดียว (Unique) ไม่ซ้ำ คือชี้เฉพาะรายการนั้น เช่น รหัสนิสิต สามารถใช้เป็นคีย์หลัก เพราะรหัสนิสิตจะไม่ซ้ำกัน อย่างไรก็ตาม ผู้ใช้ยังสามารถกำหนดเขตข้อมูลอื่น ๆ (ยกเว้นข้อมูลบางประเภท เช่น OLE, Memo) เป็นดรรชนี เพื่อประโยชน์ในการเข้าถึง หรือค้นคืนข้อมูลได้รวดเร็วขึ้น ผู้ใช้สามารถสร้างความสัมพันธ์ระหว่างตารางเพื่อเชื่อมโยงข้อมูลได้ ความสัมพันธ์ในฐานข้อมูลมีได้ 2 แบบคือ

1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-one, 1:1 relationship)
2. ความสัมพันธ์แบบหนึ่งต่อหลาย (One-to-many, 1:N relationship)

ลักษณะของความสัมพันธ์ทั้งสองแบบนี้จะอธิบายในบทต่อไป

นอกจากนี้ผู้ใช้อาจสร้างความสัมพันธ์ระหว่างตารางและกำหนดให้แอคเซสตรวจสอบความสัมพันธ์ของข้อมูล 2 ตารางนั้นให้อัตโนมัติด้วย เพื่อให้เกิดบูรณภาพอ้างอิง (Referential integrity) ของข้อมูล กล่าวคือจะเพิ่มข้อมูลในตารางรองได้ต่อเมื่อได้มีข้อมูลที่สัมพันธ์กันในตารางหลักแล้วเท่านั้น หรือจะลบข้อมูลในตารางหลักก่อนลบข้อมูลในตารางรองไม่ได้ ดังเช่น ในระบบฐานข้อมูลวิทยานิพนธ์ ความสัมพันธ์ระหว่างตารางนิสิตที่ลงทะเบียนเรียนวิทยานิพนธ์(ตารางหลัก) และตารางวิทยานิพนธ์ (Thesis) จะต้องมีชื่อนิสิตในตารางหลักก่อนจึงสามารถเพิ่มหัวข้อวิทยานิพนธ์ของนิสิตนั้นในตารางวิทยานิพนธ์ได้ และในทางกลับกันจะลบชื่อนิสิตนั้นจากตารางหลักได้เมื่อได้ลบระเบียนหัวข้อวิทยานิพนธ์ในตารางวิทยานิพนธ์แล้ว เป็นต้น แอคเซสจะดูแลข้อมูลให้เป็นไปตามกฎบูรณภาพอ้างอิงนี้ เมื่อผู้ใช้ทำงานผิดขั้นตอน ระบบจะมีข้อความเตือนให้ทราบอัตโนมัติ

2. ข้อคำถาม หรือคิวรี (Queries) คือออบเจคที่เก็บข้อมูลที่ได้จากการกำหนดความต้องการการใช้ข้อมูลซึ่งอาจเป็นข้อมูลที่มาจากหนึ่งหรือหลายตาราง หรือจากคิวรี ด้วยเงื่อนไขต่าง ๆ โดยใช้ภาษาข้อคำถามหรือ เอสคิวแอล (Structured query language - SQL) หรือใช้เครื่องมือที่เรียกว่า Graphical query by example (GQBE) เพื่อเลือกข้อมูล โดยการกำหนดเงื่อนไขของข้อมูลที่ต้องการ ซึ่งผู้ใช้สามารถเรียนรู้ได้ง่ายและรวดเร็ว แอคเซสจะนำข้อมูลจาก GQBE มาเขียนเป็น SQL ให้โดยอัตโนมัติ ผู้ใช้ระบบจึงไม่จำเป็นต้องเรียนรู้ SQL

คิวรีมีหลายประเภทตามวัตถุประสงค์ต่าง ๆ กัน คือ

2.1 คิวรีสำหรับเลือกข้อมูล เรียกว่า Select query ใช้ประโยชน์ในการแทรกข้อมูล (Insert) ปรับปรุง (Update) หรือลบข้อมูล (Delete) จากตาราง ซึ่งจะทำได้ทีละ 1 ระเบียน โดยใช้ร่วมกับฟอร์ม (ออบเจคหนึ่งซึ่งจะอธิบายต่อไป) เพื่อแสดงข้อมูลบนจอภาพ หรือใช้ Select query เลือกข้อมูลเพื่อการวิเคราะห์ ซึ่งสามารถแสดงได้ทั้งทางจอภาพ และรายงาน (ออบเจคหนึ่งซึ่งจะอธิบายต่อไป)

2.2 สร้างคิวรีจากผลรวมของข้อมูลใน 2 เงื่อนไข เปรียบเทียบกัน เรียกว่า Crosstab query เช่น ผลรวมของการขายสินค้าแต่ละประเภทในแต่ละไตรมาส เป็นต้น

2.3 คิวรีเพื่อสร้างตาราง เรียกว่า Make table query เป็นคิวรีที่เลือกข้อมูลและนำข้อมูลที่ได้สร้างเป็นตารางใหม่

2.4 คิวรีเพื่อการปรับปรุง เรียกว่า Update query สร้างคิวรีนี้ขึ้นเพื่อนำข้อมูลนั้นไปปรับปรุงข้อมูลบางเขตข้อมูล (Field) หรือทุกเขตข้อมูล (All Fields) ในตาราง

2.5 คิวรีเพื่อเพิ่มข้อมูล เรียกว่า Append query สร้างคิวรีนี้ขึ้นเพื่อนำข้อมูลที่ได้ไปเพิ่ม (Add) ต่อท้ายตารางที่ต้องการ

2.6 คิวรีเพื่อลบข้อมูล เรียกว่า Delete query สร้างคิวรีนี้ขึ้นเพื่อเลือกข้อมูลที่ต้องการลบออกจากตาราง

คิวรี 2.3 - 2.6 นั้นเรียกว่า คิวรีกระทำการ หรือ Action query เป็นคิวรีที่ใช้ประโยชน์ในการสร้างตาราง หรือทำการปรับปรุง (Update) เพิ่ม (Append) หรือลบ (Delete) ข้อมูลเป็นชุดหรือทีละหลายระเบียน การสร้างคิวรีทุกชนิดสามารถกำหนดเงื่อนไข (Criteria) เพื่อเลือกเฉพาะข้อมูลที่ต้องการ และสามารถกำหนดการเรียงลำดับของข้อมูล (Sorting) ได้ด้วย

3. ฟอร์ม (Forms) คือออบเจคที่ใช้แสดงข้อมูล หรือกลุ่มข้อมูลจากตารางหรือคิวรี โดยแสดงผลบนจอภาพ หรือสร้างแบบฟอร์มให้มีรูปแบบชัดเจน เข้าใจง่ายและอ้างถึงตารางหรือคิวรีที่

ศูนย์วิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

ต้องการบันทึก หรือปรับปรุงข้อมูลในตารางหรือคิวรีนั้น และกำหนดกฎเกณฑ์สำหรับตรวจสอบข้อมูลแต่ละตัวเพื่อให้ได้ข้อมูลที่ถูกต้องก่อนที่จะบันทึก หรือปรับปรุงในฐานข้อมูล หรืออาจสร้างฟอร์ม (โดยไม่ใช้ตารางหรือคิวรี) เพื่อให้ผู้ใช้ระบบใส่เงื่อนไขการเลือกข้อมูล ซึ่งมีการเปลี่ยนแปลงไปตามความต้องการ เช่น การดูยอดขายตามวันที่กำหนด เป็นต้น

นอกจากใช้ฟอร์มแสดงผลบนจอภาพแล้วยังสามารถสร้างฟอร์ม และพิมพ์เป็นแบบฟอร์มมาตรฐานของบริษัทได้ เช่น แบบฟอร์มใบเสร็จรับเงินของบริษัท เป็นต้น

4. รายงาน (Reports) เป็นออบเจคที่ใช้แสดงข้อมูลเป็นกลุ่มจากตาราง หรือคิวรี โดยสามารถคำนวณยอดรวมแยกตามกลุ่ม แสดงค่าเฉลี่ยของข้อมูลแต่ละกลุ่ม รายงานสามารถแสดงรายละเอียดของข้อมูล และยอดรวมเป็นระดับได้มากกว่าการใช้ฟอร์ม

5. แมโคร (Macros) เป็นออบเจคที่ใช้เก็บการทำงานต่าง ๆ ไว้เป็นชุดของงาน งาน (Task) แต่ละงานนี้แอคเซสเรียกว่า แอคชัน (Action) แอคเซสจะทำงานต่าง ๆ ตามลำดับที่กำหนดไว้ในแมโครโดยอัตโนมัติ เมื่อเกิดเหตุการณ์ (Event) นั้น ๆ ขึ้น หรือเงื่อนไขที่กำหนดเป็นจริง งานต่าง ๆ ที่กำหนดไว้ในแมโครจะเป็นงานง่าย ๆ ไม่ซับซ้อน และใช้ร่วมกับฟอร์ม หรือรายงาน โดยระบุไว้ที่พรอบเพอตี (Properties) ของฟอร์มหรือรายงาน

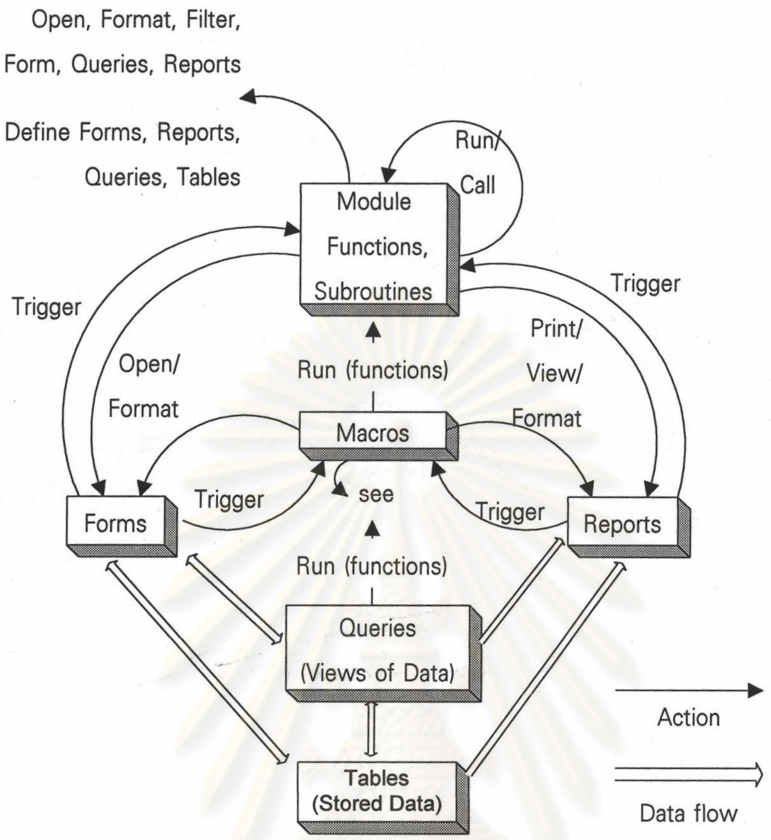
เหตุการณ์ต่าง ๆ นั้นคือ การเปลี่ยนแปลงสภาวะ (State) ของไมโครซอฟต์ออบเจค เช่น เมื่อมีการเรียกใช้ฟอร์ม (Open form) หรือเมื่อมีการเปลี่ยนแปลงข้อมูลแต่ละคอนโทล (Control คือออบเจคที่เก็บข้อมูลบนฟอร์มหรือรายงาน หรือ Data field นั่นเอง) เป็นต้น

6. มอดูล (Modules) คือออบเจคที่เก็บขั้นตอนการดำเนินงานต่าง ๆ (Procedure) โดยใช้สำหรับงานที่มีขั้นตอนหรือเงื่อนไขซับซ้อนและไม่สามารถใช้แมโครได้ มอดูลนี้จะเขียนด้วยภาษาแอคเซสเบสิค (Access basic) สามารถเรียกใช้มอดูลได้จากแมโคร หรือเมื่อมีเหตุการณ์ หรือมีเงื่อนไขตามที่กำหนด หรือเรียกใช้มอดูลจากฟอร์มและรายงานโดยกำหนดไว้ที่พรอบเพอตีของฟอร์มและรายงานเช่นกัน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 3.5 เป็นแนวคิดที่แสดงความสัมพันธ์ระหว่างออบเจคทั้ง 6 ชนิดของแอคเซส โดยแสดงถึงการไหลของข้อมูล (Data flow) และการเกิดทริกเกอร์ (Trigger) เมื่อมีเหตุการณ์ (Event) ต่าง ๆ เกิดขึ้น

ระบบฐานข้อมูลเมื่อได้นำไปใช้งานแล้ว เพื่อให้ระบบสามารถทำงานได้อย่างมีประสิทธิภาพ และเพื่อป้องกันการสูญเสียหรือสูญหายของข้อมูล อาจด้วยวิธีข้อมูลถูกทำลาย ถูกขโมย โดย



รูปที่ 3.5 แสดงความสัมพันธ์ระหว่างออบเจคทั้ง 6 ชนิดของแอคเซส (Access) (Viescas, 1994)

บุคคลที่ไม่ได้รับอนุญาต หรือเกิดจากความผิดพลาดของระบบฮาร์ดแวร์ จึงควรดูแลรักษาระบบด้วยการสำรองข้อมูลเป็นประจำตามความเหมาะสม เช่น สำรองข้อมูลทุกวัน ทุกสัปดาห์ หรือทุกสิ้นเดือน ขึ้นอยู่กับการใช้งานระบบนั้น ๆ เป็นต้น เมื่อมีการสูญเสีย หรือสูญหายของข้อมูลก็สามารถนำข้อมูลสำรองชุดล่าสุดมาใช้งาน เพื่อเป็นการป้องกันการสูญเสียหรือสูญหายของข้อมูลจากบุคคลที่ไม่ได้รับอนุญาต และเพื่อให้การทำงานของระบบฐานข้อมูลแอคเซสเป็นไปด้วยความเรียบร้อย แอคเซสจึงเตรียมโปรแกรมอรรถประโยชน์ต่าง ๆ ไว้ดังนี้

1. ระบบรักษาความปลอดภัย (Data security)
2. การเข้ารหัสฐานข้อมูล (Encrypting database)
3. การอัดแน่นฐานข้อมูล (Compacting the database)
4. การซ่อมแซมฐานข้อมูล (Repairing the database)

ระบบการรักษาความปลอดภัย (Data security)

ระบบการรักษาความปลอดภัยของแอคเซสมีวิธีป้องกันข้อมูล 4 ระดับ (Perschke and Liczbanski, 1993) คือ

1. ผู้ใช้ระบบ (Users) สามารถกำหนดได้ว่าใครบ้างที่มีสิทธิในการใช้ฐานข้อมูลนั้น
2. กลุ่ม (Groups) ผู้ใช้ระบบจะถูกแบ่งเป็นกลุ่ม ผู้ออกแบบสามารถกำหนดความสำคัญของแต่ละกลุ่มแตกต่างกันไป เพราะแต่ละกลุ่มจะมีความรับผิดชอบต่างกัน แอคเซสแบ่งผู้ใช้ออกเป็น 3 กลุ่มคือ

   2.1 กลุ่มผู้ใช้ (User) เป็นผู้ใช้ข้อมูลในระบบ จะต้องกำหนดบัญชี (Account) ของผู้ใช้แต่ละคนในกลุ่มนี้

   2.2 กลุ่มผู้ดูแลระบบ (Admins) คือกลุ่มผู้ดูแลรักษาระบบฐานข้อมูล

   2.3 กลุ่มผู้ใช้ชั่วคราว (Guest) คือผู้ใช้ข้อมูลที่ไม่ต้องการกำหนดบัญชีให้ หรือเป็นผู้ใช้ที่ใช้ข้อมูลเป็นครั้งคราว
3. รหัสผ่าน (Password) รหัสผ่านจะเป็นรหัสลับของแต่ละคนที่ได้รับอนุญาต ผู้ได้รับอนุญาตให้ใช้ระบบต้องใส่รหัสผ่านให้ถูกต้องจึงจะเข้าไปใช้ระบบได้
4. การอนุญาต (Permissions) เป็นการอนุญาตให้แต่ละคนสามารถปฏิบัติงานกับออบเจคอะไรบ้าง และมีสิทธิมากน้อยอย่างไร ผู้ดูแลระบบจะเป็นผู้กำหนดสิทธิต่าง ๆ ดังนี้

   4.1 อ่านข้อกำหนดได้ (Read definition)

   4.2 ปรับปรุงข้อกำหนด (Modify definition)

   4.3 ประมวลผลได้ (Execute)

   4.4 อ่านข้อมูลได้ (Read data)

   4.5 ปรับปรุงข้อมูล (Modify data)

   4.6 มีสิทธิ์ทำทุกอย่างตั้งแต่ข้อ 4.1 - 4.5

การเข้ารหัสฐานข้อมูล (Encrypting the database) (Perschke and Liczbanski)

นอกจากระบบรักษาความปลอดภัยแล้ว แอคเซสยังเตรียมขั้นตอนของการเข้า/ถอดรหัสลับฐานข้อมูล (Encrypting / Decrypting database) เพื่อแปลงข้อมูลให้อยู่ในรูปแบบที่อ่านไม่รู้เรื่อง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

โดยโปรแกรมประยุกต์อื่นที่ไม่ใช่แอคเซส ซึ่งเป็นการป้องกันการใช้ข้อมูลโดยมิได้รับอนุญาตอีกชั้นหนึ่ง

การอัดแน่นฐานข้อมูล (Compacting the database) (Perschke and Liczbanski)

แอคเซสจะเก็บข้อมูลชนิดข้อความ (Text) ตามเนื้อหาของข้อมูล ข้อมูลชนิดข้อความจึงมีความยาวแตกต่างกัน (Variable length) เมื่อมีการบรรณาธิกรข้อความจะทำให้การเก็บข้อมูลทางกายภาพแบ่งเป็นส่วน ๆ (Fragment) เพื่อให้การเก็บข้อมูลมีประสิทธิภาพมากขึ้น และทำให้การทำงานรวดเร็วขึ้น แอคเซสได้เตรียมโปรแกรมอัดแน่นไว้ให้ โปรแกรมอัดแน่นจะรวมส่วนต่าง ๆ ของเขตข้อมูลเข้าด้วยกัน และทำการลบข้อมูลซึ่งโปรแกรมประยุกต์ได้สั่งลบไปแล้ว แต่ทางกายภาพนั้นข้อมูลเหล่านี้ยังคงจองเนื้อที่ไว้ทำให้โปรแกรมประยุกต์ไม่สามารถใช้เนื้อที่เหล่านี้ได้ ข้อมูลเหล่านี้ได้แก่

1. แฟ้มชั่วคราว (Temporary files)
2. ระเบียนที่ลบแล้ว (Deleted records)
3. ลบค่าว่าง (Null values) และที่ว่าง (Blanks) ในระเบียน
4. ออบเจคที่ลบแล้ว (Deleted objects)

การอัดแน่นจะเป็นการลบข้อมูลทางกายภาพ ซึ่งจะทำให้เนื้อที่เหล่านี้สามารถนำมาใช้ใหม่ได้

การซ่อมแซมฐานข้อมูล (Repairing the database) (Perschke and Liczbanski)

แอคเซสจะตรวจสอบความวิบัติของฐานข้อมูล (Corrupted database) ทุกครั้งเมื่อเริ่มใช้ฐานข้อมูล และแอคเซสจะใช้โปรแกรมอรรถประโยชน์ช่วยซ่อมแซมฐานข้อมูลให้กลับสู่สภาพที่ใช้งานต่อได้ อย่างไรก็ตามการซ่อมแซมนี้จะสามารถทำได้เฉพาะบางกรณีเท่านั้น เช่น เกิดไฟดับขณะใช้งาน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย