

บทที่ 2

หลักการและทฤษฎี

ในบทนี้จะกล่าวถึงหลักการ และทฤษฎีที่ใช้ในการสร้าง Web Page ระบบการจัดการการเรียนการสอนในคณะวิศวกรรมศาสตร์ ด้วย Active Server Page (ASP) Script ซึ่งเป็นอีกวิธีหนึ่งในการสร้าง Web Page แบบสามารถตอบโต้กับผู้ใช้ได้ (Inter Active) วิธีการติดต่อกับฐานข้อมูลด้วยภาษา SQL โดยใช้ฐานข้อมูลของ Microsoft Access การสร้างฐานข้อมูล และเครื่องมือในการสร้าง Web Page ดังนี้

2.1 Active Server Page (ASP) เบื้องต้น [1], [2], [3]

Active Server Pages เป็นเอกสาร HTML ที่ได้เพิ่มความสามารถบางอย่างเข้าไป ดังนั้น Active Server Pages จึงสามารถทำงานหลายๆ สิ่งที่ HTML ทำได้เช่น Active Server Pages สามารถบรรจุ HTML tags ซึ่งจะถูกแปล (Interpret) และแสดงผลโดย Browser และ อีกหลายๆ สิ่งที่จะใช้บรรจุให้กับหน้าเอกสาร HTML ธรรมดาเช่น Java applets, ตัวอักษรกระพริบ, Script ที่ใช้ประมวลผลทาง Client (Browser), ActiveX control ทาง Client เหล่านี้สามารถนำมาไว้ที่ Active Server Pages ได้ แต่ Active Server Pages ยังมีคุณลักษณะ (features) ที่สำคัญ 4 อย่าง คือ

1. Active Server Pages สามารถบรรจุ Script ที่ใช้ประมวลผลทาง Server ได้ การทำเช่นนี้ได้ทำให้เกิดประโยชน์มาก เพราะจะทำให้เราสามารถสร้างหน้าเอกสารที่เป็น dynamic ได้ ตัวอย่างง่ายๆ ที่แสดงถึงประโยชน์ของคุณลักษณะนี้คือเราสามารถที่จะสร้างเอกสาร (page) ที่แสดงข้อความทักทายที่แตกต่างกันไป ในแต่ละเวลาของวันได้
2. Active Server Pages ได้เตรียม built-in object มากมาย การที่มี built-in object ใน Active Server Pages ช่วยให้ Script ของเรามีประสิทธิภาพในการใช้งานมากขึ้น นั่นเพราะ Object ต่างๆ เหล่านี้จะทำให้เราสามารถรับ-ส่ง ข้อมูล (data) ระหว่าง Server กับ Client (Browser) ได้ ตัวอย่างเช่น การใช้ Object "Request" เราสามารถรับข้อมูลจากผู้ใช้ที่ส่งมาทางฟอร์ม (Form) ของ HTML และส่งข้อมูลนั้นต่อไปให้กับส่วนของ Script ที่ต้องการได้อย่างง่ายดาย
3. Active Server Pages สามารถเพิ่ม Component ที่ต้องการได้ ไม่เฉพาะ Component มาตรฐานที่ Active Server Pages ได้เตรียมไว้ตอนที่ Install เท่านั้น Active Server Pages ยังสามารถทำการเพิ่ม Component ที่ต้องการเข้าไปได้อีก
4. Active Server Pages สามารถทำการติดต่อกับฐานข้อมูล (Database) ดังเช่น Microsoft SQL Server หรือ Microsoft Access ได้เป็นอย่างดี โดยการใช้ชุดของ Object พิเศษ (Object เหล่านี้มี

เป็นมาตรฐานอยู่แล้วใน Active Server Pages) ที่เรียกว่า ActiveX Data Object (ADO) คุณลักษณะในข้อนี้ทำให้ Active Server Pages มีประสิทธิภาพมากที่สุดในการที่จะนำไปใช้งาน ดังนั้นด้วยคุณลักษณะทั้ง 4 ข้อที่กล่าวมา ทำให้กล่าวได้ว่า Active Server Pages นั้นคือหน้าเอกสาร HTML (pages) มาตรฐานที่ได้เพิ่มการทำงานของ Script ที่ประมวลผลทาง Server โดยมี Object และ Component เพิ่มเข้ามาช่วยในการทำงาน ทำให้สามารถสร้าง Web Site ที่มี หน้าเอกสาร (pages) แบบ Dynamic ได้

ในการใช้งาน Active Server Pages นี้เราสามารถเข้ากับ Web Server ที่มาจาก Microsoft ดังเช่น Internet Information Server (IIS) สำหรับ Window NT Server, Microsoft's Personal Web Server สำหรับ Window 95/98, Peer Web Server สำหรับ WindowNT Workstation หรือ ของบริษัทอื่นๆ ที่ไม่ใช่ Microsoft เช่น Chili!ASP ของ Chili!Soft, FastTrack servers, LotusDomino, Go Servers และอื่นๆอีกมากมาย (ข้อมูลเพิ่มเติมสามารถหาได้จาก Web Site ของ Chili!Soft)

2.1.1 ประโยชน์ของ Active Server Pages

Active Server Pages สามารถทำประโยชน์ได้อย่างไม่มีขีดจำกัด รายการด้านล่างนี้ จะแสดงถึงตัวอย่างบางส่วนของ Active Server Pages สามารถทำได้

1. รับข้อมูลจากฟอร์ม HTML และเก็บข้อมูลนั้นที่ฐานข้อมูล (Database)
2. สร้างหน้าเอกสาร (Pages) ส่วนบุคคล ที่จะแสดงรายละเอียดแตกต่างกันเป็นส่วนเฉพาะ ของแต่ละบุคคล
3. แสดงหน้าเอกสาร (Pages) ที่แตกต่างกันสำหรับความสามารถที่มีของแต่ละ Browser ของผู้ใช้
4. ทำตัวเชื่อมโยง (Link) ร่วมกันของหลายๆ หน้าเอกสาร ทำให้เกิดความง่ายในการค้นหา
5. ทำตัวนับ (Counter) ให้กับหน้าเอกสาร (Pages) มากกว่าหนึ่งหน้า
6. เก็บข้อมูลเกี่ยวกับกิจกรรมที่ผู้ใช้ทำต่อ Web Site ลงบน log ไฟล์ได้

ดังนั้นเป็นเพียงตัวอย่างง่ายๆ ที่ Active Server Pages สามารถทำได้ อาจกล่าวได้ว่า สิ่งต่างๆ ที่ผู้พัฒนา Web Site ต้องการสามารถหาอุปกรณ์เหล่านั้นได้จาก Active Server Pages เลย์ที่เดียว

2.1.2 การทำงาน Active Server Pages

แนวทาง ที่ดีที่สุดที่จะเข้าใจได้ว่า Active Server Pages ทำงานอย่างไรนั้น คือ การพิจารณา Web Server ที่สนับสนุนและไม่สนับสนุน Active Server Pages เปรียบเทียบกัน ดังนั้นในที่นี้จึงขอใช้ Internet Information Server (IIS) (โปรแกรม Web Server ที่มาพร้อมกับ Window NT Server) เป็น Web

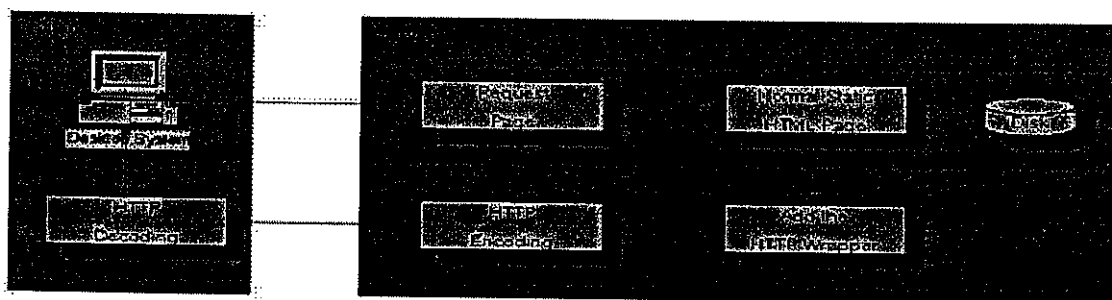
การอธิบาย จะเริ่มจากการทำงานโดยทั่วไปของ IIS ที่ทำงานกับ Static Page ไปจนถึงการทำงาน ของ IIS ที่สร้าง Dynamic Page ดังนี้

โดยทั่วไปของ IIS ก่อนที่จะมี Active Server Pages จะเป็นการให้บริการกับหน้าเอกสาร HTML ที่คงที่ (static HTML page) ดังนั้นเมื่อผู้ใช้งานร้องขอหน้าเอกสารที่ Web Site นั้นๆ IIS จะทำการดึงข้อมูลหน้าเอกสาร (page) ที่ต้องการจาก Disk หรือหน่วยความจำ (ซึ่งกล่าวได้ว่าเป็นลักษณะของ Static HTML) แล้วส่งไปให้ Browser ของผู้ใช้ (ผู้ร้องขอหน้าเอกสารนั้น) ดังนั้น IIS จึงมีหน้าที่หลักเพียง เป็น ตัวกลางระหว่าง Browser และไฟล์ที่อยู่ใน Disk บนเครื่อง Server เท่านั้น

2.1.3 สรุปขั้นตอนการทำงานของ IIS ต่อ Static HTML

1. ผู้ใช้ใส่ Internet Address ของไฟล์ HTML ที่ Address บาร์ของ Web Browser และกด Enter เพื่อ ร้องขอหน้าเอกสาร (page) ที่ต้องการ ตัวอย่างเช่น <http://www.payap.ac.th/hello.html>
2. ตัว Browser จะส่งคำร้องขอสำหรับหน้าเอกสารนั้นแก่ Web Server (ในที่นี้คือ IIS)
3. ตัว Web Server เมื่อได้รับการร้องขอหน้าเอกสาร และจดจำได้ว่าหน้าเอกสารที่ร้องขอเป็น ไฟล์ HTML เพราะว่าไฟล์ที่ร้องขอ มามีส่วนขยาย (นามสกุลไฟล์) เป็น .htm หรือ .html
4. ตัว Web Server รับหน้าเอกสารที่ต้องการจาก Disk หรือ หน่วยความจำแล้วส่งกลับคืนไปยัง Browser ที่ร้องขอมา

ตัวไฟล์ HTML ที่ส่งมาจะถูกแปลความโดย Browser ของบุคคลที่ร้องขอหน้าเอกสารนั้น แล้ว แสดงผลบนหน้าจอให้เห็น ต่างจาก Active Server Pages ด้วยในขณะที่ IIS สามารถทำงานบริการหน้า เอกสาร Static HTML Active Server Pages จะยอมให้ IIS ให้บริการหน้าเอกสารที่เป็นแบบ Dynamic ด้วย การใช้ Active Server Pages หน้าเอกสารสามารถ ถูกสร้าง ขึ้นมาเพื่อตอบสนองสิ่งที่ผู้ใช้งานร้องขอมา ดังนั้น ตัว Web Server เองจึงกลายเป็นตัวสร้างหน้าเอกสารไปด้วย

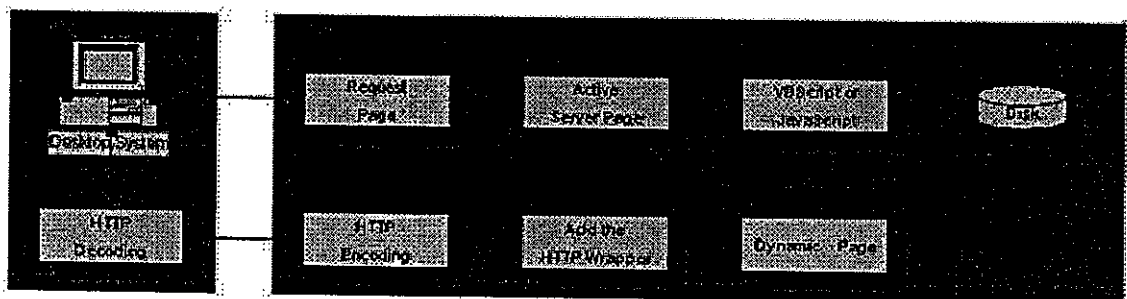


รูปที่ 2.1 แสดงการทำงานระหว่าง Server และ Client ของ Static HTML page

สรุป สามารถแตกขั้นตอนได้ดังนี้

1. ผู้ใช้ใส่ Internet Address ของไฟล์ Active Server Pages ที่ Address บาร์ของ Web Browser แล้ว กด Enter เพื่อร้องขอหน้าเอกสาร Active Server Pages ตัวอย่างเช่น <http://www.payap.ac.th/hello.asp>
2. ตัว Browser จะส่งคำร้องขอหน้าเอกสาร Active Server Pages ไปยัง IIS

3. ตัว Web Server จะรับคำร้องขอ และจดจำได้ว่าเป็นการร้องขอไฟล์ Active Server Pages เพราะว่าเป็นการร้องขอไฟล์ที่มี ส่วนขยาย (นามสกุลไฟล์) เป็น .asp
4. ตัว Web Server จะทำการรับไฟล์ ASP ที่เหมาะสมจาก Disk หรือหน่วยความจำ
5. ตัว Web Server จะทำการส่งไฟล์นั้นไปยังโปรแกรมพิเศษ ที่ชื่อว่า ASP.dll ไฟล์ Active Server Pages จะถูกประมวลผลจาก บนลงล่าง (Top to Bottom) และทุกๆ คำสั่งที่มีในไฟล์ จะทำการประมวลผล แล้วเก็บผลลัพธ์ ของการประมวลไว้ในรูปไฟล์ HTML ธรรมดา
6. ไฟล์ HTML ที่ได้จะถูกส่งกลับไปยัง Browser ที่ร้องขอ (ร้องขอมาในรูป .asp)
7. ไฟล์ HTML จะถูกแปลความโดย Browser ของผู้ร้องขอหน้าเอกสารนั้น และแสดงผลลัพธ์ไปที่หน้าจอ



รูปที่ 2.2 แสดงการทำงานระหว่าง Server และ Client ของ Active Server Pages

ในมุมมองของ Web Server แล้ว Active Server Pages จะมีความแตกต่างกับ หน้าเอกสารธรรมดาอย่างมาก เพราะ หน้าเอกสารธรรมดา (normal HTML page, static HTML page) จะถูกส่งไปให้ Browser โดยไม่มีการประมวลผลใดๆ เลย แต่สำหรับ Active Server Pages แล้ว คำสั่ง (Command) ที่มีอยู่ใน Active Server Pages จะต้องถูกประมวลผล (Execute) เพื่อที่จะสร้างผล ลัพธ์ที่เป็นหน้าเอกสาร HTML ซึ่งเป็นสิ่งที่ทำให้ Active Server Pages บรรลุไปด้วยส่วนที่เป็น Dynamic

ในมุมมองของ Browser กล่าวได้ว่า Active Server Pages นั้นจะเหมือนกับหน้าเอกสารธรรมดาอย่างมาก สิ่งเดียวที่ แตกต่างกันก็คือ Active Server Pages จะมีส่วนขยายเป็น .asp แทนที่จะเป็น .htm หรือ .html ที่เป็นแบบนี้เพราะว่า เมื่อมีการร้องขอ หน้าเอกสารที่เป็น Active Server Pages สิ่งที่จะส่งกลับคืนมานั้นก็คือหน้าเอกสารที่เป็น HTML ธรรมดา ประโยชน์ที่ได้รับจากลักษณะ เช่นนี้คือ Active Server Pages สามารถนำไปแสดงได้กับทุก Browser ที่มีอยู่

2.2 การติดต่อกับฐานข้อมูล [4], [5], [6]

2.2.1 เพิ่มข้อมูล

ข้อมูลมักจะเก็บในแฟ้มข้อมูลที่นิยมกันเช่น Access (*.mdb) หรือ Foxbase (*.dbf) ASP จะอาศัย ActiveX อีกตัว คือ ADO DB หรือ ActiveX Data Object DataBase ในการติดต่อและจัดการกับแฟ้มข้อมูล

2.2.2 ขั้นตอนการติดต่อแฟ้มข้อมูล

ปัจจุบันการติดต่อกับแฟ้มข้อมูลภายนอก หมายถึง โปรแกรมต่างๆที่ต้องการติดต่อกับแฟ้มข้อมูลภายนอกตัวเองเช่น โปรแกรม Access อยากจะติดต่อใช้งานแฟ้มข้อมูลของ SQL server เป็นต้น นั้น หมายถึง Access กำลังติดต่อแฟ้มข้อมูลภายนอก หรือ ASP ซึ่งไม่มีแฟ้มข้อมูลของตัวเอง ดังนั้นก็ต้องเป็นการติดต่อแฟ้มข้อมูลภายนอก การติดต่อแฟ้มข้อมูลภายนอกของ ASP จะต้องอาศัย driver ของแฟ้มข้อมูลนั้นๆเข้ามาช่วย โดยที่วิธีการติดต่อกับแฟ้มข้อมูลปัจจุบันมี 3 วิธีคือ

- ODBC
- DSNLess
- OLEDB

ในที่นี้ขออ้างถึงวิธี DSN เพียงอย่างเดียว เหตุเพราะว่าเป็นวิธีที่กลุ่มของเราใช้ในโครงงาน

2.2.2.1 วิธีการเชื่อมต่อแบบ DSN

เป็นการสร้างดาต้าซอร์ส (Data Source) ให้อยู่ในรูปของ File DSN หรือ System DSN เพื่อให้การเชื่อมต่อกับฐานข้อมูลสามารถจัดการและบริหารได้จากศูนย์กลางด้วย ODBC Data Source Administrator เพียงแห่งเดียว สำหรับวิธีการสร้าง DSN แบบ File DSN เราจะขอลำถึงขั้นตอนการสร้างพอสังเขปดังนี้

1. เมื่อเราเข้า ODBC Data Source Administrator เราจะเห็นแท็บ File DSN คลิกปุ่ม Add.. จะได้หน้าจอเลือก Driver โดยเราสามารถเลือก Advanced.. เพื่อแก้ไข Driver ได้ ซึ่งโดยปกติแล้วเราจะทิ้งค่า Default ที่ระบบกำหนดไว้ให้ จากนั้น คลิก OK
2. คลิก Next เพื่อเข้าสู่การตั้งชื่อ DSN
3. คลิก Next เพื่อเข้าสู่หน้าจอสรุปการกำหนดข้อมูลที่ใช้ในการสร้าง DSN
4. คลิก Finish เพื่อเลือกติดต่อกับฐานข้อมูลที่ต้องการ สำหรับ MS Access จะพบว่าไม่สามารถตั้งชื่อ DSN หรือใส่คำอธิบายที่ช่อง Data Source Name และ Discription ได้ตามลำดับ ทั้งนี้เนื่องจากการตั้งชื่อไว้ในขั้นตอนที่ผ่านมานั่นเอง ส่วน SQL Server ไม่จำเป็นต้องระบุชื่อของ DSN

5. สำหรับ File DSN ที่สร้างได้จะถูกเก็บรายละเอียดของการเชื่อมต่อ ให้อยู่ในรูปแบบของเท็กซ์ไฟล์ ซึ่งมีนามสกุลเป็น .dsn โดยดีฟอลต์จะอยู่ที่ C:\Program Files\Common Files\ODBC\Data Sources หากเปิดไฟล์เหล่านี้ด้วยโปรแกรม NotePad จะเห็นถึงสตริงข้อความที่ใช้ในการติดต่อกับฐานข้อมูล
6. สำหรับวิธีการเรียกใช้ File DSN ในทุกๆ ประเภทสามารถเลือกได้จากรูปแบบดังนี้

Syntax: DSN=<DSN name>;UID=<user name>;PWD=<password>

ตัวอย่าง DSN=biblio.dsn;UID=admin;PWD=password ในกรณีที่ file DSN เป็น file ชื่อ biblio.dsn

ในกรณีการสร้าง DSN file แล้วจะมีการสร้างรายละเอียดไว้ภายในเช่น รหัสผ่านดังนั้นหากไม่ได้ทำการระบุค่าดังกล่าวนี้ แสดงว่าจะเป็นการกำหนดด้วยค่า default โปรดสังเกตเครื่องหมายเท่ากับระหว่างตัวแปรจะต้องอยู่ติดกันเสมอ มิเช่นนั้นอาจเกิดข้อผิดพลาดขึ้นได้ แล้วจะมีผลต่อ Connection String ซึ่งจะกล่าวต่อไป

2.2.2.2 การติดต่อกับ Connection String

เป็นการสร้างการเชื่อมต่อกับแหล่งเก็บข้อมูลให้อยู่ในส่วนของสตริงคำสั่ง ประโยชน์คือสตริงคำสั่งจะถูกเก็บอยู่ภายในเว็บเพจ ทำให้ไม่ต้องคอนฟิกเพิ่มเติมให้กับสภาพแวดล้อมของเว็บเซิร์ฟเวอร์ ตัวอย่างเช่น การอัปโหลดเว็บเพจไปยังเซิร์ฟเวอร์ของ ISP (Internet Server Provider) ที่ไม่สามารถสร้าง DSN ของ ODBC ได้ตามอำเภอใจ

ส่วนข้อเสียคือเมื่อมีเว็บเพจมากขึ้นจะทำให้การจัดการเชื่อมต่อเหล่านั้นยากขึ้นไปเป็นลำดับ สำหรับปัญหาในส่วนนี้สามารถแก้ไขได้โดยการเขียนสคริปต์ ให้มีระบบด้วยการสร้างสตริงคำสั่งในการเชื่อมต่อกับแหล่งเก็บข้อมูลให้อยู่เป็น file แยก ต่างหาก และสามารถถูกเรียกใช้งานจากเว็บเพจอื่นๆ ได้ด้วยวิธีการ Include ไฟล์เข้ามาเท่านั้นเอง นอกจากนี้เรายังสามารถเลือกใช้วิธีการกำหนดสตริงคำสั่งของการเชื่อมต่อไว้ที่ตัวแปรระดับ Session หรือ Application ที่อาจมีการสร้างทิ้งไว้ก่อนในไฟล์ global.asa ได้อีกด้วย

สำหรับวิธีการเรียกใช้การเชื่อมต่อแบบสตริงคำสั่งนี้แบ่งออกได้ตามลำดับชั้นในการเข้าถึงแหล่งเก็บข้อมูลที่ผ่านมาโดยหากเป็นการเชื่อมต่อแบบ ODBC จะมีวิธีการที่แตกต่างกันไป

2.2.2.3 ADODB

ASP จะใช้ ActiveX ที่ชื่อ ADODB ในการติดต่อกับแฟ้มข้อมูล มาตรฐานนี้ขอยกตัวอย่าง ของการติดต่อกับแฟ้มข้อมูล Access โดยใช้การติดต่อแบบ DSNLess จะต้องเขียนดังนี้

1. set conn = CreateObject("ADODB.connection")
2. conn.open "Driver={Microsoft Access Driver (*.mdb)};DBQ=c:\mydataaccess\data1.mdb"

หากไม่เกิด error ใดๆ เราก็จะติดต่อเข้าไปใช้งาน data1.mdb ได้แล้ว จากนั้นก็ต้องระบุว่าเราจะเอาข้อมูลจากตารางอะไร แบบไหน ด้วยการกำหนดผ่านคำสั่งภาษา SQL Query หลักที่ใช้กันจะเป็น Action Query มี 4 แบบ คือ

- Select query ใช้ในการนำข้อมูลมาแสดง
- Update query ใช้ในการแก้ไขข้อมูล
- Delete query ใช้ในการลบข้อมูล
- Insert query ใช้ในการเพิ่มข้อมูล

เริ่มจาก select query สมมติว่าเราจะนำข้อมูลจาก ตารางชื่อ table1 มาแสดงก็จะระบุคำสั่งเป็น Select * from table1 จากนั้นก็สั่งให้สร้างตารางผลลัพธ์ของข้อมูลออกมา เรียกว่าสร้าง recordset ซึ่งจะเป็น set ของ record นั้นเอง ตัว recordset เองก็จะมีวิธีการสร้างได้อีกหลายวิธี เช่นสร้างแบบให้อ่านอย่างเดียวย ห้ามแก้ไขเป็นต้น ทำต่อจากการสร้างการติดต่อเพิ่มข้อมูล คือ ข้อ 1 กับ 2

3. sql="Select * from table1"

4. set rs = conn.execute(sql)

จะได้ ตารางผลลัพธ์หรือได้ข้อมูลมาแล้ว โดยที่ pointer จะชี้อยู่ที่รายการแรก หากจะดูข้อมูลจาก field ใดๆก็ระบุชื่อ field ได้เลยด้วยคำสั่ง rs("ชื่อfield") ซึ่งควรจะรู้มาก่อนว่าในตารางนั้นมี field อะไรบ้าง หากไม่ทราบก็ต้องใช้การอ้างอิงผ่านหมายเลขแทน โดยหมายเลขจะเริ่มจาก 0 คือเป็น field แรก ไปเรื่อยเท่ากับจำนวน field -1 เพราะว่า เริ่มจาก 0 การหาจำนวน field จะใช้คำสั่ง rs.fields.count

2.2.3 ปัญหาที่ควรทราบ

เนื่องจากไมโครซอฟท์ชอบเขียนโปรแกรมแบบที่ยึดหยุ่น กล่าวคือ งานแบบเดียวกัน เราเขียนคำสั่งได้หลายวิธี ทำนองเดียวกันการเขียนที่ไม่เหมือนกัน ก็อาจจะได้ผลลัพธ์ที่ไม่เหมือนกัน การเปิดแบบ set rs = conn.execute(sql) จะสามารถเรียกใช้ property บางอย่างได้ เช่นไม่มี property บอกว่าทั้งหมดมีกี่รายการเป็นต้น แต่ที่ได้มาก็น่าจะเพียงพอทำงานได้ คือ

- rs.fields.count บอกจำนวน field
- rs.fields.name บอกชื่อ field
- rs.fields.value ข้อมูลใน field นั้น
- rs.movenext เลื่อนรายการไปข้างหน้า 1 รายการ
- rs.eof รายการสุดท้าย หรือหมดรายการ

2.2.4 เขียน asp script ที่จะติดต่อกับ database

เอาความรู้ทั้งหมดมาประมวล แล้วเขียนเป็น asp script เพื่อดูข้อมูลในแฟ้มข้อมูล จะเอามาข้อมูลจาก data1.mdb โดยเลือกเอาตารางชื่อ pagecount ซึ่งเป็นตารางที่เก็บค่า ตัวนับ ขึ้นมาแสดงดังนี้

```
<%
set conn = CreateObject("ADODB.connection")
conn.open "Driver={Microsoft Access Driver (*.mdb)};DBQ=c:\data\data1.mdb"
sql="Select * from pagecount"
set rs = conn.execute(sql)
response.write "<table border=1><tr bgcolor=skyblue>"
for i = 0 to rs.fields.count-1
response.write( "<td>" & rs(i).name & "</td>")
next
do while not rs.eof
response.write "</tr><tr>"
for i = 0 to rs.fields.count-1
response.write( "<td>" & rs(i).value & "</td>")
next
rs.movenext
loop
response.write "</tr></table>"
rs.close
conn.close %>
```

และมีผลลัพธ์ดังนี้

pagename	pagecount	ldate
mcu1	151724	5/2/2002 1:37:32
psymain	4026	5/2/2002 1:09:01
mednet	28625	4/2/2002 23:58:54
medlib	8997	25/12/2000 14:26:39
keha	2217	4/2/2002 9:51:36
addict	5	19/4/2001 13:25:35

ผลลัพธ์ (ต่อ)

pagename	pagecount	ldate
ftjmain	22736	5/2/2002 1:02:48
ped2000	168	25/4/2001 11:42:25
medlink	17413	5/2/2002 1:04:25
ebmain	8509	4/2/2002 22:11:53
wmail	16156	30/8/2001 14:23:38
eye	301	9/11/2001 15:23:07
download	3859	13/10/2001 23:29:12
mpc	1387	2/2/2002 12:51:22
mllibphp	25026	4/2/2002 20:39:01
ktalk	7434	4/2/2002 23:54:35
faq	5283	4/2/2002 18:21:43
phpole	2258	4/2/2002 20:04:10
phpodbc2	2740	3/2/2002 12:19:54
phpodbc3	2352	3/2/2002 10:42:28
phpcookie	2638	4/2/2002 22:26:05
maneerat	25	29/1/2002 13:55:00
saacmu	62	5/6/2001 11:07:12
phpodbc4	641	5/2/2002 0:54:59
gdcom	1181	4/2/2002 23:18:55
badblue	1268	4/2/2002 20:01:14
toxicplant	270	3/2/2002 19:29:18
pharmmeet	8	4/2/2002 13:14:54

2.2.5 ติดต่อ database

ขั้นตอนการติดต่อกับ database มีขั้นตอนดังนี้

- set conn = CreateObject("ADODB.connection")
- conn.open "Driver={Microsoft Access Driver (*.mdb)}; DBQ = c:\mydataaccess\data1.mdb"

จากนั้นก็จะเป็นการกำหนด sql statement เพื่อให้ทำงานกับ database ดังกล่าว สังเกต ทั้ง 3 งานต่อไปนี้ จัดอยู่ในประเภทเป็นงานที่ไม่ต้องการการแสดงผลหรือออกมาเป็น ตารางผลลัพธ์หรือ recordset ดังนั้นไม่จำเป็นต้องกำหนดให้มีการสร้าง recordset ไม่ว่าจะเป็น

- set rs = CreateObject("ADODB.Recordset") หรือ
- set rs=conn.execute(sql)

โดยจะมุ่งไปที่คำสั่ง conn.execute(sql) เฉย ตามที่บอกว่าไม่ต้องการให้สร้าง recordset

2.2.6 การเพิ่มข้อมูล

sql ที่ใช้ในการเพิ่มข้อมูลคือ Insert into ชื่อ table (field list) values (value list) ดังกรณีที่เราจะเพิ่มรายการให้กับ database ที่เราเปิดการติดต่อมาจากขั้นตอนแรกแล้ว คือ table ชื่อ table1 โดยที่ field list คือ pagename กับ pagecount และสมมติว่าเราจะกำหนดค่า pagename เป็น mytest และ pagecount เป็น 1 ก็จะกำหนด sql เป็น ดังนี้

- กำหนดแบบอ้างอิงค่าจากตัวแปร วิธีนี้เหมาะกับการนำไปใช้ในการเขียน web page ที่เราต้องรับค่าที่ผ่านมาจาก form อีกทีเช่นให้เขากรอก form ก่อนแล้วค่อยส่งไปบันทึก
- vPageName="mytest"
- vPagecount=1
- sql="Insert into table1 (pagename,pagecount) values ('" & vPageName & "','" & vPageCount & "')"

หรือ กำหนดตรงๆ

- sql="Insert into table1 (pagename,pagecount) values ('mytest','1')"

TIP : การเขียน sql แบบแรก ต้องรับค่าจาก form แต่ในการทำงานจริงๆ ค่า sql จาก แบบแรก จะต้องเทียบเท่าได้กับแบบที่สอง นั่นคือ sql="Insert into table1 (pagename,pagecount) values ('" & vPageName & "','" & vPageCount & "')" จะต้องถูกแปลงไปเป็น sql="Insert into table1 (pagename,pagecount) values ('mytest','1')"

ได้ตรงกัน โดย asp จะทำการแปลงให้ แต่ก็ควรตรวจสอบ แสดงผลการแปลงดูก่อนด้วย

- response.write "SQL statement = " & sql

และหากผลลัพธ์คือการแสดงผลตรงกับที่เราต้องการก็แสดงว่าเราเขียน sql แบบแรกถูกแล้ว ก็ไปต่อ ดังนี้

- conn.execute(sql)

ข้อมูลจะถูกบันทึกเพิ่มเข้าไปใน table1 โดย ข้อมูลก็ต้องมีคุณสมบัติที่ตรงกับแต่ละ field ด้วย

2.2.7 การแก้ไขข้อมูล

sql ที่ใช้ในการแก้ไขข้อมูลคือ update table ชื่อtable set ชื่อfield=ค่าใหม่ where เงื่อนไข เช่น หากข้อมูลที่เราเพิ่มเข้าไปเราต้องการแก้ค่า pagecount จาก 1 เป็น 100 ก็กำหนดเป็น

- sql="update table table1 set pagecount=100 where pagename='mytest'"

ส่วนเวลานำไปใช้จริงก็ต้องแก้ sql ให้รับค่าแบบกำหนดตัวแปรก่อนเช่นกัน เพราะจะต้องรับค่าจาก form ดังนี้

- vPageName="mytest"
- vPagecount=100
- sql="update table table1 set pagecount= " & vPageCount & " where pagename=" & vPageName & ""

คำเตือน : หากไม่ระบุเงื่อนไข จะหมายถึงการแก้ไขค่าทุกรายการ และ set field= จะทำทีละ 1 field หรือหลายๆ field ก็ได้โดยแยกแต่ละ field ด้วย comma เช่น set field1 = value1, field2 = value2, ...

2.2.8 การลบรายการ

sql ที่ใช้ในการลบข้อมูลคือ delete from ชื่อtable where เงื่อนไข จากที่แก้ไขก่อนหน้านี้ โดย

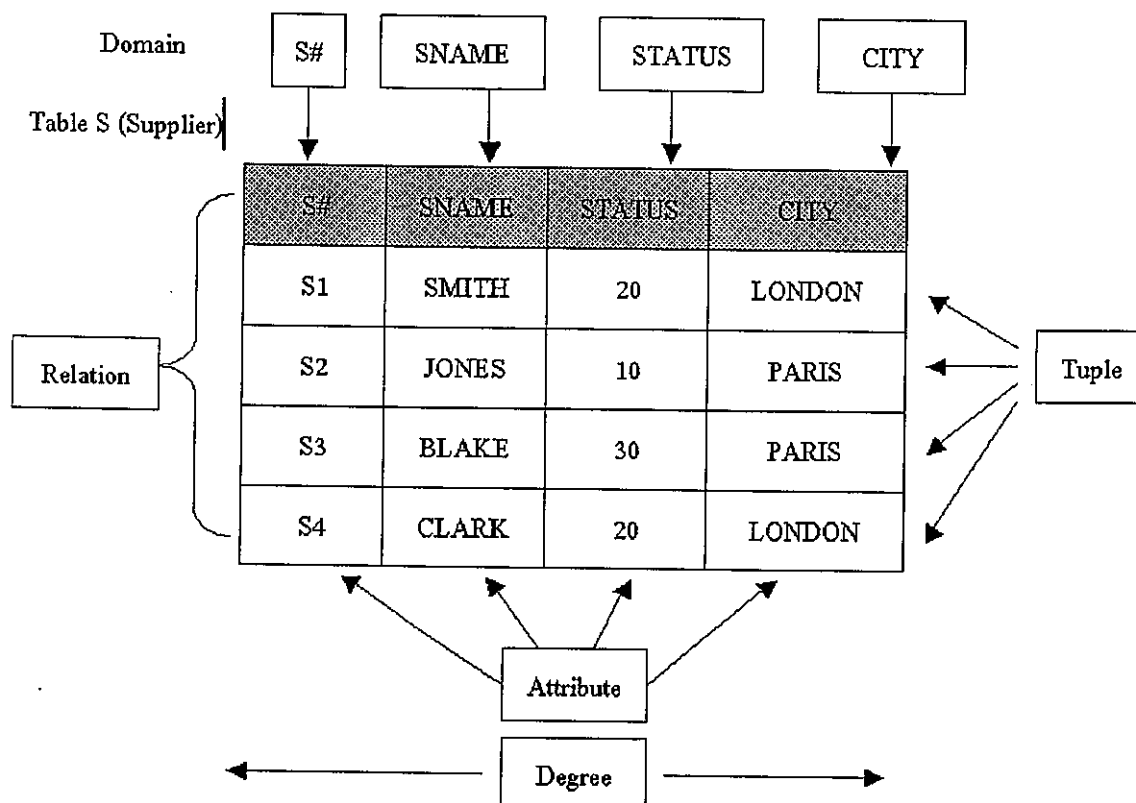
- sql="delete from table1 where pagename='mytest'"
หรือโดยการรับจาก form โดย
- vPageName="mytest"
- sql="delete from table1 where pagename=" & vPageName & ""

2.3 ระบบฐานข้อมูลเชิงสัมพันธ์(Relational Database Model) [7]

ระบบฐานข้อมูลเชิงสัมพันธ์ได้รับความนิยมมากในปัจจุบันเนื่องจากคุณลักษณะ 2 ประการ คือ ประการแรก ข้อมูลที่เก็บภายในสื่อจะไม่ขึ้นอยู่กับหรือผูกติดกับผู้ใช้คนใดคนหนึ่งโดยเฉพาะ และ ประการที่ 2 ได้แก่ ลักษณะการออกแบบข้อมูลสามารถที่จะนำเอาหลักการทางคณิตศาสตร์ไปประยุกต์ใช้ได้ทันที ทำให้ระบบการทำงานทำความเข้าใจได้โดยง่าย มีความยืดหยุ่นสูง จึงสามารถแก้ไขได้โดยง่ายเหมาะสำหรับระบบฐานข้อมูลระดับกลางและเล็ก

2.3.1 นิยามที่ใช้ใน Relational Database Model

- Relation หมายถึง ตาราง (Table) ใน Model แบบอื่นๆ
- Tuple หมายถึง แถว (Row) หรือ Record
- Cardinality หมายถึง จำนวนแถวในตาราง
- Attribute หมายถึง คอลัมน์ (Column) หรือ Field
- Degree หมายถึง จำนวน Column ในตาราง
- Domain หมายถึง ขอบเขตของค่าต่างๆ ที่เป็นไปได้



รูปที่ 2.3 แสดงตัวอย่างของ Relational Database Model

2.3.2 ความสัมพันธ์ระหว่างค่าภายใน Relation

กำหนดให้ R เป็น Relation ที่เป็นตาราง 2 มิติ ประกอบไปด้วย Domain $D_1, D_2, \dots, D_n$ มีค่าที่สามารถระบุได้จากชื่อของ Column และค่าของข้อมูลภายใน Column ทั้งสองส่วนนี้เรียกว่า Attribute ซึ่งจะแทนด้วยสัญลักษณ์ A

แต่ละ Column ประกอบไปด้วย Attribute จำนวนหนึ่งซึ่งสามารถเขียนได้ในรูปความสัมพันธ์กับโดเมนได้ดังนี้

$$\{(A_1:D_1), (A_2:D_2), \dots, (A_n:D_n)\}$$

รีเลชัน R

D1	D2	D3	...	Dn
A1	A2	A3	...	An

หาก Attribute A_j เป็นส่วนที่สามารถอ้างอิงถึง Domain D_j ($j=1, 2, \dots, n$) ได้เพียง Domain เดียวนั้นก็หมายความว่า การระบุ Domain D_j โดยอาศัย Attribute A_j สามารถแสดงความแตกต่างจากค่า Domain อื่นๆได้

ส่วนของข้อมูลภายใน Column ประกอบไปด้วย Set ของ Tuple ซึ่งแต่ละ Tuple ประกอบไปด้วย Attribute จำนวนหนึ่ง สามารถระบุค่าได้โดยการเขียน ให้อยู่ในรูปของความสัมพันธ์ระหว่างกัน ได้ดังนี้

$$\{(A_1:V_{i1}), (A_2:V_{i2}), \dots, (A_n:V_{in})\}$$

V {	A		
	A1 : V _{i1}		
	A2 : V _{i2}		

รูปที่ 2.4 แสดงความสัมพันธ์ของ relation

($I=(1, 2, \dots, m)$) เมื่อ m มีค่าเท่ากับจำนวน Tuple ใน Set นั้นๆ ในแต่ละ Tuple จะมีค่า Attribute แสดงความสัมพันธ์ในรูปของ $(A_j : V_{ij})$ ที่ถูกระบุในแต่ละ Column ส่วนค่า V_{ij} จะเป็นค่าที่ได้มาจาก Domain D_j ที่สามารถแสดงความเป็นเอกลักษณ์ Unique ได้โดยอาศัยความสัมพันธ์กับ Attribute A_j

ค่าของ m และ n ถูกเรียกว่า Cardinality (จำนวนแถว) และ Degree (จำนวน Column) ตามหลักความสัมพันธ์ภายใน Relation สำหรับจำนวนค่าภายใน Cardinality จะเปลี่ยนแปลงโดยขึ้นอยู่กับช่วงเวลาใดเวลาหนึ่ง (ได้แก่การเพิ่มหรือลดจำนวน Record) ส่วนการเปลี่ยนแปลงของ Degree จะไม่ขึ้นอยู่กับปัจจัยนี้

จาก Relation Supplier ประกอบด้วย Domain 4 ค่าได้แก่

1. Supplier ID
2. Supplier Name
3. Supplier Status
4. Supplier City

โดยมีส่วนของ Column ที่สามารถระบุได้ดังนี้

(S#, SNAME, STATUS, CITY)

(S1, SMITH, 20, LONDON)

ส่วนการระบุภายใน Relation สามารถเขียนให้อยู่ในรูปความสัมพันธ์ได้ดังนี้

(S# : S1)

(SNAME : SMITH)

(STATUS : 20)

(CITY : LONDON)

ถ้าค่า S1, SMITH, 20 และ LONDON ตามลำดับ จะเป็นค่า Attribute ของ Domain S#, SNAME, STATUS และ CITY ซึ่งประกอบกันเป็น Tuple หรือ Record ที่มีรากฐานมาจาก Relation เดียวกัน คือ Relation Supplier นั่นเอง

ตารางที่ 2.1 แสดง relation table

Relation S (Supplier)

S#	SNAME	STATUS	CITY
S1	SMITH	20	LONDON
S2	JONES	10	PARIS
S3	PAUL	30	PARIS

Relation SP (Supplier & Part)

S#	P#	Qty
S1	P1	300
S2	P2	200
S3	P3	400

ตารางที่ 2.1 แสดง relation table (ต่อ)

Relation P (Part)

P#	PNAME	COLOR	WEIGHT	CITY
P1	NUT	RED	12	LONDON
P2	BOLT	GREEN	17	PARIS
P3	SCREW	BLUE	17	ROME

2.3.3 Domain

เป็นค่าขอบข่ายที่ควรจะเป็นค่าในแต่ละ Attribute หากค่าของ Attribute มาจาก Domain เดียวกัน เช่น หมายเลขชิ้นส่วนใน Relation P(P.P#) และหมายเลขชิ้นส่วนใน Relation SP(SP.P#) ที่มีที่มาจาก Domain เดียวกัน ดังนั้นค่า Attribute ทั้งสองสามารถนำมาทำการเปรียบเทียบรวมหรือแม้แต่ทำการ Union กันได้ นอกจากนี้ยังมี Domain หนึ่งที่เรียกว่า Composite เป็น Domain หลักที่รวมเอา Domain ย่อยไว้ด้วยกัน เช่น Domain ของ Date จะประกอบไปด้วย Domain ของ Month ,Day,Year ไว้ด้วยกัน โดยมีขอบข่ายของค่าดังนี้ 1-12,1-31,0-9999

2.3.4 คีย์หลัก (Primary Key)

เป็นสัญลักษณ์ความเป็นเอกลักษณ์ (Unique) ของ Relation นั้นๆ เช่น จากตารางทั้งสาม จะมี Key หลักได้แก่ S#,P# และ S#P# จากตาราง S ,P และ SP ตามลำดับ แต่บางครั้งใน Relation หนึ่งๆ อาจมีลักษณะความเป็นเอกลักษณ์ ได้หลายตัว เช่น ใน Relation ของ Supplier จะมีค่าของ Supplier Number (.S#) และ Supplier Name (S.SNAME) เป็นสัญลักษณ์แสดงความเป็นเอกลักษณ์ในกรณีทั้งสองค่านี้ ถูกเรียกว่า Candidate Key ซึ่งเมื่อตัวใดตัวหนึ่งถูกเลือกให้เป็น Key หลักตัวที่เหลือจะถูกเรียกว่า Alternate Key

หากกำหนดให้ Relation R ประกอบไปด้วย Attribute K และอื่นๆ ค่า K จะถือได้ว่าเป็น Candidate Key ก็ต่อเมื่อมีคุณสมบัติดังต่อไปนี้

1. Uniqueness

ในช่วงเวลาที่กำหนดจะต้องไม่มีค่าใน Tuple ใดๆ ใน Relation R ที่มีค่าเหมือนกับ K

2. Minimality

หาก Attribute K มีลักษณะเป็น Domain ร่วม (Composite) ค่า Domain ย่อยเหล่านั้นเมื่อตัดออกจะทำลายคุณสมบัติความเป็นเอกลักษณ์ของ Domain ร่วมให้สูญเสียไปได้ โดยปรกติแล้ว Relation แต่ละตัวจะมีจำนวน Candidate Key 1 ตัวหรือมากกว่าก็ได้ ทั้งนี้เนื่องจากคุณลักษณะของ Tuple จะไม่ยอมให้มีค่าที่ซ้ำกันได้ ดังนั้นแต่ละ Field จึงสามารถเป็น Candidate Key ได้โดยไม่ทำลายคุณสมบัติเดิมเช่น

ตารางที่ 2.2 Relation Chemical Element

Name	Symbolic	Atomic Number		
.....				

จากตารางธาตุเห็นได้ว่า ไม่ว่าจะเป็นชื่อแต่ละธาตุ ลักษณะทางเคมีหรือค่า AtomicNumber ทั้งหมดสามารถเป็น Key หลักได้ ทั้งนี้เนื่องจาก สำหรับแต่ละ Attribute สามารถแสดงความเป็นเอกลักษณ์ของ แต่ละ Relation ได้ ดังนั้นเมื่อค่าใดค่าหนึ่งเป็น Keyหลัก ส่วนที่เหลือก็จะเป็น Alternate Key โดยอัตโนมัติ

2.3.5 คีย์ นอก (Foreign Key)

เป็น Attribute ใน relation หนึ่งที่ใช้ในการอ้างอิงถึง Attribute เดียวกันในอีก Relation หนึ่ง โดยที่ Attribute นี้จะมีค่าเป็น Key หลักใน Relation ที่ถูกอ้างอิงถึง ตามที่มี Attribute นี้ปรากฏอยู่ใน Relation ทั้งสอง เพื่อประโยชน์ในการเชื่อมข้อมูลทั้งสองซึ่งกันและกัน

จาก Relation ของฐานข้อมูลมี Supplier And Parts จะเห็นว่า Attribute S# ใน Relation SP เป็นค่าที่แสดงการเชื่อมโยงระหว่าง 2 Relation S และ SP นั่นคือ Attribute ดังกล่าวเป็น Attribute ที่เป็น Key หลักซึ่งปรากฏ อยู่ใน Relation S เช่นเดียวกับในกรณีของ Attribute P# ใน Relation SP ซึ่งทั้ง Attribute S# และ P# ใน Relation SP จะถูกเรียกว่า Key นอก (Foreign Key)

2.3.6 ความสัมพันธ์ระหว่าง Relation

ในระบบข้อมูลเชิงสัมพันธ์ข้อมูลทั้งหลายจะถูกเก็บอยู่ในรูปของ Relation แยกออกจากกัน และสร้างความสัมพันธ์ระหว่าง Relation โดยอาศัย Key หลัก โดยการกำหนดค่า Attribute ที่เป็น Key หลักเพื่อแสดงความสัมพันธ์ไปยังอีก Relation หนึ่งนั่นเอง

ชนิดความสัมพันธ์ระหว่าง Relation ในระบบฐานข้อมูลเชิงสัมพันธ์จะมีดังต่อไปนี้

2.3.6.1 ความสัมพันธ์แบบ 1 ต่อ กลุ่ม (1 to many Relationships)

หมายถึง ค่าของ Record A จะมีความสัมพันธ์กับค่าของ Record B ได้มากกว่า 1 Record แต่ของ Record B จะมีความสัมพันธ์ กับค่าของ Record A ได้เพียง Record เดียว

2.3.6.2 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (many to many Relationships)

หมายถึง ค่าของทั้ง Record A และ Record B สัมพันธ์กัน ได้มากกว่า 1 Record

2.3.6.3 ความสัมพันธ์แบบ 1 ต่อ 1 (1 to 1 Relationships)

หมายถึง ค่าของ Record A มีความสัมพันธ์กับค่าของ Record B ได้เพียง 1 Record เท่านั้น ซึ่งมีโอกาสเกิดขึ้นได้ยาก

2.3.7 กฎที่เกี่ยวข้องกับ Key ในระบบฐานข้อมูลสัมพันธ์

เนื่องจาก Relation ต่างๆ ในฐานข้อมูลเชิงสัมพันธ์ จะอ้างอิงกับความสัมพันธ์ของข้อมูล โดยใช้ Key เป็นหลัก ดังนั้นในการออกแบบระบบ จึงจำเป็นที่จะต้องทำการกำหนดกฎเกณฑ์ที่เกี่ยวข้องกับ Key ในฐานข้อมูลเชิงสัมพันธ์ดังต่อไปนี้

2.3.7.1 กฎความบูรณาภาพของ Entity (The Entity Integrity Rule)

กำหนดว่า Attribute ใดที่จะเป็น Key หลักข้อมูล Attribute นั้นจะต้องไม่เป็นค่าว่าง (Null) ซึ่งหมายถึงค่าที่ไม่สามารถระบุได้อย่างชัดเจน หรือค่าที่ไม่อยู่ในขอบเขตของ Domain เช่น กำหนดให้รหัสของ Supplier S# ทั้งนี้ หากทำการกำหนดรหัสในภายหลัง Attribute ดังกล่าวจะถือว่าเป็นค่าว่างทันที เมื่อมีการแก้ไขในระบบเกิดขึ้น จะทำให้เกิดความผิดพลาดตามไปด้วย

2.3.7.2 กฎความบูรณาภาพของการอ้างอิง (The Referential Integrity Rule)

การอ้างอิงระหว่าง Relation ในฐานข้อมูลเชิงสัมพันธ์ จะใช้ Key นอกของ Relation หนึ่ง ทำการเชื่อมโยงกับค่าของ Attribute ที่เป็น Key หลักของอีก Relation หนึ่ง เพื่อใช้ตรวจสอบข้อมูลอื่นที่เกี่ยวข้อง ดังนั้นในขั้นตอนการออกแบบระบบ จึงจำเป็นที่จะต้องสร้างระบบป้องกัน และตรวจสอบ Key ต่างๆ เพื่อที่จะสามารถรับประกันความมีตัวตนของทุก Key ที่มีการอ้างอิงถึงกันได้

ในกรณีที่ต้องการลบ หรือแก้ไขข้อมูลของ Attribute ที่เป็น Key หลักใน Relation หนึ่ง ที่ถูกอ้างอิงจาก Key นอกในอีก Relation หนึ่ง จะสามารถกระทำได้ ขึ้นอยู่กับการออกแบบฐานข้อมูลซึ่งสามารถกำหนดได้เป็น 4 ลักษณะคือ

1. การลบ หรือแก้ไขข้อมูลแบบมีขีดจำกัด (Restrict)

การลบ หรือแก้ไขข้อมูลจะกระทำได้อีกต่อเมื่อข้อมูลของ Key หลัก ใน Relation หนึ่ง ไม่มีข้อมูลที่ถูกอ้างอิง โดย Key นอกของอีก Relation หนึ่ง

2. การลบ หรือแก้ไขข้อมูลแบบถูกโซ่ (Cascade)

ถ้ามีการลบ หรือแก้ไขข้อมูลของ Key หลัก ใน Relation หนึ่ง ต้องทำการลบ หรือแก้ไข ข้อมูลของ Key นอกที่อ้างอิงถึง Key หลักนั้นด้วย

3. การลบ หรือแก้ไขข้อมูลโดยเปลี่ยนเป็นค่าว่าง (Nullify)

การลบ หรือแก้ไขข้อมูลจะกระทำได้อีกต่อเมื่อข้อมูลของ Key นอกในข้อมูลที่ถูกอ้างอิงถึงเป็นค่าว่างก่อนจากนั้นจึงทำการลบ หรือแก้ไขจาก Key หลักออกไปภายหลัง

4. การลบ หรือแก้ไขข้อมูลโดยใช้ค่ากำหนด (Default)

มีการลบ หรือแก้ไขข้อมูลของ Key หลัก สามารถทำได้โดย กำหนดค่าของ Key นอกเพื่อปรับค่าให้เป็นค่า Default ซึ่งผู้ใช้สามารถกำหนดขึ้นเองได้

2.3.8 คุณสมบัติในการจัดเก็บข้อมูลแบบสัมพันธ์ (Property Of Relation)

2.3.8.1 ข้อมูล และแถวจะไม่ซ้ำกัน (No Duplicate Tuples)

2.3.8.2 การเรียงลำดับของข้อมูลแต่ละแถวไม่เป็นสาระที่สำคัญ (Tuples are Unordered)

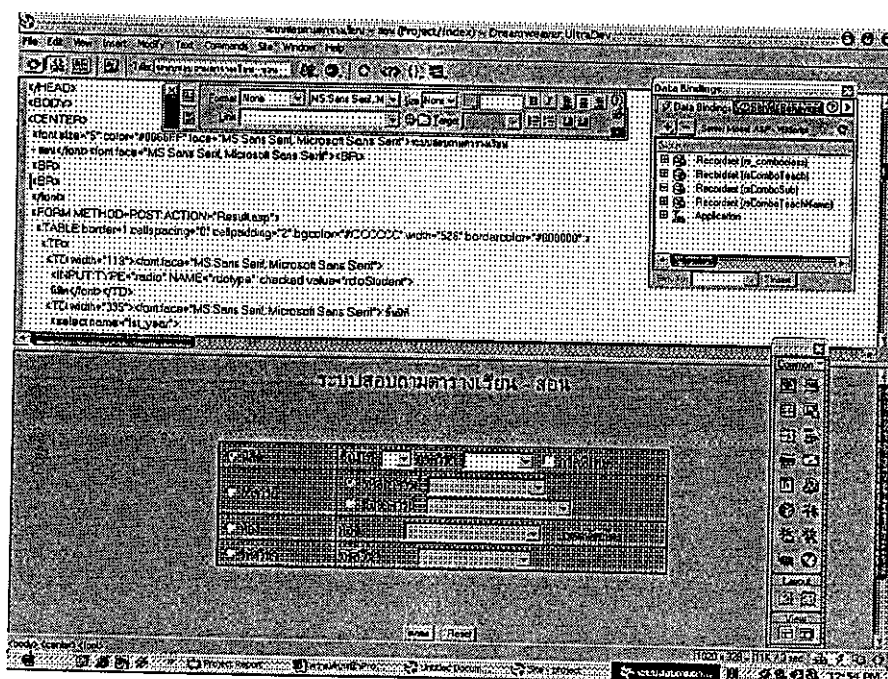
2.3.8.3 การเรียงลำดับของ Attribute ไม่เป็นสาระที่สำคัญ (Attribute are Unordered)

2.3.8.4 Attribute หนึ่งๆ จะบรรจุข้อมูลได้เพียงค่าเดียว (Attribute Values are atomic)

2.4 เครื่องมือที่ช่วยในการเขียนเว็บเพจ [8], [9], [10]

Marcromedia Dreamweaver Ultradev 4.1

เป็นเครื่องมือที่ใช้ในการสร้าง Web Page ที่มี Tools ช่วยในการเขียนโค้ดภาษา HTML และยังช่วยในการเขียน Active Server Page (ASP) ได้



รูปที่ 2.5 แสดงตัวอย่างโปรแกรม Marcromedia Dreamweaver Ultradev 4.1