

บทที่ 2

หลักการและทฤษฎี

2.1 หลักในการแยกปัญหาหรืออาการเสียของคอมพิวเตอร์เป็นหมวดหมู่

2.1.1 ความหมายของปัญหาหรืออาการเสียของคอมพิวเตอร์

ปัญหาหรืออาการเสียของคอมพิวเตอร์ที่ใช้อ้างถึงในโปรแกรม หมายถึง อาการที่เกิดขึ้นกับเครื่องคอมพิวเตอร์แล้วทำให้เครื่องคอมพิวเตอร์หยุดทำงานหรือทำงานช้าชงจนไม่สามารถใช้เครื่องคอมพิวเตอร์ได้ตามปกติได้

2.1.2 การตีกรอบปัญหา (Bracket) [1,2]

การตีกรอบปัญหาการทำงานของเครื่องคอมพิวเตอร์ ก่อนอื่นก็ต้องทราบปัญหาที่เกิดขึ้นเป็นอันดับแรก เช่นสมมติว่ามีปัญหาที่เกิดจากการพิมพ์ จะต้องทำการพิจารณาว่าปัญหาอาจเกิดขึ้นกับส่วนประกอบใดบ้างของเครื่องคอมพิวเตอร์ที่เกี่ยวข้องกับการพิมพ์ สมมติว่าพิจารณาแล้วได้สาเหตุของปัญหาเป็น 5 ลักษณะ ได้แก่ ปัญหาซอฟต์แวร์ที่ใช้ ปัญหาไดรเวอร์ของเครื่องพิมพ์ (Printer Driver) (คือซอฟต์แวร์ที่ทำหน้าที่ติดต่อกับวินโดวส์ให้สัมพันธ์กับเครื่องพิมพ์) ปัญหาฮาร์ดแวร์ ปัญหาสายพ่วง และปัญหาที่ตัวเครื่องพิมพ์

เมื่อทำการทราบถึงสาเหตุของปัญหาที่คิดว่าน่าจะเป็นไปได้แล้ว ก็ทำการทดสอบแต่ละปัญหาเพื่อทดสอบว่าเกิดจากสาเหตุใด สาเหตุที่ทดสอบแล้วไม่ใช่สาเหตุของปัญหาก็จะถูกกำจัดออกไป แต่จะถูกเก็บไว้ในแผนผังของปัญหา เพื่อจัดเก็บไว้ในฐานข้อมูล

2.1.3 การแบ่งหมวดหมู่ของปัญหา

เนื่องจากปัญหาที่เกิดขึ้นกับเครื่องคอมพิวเตอร์อาจเกิดขึ้นโดยมีความเกี่ยวข้องกันระหว่างปัญหาทางด้านซอฟต์แวร์และปัญหาทางด้านฮาร์ดแวร์ บางปัญหาจึงไม่สามารถระบุได้ว่าเกิดจากฮาร์ดแวร์หรือซอฟต์แวร์ ต้องอาศัยการตรวจสอบอุปกรณ์และฮาร์ดแวร์ที่เกี่ยวข้องที่ละอุปกรณ์ การแบ่งหมวดหมู่นั้นจะทำการแบ่งตามปัญหาที่พบ

2.2 การพัฒนาโปรแกรมด้วย Microsoft Visual Basic 6.0 [3,5]

2.2.1 ประวัติความเป็นมาและความหมายของ Visual Basic

Visual Basic เป็นภาษาคอมพิวเตอร์ (Programming Language) ที่พัฒนาโดยบริษัทไมโครซอฟท์ ซึ่งเป็นบริษัทที่สร้างระบบปฏิบัติการ Windows 95/98 และ Windows NT ที่เราใช้กันอยู่ในปัจจุบัน โดยตัวภาษาเองมีรากฐานมาจากภาษา Basic ซึ่งย่อมาจาก Beginner's All Purpose Symbolic Instruction หมายถึง "ชุดคำสั่งหรือภาษาคอมพิวเตอร์สำหรับผู้เริ่มต้น"

ภาษาคอมพิวเตอร์ (Programming Language) คือรูปแบบของชุดคำสั่งต่างๆ ที่เราใช้ติดต่อกับเครื่องคอมพิวเตอร์เพื่อให้ทำงานตามที่เราร้องขอ โดยภาษาคอมพิวเตอร์นี้จะประกอบไปด้วยไวยากรณ์และคำสั่งต่างๆ ซึ่งเมื่อเราเขียนโปรแกรมด้วยภาษาเหล่านี้แล้วคำสั่งต่างๆ จะถูกแปลงให้เป็นรูปแบบที่เครื่องคอมพิวเตอร์เข้าใจได้

เนื่องจากภาษาในขณะนั้นเป็นแบบ Text Mode คือเป็นตัวอักษรล้วนๆ ไม่มีภาพกราฟฟิก ประกอบกับมีการกำเนิดขึ้นของระบบปฏิบัติการ Windows ซึ่งเป็นภาพกราฟฟิกสวย ภาษาที่ใช้ในการเขียนระบบปฏิบัติการ Windows คือ Visual Basic นั่นเอง

Visual Basic เวอร์ชันแรกคือเวอร์ชัน 1.0 ออกมาใช้งานในปี 1991 โดยในช่วงแรกยังไม่มีความสามารถต่างจากภาษา QBASIC มากนัก แต่จะเน้นเรื่องเครื่องมือที่ช่วยในการเขียนโปรแกรมบนวินโดวส์ ซึ่งปรากฏว่า Visual Basic ได้รับความนิยมเป็นอย่างมาก ไมโครซอฟต์จึงพัฒนา Visual Basic ให้ดีขึ้นเรื่อยๆ ทั้งในด้านประสิทธิภาพ ความสามารถ และเครื่องมือต่างๆ เช่น เครื่องมือตรวจสอบแก้ไขโปรแกรม (debugger) สภาพแวดล้อมของการพัฒนาโปรแกรม การเขียนโปรแกรมแบบหลายวินโดวส์ย่อย (MDI) และอื่นๆ

สำหรับ Visual Basic เวอร์ชัน 6.0 ซึ่งออกมาในปี 1998 ได้เพิ่มความสามารถในการเขียนโปรแกรมติดต่อกับระบบเครือข่ายอินเทอร์เน็ต การเชื่อมต่อกับฐานข้อมูล รวมทั้งปรับปรุงเครื่องมือและการเขียนโปรแกรมเชิงวัตถุ (Object Programming) และมีการเพิ่มเครื่องมือต่างๆ อีกมากมาย

2.2.2 สาเหตุที่เลือกใช้ Visual Basic ในการพัฒนาโปรแกรม

เนื่องจาก Visual Basic มีข้อดีหลายประการดังนี้

1. ง่ายต่อการเรียนรู้และเหมาะสำหรับผู้เริ่มต้น ทั้งในเรื่องไวยากรณ์ของภาษาและเครื่องมือที่ใช้งาน
2. ได้รับความนิยมของตัวภาษา โดยอาจกล่าวได้ว่าภาษา Basic เป็นภาษาที่มีคนเรียนรู้และใช้งานมากที่สุดในประวัติศาสตร์ของคอมพิวเตอร์
3. มีการพัฒนาอย่างต่อเนื่อง การปรับปรุงประสิทธิภาพในด้านของตัวภาษาและความเร็วของการประมวลผล และในเรื่องของความสามารถใหม่ๆ เช่น การติดต่อกับระบบฐานข้อมูล การเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต
4. ผู้พัฒนาสำคัญของ Visual Basic คือบริษัท ไมโครซอฟต์ซึ่งจัดว่าเป็นยักษ์ใหญ่ของวงการคอมพิวเตอร์ในปัจจุบัน เราจึงมั่นใจได้ว่า Visual Basic จะยังมีการพัฒนา ปรับปรุงและคงใช้งานอยู่ไปอีกนาน

2.3 ระบบฐานข้อมูล [7]

ระบบฐานข้อมูล = ฐานข้อมูล + ระบบจัดการฐานข้อมูล (DBMS)

2.3.1 ความหมายของฐานข้อมูล

ฐานข้อมูล คือการจัดเก็บข้อมูลที่สัมพันธ์กันอย่างเป็นระเบียบ ซึ่งเราสามารถนำไปทำการประมวลผลได้ เช่น การเพิ่มข้อมูล การสร้างรายงานเกี่ยวกับข้อมูล และการแสดงผลข้อมูล เป็นต้น

2.3.2 ความหมายและหน้าที่ของระบบจัดการฐานข้อมูล

ระบบจัดการฐานข้อมูล (Data Base Management System) คือ ซอฟต์แวร์ที่ใช้ในการจัดการข้อมูล ให้สามารถใช้ร่วมกันจากความต้องการใช้งานของผู้ใช้หลายคนได้อย่างมีประสิทธิภาพ

2.3.3 หน้าที่ของระบบจัดการฐานข้อมูล

หน้าที่ของระบบจัดการฐานข้อมูลมีดังนี้

1. ทำหน้าที่แปลงคำสั่งที่ใช้จัดการกับข้อมูลภายในฐานข้อมูลให้อยู่ในรูปแบบที่ฐานข้อมูลเข้าใจได้
2. ทำหน้าที่ในการนำคำสั่งต่างๆ ซึ่งได้รับการแปลแล้วไปส่งให้ฐานข้อมูลทำงาน เช่น การเรียกใช้ข้อมูล (Retrieve) การจัดเก็บข้อมูล (Update) การลบข้อมูล (Delete) การเพิ่มข้อมูล (Add) เป็นต้น
3. ทำหน้าที่ป้องกันความเสียหายที่จะเกิดขึ้นกับข้อมูลภายในฐานข้อมูล โดยจะตรวจสอบว่าคำสั่งใดที่สามารถตอบสนองได้ และคำสั่งใดที่ต้องปฏิเสธไม่ปฏิบัติ
4. ทำหน้าที่รักษาความสัมพันธ์ของข้อมูลภายในฐานข้อมูลให้มีความถูกต้องอยู่เสมอ

2.3.4 ข้อดีและประโยชน์ของการใช้ระบบฐานข้อมูล

ข้อดีและประโยชน์ของการใช้ระบบฐานข้อมูลมีดังนี้

1. สามารถลดความซ้ำซ้อนของการเก็บข้อมูลได้
2. สามารถหลีกเลี่ยงความขัดแย้งข้อมูลได้
3. ให้การสนับสนุนการเก็บอย่างถาวรของทั้ง โปรแกรมและโครงสร้างข้อมูล
4. ให้การสนับสนุนการใช้งานในแ่งมุมที่ต่างกันสำหรับผู้ใช้ที่ต่างกัน
5. ป้องกันการใช้งานจากบุคคลที่ไม่มีสิทธิ์
6. สามารถรักษาความถูกต้องให้กับข้อมูลได้
7. สามารถกำหนดระดับความปลอดภัยให้กับข้อมูลได้
8. ทำให้ข้อมูลเป็นอิสระจากโปรแกรม คือ เมื่อต้องการแก้ไขข้อมูลจะไม่ส่งผลกระทบต่อตัวโปรแกรมเลย

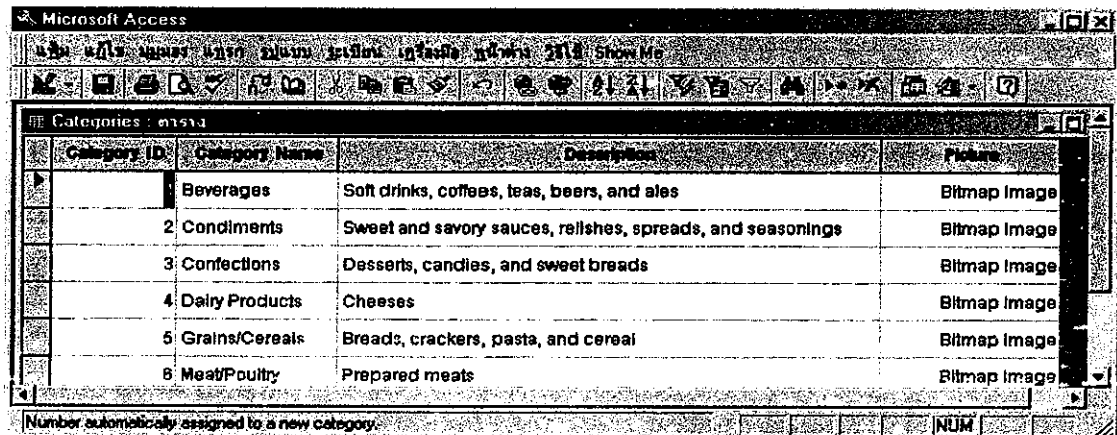
2.3.5 การออกแบบฐานข้อมูลโดยใช้ตารางแสดงความสัมพันธ์

ฐานข้อมูลนั้นนั้นมีหลายรูปแบบ แต่ที่จะนำมาใช้ก็คือ Relation Model ซึ่งข้อมูลจะอยู่ในรูปของตารางต่าง ๆ และตารางเหล่านี้จะมีการเชื่อมต่อสัมพันธ์กันเพื่อประโยชน์ในการจัดเก็บ การแก้ไขข้อมูล และการค้นหาข้อมูล คำศัพท์ที่สำคัญและเกี่ยวข้องมีดังนี้

ตาราง

ตาราง หรือ Table คือ โครงสร้างที่เราใช้เก็บข้อมูลจริงใน Database โดยเก็บในรูปแบบของตารางย่อย ๆ ที่มีความสัมพันธ์กันในแต่ละแถว (row) ในตารางซึ่งเรียกว่า Record ส่วนแต่ละคอลัมน์ (column) ในแต่ละแถวจะเรียกว่า ฟیلด์ (field) โดยแต่ละเรคคอร์ดคือข้อมูลหนึ่งชุดที่มีโครงสร้างซ้ำกัน และแต่ละฟیلด์ก็คือข้อมูลตัวเดียวกันของแต่ละเรคคอร์ด

รูปที่ 2.1 ตารางแสดงส่วนประกอบของ Record



Category ID	Category Name	Description	Picture
1	Beverages	Soft drinks, coffees, teas, beers, and ales	Bitmap Image
2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	Bitmap Image
3	Confections	Desserts, candies, and sweet breads	Bitmap Image
4	Dairy Products	Cheeses	Bitmap Image
5	Grains/Cereals	Breads, crackers, pasta, and cereal	Bitmap Image
6	Meat/Poultry	Prepared meats	Bitmap Image

ความสัมพันธ์และคีย์

ความสัมพันธ์ (Relation) เป็นหัวใจหลักของระบบฐานข้อมูลแบบ Relational Model คือ เป็นการเชื่อมโยงข้อมูลระหว่างตารางต่าง ๆ เข้าด้วยกัน โดยอาศัยฟیلด์ที่มีค่าตรงกันในแต่ละตารางเป็นตัวเชื่อมหรือที่เรียกว่า คีย์ (Key) โดยปกติแต่ละตารางจะมีคีย์หรือข้อมูลที่จะเป็นกุญแจสำหรับการเข้าถึงหรือบ่งชี้ว่าเรคคอร์ดใดเป็นเรคคอร์ดใด หรือต่างจากเรคคอร์ดอื่นอย่างไร เพื่อความถูกต้องในการอ้างอิง

- **Primary Key** เป็นคีย์หลักที่ใช้ในการจัดแยกแยะข้อมูลแต่ละเรคคอร์ดในตารางนั้นออกจากกัน โดย Primary Key นี้จะต้องมีอยู่ในเรคคอร์ดทุกเรคคอร์ด จะปล่อยว่างไม่ได้และต้องไม่ซ้ำกัน อีกด้วยคือจะต้องเป็นหนึ่งเดียว (Unique)

- **Foreign Key** เป็นฟیلด์ที่เก็บอยู่หลายตารางจึงสามารถใช้เป็นคีย์ในการเชื่อมโยงข้อมูลระหว่างตารางเข้าด้วยกัน คือ เป็นการเชื่อมโยงข้อมูลระหว่างตาราง 2 ตาราง เพื่อทำการอ้างอิงถึงข้อมูลตัวเดียวกัน

2.3.6 ชนิดของความสัมพันธ์ระหว่างตาราง

ความสัมพันธ์ของข้อมูลระหว่างตาราง สามารถแยกออกได้เป็น 3 ชนิด คือ

1. One-to-One หมายถึงข้อมูล 1 เร็คคอร์ดที่อยู่ในตารางหนึ่ง มีความสัมพันธ์หรือเชื่อมโยงกับอีกตารางหนึ่ง โดยที่สามารถอ้างอิงได้เพียง 1 เร็คคอร์ดเท่านั้น เช่น คน 1 สามารถมีเลขประจำตัวบัตรประชาชนได้เพียง 1 หมายเลข ในทางกลับกันเลขประจำตัวบัตรประชาชนก็เป็นของคนเพียงคนเดียวเท่านั้น

2. One-to-Many หมายถึง ข้อมูล 1 เร็คคอร์ดที่อยู่ในตารางหนึ่ง มีความสัมพันธ์หรือเชื่อมโยงกับอีกตารางหนึ่ง โดยที่สามารถอ้างอิงได้มากกว่า 1 เร็คคอร์ด เช่น มหาวิทยาลัยมีนักศึกษาได้หลายคน ในทางกลับกันมีนักศึกษาหลายคนที่ศึกษาอยู่ในมหาวิทยาลัยเดียวกัน

3. Many-to-Many หมายถึง ข้อมูล 1 เร็คคอร์ดที่อยู่ในตารางหนึ่ง มีความสัมพันธ์หรือเชื่อมโยงกับอีกตารางหนึ่ง โดยที่สามารถอ้างอิงได้มากกว่า 1 เร็คคอร์ด เช่น นักศึกษา 1 คน สามารถลงทะเบียนได้หลายวิชา และในทางกลับกันแต่ละวิชาที่มีนักศึกษาหลายคน

2.4 การเชื่อมต่อฐานข้อมูลกับ Visual Basic [7]

ในการติดต่อกับฐานข้อมูลของโปรแกรมช่วยวินิจฉัยอาการเสียของคอมพิวเตอร์มีการติดต่อแบบต่าง ๆ 3 แบบ ที่นำมาใช้ในโปรแกรมมีดังนี้

1. ติดต่อโดยอาศัยคอนโทรลด้านฐานข้อมูล
2. ติดต่อโดยใช้ออบเจกต์ Data Access Object (DAO)
3. เข้าถึงข้อมูลโดยอาศัยเทคโนโลยี OLEDB

2.4.1 ติดต่อโดยอาศัยคอนโทรลด้านฐานข้อมูล

คอนโทรลด้านฐานข้อมูลที่ใช้คือ กลุ่มของคอนโทรลมาตรฐานที่ใช้โดยทั่วไป เช่น คอนโทรล TextBox, PictureBox, Image, ListBox, ComboBox เป็นต้น โดยใช้คอนโทรล Data (Data Control) เป็นตัวเชื่อมโยงระหว่างฐานข้อมูลกับกลุ่ม Bound Controls ซึ่งใน Properties ของกลุ่มนี้จะมีค่าที่ขึ้นต้นด้วยคำว่า Data เช่น DataField, DataSource เป็นต้น

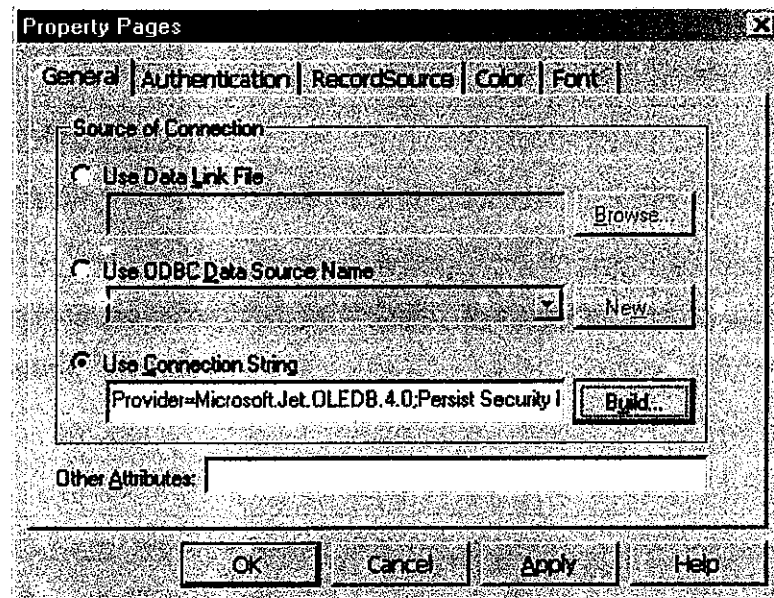
2.4.2 ติดต่อโดยใช้ออบเจกต์ Data Access Object (DAO)

แนวความคิดนี้มีลักษณะในการติดต่อหรือเข้าถึงฐานข้อมูลในฐานข้อมูล ผ่านองค์ประกอบต่าง ๆ ในฐานข้อมูล เช่น ฟีลด์(Field), เร็คคอร์ด(Record), ความสัมพันธ์ระหว่างตาราง(Relation) เป็นต้น โดยแต่ละองค์ประกอบจะประกอบไปด้วยออบเจกต์ และควบคุมออบเจกต์ต่าง ๆ ด้วยการเขียนโค้ด

2.4.3 เข้าถึงข้อมูลโดยอาศัยเทคโนโลยี OLEDB

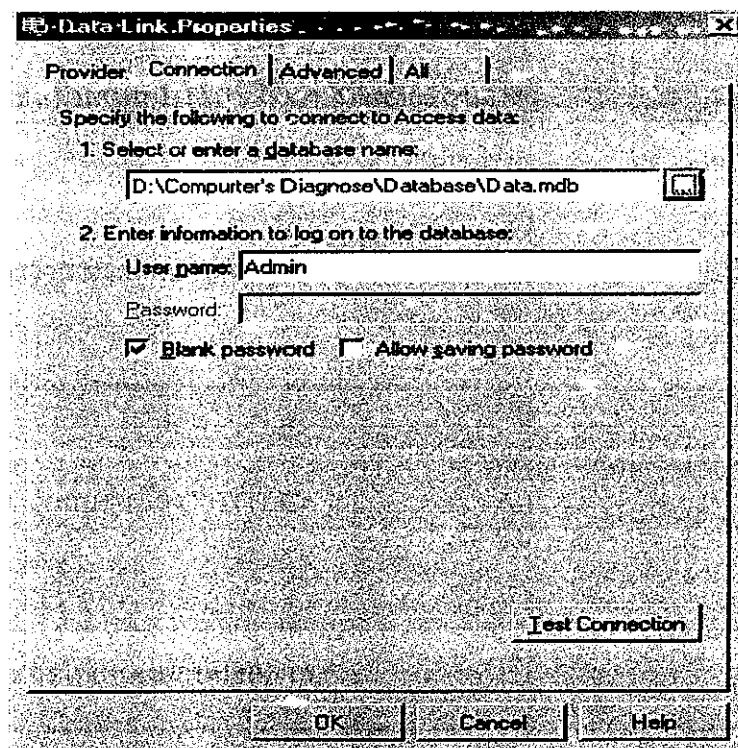
เป็นรูปแบบการติดต่อกับฐานข้อมูลผ่านทางกลุ่มออบเจกต์ในโมเดล ADO ซึ่งใช้ OLEDB Provider เป็นกลไกในการเข้าถึงข้อมูลในฐานข้อมูลแทน JET Engine โดยเปลี่ยนการติดต่อฐาน

ข้อมูลแบบเก่า ที่ต้องกำหนดชนิดของฐานข้อมูลมาเป็นการมองที่การเชื่อมต่อ (Connection) เข้ากับฐานข้อมูลดังตัวอย่างต่อไปนี้



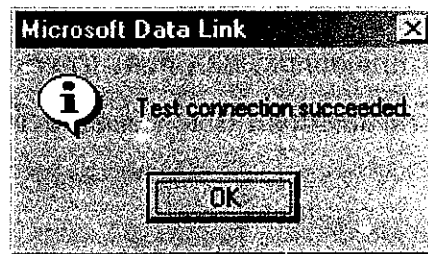
รูปที่ 2.2 แสดงการเชื่อมต่อแบบ OLEDB

คลิก Build... เพื่อทำการเปิดช่องทางการเชื่อมต่อ



รูปที่ 2.3 แสดงการเชื่อมต่อกับฐานข้อมูลชื่อ Data.mdb

จากนั้นกด Test Connection เพื่อทำการทดสอบว่าเส้นทางที่เชื่อมต่อที่จัดสร้างขึ้นนั้นใช้งานได้หรือไม่ ถ้าใช้งานได้ก็จะปรากฏข้อความดังรูป



รูปที่ 2.4 แสดงการเชื่อมต่อที่ล้มเหลว

2.5 ความรู้เบื้องต้นเกี่ยวกับ HTML [4]

2.5.1 เบื้องต้นเกี่ยวกับ HTML

HTML (Hypertext Markup Language) เป็นภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่งของคอมพิวเตอร์ เพื่อตอบสนองการแสดงผลบนจอภาพในระบบอินเทอร์เน็ต (Internet) ในลักษณะของเว็บเพจ ซึ่งสามารถแสดงผลได้ทั้งภาษาไทยและภาษาอังกฤษ รวมทั้งการแสดงรูปทรงกราฟฟิก ภาพนิ่ง ภาพเคลื่อนไหว เสียง รวมทั้งการเชื่อมโยงไปยังเว็บไซต์อื่นๆ ในระบบอินเทอร์เน็ต

การเขียนโปรแกรมด้วยภาษา HTML สามารถสร้างและแก้ไขไฟล์ข้อความด้วย Text Editor ต่างๆ ได้แก่ Notepad, SideKick, Word Processing ต่างๆ ได้

HTML สามารถทำงานได้บนเว็บเบราว์เซอร์ทุกตัวที่สนับสนุน HTML ได้แก่ Netscape รุ่นต่างๆ , Microsoft Internet Explorer, American Online เป็นต้น

2.5.2 โครงสร้างของภาษา HTML

รูปแบบการจัดวางที่แสดงต่อไปนี้เป็นรูปแบบมาตรฐาน ประกอบไปด้วย 4 คำสั่งหลัก ดังต่อไปนี้

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> ข้อความที่ต้องการนำเสนอบนแถบชื่อเรื่อง </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
    ข้อความหรือคำสั่งใดๆ ที่ต้องการให้แสดงผลบนจอภาพ
```

```
</BODY>
```

```
</HTML>
```

2.6 ความรู้เบื้องต้นเกี่ยวกับ ASP [6]

2.6.1 ที่มาและประวัติของ ASP

ASP ย่อมาจากคำว่า Active Server Pages เป็นภาษาสคริปต์ชนิดหนึ่ง ที่บริษัทไมโครซอฟต์คิดขึ้นมา เพื่อใช้สร้างเอกสารบนเว็บได้เช่นเดียวกับ HTML หรือภาษา CGI โดยที่ ASP ได้ถูกออกแบบให้สามารถประมวลผลได้ทั้งฝั่งไคลเอ็นต์ (Client Side) และฝั่งเซิร์ฟเวอร์ (Server Side) โดยที่ทั้งสองฝั่งจะต้องให้การสนับสนุนภาษา ASP ด้วย

2.6.2 โครงสร้างของภาษา ASP

อันที่จริงแล้วภาษา ASP ก็คือภาษา HTML ที่มีการทำงานร่วมกับภาษา VBScript โดยไฟล์ที่ทำการบันทึกจะต้องเป็นไฟล์ที่มีนามสกุล .asp ข้อสังเกตว่าเริ่มมีการใช้ภาษา ASP คือ สังเกตดูเครื่องหมาย <% แทนการเริ่มต้นของภาษา ASP และเครื่องหมาย %> แทนการสิ้นสุดการทำงานของภาษา ASP ดังตัวอย่าง

```
<html><body>
    <%
        คำสั่งหรือการเขียนโค้ดภาษา HTML เพื่อแสดงเอกสารข้อความ
    %>
</body></html>
```

2.6.3 การทดสอบตัวอย่างเว็บเพจบนพีซีเซิร์ฟเวอร์

เพื่อทำการทดสอบการทำงานของโปรแกรมว่าเป็นไปตามที่เราต้องการหรือไม่ และเพื่อทำการทดสอบหาข้อผิดพลาดก่อนที่จะนำเว็บเพจของเราอัปโหลดขึ้นใช้งานบนเครื่องเซิร์ฟเวอร์จริงๆ ซึ่งถ้ามีข้อผิดพลาดก็จะสามารถทำการแก้ไขได้ทันที

การทดสอบสามารถทำการทดสอบจากโปรแกรม PWS ที่เราต้องทำการติดตั้งบนเครื่องคอมพิวเตอร์ก่อน (PWS สำหรับเครื่องที่ใช้ระบบปฏิบัติการ Windows 98/Me) (IIS สำหรับเครื่องที่ใช้ระบบปฏิบัติการ Windows 2000/NT 4.0) สำหรับการติดตั้งและทดสอบการใช้งาน PWS จะทำการสาธิตการติดตั้งไว้ในภาคผนวก

2.7 การติดต่อฐานข้อมูลกับ ASP [6]

2.7.1 การติดต่อโดยตรงที่ Driver ของฐานข้อมูลที่ใช้ ODBC

มีรูปแบบการใช้งานดังนี้

```
ตัวแปรข้อความ = "Driver = {ชนิดของ Driver}; DBQ=ชื่อและที่อยู่ของฐานข้อมูล"
```

ตัวอย่างเช่น ในกรณีที่เราต้องการติดต่อฐานข้อมูลที่อยู่ที่ C:\v\tmp2\data.mdb และเราจะใช้ Driver ที่มีชื่อว่า Microsoft Access Driver (*.mdb) เราจะต้องสร้างตัวแปรข้อความ ได้ดังนี้

```
"Driver = {Microsoft Access Driver (*.mdb)}; DBQ= C:\v\tmp2\data.mdb" หรือ
"DBQ= C:\v\tmp2\data.mdb; Driver = {Microsoft Access Driver (*.mdb)}"
```

2.7.2 การติดต่อด้วย JET 3.51

เป็นเครื่องมือที่ใช้ติดต่อกับฐานข้อมูล Access โดยเฉพาะซึ่ง JET 3.51 ใช้ติดต่อกับ Access 97 และรุ่นต่ำกว่า มีรูปแบบการใช้งานดังนี้

```
ตัวแปรข้อความ= ""Data Source=ชื่อที่อยู่ของฐานข้อมูล
ข้อมูล;Provider=Microsoft.Jet.OLEDB.3.51"
```

จากตัวอย่างที่อยู่ของฐานข้อมูลในหัวข้อ 2.7.1 สามารถเปลี่ยนมาเป็นวิธีการติดต่อแบบ JET 3.51 ได้ดังนี้

```
Set Obj=Server.CreateObject("ADODB.Connection")
Obj.Open ""Data Source= C:\v\tmp2\data.mdb;Provider=_
Microsoft.Jet.OLEDB.3.51"
Set Rs=Server.CretateObject("ADODB.RecordSet")
Rs.Open "customer",Obj,2,3,-1
```

2.7.3 การติดต่อด้วย JET 4.0

เป็นเครื่องมือที่ใช้ติดต่อกับฐานข้อมูล Access รุ่นล่าสุดเพื่อใช้ติดต่อกับ Access 2000 และรุ่นต่ำกว่า แต่ไม่สามารถนำ Access รุ่นต่ำกว่า 2000 มาใช้กับ JET 4.0 ได้ มีรูปแบบการใช้งานดังนี้

```
ตัวแปรข้อความ= ""Data Source=ชื่อที่อยู่ของฐานข้อมูล
ข้อมูล;Provider=Microsoft.Jet.OLEDB.4.0"
```

จากตัวอย่างที่อยู่ของฐานข้อมูลในหัวข้อ 2.7.1 สามารถเปลี่ยนมาเป็นวิธีการติดต่อแบบ JET 4.0 ได้ดังนี้

```
Set Obj=Server.CreateObject("ADODB.Connection")
Obj.Open ""Data Source= C:\v\tmp2\data.mdb;Provider=_
Microsoft.Jet.OLEDB.4.0"
Set Rs=Server.CretateObject("ADODB.RecordSet")
Rs.Open "customer",Obj,2,3,-1
```

นอกจากนี้ยังมีการติดต่อแบบอื่นอีกมากมาย เนื่องจากการติดต่อแบบอื่นไม่ได้นำมาใช้ในการเขียนโปรแกรมจึงไม่ขอนำมาอธิบาย สำหรับผู้ที่สนใจอาจไปศึกษาต่อในหัวข้อการติดต่อฐานข้อมูลบนเซิร์ฟเวอร์ระยะไกลด้วย DNS และไม่ใช่ DNS ตามเอกสารอ้างอิงที่ระบุไว้ในหัวข้อ 2.7

2.8 เบื้องต้นกับภาษา SQL [8]

SQL (Structured Query Language) คือ ภาษามาตรฐานกลางที่ใช้สำหรับการจัดการกับฐานข้อมูล โดยเฉพาะฐานข้อมูลประเภท RDMB (Relation Database Management System) จะรู้จักภาษา SQL เป็นอย่างดี เราจะใช้ SQL เพื่อที่จัดการกับฐานข้อมูลจากตารางหลายๆ ตารางแสดงร่วมกันได้ เป็นต้น เป็นภาษาที่ใช้จัดการข้อมูลในฐานข้อมูลได้มีประสิทธิภาพมากที่สุด เราจะต้องใช้ ภาษา SQL เพื่อทำการ คิวรี (Query) ข้อมูลที่อยู่ในตาราง

2.8.1 โครงสร้างของภาษา SQL

โครงสร้างของภาษา SQLประกอบด้วย 3 ส่วนใหญ่ คือ

1. Data Definition Language (DDL) เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับการจัดการโครงสร้างของฐานข้อมูล เช่น การสร้างฐานข้อมูล, การปรับปรุงฐานข้อมูล เป็นต้น
2. Data Manipulation Language (DML) เป็นกลุ่มคำสั่งในภาษา SQL ที่จะใช้สำหรับจัดการ

ข้อมูลในฐานะข้อมูล เช่น การแสดงข้อมูลแบบมีเงื่อนไข , การลบข้อมูล , การเพิ่มข้อมูล และการแสดงข้อมูลที่ได้จากตารางหลายตาราง เป็นต้น

3. กลุ่มฟังก์ชันของ Aggregate Function เป็นฟังก์ชันพิเศษของภาษา SQL ที่ทำหน้าที่เฉพาะอย่างเช่น การหาผลรวมเร็คคอร์ด , ค่าสูงสุด , ค่าต่ำสุด เป็นต้น เป็นกลุ่มฟังก์ชันที่มีประโยชน์มาก เพราะช่วยลดภาระ

2.8.2 คำสั่งพื้นฐานของคำสั่งในกลุ่ม DML

คำสั่ง ในกลุ่มของ DML จะมีพื้นฐานอยู่ 4 คำสั่ง ดังนี้

- DELETE เป็นคำสั่งที่ใช้สำหรับการลบข้อมูลหรือลบเร็คคอร์ดใดๆ ในตาราง
- INSERT เป็นคำสั่งที่ใช้เพิ่มข้อมูลหรือเพิ่มเร็คคอร์ดใดๆ เข้าไปในตาราง
- SELECT เป็นคำสั่งที่ใช้สำหรับเลือกข้อมูลหรือแสดงเร็คคอร์ดใดๆ ที่ต้องการจากตาราง อาจจะมาจกตารางเดียว หรือหลายตารางก็ได้
- UPDATE ใช้สำหรับการแก้ไขข้อมูลหรือการแก้ไขเร็คคอร์ดใดๆ ในตารางตัวดำเนินการ (Operater)

2.8.3 ตารางตัวดำเนินการ

ตารางที่ 2.1 แสดงตัวดำเนินการ

ตัวดำเนินการ	ความหมาย
=	เท่ากับ
<>	ไม่เท่ากับ
<	น้อยกว่า
>	มากกว่า
<=	น้อยกว่าหรือเท่ากับ
>=	มากกว่าหรือเท่ากับ
Like	เป็นการเปรียบเทียบโดยใช้ตัวอักษรพิเศษเข้ามาช่วย

2.8.4 ตัวอย่างการใช้คำสั่ง SQL ที่ใช้ในโปรแกรมช่วยวินิจฉัยอาการเสียของคอมพิวเตอร์

คำสั่ง Select

ใช้สำหรับการดึงข้อมูล (Retrieve Data) ที่เราต้องการจากตารางที่ได้ระบุไว้ เป็นคำสั่งที่มีความยืดหยุ่นสูงมาก เพราะว่าเงื่อนไขในการนำข้อมูลมาจากตารางมีมากมายหลายลักษณะแต่มีรูปแบบอยู่ 2 ลักษณะคือ

```
SELECT * From ชื่อตาราง
```

หรือ

```
SELECT ฟิลด์ที่1 ,ฟิลด์ที่2 ...,ฟิลด์ที่ n From ชื่อตาราง WHERE เงื่อนไข
```

โดยที่ชื่อตาราง จะหมายถึง ชื่อตารางที่ต้องการดึงข้อมูลที่ต้องการดึงข้อมูล

ส่วนตัวแปร ฟิลด์ที่1,ฟิลด์ที่2 ...,ฟิลด์ n จะหมายถึง ชื่อฟิลด์ที่ต้องการดึงข้อมูล ถ้ามีมากกว่า 1 ฟิลด์ใช้เครื่องหมาย, คั่นระหว่างฟิลด์

สำหรับเงื่อนไข หมายถึง เงื่อนไขในการดึงข้อมูล อาจเป็นเงื่อนไขทางคณิตศาสตร์ หรือเป็นคำสั่ง SELECT ซ้อนอยู่ข้างในก็ได้

สำหรับการใช้งานของ SELECT ในโปรแกรมวินจี้ดยาการเสียของคอมพิวเตอร์เป็นดังนี้

```
Private Sub Command1_Click()
    Dim count As Integer
    Data1.RecordSource = "Select * from sub where question " & " Like " & "*" &
    Text1.Text & "*"
    Data1.Refresh
    'add new 1 to listbox
    With List1
        .Clear
        count = 0
        Do While Not Data1.Recordset.EOF
            .AddItem Data1.Recordset.Fields("question"), count
            count = count + 1
            Data1.Recordset.MoveNext
        Loop
        Data1.Refresh
    End With
End Sub
```

(ใน source code นี้จะอยู่ในส่วนของปุ่มของการค้นหาการเสียชีวิต)

```
Data1.RecordSource = "Select * from sub where question " & " Like " & "*" & Text1.Text  
& "*"
```

จากกรอบสี่เหลี่ยมนี้บอกรายละเอียดของการทำงานของภาษา SQL โดยที่คอนโทรลของตัวที่จัดการฐานข้อมูลซึ่งได้ทำเป็นตัวแปรคือ Data1 จะทำการดึงข้อมูลทุกฟิลด์จากรางที่ชื่อว่า sub โดยที่เงื่อนไขของการดึงข้อมูลคือ ซึ่งค่าในช่องค้นหา (Text1.text) จะเปรียบเทียบโดยตัวอักษรที่ทำการใส่เข้ามาในช่องค้นหากับค่าในของฟิลด์ที่ชื่อ question ถ้าพบค่าที่เหมือนกันระหว่าง Text1.text กับ ค่าในของฟิลด์ question ก็จะแสดงฟิลด์ทุกฟิลด์ออกมา