

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

ระบบเมนูอิเล็กทรอนิกส์
Electronic Menu System



H002469



วัน เดือน ปี..... 22 ก.พ. 2550
เลขทะเบียน..... 02469
เลขเรียกหนังสือ... ๑๗.๕๒๕๑๘ ๒540
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการศึกษาระณีพิเศษ
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 1 ปีการศึกษา 2540
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีสารสนเทศพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|------------------|---|
| ชื่อหัวข้อ | ระบบเมนูอิเล็กทรอนิกส์ |
| นักศึกษา | นายสมประสงค์ จิตราคณี |
| อาจารย์ที่ปรึกษา | อาจารย์ บรรจง ปิยะธำรง |
| ระดับการศึกษา | วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ |
| แขนงวิชา | การจัดการเทคโนโลยีสารสนเทศ |
| พ.ศ. | 2540 |

บทคัดย่อ

ระบบเมนูอิเล็กทรอนิกส์ เป็นระบบงานคอมพิวเตอร์ที่ได้รับการพัฒนามาเพื่อนำไปใช้สำหรับการให้บริการแก่ลูกค้าที่ต้องการสั่งอาหารและเครื่องดื่ม โดยเลือกรายการที่ต้องการสั่งจากหน้าจอคอมพิวเตอร์ซึ่งแสดงภาพกราฟิก เป็นรูปเมนูอาหารและเครื่องดื่มที่มีรายละเอียดของรายการและภาพประกอบคล้ายกับสมุดเมนูอาหาร ๑ ที่ใช้อยู่ในห้องอาหารทั่วไป ลักษณะการทำงานของโปรแกรมส่วนใหญ่ใช้ภาพลักษณะ เป็นส่วนติดต่อกับผู้ใช้เพื่อให้โปรแกรมมีคุณสมบัติเด่นที่ง่ายต่อการใช้ และมีความสวยงามไม่แพ้สมุดเมนูอาหารแบบเดิม

การพัฒนาโปรแกรมในส่วนของคุณลักษณะข้อมูลเป็นลักษณะฐานข้อมูลแบบรีเลชัน ที่ตัวข้อมูลของระบบถูกจัดเก็บในรูปแบบของตารางที่มีความสัมพันธ์ซึ่งกันและกันภายในฐานข้อมูล การเขียนโปรแกรมชนิดแนวการเขียนโปรแกรมเชิงวัตถุ ที่โมดูลต่าง ๆ ของระบบจะมีความสัมพันธ์กับเหตุการณ์ภายในระบบ ลักษณะการทำงานของโปรแกรมโดยรวมเป็นระบบงานแบบไคลเอนต์ / เซิร์ฟเวอร์ ที่มีการแบ่งหน้าที่ประมวลผลระหว่างคอมพิวเตอร์ที่ต่อเชื่อมกันเป็นระบบเครือข่าย

เครื่องมือพัฒนาระบบงานที่ใช้ในสร้างระบบงานนี้ ผู้พัฒนาเลือกใช้โปรแกรมสำเร็จรูปบริหารฐานข้อมูลขนาดกลางที่สามารถทำงานบนเครื่องไมโครคอมพิวเตอร์ทั่วไปได้ดี ด้วยเหตุผลของราคาซอฟต์แวร์ และ ความยากง่ายในการศึกษาการใช้งานเครื่องมือพัฒนาดังกล่าวภายในเวลาที่กำหนด

| | |
|-----------------------|---|
| Title | Electronic Menu |
| Student | Mr. Somprasong Chittrakanee |
| Advisor | Mr. Bunjong Piyathamrong |
| Level of Study | Master of Science in Information Technology |
| Major | Information Technology Management |
| Year | 1997 |

ABSTRACT

Electronic Menu is an application software which was developed to be used to serve the clients for taking an order of food & beverage items from the menu list in the restaurant industry. In the system, the clients are able to choose any particular food & beverage items they wish to order directly from a touched screen computer monitor. This program was designed using Graphical User Interface or GUI mostly. The Electronic Menu System will present the user with a food & beverage menu simulation which looks alike the real menu book in the old fashion menu style.

The program was designed using relational database concept for the entire system data manipulation. All data were kept in a table where relationship was established among associated tables in the same database. The system code was written base on Object Oriented Programming or OOP method. Each modules were associated with a particular events in the system, either by the user or system initialized or both. Client / Server Application idea was also applied in designing stage, as application was allowed to be segmented into tasks.

A popular PC-DBMS software package "MS Visual FoxPro" from Microsoft Corporation is the main development tools of Electronic Menu System.

กิตติกรรมประกาศ

โครงการฉบับนี้สำเร็จลงด้วยดีเนื่องมาจากความร่วมมือ และความช่วยเหลือจากบุคคลหลายฝ่าย โดยเฉพาะอาจารย์บรรจง ปิยะธำรง อาจารย์ที่ปรึกษาโครงการ ที่คอยให้คำแนะนำ และให้คำปรึกษา ทั้งทางด้านวิชาการ และด้านอื่น ๆ ตลอดของการทำโครงการนี้ ซึ่งต้องขอขอบพระคุณท่านอาจารย์เป็นอย่างยิ่ง ส่วนบุคคลสำคัญอีกสองท่านที่คอยให้กำลังใจ และช่วยเหลือข้าพเจ้ามาตลอด คือ บิดา มารดา อันเป็นที่รักและเคารพยิ่งของข้าพเจ้า ซึ่งในชีวิตของข้าพเจ้าไม่ว่าจะกระทำการใด ๆ ท่านทั้งสองได้เป็นที่พึ่งของข้าพเจ้าอยู่ตลอดเวลาทั้งกำลังกาย และกระตุ้นให้เกิดกำลังใจยามข้าพเจ้าเหนื่อยล้า หรือรู้สึกท้อถอย ทำให้ข้าพเจ้าสามารถฟันฝ่าอุปสรรคชีวิตนานับประการ จนมาถึง ณ จุดนี้ได้ พระคุณอันยิ่งใหญ่ ข้าพเจ้าจะขอระลึกถึงและจดจำตราบจนลมหายใจสุดท้าย

ขอขอบคุณ พี่ เพื่อน น้อง และเจ้าหน้าที่ ชาวลาดกระบัง ทุกคนที่คอยแนะนำ และช่วยเหลือข้าพเจ้ามาตลอดเวลาที่ทำโครงการนี้

ขอขอบคุณคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) ที่ให้ความช่วยเหลือทั้งทางด้านความรู้ สำหรับตำรา และเครื่องมือต่าง ๆ รวมทั้งเหล่าคณาจารย์ทั้งหลาย ที่ได้ช่วยประสิทธิ์ประสาทวิชาความรู้ทั้งหมดให้แก่ข้าพเจ้าตั้งแต่ปฐมวัยจนถึงปัจจุบัน ซึ่งได้นำมาใช้เพื่อความสำเร็จของโครงการชิ้นนี้

ขอขอบคุณบรรดาเหล่าสมาชิก ยูสเซอร์กรุ๊ปทั้งหลายในอินเทอร์เน็ต โดยเฉพาะอย่างยิ่ง ฟอกซ์ยูสเซอร์เน็ตกรุ๊ป (Fox User Net Group) ที่ให้ความช่วยเหลือทั้งชี้แนะเครื่องมือ (Tools/Utility) และเทคนิคการเขียนโปรแกรม รวมทั้งตอบคำถามไขข้อข้องใจต่าง ๆ ให้แก่ข้าพเจ้าตลอดมา

สุดท้ายขอขอบคุณ เจ้าของโฮมเพจ (Home Page) หรือ World Wide Web ต่าง ๆ ที่อนุญาตให้ข้าพเจ้าได้นำโปรแกรมย่อย (Subroutine) หรือ คลาส (Class) ต่าง ๆ ของท่านมาใช้ในโครงการนี้

สมประสงค์ จิตราคณี

พฤศจิกายน 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

| | |
|---|----|
| บทคัดย่อภาษาไทย | 1 |
| บทคัดย่อภาษาอังกฤษ | 2 |
| กิตติกรรมประกาศ | 3 |
| สารบัญ | 4 |
| สารบัญตาราง | 7 |
| สารบัญภาพ | 8 |
| บทที่ | |
| 1. บทนำ | 10 |
| 1.1 ความเป็นมา | 10 |
| 1.2 วัตถุประสงค์ | 11 |
| 1.3 ขอบเขตของโครงการ | 11 |
| 1.4 ขั้นตอนการศึกษาและส่วนประกอบของเอกสารโครงการ | 11 |
| 2. ทฤษฎีที่เกี่ยวข้อง | 13 |
| 2.1 ฐานข้อมูลแบบรีเลชันแนล (Relational Database Theory) | 13 |
| 2.1.1. ประวัติความเป็นมาของการจัดการฐานข้อมูล | 13 |
| 2.1.2. ประโยชน์ของการประมวลผลด้วยฐานข้อมูล | 14 |
| 2.1.3. สถาปัตยกรรมของระบบฐานข้อมูล | 17 |
| 2.1.4. ระบบจัดการฐานข้อมูล (Database Management System : DBMS) | 20 |
| 2.1.5. หน้าที่ของระบบจัดการฐานข้อมูล | 21 |
| 2.1.6. โมเดลเชิงสัมพันธ์ (Relational Model) | 21 |
| 2.1.7. โครงสร้างข้อมูล (Data Structure) | 22 |
| 2.1.8. การจัดเก็บข้อมูลและความสัมพันธ์ของข้อมูลในฐานข้อมูล | 24 |
| 2.2 ภาษา SQL (Structural Query Language) | 25 |
| 2.2.1. ภาษาที่ใช้ติดต่อกับระบบจัดการฐานข้อมูล | 25 |
| 2.2.2. ประวัติความเป็นมาของภาษา SQL | 25 |
| 2.2.3. การใช้ภาษา SQL | 26 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

| บทที่ | หน้า |
|--|------|
| 2. ทฤษฎีที่เกี่ยวข้อง (ต่อ) | |
| 2.3 ระบบเครือข่าย (Networking) | 28 |
| 2.3.1. วิวัฒนาการของระบบเครือข่าย | 28 |
| 2.3.2. การใช้งานระบบเครือข่าย | 29 |
| 2.3.3. ศูนย์บริการไฟล์ (File Server) | 30 |
| 2.3.4. ข้อดีของระบบเครือข่าย | 30 |
| 2.4 ระบบงานแบบ ไคลเอนต์ / เซิร์ฟเวอร์ | 31 |
| 2.4.1. ระบบ ไคลเอนต์ / เซิร์ฟเวอร์ | 31 |
| 2.4.2. องค์ประกอบของระบบ ไคลเอนต์ / เซิร์ฟเวอร์ | 31 |
| 2.4.3. ประโยชน์ของการพัฒนาระบบงานแบบ ไคลเอนต์ / เซิร์ฟเวอร์ | 32 |
| 2.5 โปรแกรมเชิงวัตถุ (Object-Oriented Programming) | 36 |
| 2.5.1. ประวัติภาษาออบเจกต์โอเรียนเตด (Object-Oriented) | 36 |
| 2.5.2. การเขียนโปรแกรมแบบ โปรซีเยอร์เรอร์โปรแกรมมิ่ง (Procedural Programming) | 37 |
| 2.5.3. การเขียนโปรแกรมแบบออบเจกต์โอเรียนเตด (Object-Oriented Programming) | 38 |
| 2.5.4. หลักการของ โอโอพี (Object-Oriented Programming) | 39 |
| 2.5.5. ศัพท์พื้นฐานในแนวคิดแบบ โอโอพี OOP | 40 |
| 2.5.6. ยูสเซอร์อินเตอร์เฟตแบบออบเจกต์โอเรียนเตด (Object-Oriented User Interface) | 41 |
| 2.6 เครื่องมือที่ใช้ในการพัฒนา (Development Tools) | 43 |
| 2.6.1. วิวอลฟอกซ์โปร (Visual FoxPro) | 43 |
| 2.6.2. วิวอลฟอกซ์โปรกับการจัดการฐานข้อมูล | 44 |
| 2.6.3. วิวอลฟอกซ์โปรกับภาษา SQL | 48 |
| 2.6.4. วิวอลฟอกซ์โปรกับระบบเครือข่าย | 49 |
| 2.6.5. วิวอลฟอกซ์โปรกับไคลเอนต์/เซิร์ฟเวอร์ | 49 |
| 2.6.6. วิวอลฟอกซ์โปรกับโปรแกรมเชิงวัตถุ | 50 |

สารบัญ (ต่อ)

| บทที่ | หน้า |
|---|------|
| 3. การออกแบบและการพัฒนาระบบงาน | 52 |
| 3.1 บทนำ | 52 |
| 3.2 ความสามารถพื้นฐานของระบบที่ต้องการ (Essential Requirements) | 52 |
| 3.3 การทำงานของระบบโดยรวม (System Narrative) | 53 |
| 3.4 สถาปัตยกรรมของระบบ | 55 |
| 3.5 การเขียนโปรแกรม | 65 |
| 4. การใช้งานระบบเมนูอิเล็กทรอนิกส์ | 73 |
| 4.1 ความต้องการของระบบ (System Requirements) | 73 |
| 4.2 การติดตั้งโปรแกรม (Installation) | 75 |
| 4.3 การใช้งาน | 77 |
| 5. การประเมินผล และ แนวทางในการปรับปรุงพัฒนาต่อไปในอนาคต | 84 |
| 5.1 การประเมินผล | 84 |
| 5.2 แนวทางในการพัฒนาต่อไปในอนาคต | 84 |
| 6. สรุป ผลที่ได้จากการพัฒนาโครงการ | 87 |
| บรรณานุกรม | 89 |
| ประวัติผู้เขียน | 90 |

สารบัญตาราง

| ตารางที่ | หน้า |
|--|------|
| 2.1 แสดง Entity หรือ ตารางที่มีความสัมพันธ์กัน | 24 |
| 2.2 สรุปคำสั่ง SQL | 27 |
| 3.1 แสดงโครงสร้างของไฟล์ Menu Item Type | 56 |
| 3.2 แสดงโครงสร้างของไฟล์ Menu Item | 57 |
| 3.3 แสดงโครงสร้างของไฟล์ Menu Item Picture | 58 |
| 3.4 แสดงโครงสร้างของไฟล์ Captain Order | 59 |
| 3.5 แสดงโครงสร้างของไฟล์ Promoted Item | 59 |



สารบัญภาพ

| ภาพที่ | หน้า |
|--|------|
| 2.1 แสดง ระดับของข้อมูล 3 ระดับ | 18 |
| 2.2 ตัวอย่างข้อมูล 3 ระดับ | 18 |
| 2.3 ระบบจัดการฐานข้อมูล | 20 |
| 2.4 แสดงความสัมพันธ์ข้อมูลพนักงาน | 22 |
| 2.5 แสดงลักษณะของการพัฒนาโปรแกรมใน 2 นิยาม | 38 |
| 2.6 แสดงรายละเอียดในพจนานุกรมข้อมูล (Data Dictionary) | 45 |
| 2.7 แสดงการสร้าง Referential Integrity | 46 |
| 2.8 แสดงการสร้าง Validation Rule | 47 |
| 2.9 หน้าต่าง Project Manager แสดงการสร้างยูเซอร์วิว | 48 |
| 2.10 ตัวอย่างการใช้ภาษา SQL ใน Visual FoxPro | 49 |
| 2.11 แสดง เครื่องมือช่วยการพัฒนาต่าง ๆ ที่มีใน Visual FoxPro | 51 |
| 3.1 แสดง Document Flow ของเอกสาร Captain Order | 54 |
| 3.2 แสดงความสัมพันธ์ของตารางต่าง ๆ ในฐานข้อมูล | 60 |
| 3.3 แสดงการแบ่งหน้าที่การประมวลผลของ Front End กับ Back End Computer | 61 |
| 3.4 สมุดเมนูจาก โปรแกรม | 63 |
| 3.5 แสดงโมดูลย่อยต่าง ๆ ของระบบ | 64 |
| 3.6 แสดงการออกแบบหน้าต่างฟอร์มและปุ่มกด “คลิก” | 66 |
| 3.7 แสดงหน้าต่างการเขียนโปรแกรมสำหรับปุ่ม “คลิก” | 66 |
| 3.8 Form Designer พร้อม Form Control Windows แสดงหน้าต่างการเพิ่ม Class Library | 68 |
| 3.9 คำอธิบายการใช้คำสั่ง Set Classlib เพื่อเพิ่ม Class Library ในโปรแกรม | 69 |
| 3.10 แสดงหน้าต่างการกำหนด Referential Integrity | 70 |
| 3.11 รายละเอียดคำสั่งในไฟล์ Store Procedure | 70 |
| 3.12 ตัวอย่างการใช้ SQL ในโมดูลหนึ่งของระบบ | 72 |
| 4.1 หน้าต่าง Run สำหรับเรียกโปรแกรม Setup | 75 |
| 4.2 หน้าต่างแรกของการติดตั้งระบบเมนูอิเล็กทรอนิกส์ | 76 |
| 4.3 หน้าต่าง เสร็จสิ้นกระบวนการติดตั้งระบบเมนูอิเล็กทรอนิกส์ | 76 |

สารบัญภาพ (ต่อ)

| ภาพที่ | หน้า |
|---|------|
| 4.4 หน้าต่าง การ Log in เพื่อเข้าสู่ระบบเมนูอิเล็กทรอนิกส์ | 77 |
| 4.5 แสดงหน้าจอแรกของระบบเมนูอิเล็กทรอนิกส์ | 78 |
| 4.6 แสดงรายการอาหารพิเศษประจำวัน ด้านขวาของจอภาพ | 79 |
| 4.7 แสดงรูปสมุดเมนูอาหาร และเครื่องดื่ม | 80 |
| 4.8 แสดง เมนูที่เปิดออกเป็น 2 หน้า | 80 |
| 4.9 แสดงรายละเอียดอาหารที่เปลี่ยนไปจากการซื้อรายการจากด้านขวา | 81 |
| 4.10 แสดงการเลือกจำนวนอาหารหรือเครื่องดื่มที่ต้องการสั่ง | 82 |
| 4.11 แสดงรายการอาหารและเครื่องดื่มที่ส่งไปแล้ว | 83 |
| 4.12 แสดงรายละเอียดค่าอาหารและเครื่องดื่มที่ต้องชำระ | 83 |

บทที่ 1

บทนำ

1.1 ความเป็นมา

การบริหารงานห้องอาหาร เป็นงานบริหารอีกแขนงหนึ่งที่มีความสำคัญและน่าสนใจไม่น้อยไปกว่างานบริหารในด้านอื่น โดยเฉพาะอย่างยิ่งในธุรกิจโรงแรมเพราะในแต่ละแห่งมักจะประกอบไปด้วยห้องอาหารที่ตั้งอยู่ภายในไม่น้อยกว่า 2-3 ที่ ซึ่งแน่นอนรายได้จากห้องอาหารเหล่านี้เป็นส่วนหนึ่งของรายได้หลักของธุรกิจโรงแรมนอกเหนือไปจากรายได้ค่าห้องพัก

อย่างไรก็ตามการบริหารห้องอาหารภายในโรงแรมเพื่อให้บรรลุวัตถุประสงค์ตามที่ตั้งไว้ย่อมต้องอาศัยปัจจัยสำคัญที่เกี่ยวข้องหลายประการ เช่น คุณภาพและราคาของอาหาร, การให้บริการของพนักงาน, รสชาติของอาหารและความหลากหลายของรายการอาหารที่มีให้เลือกในเมนู ความรวดเร็วของการให้บริการ, ความรวดเร็วของการรับคำสั่งจากลูกค้า, ความรวดเร็วของการเสิร์ฟ และความรวดเร็วของการเก็บเงิน ฯลฯ เหล่านี้ล้วนมีผลกระทบต่อความนิยมชมชอบของลูกค้าผู้จะมาใช้บริการกับห้องอาหารนั้น ๆ ทั้งสิ้น

นอกเหนือจากปัจจัยดังกล่าวข้างต้นแล้ว ความสำเร็จในการบริหารห้องอาหารยังต้องอาศัยข้อมูลอื่น ๆ ที่เกี่ยวข้องเพื่อให้ผู้จัดการห้องอาหารหรือผู้เกี่ยวข้อง นำไปใช้ตัดสินใจในการกำหนดแนวทางในการบริหาร เช่น ปรับปรุงความเร็วของการให้บริการ ปรับปรุงรายการอาหารในเมนูให้เหมาะสม ทั้งต่อความต้องการหรือรสนิยมของลูกค้าเอง และ กำไรต่อหน่วยที่ต้องการตามเป้าหมาย ซึ่งข้อมูลเหล่านี้จำเป็นต้องมีกระบวนการจัดเก็บและนำเสนอที่ถูกต้องเหมาะสม ทั้งความเหมาะสมในแง่ของเวลาและความเหมาะสมในเนื้อหาของข้อมูล

ปัจจุบันด้วยความก้าวหน้าของเทคโนโลยีคอมพิวเตอร์ โรงแรมส่วนใหญ่ได้เริ่มนำระบบคอมพิวเตอร์เข้ามาช่วยในการบริหารโรงแรม เพื่อเพิ่มประสิทธิภาพของการให้บริการและความทันสมัยตามยุคสมัย ทั้งในแผนกบริการส่วนหน้า (Front Office) และ แผนกบริการส่วนหลัง (Back Office) หรือส่วนสนับสนุนอื่น ในส่วนของห้องอาหารซึ่งถือว่าอยู่ในแผนกบริการส่วนหน้า ระบบคอมพิวเตอร์ที่นำมาใช้ส่วนใหญ่จะยังคงใช้เฉพาะส่วนของการเก็บเงิน หลังจากลูกค้าใช้บริการเรียบร้อยแล้วเท่านั้น ในส่วนของการจัดทำโปรแกรมคอมพิวเตอร์แบบเมนูอัตโนมัติที่ลูกค้าสามารถเลือกรายการอาหารได้โดยตรงจากหน้าจอคอมพิวเตอร์ แทนเมนูจริงที่ทำเป็นเล่มหรือสมุดแบบเดิมยังมีน้อย โครงการฉบับนี้จะนำเสนอแนวคิด/วิธีการและการพัฒนาระบบงานดังกล่าวเพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำไปใช้แทนระบบเดิม (ในอนาคต) โดยอาศัยทฤษฎีการออกแบบระบบงานแบบฐานข้อมูลสัมพันธ์ (Relational Database) และ ไคลเอนต์เซิร์ฟเวอร์ (Client / Server Application) ประกอบกับทฤษฎีอื่นที่เกี่ยวข้อง โดยเครื่องมือที่ใช้ในการพัฒนา (Development Tools) จะใช้ภาษาคำต่ำแบบบนเครื่องไมโครคอมพิวเตอร์ (PC-DBMS) ที่ชื่อว่า ไมโครซอฟต์วิซวลฟอกซ์โปร (Microsoft Visual FoxPro) เป็นหลักในการพัฒนา

1.2 วัตถุประสงค์

วัตถุประสงค์ของโครงการ มีดังนี้

1. เป็นการศึกษาค้นคว้างานวิชาการด้วยตนเอง ภายในช่วงเวลาที่กำหนด
2. ผสมผสานหลักการและทฤษฎีระบบสารสนเทศ เช่น ทฤษฎีฐานข้อมูล (Database Systems), ทฤษฎีการออกแบบระบบงาน (System Design), ทฤษฎีระบบเครือข่าย (Networking Systems), โปรแกรมเชิงวัตถุ (Object-Oriented Programming) ฯลฯ มาประยุกต์ใช้ในการสร้างหรือปรับปรุงระบบงานเดิมให้ดียิ่งขึ้น
3. ศึกษาเพื่อพิจารณาคัดเลือกใช้เครื่องมือในการพัฒนาระบบงาน (Development Tools) ประเภท PC-DBMS ที่มีอยู่หลากหลาย เช่น MS Visual FoxPro, MS Visual Basic, MS Access, Delphi ฯลฯ ที่เหมาะสมมาใช้ในการพัฒนาระบบงานที่ต้องการ
4. นำซอฟต์แวร์สำเร็จรูปประเภท PC-DBMS ที่ชื่อ Visual FoxPro มาใช้เพื่อพัฒนาระบบงานเมนูอิเล็กทรอนิกส์

1.3 ขอบเขตของโครงการ

โครงการฉบับนี้จะเริ่มตั้งแต่การออกแบบระบบ (System Design) และเขียนโปรแกรม (Coding) โดยอธิบายถึงทฤษฎีสารสนเทศต่าง ๆ ที่เกี่ยวข้อง ซึ่งประกอบด้วย ทฤษฎีฐานข้อมูลแบบรีเลชัน, ภาษาเอสคิวแอล, ระบบเครือข่าย, ระบบงานแบบไคลเอนต์เซิร์ฟเวอร์, โปรแกรมเชิงวัตถุ และเหตุผลที่นำวิซวลฟอกซ์โปรมาใช้เป็นหลักในการพัฒนารวมทั้งอธิบายขั้นตอน/วิธีการนำทฤษฎีต่าง ๆ มาประยุกต์ใช้กับวิซวลฟอกซ์โปร เพื่อสร้างเป็นระบบงานที่กำลังศึกษา

1.4 ขั้นตอนการศึกษาและส่วนประกอบของเอกสารโครงการ

โครงการนี้จะเริ่มต้นด้วยการศึกษาทฤษฎีสารสนเทศพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับการพัฒนาระบบงานเมนูอิเล็กทรอนิกส์ ซึ่งประกอบไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์โดยมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ภาษาเอสคิวแอล (Structural Query Language : SQL)
3. ระบบเครือข่าย (Networking System)
4. ระบบงานแบบไคลเอนต์เซิร์ฟเวอร์ (Client / Server Computing)
5. โปรแกรมเชิงวัตถุ (Object-Oriented Programming : OOP)

จากนั้นจะศึกษาเครื่องมือที่จะนำมาใช้ในการพัฒนาระบบงานนี้ ซึ่งได้แก่ ไมโครซอฟต์ วิวอลฟอกซ์โปร เวอร์ชัน 5.0 รายละเอียดจะกล่าวในบทที่ 2

สำหรับบทที่ 3 จะเป็นการนำความรู้ที่ได้จากบทที่ 2 มาประยุกต์ใช้งานร่วมกับเครื่องมือช่วยในการออกแบบระบบ (Computer Aided Software Engineering : CASE) อีกตัวหนึ่ง สำหรับโครงการนี้เลือกใช้ โปรแกรมอีซีเคสต์ (Easy CASE for Microsoft Windows) หลังจากได้โครงร่างของระบบงานจาก โปรแกรมอีซีเคสต์ ซึ่งได้แก่ แผนภูมิการไหลของข้อมูล (Data Flow Diagram), แผนภูมิความสัมพันธ์ของข้อมูลในระบบ (Entity Relationship Diagram : E/R Diagram) ฯลฯ แล้วจึงเริ่มลงมือพัฒนาระบบงาน (Coding) โดยใช้โปรแกรม “วิวอลฟอกซ์โปร” เป็นหลัก

ส่วนบทที่ 4 จะแสดงวิธีการใช้งานของระบบเมนูอิเล็กทรอนิกส์ที่ได้พัฒนาขึ้น

บทที่ 5 เป็นการประเมินผลระบบงานที่พัฒนาขึ้นมาโดยพิจารณาในแง่ การทำงานของระบบโดยรวม (Performance Evaluation) และการตอบสนองของระบบ (Response Time) รวมทั้งแนวทางในการปรับปรุงพัฒนาต่อไปในอนาคต

บทที่ 6 ซึ่งเป็นบทสุดท้าย จะเป็นบทสรุปผลที่ได้รับจากโครงการที่ได้ศึกษานี้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 ฐานข้อมูลแบบรีเลชันแนล (Relational Database Theory)

2.1.1 ประวัติความเป็นมาของการจัดการฐานข้อมูล ¹

ต้นกำเนิดของระบบฐานข้อมูลเกิดขึ้นจากโครงการอพอลโลของสหรัฐอเมริกา อันเป็นโครงการส่งมนุษย์อวกาศไปดวงจันทร์ เมื่อประมาณ 20 -30 ปีที่แล้ว เบื้องหลังการจัดการระบบข้อมูลในโครงการนี้ก็เกิดจากการว่าจ้างบริษัท ไอบีเอ็ม ให้พัฒนาระบบการดูแลข้อมูลขึ้นมาอันได้แก่ระบบที่เรียกว่า GUAM (Generalized Update Access Method) ซึ่งถือว่าเป็นต้นกำเนิดของระบบการจัดการฐานข้อมูล ดังนั้นในอีก 2 ปีถัดมา ไอบีเอ็ม จึงได้พัฒนาการจัดการข้อมูลขึ้นมาใหม่เพื่อการใช้งานในวงการธุรกิจทั่ว ๆ ไป อันได้แก่ระบบ DL/I (Data Language / I) ต่อจากนั้น ไอบีเอ็มก็ได้เสริมสร้าง DL/I เพิ่มเติมขึ้น และในที่สุดก็ได้มาซึ่งระบบ IMS (Information Management System) ซึ่งยังคงใช้กันมาจนถึงปัจจุบัน

นอกจากบริษัทไอบีเอ็มแล้ว ก็ยังมีบริษัท GE (General Electric) ที่มีส่วนในการพัฒนาระบบฐานข้อมูล อันได้แก่ระบบที่ชื่อว่า IDS (Integrated Data Store) โดยมี Charles Bachman เป็นหัวเรือใหญ่ในทีมนั้น (Charles Bachman นี้ นับเป็นอีกผู้หนึ่งที่มีชื่อเสียงในวงการดาต้าเบสอยู่มาก) และ IDS ก็ได้เริ่มใช้งานในช่วงปี พ.ศ. 2509 และเป็นต้นกำเนิดของระบบโคดาซิลล์ (CODASYL) หรือ โมเดลแบบเน็ตเวิร์คที่ยังนิยมใช้กันแพร่หลายจนถึงปัจจุบัน

ระบบโคดาซิลล์ (CODASYL) หรือที่รู้จักกันในนาม โมเดลแบบเน็ตเวิร์ค เกิดขึ้น ตั้งแต่ พ.ศ. 2517 โดยการฟอร์มตัวของคณะกรรมการชุดหนึ่งอันเป็นทีมเดียวกับชุดที่พัฒนาภาษาโคบอล และได้ร่างข้อกำหนดมาตรฐานสำหรับโคดาซิลล์ขึ้นสำเร็จเมื่อปี พ.ศ. 2515

ในช่วงเวลาใกล้เคียงกับที่โคดาซิลล์ฉบับมาตรฐาน ได้รับการบัญญัติขึ้น ดร. คอดด์ (E.F. Codd) ก็ได้เสนอผลงานทางวิชาการเกี่ยวกับโมเดลใหม่อีกโมเดลหนึ่ง อันได้แก่ โมเดลเชิงสัมพันธ์ (Relational Model) หลักทฤษฎีของดร. คอดด์ ได้รับความสนใจจากนักวิจัยและบริษัทผู้ค้าเครื่องคอมพิวเตอร์เป็นอย่างยิ่ง และผลงานที่มีความสำคัญยิ่งต่อวงการดาต้าเบสที่พัฒนาขึ้นตามแนวความคิดของ ดร.คอดด์ ก็ได้แก่ระบบ R ซึ่งเป็นระบบที่สร้างขึ้นโดยบริษัทไอบีเอ็ม แต่กว่าที่โมเดลเชิงสัมพันธ์จะได้ก้าวเข้าสู่วงการธุรกิจก็ต้องใช้เวลาถึง 10 ปี และระบบ R ก็เป็นเพียงระบบ

ต้นแบบที่ยังคงใช้อยู่เฉพาะในห้องวิจัยเท่านั้น โดยไอบีเอ็มได้สร้างระบบ DB2 ขึ้นมาแทน เพื่อนำออกสู่งานธุรกิจ

ในช่วง พ.ศ. 2525 เป็นต้นมา ถือเป็นยุคทองของระบบฐานข้อมูลที่ได้ก้าวเข้าสู่ตลาดคอมพิวเตอร์อย่างเต็มตัว จนกระทั่งในปัจจุบัน ก็ได้มีซอฟต์แวร์ที่เกิดขึ้นมาเกื้อหนุนการทำงานของระบบการจัดการฐานข้อมูลอีกมากมาย ไม่ว่าจะเป็นระบบพจนานุกรมข้อมูล (Data Dictionary) ซอฟต์แวร์ที่ช่วยในการออกแบบ และสร้างรายงาน และภาษาเรียกค้น เป็นต้น และการเจริญเติบโตของระบบฐานข้อมูลก็คงไม่หยุดนิ่ง พร้อม ๆ กับที่เทคโนโลยีคอมพิวเตอร์รุดหน้าไปอย่างรวดเร็ว เครื่องไม้เครื่องมือทางซอฟต์แวร์ที่จะช่วยในการประมวลผลข้อมูลที่จะเกิดขึ้นอยู่เรื่อย ๆ ไม่ว่าจะเป็นภาษารุ่นที่ 4 (Fourth Generation Language) หรือที่นับว่าใหม่ที่สุดในตอนนี้ก็ได้แก่ระบบ CASE (Computer Aided Software Engineering) และการผนวกเอาระบบค้ำค่าเบสเข้ามาใช้ร่วมกับระบบฐานรอบรู้ เป็นต้น

2.1.2 ประโยชน์ของการประมวลผลด้วยฐานข้อมูล ¹

ประโยชน์จากการใช้ฐานข้อมูลในการประมวลผลมีมากมาย พอสรุปได้ดังนี้

1. ลดความซ้ำซ้อนของข้อมูล
2. สามารถหลีกเลี่ยงความขัดแย้งของข้อมูลได้ในระดับหนึ่ง
3. สามารถใช้ข้อมูลร่วมกันได้
4. สามารถควบคุมความเป็นมาตรฐานได้
5. สามารถจัดหาระบบความปลอดภัยที่รัดกุมได้
6. สามารถควบคุมความคงสภาพของข้อมูลได้
7. สามารถสร้างสมดุลในความขัดแย้งของความต้องการได้
8. เกิดความเป็นอิสระของข้อมูล

1. ลดความซ้ำซ้อนของข้อมูล (Redundancy can be reduced)

การประมวลผลโดยใช้ไฟล์ธรรมดานั้นจำเป็นที่ผู้ใช้แต่ละกลุ่มจะต้องมีไฟล์ส่วนตัวเอาไว้ ดังนั้นจึงเกิดเหตุการณ์ที่ข้อมูลชนิดเดียวกันถูกเก็บไว้หลาย ๆ แห่ง หรือที่เราเรียกกันว่าความซ้ำซ้อน การนำข้อมูลทั้งหมดมาเก็บไว้ที่เดียวกันในฐานข้อมูลนี้เป็นการ ลด ความซ้ำซ้อนลงไปได้ คำว่า ลด ในที่นี้ไม่ได้หมายถึงการ ขจัด เพราะอาจมีงานบางประเภทที่เราอาจจะต้องเก็บข้อมูลชุดเดียวกันไว้มากกว่า 1 แห่ง อย่าง

ไรก็ตามการใช้ระบบฐานข้อมูลจะทำให้เราสามารถควบคุมการเกิดความซ้ำซ้อน เพราะ

ถึงแม้ว่าจะต้องเก็บข้อมูลชุดเดียวกันไว้มากกว่า 1 แห่ง โปรแกรมจัดการฐานข้อมูล (Database Management System : DBMS) ก็จะเป็นตัวที่ทราบอยู่ตลอดเวลาว่ามีความซ้ำซ้อนอยู่ที่ใดบ้าง

2. สามารถหลีกเลี่ยงความขัดแย้งของข้อมูลได้ระดับหนึ่ง (Inconsistency can be avoided....to some extent)

ดังได้กล่าวมาแล้วในข้อแรก การเก็บข้อมูลไว้หลาย ๆ แห่งอาจจะก่อให้เกิดปัญหาว่า การแก้ไขข้อมูลชุดเดียวกันนี้อาจทำไม่เหมือนกันในทุก ๆ แห่ง ทำให้เกิดปัญหาว่า ข้อมูลชุดเดียวกันอาจมีค่าในแต่ละแห่งไม่ตรงกัน ดังนั้นถ้าใช้ระบบฐานข้อมูลทำให้เราสามารถลดความซ้ำซ้อนลงไปได้ โดยมี DBMS เป็นตัวควบคุมดูแลว่าเมื่อเกิดการแก้ไขข้อมูลขึ้นเมื่อใด จะต้องแก้ไขให้เหมือนกันครบทุกแห่ง

3. สามารถใช้ข้อมูลร่วมกันได้ (The Data can be shared)

การใช้ข้อมูลร่วมกันได้นี้ ไม่ได้จำกัดอยู่เฉพาะโปรแกรมที่ใช้ข้อมูลอยู่ในปัจจุบันเท่านั้น แต่จะหมายถึงโปรแกรมประยุกต์ที่จะพัฒนาขึ้นมาใหม่ด้วย ที่สามารถจะใช้ข้อมูลที่มีอยู่ได้เลย โดยไม่จำเป็นต้องเพิ่มเติมแก้ไขโปรแกรมใหม่เข้าไปในระบบอีก

4. สามารถควบคุมความมาตรฐานได้ (Standards can be enforced)

จากการที่เรานำข้อมูลมาเก็บรวมกันไว้ในฐานข้อมูลเช่นนี้ ทำให้ผู้ที่ทำหน้าที่ควบคุมดูแลการใช้ระบบฐานข้อมูล (Data Base Administrator หรือ DBA) สามารถกำหนดมาตรฐานของข้อมูลขึ้นมาได้ เช่น ให้ใช้หน่วยมาตรการวัดที่เหมือนกัน หรือรูปแบบในการเขียนวันที่ให้เหมือนกัน เป็นต้น ซึ่งการที่ข้อมูลล้วนใช้มาตรฐานเดียวกันเช่นนี้ ทำให้การแลกเปลี่ยนข้อมูลในระหว่างระบบเป็นไปอย่างสะดวกและถูกต้อง

5. สามารถจัดหาระบบความปลอดภัยที่รัดกุมได้ (Security restrictions can be applied)

ระบบความปลอดภัยในที่นี้หมายถึง การป้องกันไม่ให้ผู้ใช้ที่ไม่มีสิทธิมาใช้ข้อมูลในระบบได้ เนื่องจาก DBA เป็นผู้ที่ควบคุมการใช้ข้อมูล เขาจึงสามารถกำหนดสิทธิการใช้ให้แก่ผู้ใช้งานใด ๆ ก็ได้ตามความเหมาะสม และผู้ใช้แต่ละคนก็อาจจะใช้ข้อมูลได้ในระดับที่ต่างกัน หรือพูดอีกนัยหนึ่งคือ ผู้ใช้แต่ละคนจะมองเห็นข้อมูลด้วยวิธีที่ต่างกัน โดยที่ถ้าหาก DBA ไม่ได้รวมข้อมูลใดไว้ในวิวของผู้ใช้แล้วผู้ใช้นั้นก็จะไม่มีสิทธิเรียกใช้ข้อมูลส่วนนั้น นอกจากนี้ DBA ยังสามารถกำหนดรหัสลับในการเรียกใช้ข้อมูลบางส่วนได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. สามารถควบคุมความคงสภาพของข้อมูลได้ (Integrity can be maintained)

นอกเหนือจากความขัดแย้งของข้อมูลดังกล่าวมา ซึ่งถือเป็นลักษณะหรือสาเหตุหนึ่งที่ทำให้ข้อมูลไม่มีความคงสภาพ กรณีของความขัดแย้งนี้จะเกิดขึ้นได้ก็เมื่อข้อมูลมีความซ้ำซ้อนเท่านั้น แต่ในอีกแง่หนึ่งของความคงสภาพ อาจเกิดขึ้นได้แม้จะไม่มีข้อมูลซ้ำซ้อน เช่น ข้อมูลอายุของบุคคลอาจจะมีค่า 200 แทนที่จะเป็น 20 ซึ่งความผิดพลาดแบบนี้เกิดขึ้นได้ง่าย ๆ จากความผิดพลาดในการพิมพ์ข้อมูลได้ ในลักษณะของความไม่ถูกต้อง เช่นนี้ ผู้ที่ออกแบบระบบฐานข้อมูลสามารถใส่กฎเกณฑ์เพื่อควบคุมความคงสภาพไว้เพื่อป้องกันความผิดพลาด เช่น ตามตัวอย่างนี้ก็อาจจะใส่กฎว่า ค่าของอายุจะต้องเป็นตัวเลขระหว่าง 18 ถึง 60 เป็นต้น ดังนั้น เมื่อมีการใส่ข้อมูลใหม่หรือแก้ไขข้อมูล DBMS ก็จะสามารถดูแลให้ข้อมูลดังกล่าวถูกต้องตามกฎเกณฑ์ที่ตั้งไว้

7. สามารถสร้างสมดุลในความขัดแย้งของความต้องการได้ (Conflicting requirements can be balanced)

การที่ผู้ใช้ทั้งหมดใช้ข้อมูลจากฐานข้อมูลร่วมกันเช่นนี้ ทำให้ DBA ทราบถึงความต้องการและความสำคัญของผู้ใช้งานทั้งหมด จึงสามารถกำหนดโครงสร้างฐานข้อมูลเพื่อให้บริการที่ดีที่สุดได้ เช่น เลือกเก็บข้อมูลที่จะต้องใช้บ่อย ๆ ไว้ในสื่อข้อมูลที่มีความเร็วเป็นพิเศษ เป็นต้น เป็นการสร้างสมดุลของความต้องการไม่ให้เกิดความขัดแย้งในหมู่ผู้ใช้ เพราะการออกแบบนั้นกระทำบนแนวทางที่มุ่งจะให้ประโยชน์ส่วนรวมสูงที่สุดอยู่แล้ว

8. เกิดความเป็นอิสระของข้อมูล (Data independence)

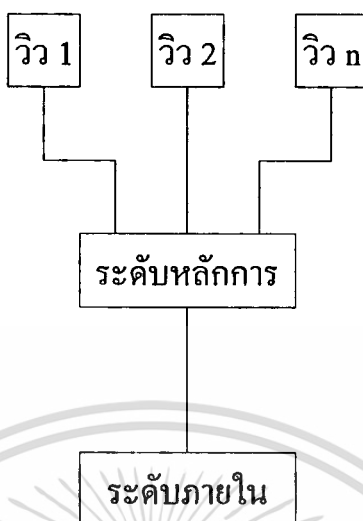
ความเป็นอิสระของข้อมูล ในที่นี้หมายถึง ความเป็นอิสระต่อการจัดเก็บและวิธีการเรียกใช้ข้อมูล เพราะข้อมูลที่ไม่เป็นอิสระนั้น เมื่อนำมาประยุกต์ใช้ยังมีความผูกพันอยู่กับวิธีการจัดเก็บและโปรแกรมที่เรียกใช้ข้อมูล ซึ่งในลักษณะการเขียนโปรแกรมประยุกต์บางประเภท อาจจำเป็นต้องใส่เทคนิคการจัดเก็บ และวิธีเรียกใช้ข้อมูลไว้ในตัวโปรแกรมเลย นั่นก็หมายความว่า ถ้าเกิดต้องมีการเปลี่ยนแปลงวิธีการจัดเก็บ หรือการเรียกใช้ข้อมูลแล้ว ผู้ใช้ก็จำเป็นที่จะต้องสร้างวิธีการประยุกต์ใช้ขึ้นมาใหม่ ซึ่งเป็นความไม่สะดวกอย่างยิ่ง และทำให้เราหมดโอกาสที่จะปรับปรุงโครงสร้างของข้อมูล เพื่อให้ใช้งานได้อย่างมีประสิทธิภาพมากขึ้น

2.1.8 สถาปัตยกรรมของระบบฐานข้อมูล²

โดยทั่วไประบบฐานข้อมูลมักจะถูกแบ่งออกเป็น 3 ระดับ คือ

- ระดับนอกสุด (External Level) เป็นระดับที่ใกล้กับผู้ใช้มากที่สุด เพราะในระดับนี้จะเป็นระดับข้อมูลที่จะมองเห็นได้จากการใช้งานของผู้ใช้แต่ละคน ในระดับนี้ภาพหรือ วิว (View) ที่ผู้ใช้เห็นจะเป็นเพียงส่วนที่เกี่ยวข้องกับผู้ใช้เท่านั้น ซึ่งระดับนี้จะเป็นส่วนที่กำหนดขอบเขตและรูปแบบที่ผู้ใช่มองเห็นว่าเป็นอย่างไร ในระบบฐานข้อมูลสัมพันธ์ ผู้ใช้จะเห็นข้อมูลในรูปของตารางเท่านั้น ภาพที่เห็นนี้บางครั้งก็เรียกว่า ภาพนอกสุด (External View) และผู้ใช้จะสามารถเรียกค้นหรือแก้ไข (หากทำได้) ก็เฉพาะในส่วนที่มองเห็นนี้เท่านั้น
- ระดับกลาง หรือ ระดับหลักการ (Conceptual Level) เป็นระดับที่อยู่ตรงกลางระหว่างระดับนอกสุดและระดับในสุด ซึ่งจะอธิบายต่อไป ภาพที่เห็นในระดับนี้เรียกว่า ภาพระดับกลาง (Conceptual View) ซึ่งเป็นระดับการมอง ตาราง (Entity) และความสัมพันธ์ระหว่าง Entity ทั้งหมดรวมทั้ง กฎเกณฑ์ต่าง ๆ เกี่ยวกับข้อมูลและ ผู้มีสิทธิจะใช้ ฯลฯ ข้อมูลในระดับนี้จะอยู่ในความสนใจของ DBA เพราะเขาจะต้องเป็นผู้ออกแบบและควบคุมการใช้งานข้อมูล
- ระดับในสุด (Internal Level) เป็นระดับที่ต่ำสุด อันได้แก่ระดับของการจัดเก็บข้อมูลจริง ๆ (Physical Storage) ซึ่งเนื้อหาในส่วนนี้เกี่ยวข้องกับโครงสร้างของข้อมูลที่จัดเก็บอยู่ในหน่วยเก็บข้อมูลจริง เช่น ตำแหน่ง (Address) ของข้อมูลที่เก็บอยู่ในหน่วยเก็บข้อมูล ลักษณะหรือโครงสร้างของอินเด็กซ์ไฟล์ เป็นแบบ เรียงลำดับ (Sequential) หรือ แบบสุ่ม (Random)

โดยในระบบฐานข้อมูลจะต้องมีบุคคลหรือกลุ่มบุคคลที่ทำหน้าที่เป็น ผู้บริหารฐานข้อมูล (DataBase Administrator : DBA) ซึ่งจะต้องคอยกำหนดวิธีการเก็บข้อมูล ออกแบบฐานข้อมูลและคอยดูแลรักษาระบบฐานข้อมูลให้เป็นไปตามที่ต้องการ



ภาพที่ 2.1 แสดง ระดับของข้อมูล 3 ระดับ

เพื่อที่จะให้เข้าใจและมองเห็นความแตกต่างของข้อมูลทั้ง 3 ระดับได้ดียิ่งขึ้น พิจารณาตัวอย่างข้างล่างนี้ ซึ่งเป็นการแสดงระดับหลักการของฐานข้อมูลสำหรับฝ่ายการพนักงานซึ่งมีวิธีการจัดเก็บข้อมูลจริง ๆ ดังที่แสดงไว้ในส่วนของระดับภายใน ส่วนในระดับภายนอกนั้นจะเป็นผู้ที่มาใช้ข้อมูล ในตัวอย่างนี้สมมติให้มีอยู่ 2 กลุ่ม กลุ่มแรกใช้ภาษาปาสคาลในการเรียกใช้ข้อมูล ส่วนกลุ่มที่สองใช้ภาษาโคบอล¹

| ระดับภายนอก (ปาสคาล) | ระดับภายนอก (โคบอล) |
|--|--|
| <pre> var empp : record emp# : string Sal : integer end; </pre> | <pre> 01 EMPC. 02 EMPNO PIC x(6). 02 DEPTNO PIC x(4). </pre> |
| ระดับหลักการ EMPLOYEE EMPLOYEE_NUMBER CHARACTER(6) DEPARTMENT_NUMBER CHARACTER(4) SALARY NUMERIC(5) | |
| ระดับภายใน STORED_EMP LENGTH = 18 PREFIX TYPE = BYTE(6), OFFSET = 0 EMP# TYPE = BYTE(6), OFFSET = 6, INDEX = EMPX DEPT# TYPE = BYTE(4), OFFSET = 12 PAY TYPE = FULLWORD, OFFSET = 16 | |

ภาพที่ 2.2 ตัวอย่างข้อมูล 3 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปข้างต้น อธิบายได้ดังนี้ :-

1. ในระดับหลักการ จะเห็นว่าเป็นการเก็บข้อมูลของ entity หรือ ตาราง ของพนักงาน (EMPLOYEE) ซึ่งมีคุณสมบัติหรือ attribute ประกอบด้วย รหัสพนักงาน (EMPLOYEE_NUMBER) รหัสแผนก (DEPARTMENT_NUMBER) และเงินเดือน (SALARY) ซึ่ง attribute แต่ละตัวมีลักษณะและขนาด ดังนี้ คือ

- รหัสพนักงาน เป็นข้อมูลประเภท ตัวอักษร ขนาด 6 ไบต์ (Bytes)
- รหัสแผนก เป็นข้อมูลประเภท ตัวอักษร ขนาด 4 ไบต์ (Bytes)
- เงินเดือน เป็นข้อมูลประเภท ตัวเลข ขนาด 5 ไบต์ (Bytes)

2. ระดับภายใน เป็นระดับจัดเก็บข้อมูลจริง ซึ่งระบบฐานข้อมูลใช้เนื้อที่ทั้งสิ้น 18 ไบต์ ในการเก็บข้อมูลของ entity/ตาราง โดยมีรายละเอียดประกอบด้วย

- ส่วนของ PREFIX มีไว้เพื่อเก็บข้อมูลสำหรับการควบคุมในการเข้าถึง (Access) ข้อมูล เช่น พอยน์เตอร์ เป็นต้น จะต้องใช้เนื้อที่ 6 ไบต์
- ส่วนของ EMP# ซึ่งใช้เก็บข้อมูล รหัสพนักงาน ใช้เนื้อที่อีก 6 ไบต์ และได้กำหนดรหัสพนักงานเป็นฟิลด์ดัชนี (index)
- ส่วนของ DEPT# ใช้เก็บข้อมูล รหัสแผนกของพนักงาน ใช้เนื้อที่ 4 ไบต์
- ส่วนของ PAY ใช้เก็บข้อมูลเกี่ยวกับเงินเดือนพนักงาน ใช้เนื้อที่ 2 ไบต์

รายละเอียดในส่วนที่กล่าวมาข้างต้นนี้ในแง่ของผู้ใช้แล้วไม่จำเป็นต้องทราบถึงรายละเอียดต่าง ๆ เหล่านี้เลย

3. ระดับภายนอก เป็นระดับการใช้งานของผู้ใช้ซึ่งในที่นี้มี 2 กลุ่ม โดยที่ผู้ใช้กลุ่มแรกจะมองฐานข้อมูลเสมือนหนึ่งว่ามีคุณลักษณะของพนักงานอยู่เพียง 2 attribute หรือ ฟิลด์ คือ รหัสพนักงานและเงินเดือน เท่านั้น ในขณะที่ผู้ใช้กลุ่มที่ 2 จะสนใจเฉพาะรหัสพนักงานและรหัสแผนกข้อสังเกต ผู้ใช้แต่ละกลุ่ม

- สามารถเขียนโปรแกรมประยุกต์เพื่อใช้ข้อมูลด้วยภาษาที่แตกต่างกัน ในที่นี้คือ ภาษาปาสคาล และ ภาษาโคบอล
- อ้างอิงถึงข้อมูลด้วยชื่อที่แตกต่างกันก็ได้ สังเกตการอ้างถึงข้อมูลรหัสพนักงานของแต่ละภาษา (ปาสคาล ใช้ emp# ในขณะที่ โคบอล ใช้ EMPNO)

ทั้งนี้จะเป็นหน้าที่ของ DBMS (จะกล่าวต่อไป) ที่จะทำการเชื่อม (map) ในแต่ละระดับคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

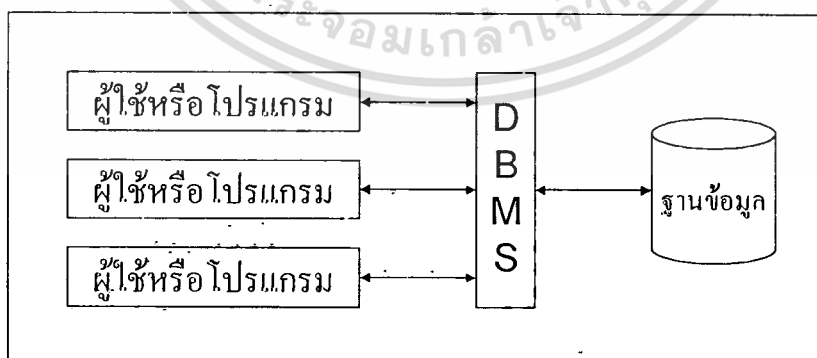
- การอ้างถึงข้อมูลใน ระดับภายนอก กับ ระดับหลักการ และ
- ยังเชื่อมข้อมูล ใน ระดับหลักการ เข้ากับข้อมูลที่จัดเก็บจริงใน ระดับภายใน อีกด้วย

จุดประสงค์ของการแบ่งระดับของข้อมูลออกเป็น 3 ระดับเช่นนี้ คือ

1. ผู้ใช้ไม่ต้องพะวงกับรายละเอียดต่าง ๆ ในการเก็บข้อมูล และไม่จำเป็นต้องรับทราบข้อมูลในส่วนอื่น ๆ ที่ไม่เกี่ยวข้อง หรือตนเองไม่ได้ใช้
2. ความเป็นอิสระของข้อมูล ดังได้กล่าวมาแล้วในตอนต้น กล่าวคือ ผู้ใช้ไม่ต้องคอยมาแก้ไขโปรแกรมใช้งานในทุก ๆ ครั้งที่เกิดการเปลี่ยนแปลงแก้ไขฐานข้อมูล ทั้งในระดับภายใน (Physical data independence) เช่น เปลี่ยนแปลงวิธีการจัดเก็บข้อมูลของฐานข้อมูลที่ใช้งานอยู่จากลำดับเชิงดัชนี (index sequential) เป็นแบบสุ่ม (direct access) หรือ การเปลี่ยนแปลงในระดับหลักการ (Logical data independence) เช่น อาจเพิ่ม entity ชนิดใหม่ลงไปในฐานข้อมูล หรือ เพิ่ม attribute เข้าไปใหม่ เป็นต้น

2.1.4 ระบบจัดการฐานข้อมูล (Database Management System : DBMS) ²

นอกจากการเก็บข้อมูลในรูปของฐานข้อมูลแล้ว ในระบบฐานข้อมูลจะมีซอฟต์แวร์ที่เรียกว่า ระบบจัดการฐานข้อมูล (Database Management System : DBMS) ซึ่งซอฟต์แวร์ตัวนี้จะเป็นตัวที่คั่นกลางระหว่างฐานข้อมูลกับผู้ใช้หรือโปรแกรมต่าง ๆ ระบบจัดการฐานข้อมูลนี้จะคอยทำหน้าที่ควบคุมดูแลฐานข้อมูลและอำนวยความสะดวกให้กับผู้ใช้เมื่อผู้ใช้ต้องการสร้างหรือเรียกคืนข้อมูล ซึ่งจะทำให้ผู้ใช้สามารถทำงานได้โดยไม่ต้องทราบถึงรายละเอียดในการเก็บข้อมูลจริง ๆ ว่าเป็นอย่างไร



ภาพที่ 2.3 ระบบจัดการฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 หน้าที่ของระบบจัดการฐานข้อมูล¹

1. ทำหน้าที่ติดต่อกับตัวจัดการระบบไฟล์ เนื่องจากการใช้งานส่วนใหญ่ของระบบฐานข้อมูล โดยเฉพาะข้อมูลขนาดใหญ่ เราไม่สามารถนำข้อมูลทั้งหมดนี้เข้ามาไว้ในหน่วยความจำหลักพร้อมกันได้ กล่าวคือข้อมูลทั้งหมดจะจัดเก็บอยู่ในดิสค์ และจะถูกนำมาสู่หน่วยความจำหลักเฉพาะส่วนที่ต้องการใช้งาน หน้าที่ในการค้นหาว่าข้อมูลที่เรากำลังต้องการนั้นเก็บอยู่ในตำแหน่งใดในดิสค์ ซึ่งเป็นฟังก์ชันการทำงานส่วนหนึ่งของระบบปฏิบัติการ (Operating System : OS) ที่เราเรียกว่า ตัวจัดการระบบไฟล์ (File Manager) ในกรณีนี้ DBMS จะทำหน้าที่ประสานการทำงานกับระบบปฏิบัติการ ในการจัดเก็บ เรียกใช้ หรือแก้ไขข้อมูล กับ ข้อมูลจริงที่เก็บอยู่ในหน่วยเก็บข้อมูลตามที่ผู้ใช้ต้องการ
2. การควบคุมความคงสภาพ DBMS จะทำหน้าที่ควบคุมค่าของข้อมูลในระบบให้อยู่ในกรอบที่ถูกต้องตามที่กำหนดไว้ล่วงหน้า เช่น อายุของพนักงานต้องมีค่าอยู่ระหว่าง 18 ถึง 20 ปี เป็นต้น
3. การควบคุมระบบความปลอดภัย ได้แก่ การป้องกันไม่ให้ผู้ที่มิได้รับอนุญาตเข้ามาใช้ หรือแก้ไขข้อมูลในส่วนที่ต้องการปกป้องไว้
4. การสร้างระบบสำรองและการฟื้นฟูสภาพ คือ หน้าที่ในการจัดทำข้อมูลสำรอง และเมื่อใดก็ตามที่มีปัญหาเกิดขึ้นไม่ว่าจะเป็นการขัดข้องของระบบไฟล์หรือเครื่องเกิดการเสียหาย DBMS จะต้องใช้ระบบข้อมูลสำรองในการฟื้นฟูสภาพ ให้ระบบข้อมูลกลับเข้าสู่สภาพที่ถูกต้องสมบูรณ์ได้
5. การควบคุมภาวะพร้อมกัน กรณีฐานข้อมูลมีการใช้หลาย ๆ คน DBMS จะทำหน้าที่ควบคุมการใช้ข้อมูลในภาวะพร้อมกัน (concurrency control) กล่าวคือ จะต้องควบคุมลำดับการทำงานให้เป็นไปอย่างถูกต้อง เช่น ระยะเวลาที่กำลังแก้ไขข้อมูลส่วนหนึ่ง ยังไม่เสร็จก็จะไม่อนุญาตให้ผู้อื่นเข้ามาเปลี่ยนแปลงแก้ไขข้อมูลนั้นได้อีก เป็นต้น

2.1.6 โมเดลเชิงสัมพันธ์ (Relational Model)¹

นับตั้งแต่ ดร.คอดด์ ได้เสนอผลงานวิจัยโมเดลของระบบฐานข้อมูลเชิงสัมพันธ์ ในปี พ.ศ. 2513 เป็นต้นมา นักวิชาการคอมพิวเตอร์ได้หันมาให้ความสนใจ และทุ่มเทความสามารถให้กับ การทำวิจัยเกี่ยวกับโมเดลนี้กันมากมาย โดยเริ่มพัฒนาระบบฐานข้อมูลแบบจำลองขึ้นมาก่อน ซึ่งระบบต้นแบบของโมเดลเชิงสัมพันธ์ที่มีชื่อเสียงที่สุดก็เห็นจะได้แก่ระบบ R ของบริษัทไอบีเอ็ม เอกสารขึ้นกระทั้งปี พ.ศ. 2523 จึงได้เริ่มมีการเผยแพร่ระบบฐานข้อมูลเชิงสัมพันธ์ออกสู่การใช้งานจริงใน

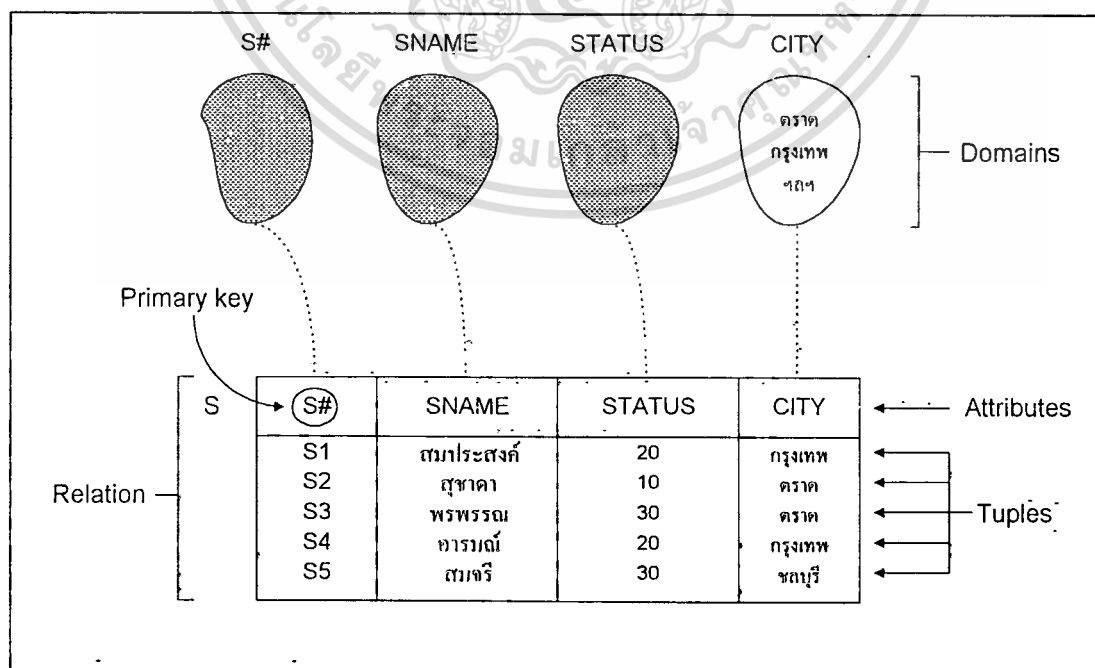
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงการธุรกิจ เริ่มต้นด้วยการนำไปใช้งานกับเครื่องระดับเมนเฟรมก่อน จนถึงปัจจุบันได้แพร่หลายลงมาใช้กับเครื่องระดับไมโครด้วย และในทุกวันนี้ก็คงเป็นที่ยอมรับกันแล้วว่า บรรดาผู้ใช้ระบบฐานข้อมูล โดยเฉพาะผู้ทำงานกับเครื่องระดับกลาง Mini Computer และระดับเล็ก Personal Computer จะมีความคุ้นเคยกับโมเดลเชิงสัมพันธ์นี้มากกว่าอีก 2 โมเดล (คือ Network Model และ Hierarchy Model)

นอกจากนั้น โมเดลเชิงสัมพันธ์ยังมีข้อได้เปรียบกว่าอีก 2 โมเดล พอสรุปได้ดังนี้

1. เป็นโมเดลที่สร้างความเข้าใจได้ง่ายในแง่มุมมองของผู้ใช้ และไม่มีใครจะมีความสลับซับซ้อนมากนัก
2. ระบบส่วนใหญ่ที่ใช้โมเดลแบบนี้มีเครื่องมือที่ช่วยให้ผู้ใช้สามารถปฏิบัติการยาก ๆ กับข้อมูลได้ด้วยคำสั่งง่าย ๆ
3. โมเดลแบบนี้มีเครื่องมือที่ช่วยให้ผู้ใช้สามารถค้นพบปัญหาที่เกิดขึ้นในการออกแบบระบบฐานข้อมูลได้โดยง่าย และยังง่ายในการแก้ไขการออกแบบที่ผิดพลาดนั้นด้วย
4. ส่วนของการจัดเก็บข้อมูลแบบกายภาพหรือการเก็บจริง นับว่ามีความแตกต่างจากข้อมูลแบบตรรก (ส่วนที่ผู้ใช้รับรู้) โดยสิ้นเชิง นับเป็นโมเดลที่มีความสอดคล้องกับหลักการของฐานข้อมูลในข้อที่จะให้ผู้ใช้ไม่ต้องพะวงกับรายละเอียดของการจัดเก็บ

2.1.7 โครงสร้างข้อมูล (Data Structure) ³



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูและบุคลากรทางการศึกษาเท่านั้น มิใช่ให้ผู้ใดนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นิยาม ฐานข้อมูลเชิงสัมพันธ์ ได้แก่การรวบรวมรีเลชันหรือตารางต่าง ๆ ที่มีความสัมพันธ์ (Relationship) ระหว่างกัน

เราเรียก รีเลชันในแต่ละแถวว่า ทัพเพิล (Tuple) และเรียกแต่ละคอลัมน์ว่า แอททริบิวต์ (Attribute) เปรียบเทียบศัพท์ต่าง ๆ ดังต่อไปนี้ ซึ่งจะหมายถึงสิ่งเดียวกัน

| | | | |
|-------------|---------|---------|-------------|
| ศัพท์เทคนิค | รีเลชัน | ทัพเพิล | แอตทริบิวต์ |
| ศัพท์ธรรมดา | ตาราง | แถว | คอลัมน์ |
| ศัพท์ทั่วไป | ไฟล์ | เรคอร์ด | ฟิลด์ |

2.1.8 การจัดเก็บข้อมูลและความสัมพันธ์ของข้อมูลในฐานข้อมูล ¹

ข้อมูลที่เก็บอยู่ในฐานข้อมูลใด ๆ ได้แก่ข้อมูลของเหล่า entity/ตารางต่าง ๆ การสร้างความสัมพันธ์ระหว่าง entity/ตาราง ในฐานข้อมูลเชิงสัมพันธ์กระทำได้โดยการกำหนดให้ entity/ตาราง ที่มีความสัมพันธ์กันมี attribute ที่เหมือนกัน และใช้ค่าของ attribute ในส่วนที่เหมือนกันนี้เป็นตัวระบุข้อมูลใน entity/ตาราง ที่มีความสัมพันธ์กัน เช่น รายการอาหาร และ ใบสั่งอาหาร เป็น entity/ตาราง 2 ตัว ที่มีความสัมพันธ์กัน โดยเราได้กำหนดให้มี attribute ที่เหมือนกันอยู่ในทั้ง 2 entity/ตาราง ในที่นี้ ได้แก่ รหัสอาหาร ดังนั้นข้อมูลในแถวใด ๆ ของตารางใบสั่งอาหารที่มีค่าของรหัสอาหารตรงกับข้อมูลในตารางรายการอาหาร ก็จะแสดงความสัมพันธ์ว่าใบสั่งอาหารใบนั้นสั่งอาหารชื่อรายการใดบ้าง

Entity/ Table : รายการอาหาร

| รหัสอาหาร | ชื่ออาหาร |
|-----------|-------------------|
| 001 | ข้าวผัดปู |
| 002 | ก๋วยเตี๋ยวคั่วไก่ |
| 003 | หมี่กะทิกุ้งสด |
| 004 | แกงเลียงกุ้งสด |
| 005 | หมูน้ำตก |

Entity/ Table : ใบสั่งอาหาร

| เลขที่โต๊ะ | รหัสอาหาร | จำนวน |
|------------|-----------|-------|
| 02 | 002 | 1 |
| 02 | 001 | 1 |
| 02 | 005 | 2 |
| 04 | 004 | 1 |
| 05 | 003 | 1 |

ตารางที่ 2.1 แสดง Entity หรือ ตารางที่มีความสัมพันธ์กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดของความสัมพันธ์ระหว่าง entity/ตาราง แบ่งออกเป็น 3 ชนิด ได้แก่ :-

- ความสัมพันธ์แบบ หนึ่ง ต่อ หนึ่ง (One to One Relationship 1 : 1)
 นิยาม : ถ้า entity A มีความสัมพันธ์แบบ หนึ่ง ต่อ หนึ่ง กับ entity B หมายความว่า ข้อมูลในแถวใด ๆ ของ entity A จะมีความสัมพันธ์กับข้อมูลใน entity B เพียงแถวเดียวเท่านั้น และในทางตรงกันข้าม ข้อมูลในแต่ละแถวของ entity B ก็จะมี ความสัมพันธ์กับข้อมูลเพียงแถวเดียวใน entity A
- ความสัมพันธ์แบบ หนึ่ง ต่อ กลุ่ม (One to Many Relationship 1 : m)
 นิยาม : ถ้า entity A มีความสัมพันธ์แบบ หนึ่ง ต่อ กลุ่ม กับ entity B หมายความว่า ข้อมูลใน 1 แถวจาก entity A สามารถมีความสัมพันธ์กับข้อมูลใน entity B ได้มากกว่า 1 แถว แต่ข้อมูลในแต่ละแถวของ entity B จะมีความสัมพันธ์กับ entity A ได้เพียง 1 แถวเท่านั้น
- ความสัมพันธ์แบบ กลุ่ม ต่อ กลุ่ม (Many to Many Relationship n : m)
 นิยาม : ถ้า entity A มีความสัมพันธ์แบบ กลุ่ม ต่อ กลุ่ม กับ entity B หมายความว่า ข้อมูลในแต่ละแถวของ entity A มีความสัมพันธ์กับข้อมูลใน entity B ได้มากกว่า 1 แถว และในขณะเดียวกัน ข้อมูลในแต่ละแถวของ entity B สามารถมีความสัมพันธ์กับ entity A ได้มากกว่า 1 แถว เช่นเดียวกัน

2.2 ภาษา SQL (Structural Query Language)

2.2.1 ภาษาที่ใช้ติดต่อกับระบบจัดการฐานข้อมูล

ภาษา SQL เป็นคำย่อมาจาก Structured Query Language เป็นภาษาที่ผู้ใช้สามารถใช้จะ ใช้สำหรับติดต่อกับระบบจัดการฐานข้อมูล (DBMS) ทั้งการเรียกดู เปลี่ยนแปลงแก้ไข เพิ่มเติม และ ลบข้อมูล ออกจากระบบฐานข้อมูล นอกจากนี้ยังสามารถใช้ในการสร้างและควบคุมการใช้ ฐานข้อมูลได้ด้วย

2.2.2 ประวัติความเป็นมาของภาษา SQL ¹

ภาษา SQL มีพื้นฐานมาจากทฤษฎีความสัมพันธ์แคลคูลัส (Relational Calculus) ซึ่งคิดค้น โดย ดร. คอดด์ เป็นภาษาที่เป็นที่นิยมใช้มากที่สุด ภาษานี้ถือกำเนิดขึ้นจากห้องปฏิบัติการวิจัยของ บริษัท ไอบีเอ็ม ที่เมือง เซ็นโจเซ่ (San Jose) มลรัฐแคลิฟอร์เนีย ในช่วงปี ค.ศ. 1970 โดยมี มิสเตอร์ แชมเบอร์ลิน (Mr. Chamberlin) เป็นผู้บัญญัตินิยามขึ้น และได้เริ่มพัฒนาระบบทดลองขึ้น บนระบบ R โดยในตอนแรกได้ใช้ชื่อว่า SEQUEL ต่อมาจึงได้เปลี่ยนเป็น SQL ในปี พ.ศ. 2523

เอกสารผลการทดลองเป็นที่น่าพอใจ ดังนั้นบริษัท ไอบีเอ็ม จึงให้การสนับสนุนอย่างจริงจังทำให้มีการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนาเรื่อยมา จนกระทั่งปี พ.ศ. 2525 บริษัท ไอบีเอ็ม จึงได้เปิดตัวระบบดังกล่าวสู่วงการธุรกิจ ภายใต้ชื่อ SQL / DS (Structured Query Language / Data System) เพื่อใช้งานกับเครื่องระดับ 370, 30xx, 43xx รวมทั้งระบบ 937x ซึ่งทั้งหมดนี้ทำงานภายใต้ระบบปฏิบัติการ DOS/VSE และ VM

นอกเหนือจาก SQL / DS แล้วระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ของไอบีเอ็มอีกตัวที่มีชื่อเสียง คือ DB2 ซึ่งก็มีรากฐานการพัฒนาโดยอาศัยต้นแบบจาก SQL ที่ทำงานภายใต้ระบบ R เช่นเดียวกัน โดยที่ DB2 เป็นระบบที่ทำงานกับระบบปฏิบัติการ MVS

2.2.3 การใช้ภาษา SQL

ภาษา SQL ไม่เพียงแต่ใช้สำหรับเรียกค้นข้อมูลเพียงอย่างเดียวแต่ยังใช้สำหรับงานด้านอื่นของฐานข้อมูลได้อีกด้วย เช่น

1. นิยามข้อมูล (Data Definition) นักบริหารฐานข้อมูล DBA สามารถใช้ความสามารถของภาษา SQL สำหรับกำหนดหรือสร้างฐานข้อมูลขึ้นมาใหม่ หรือใช้สำหรับสร้างวิว (View) สำหรับผู้ใช้แต่ละคนหรือแต่ละกลุ่มตามลักษณะการใช้งาน
2. จัดการข้อมูล (Data Manipulation) ผู้ใช้สามารถใช้ภาษา SQL สำหรับ เพิ่ม (Insert) เปลี่ยนแปลงแก้ไข (Update) และลบ (Delete) ข้อมูลจากฐานข้อมูล
3. ควบคุมการใช้งาน (Access Control) เป็นการใช้ภาษา SQL สำหรับกำหนดว่าผู้ใช้คนใดบ้างที่สามารถใช้ข้อมูลในฐานข้อมูลได้ เช่น การเรียกค้นข้อมูล หรือ ใครบ้างที่มีสิทธิในการเพิ่ม ลบ หรือแก้ไขข้อมูลได้

การเรียกใช้ภาษา SQL ทำได้ 3 วิธี ดังนี้¹

1. ออกคำสั่งแบบโต้ตอบโดยตรง (Interactive) กล่าวคือ ผู้ใช้สามารถพิมพ์ประโยคคำสั่งผ่านทางจอภาพโดยคำสั่งเหล่านี้จะถูกปฏิบัติ (Execute) ทันที
2. ส่งคำสั่งในลักษณะงานออฟไลน์ หรือ งานแบตช์ (Off Line / Batch Task) ลักษณะการใช้งานประเภทนี้เหมาะกับการสร้างรายงาน หรือประเภทของงานที่ไม่จำเป็นต้องการทราบผลโดยทันที
3. สอดแทรกหรือฝัง (Embed) ประโยคคำสั่ง SQL ไว้ในโปรแกรมประยุกต์ที่เขียนขึ้นสำหรับการใช้ข้อมูลในฐานข้อมูล ซึ่งโปรแกรมประยุกต์เหล่านี้อาจจะเขียนด้วยภาษา โคลบอล ฟอว์แทน ปาสคาล หรือ ภาษา ซี หรือในปัจจุบันแม้แต่ในโปรแกรมจัดการฐานข้อมูลบนเครื่องพีซี (PC-DBMS) ตระกูลวิซวล (Visual) ทั้งหมด เช่น Visual Basic, Visual FoxPro, Visual C++, Visual Dbase, Paradox ฯลฯ ก็สามารถใช้งานภาษา

¹ เอกสารนี้เป็นเอกสารที่ SQL ในลักษณะนี้ร่วมด้วยได้เป็นอย่างดี นั่น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|----------------------------------|---|
| คำสั่งสำหรับเรียกดูข้อมูล | |
| SELECT | เรียกดูข้อมูลจากตารางจาก 1 หรือมากกว่า 1 ตาราง |
| คำสั่งสำหรับการใช้ข้อมูล | |
| INSERT | แทรกข้อมูล 1 แถว หรือ มากกว่า 1 แถวลงในตาราง |
| UPDATE | แก้ไขข้อมูลใน 1 แถว หรือมากกว่า 1 แถว |
| DELETE | ลบข้อมูล 1 แถว หรือมากกว่า 1 แถว จากตาราง |
| คำสั่งสำหรับนิยามข้อมูล | |
| CREATE TABLE | นิยามโครงสร้างข้อมูลในตารางที่จะสร้างขึ้นใหม่ |
| DROP TABLE | ลบโครงสร้างของตารางข้อมูลออกจากระบบ |
| ALTER TABLE | เพิ่มคอลัมน์ใหม่ลงในตาราง |
| CREATE INDEX | สร้างคีย์ของตารางซึ่งประกอบด้วย 1 คอลัมน์ หรือมากกว่า |
| DROP INDEX | ยกเลิกคีย์นี้ออกจากระบบ |
| CREATE VIEW | นิยามวิวของผู้ใช้ |
| DROP VIEW | ลบวิวออกจากระบบ |

ตารางที่ 2.2 สรุปคำสั่ง SQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ระบบเครือข่าย (Networking) ⁴

เครือข่ายคอมพิวเตอร์ เป็นการเชื่อมต่อคอมพิวเตอร์หลาย ๆ เครื่องเข้าด้วยกันผ่านทางสายเคเบิล และยินยอมให้เครื่องเหล่านั้นติดต่อใช้ทรัพยากรต่าง ๆ ร่วมกัน ทรัพยากรที่กล่าวนี้ได้แก่ โปรแกรม ไฟล์ เครื่องพิมพ์ เครื่องพล็อตเตอร์ เครื่องเก็บสำรองข้อมูล โมเด็ม และแฟกซ์การ์ด เป็นต้น ระบบเครือข่ายจะช่วยให้ผู้ใช้งานบนเครื่องคอมพิวเตอร์หลาย ๆ คนทำการโอนย้ายเพื่อใช้โปรแกรมและข้อมูลร่วมกัน สามารถทำงานติดต่อร่วมกันอย่างมีประสิทธิภาพ ช่วยลดต้นทุนด้านฮาร์ดแวร์และซอฟต์แวร์ เพิ่มพูนความน่าเชื่อถือ และตัดรายจ่ายด้านสนับสนุนต่าง ๆ

เนื่องจากระบบเครือข่ายคอมพิวเตอร์ เป็นเทคโนโลยีที่ก้าวหน้าและเติบโตอย่างรวดเร็วในยุคปัจจุบัน เราจะพบเห็นระบบเครือข่ายกันมากขึ้นในแวดวงเทคโนโลยีสารสนเทศทั่วไปที่นิยมกันมากที่สุดคือ ระบบเครือข่ายท้องถิ่น หรือ แลน (LAN-Local Area Network) พบมากในระบบสำนักงานอัตโนมัติ ซึ่งในระบบแลนนี้นี้เป็นการเชื่อมต่อเครื่องพีซี ให้มีการใช้ข้อมูลข่าวสารร่วมกัน ใช้โปรแกรมประยุกต์และอุปกรณ์ต่อพ่วงต่าง ๆ ร่วมกัน

2.3.1 วิวัฒนาการของระบบเครือข่าย

ในระยะแรกวงการธุรกิจใช้เครื่องเมนเฟรมมาช่วยในการประมวลผลข้อมูล โดยจะมีขีดความสามารถในการคำนวณสูงมาก ในยุคปัจจุบันเครื่องเมนเฟรมก็ยังจำเป็นที่ให้กำลังการคำนวณสูงสำหรับงานประมวลผลของบริษัทหรือองค์กรขนาดใหญ่ เพราะเครื่องเมนเฟรมจะประกอบด้วยหน่วยประมวลผลหลายส่วน และสนับสนุนให้ผู้ใช้งานจำนวนมากมาใช้งานร่วมกันพร้อมกัน

การใช้เครื่องเมนเฟรมมีข้อเสียในเรื่องต้นทุนค่าใช้จ่ายแยกจากค่าบำรุงรักษาและค่าซอฟต์แวร์แล้ว ยังจำเป็นต้องมีบุคลากรประจำจำนวนมากมาดูแลและทำงานอยู่ภายในศูนย์คอมพิวเตอร์ ดังนั้นบริษัทต้องมีขนาดใหญ่ ๆ จึงจะสามารถแบกรับค่าใช้จ่ายของเครื่องเมนเฟรมไว้ได้

เครื่องมินิคอมพิวเตอร์ในยุคต่อมา ได้รับการยอมรับและนำมาใช้งานในทางธุรกิจกันมาก เพราะเครื่องมินิคอมพิวเตอร์จะมีขนาดที่เล็กลงกว่าเครื่องเมนเฟรมและใช้จำนวนบุคลากรสนับสนุนน้อยกว่า จึงเหมาะสำหรับบริษัทระดับกลาง แต่ก็ยังมีราคาสูงสำหรับบริษัทระดับเล็ก ๆ ทั่วไป

เมื่อเครื่องไมโครคอมพิวเตอร์ หรือที่นิยมเรียกกันว่า เครื่องพีซี ออกสู่ท้องตลาดก็ได้รับความนิยมเป็นที่ยอมรับกันทั่วไป เพราะเครื่องพีซีเรียนรู้และใช้งานง่ายและสามารถจัดหามาใช้ได้ในราคาไม่สูงจนเกินไปนัก ขีดความสามารถในการประมวลผลมีกำลังเฉพาะตัวระดับหนึ่ง แต่ก็ยังไม่มีการเชื่อมติดต่อระหว่างเครื่องพีซีหลาย ๆ เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าความเจริญก้าวหน้าของบริษัทหรือองค์กร ได้ขยายเติบโตขึ้นความจำเป็นในการประมวลผลข้อมูลก็ยิ่งมากขึ้น การเพิ่มจำนวนเครื่องพีซีให้มากขึ้นไม่ใช่หนทางแก้ปัญหา เพราะเครื่องพีซีเหล่านั้นไม่มีการเชื่อมติดต่อกัน ทางเลือกหนึ่งก็คือการหันไปใช้เครื่องเมนเฟรมหรือมินิคอมพิวเตอร์ที่ต้องลงทุนอีกมาก หนทางใหม่อีกทางหนึ่งคือ การสร้างระบบเครือข่ายให้เครื่องพีซีเหล่านั้นมาเชื่อมติดต่อใช้ทรัพยากรร่วมกัน ทำให้เกิดเป็นเทคโนโลยีใหม่ล่าสุดออกมา

2.3.2 การใช้งานระบบเครือข่าย

เครื่องพีซีปกติจะมีฮาร์ดดิสก์แยกประจำเฉพาะเครื่อง เมื่อซอฟต์แวร์มีการพัฒนาก้าวหน้ามากขึ้น ขนาดของซอฟต์แวร์จึงมีขนาดใหญ่โตมาก การเก็บโปรแกรมและข้อมูลภายในดิสก์ก็ต้องใช้ฮาร์ดดิสก์ขนาดความจุสูง หากเราสามารถนำโปรแกรมหรือข้อมูลที่แทบทุกเครื่องมีใช้งานอยู่มาบรรจบรวมกันในกล่องฮาร์ดดิสก์รวมที่อยู่ห่างไกลออกไป โดยมีการเชื่อมต่อของฮาร์ดดิสก์กับเครื่องอื่น ๆ ด้วยสายเคเบิล แล้วยินยอมให้เครื่องต่าง ๆ เรียกโปรแกรมและข้อมูลที่อยู่ในฮาร์ดดิสก์นั้น โดยเสมือนกับเป็นฮาร์ดดิสก์ของตัวเอง ผู้ใช้รายอื่นจากเครื่องอื่นต่างก็สามารถเรียกใช้โปรแกรมและข้อมูลจากฮาร์ดดิสก์เช่นกัน ลักษณะเช่นนี้ถือเป็นหลักพื้นฐานเบื้องต้นของระบบเครือข่าย ผู้ใช้ที่มีเครื่องต่อกันยังมีหน่วยประมวลผล หรือซีพียู หน่วยความจำ จอภาพและเป็นพิมพ์เป็นของตนเอง แต่มีการใช้ฮาร์ดดิสก์ร่วมกันและมองเห็นไฟล์ในฮาร์ดดิสก์ของเครื่องอื่นเสมือนเป็นไฟล์ของตัวเอง

ในระบบเครือข่ายโดยปกติจะมีคอมพิวเตอร์เครื่องหนึ่งเฉพาะเรียกว่า ศูนย์บริการไฟล์ หรือ ไฟล์เซิร์ฟเวอร์ (File Server) ซึ่งเครื่องนี้จะมีฮาร์ดดิสก์ขนาดใหญ่ทำงานด้วยความเร็วสูงกับมีหน่วยความจำจำนวนมาก เครื่องไฟล์เซิร์ฟเวอร์ นี้จะต่อพ่วงกับเครื่องพิมพ์ที่อาจมีเครื่องเดียวหรือหลาย ๆ เครื่องก็ได้ การติดต่อระหว่างเครื่องคอมพิวเตอร์เครื่องอื่นกับเครื่อง ไฟล์เซิร์ฟเวอร์ จะผ่านทางแลนดการ์ด (Local Area Network Card) และสายเคเบิลที่เชื่อมโยงติดต่อกัน แลนการ์ดเป็นอุปกรณ์ฮาร์ดแวร์ที่ใช้สำหรับโอนย้ายข้อมูลความเร็วสูงซึ่งจะติดประจำกับเครื่องคอมพิวเตอร์ทุกเครื่องที่อยู่ในเครือข่าย

ลักษณะการประมวลผล ณ เครื่องผู้ใช้งานหรือที่เรานิยมเรียกว่า สถานีงาน (Workstation) จะเป็นการสำเนาไฟล์โปรแกรมและข้อมูลจากศูนย์บริการไฟล์ ไปยังสถานีงานเหมือนการกระจายงาน ซึ่งลักษณะนี้จะแตกต่างกับการทำงานของเครื่องเมนเฟรม โดยในเครื่องเมนเฟรมหน่วยประมวลผลจะแบ่งเวลาเฉลี่ยการทำงานให้กับสถานีงานต่าง ๆ ถ้าหากมีจำนวนสถานีงานมาก การประมวลผลในภาพรวมจะช้าลง ประสิทธิภาพการทำงานก็จะน้อยลงตามไปด้วย แต่ในระบบเครือข่าย สถานีงานเพียงแต่โหลด (Load) ไฟล์โปรแกรมและข้อมูลมาประมวลผลที่สถานีเอง จากนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ระบบงานแบบ ไคลเอนต์ / เซิร์ฟเวอร์ ⁶

ภายในองค์กรขนาดใหญ่ นั้น ข้อมูลทั้งหมดที่โปรแกรมประยุกต์ต่าง ๆ จะเข้ามาเรียกใช้จะถูกเก็บไว้ในฐานข้อมูลส่วนกลางเพียงแห่งเดียว เมื่อข้อมูลมีจำนวนมากขึ้น โปรแกรมประยุกต์จะต้องทำงานซับซ้อนมากขึ้น มีผู้เรียกใช้ข้อมูลจำนวนมากขึ้นและบ่อยครั้งขึ้น การใช้เครื่องมินิคอมพิวเตอร์หรือเครื่องเมนเฟรมมาทำการประมวลผลโปรแกรมประยุกต์ส่วนกลางทั้งหมดสำหรับให้บริการแก่ผู้ใช้ทุก ๆ คนนั้น จะต้องเสียค่าใช้จ่ายสูงขึ้นไปเรื่อย ๆ โดยเฉพาะเมื่อเทียบเป็นอัตราส่วนของราคาต่อประสิทธิภาพของระบบไมโครคอมพิวเตอร์ ดังนั้นหนทางในการแบ่งเบาภาระการประมวลผลจากเครื่องมินิคอมพิวเตอร์หรือเครื่องเมนเฟรมมาสู่การประมวลผลของเครื่องไมโครคอมพิวเตอร์ที่มีราคาถูกกว่ามาก จึงได้รับความสนใจมากขึ้นเรื่อย ๆ การพัฒนาโปรแกรมประยุกต์และโปรแกรมสนับสนุนต่าง ๆ เพื่อให้เกิดการประมวลผลที่เครื่องของผู้ใช้เริ่มมีมากขึ้นพร้อมกับเทคโนโลยีของไมโครคอมพิวเตอร์ที่พัฒนาไปอย่างรวดเร็ว

2.4.1 ระบบ ไคลเอนต์ / เซิร์ฟเวอร์ ⁷

นิยาม ระบบ ไคลเอนต์ / เซิร์ฟเวอร์ คือ ระบบงานที่มีการจัดแบ่งหน้าที่การทำงาน การประมวลผล ของแต่ละงาน ให้แก่เครื่องคอมพิวเตอร์ ที่มีความเหมาะสมมากที่สุดทำการประมวลผล เพื่อให้เกิดประสิทธิภาพในการทำงานสูงสุด เช่น ไคลเอนต์ ควรจะทำงานที่เกี่ยวกับการรับ - แสดงผลทางจอภาพ , เซิร์ฟเวอร์ ควรทำงานทางด้าน จัดการหรือจัดเก็บข้อมูลในฐานข้อมูล

2.4.2 องค์ประกอบของระบบ ไคลเอนต์ / เซิร์ฟเวอร์ ⁷

การพัฒนาระบบงานประยุกต์ (Application Software Development) โดยใช้สถาปัตยกรรมแบบ ไคลเอนต์ / เซิร์ฟเวอร์ ประกอบด้วยองค์ประกอบที่สำคัญ 3 ส่วน คือ

1. ไคลเอนต์ (Client) บางครั้งเรียกว่า ตัวลูก หรือ ผู้ขอใช้บริการ คือ เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นผู้รับ - ส่ง ข้อมูล ข่าวสาร คำสั่ง จากผู้ใช้ระบบงาน (User) แล้วส่งไปให้แก่ คอมพิวเตอร์อีกเครื่องหนึ่ง ที่เรียกว่า เซิร์ฟเวอร์ (Server) หรือ ตัวแม่ หรือ ผู้ให้บริการ เพื่อ อ่านข้อมูล ประมวลผล และ ส่งกลับมายังเครื่องลูก
2. เซิร์ฟเวอร์ (Server) มักจะเรียกว่า ตัวแม่ หรือ ผู้ให้บริการ คือ เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นผู้รับ - ส่ง ข้อมูล ข่าวสาร จาก ไคลเอนต์ / เครื่องลูก เพื่อ อ่านข้อมูล ประมวลผล และส่งกลับไปที่ ไคลเอนต์ ตัวลูก หรือ ผู้ขอใช้บริการ

3. ระบบเครือข่าย (Network) คือ ระบบงานคอมพิวเตอร์ที่ประกอบด้วยเครื่องคอมพิวเตอร์ตั้งแต่ 2 เครื่องขึ้นไปที่สามารถรับ - ส่งข้อมูล ข่าวสารระหว่างกันได้โดยผ่านอุปกรณ์เชื่อมต่อทั้ง ฮาร์ดแวร์ (Hardware) และ ซอฟต์แวร์ (Software)

2.4.3 ประโยชน์ของการพัฒนาระบบงานแบบ ไคลเอนต์ / เซิร์ฟเวอร์ ⁸

1. ประหยัดงบประมาณในการลงทุน (Cost Saving)
2. เพิ่มผลผลิต หรือ ก่อให้เกิดประสิทธิผลสูง (Increase Productivity)
3. ระบบงานมีความยืดหยุ่นและขยายระบบได้ง่าย (Flexibility and Scalability)
4. มีการใช้ทรัพยากรที่มีอยู่อย่างมีประสิทธิภาพ (Resource Utilization)
5. สามารถบริหาร ควบคุมจากส่วนกลาง (Centralized Control)
6. มีคุณสมบัติที่เป็นระบบงานแบบเปิด (Open Systems)

1. ประหยัดงบประมาณในการลงทุน (Cost Saving)
 - โดยทั่วไปราคาของเครื่องคอมพิวเตอร์ และอุปกรณ์ ที่เกี่ยวข้องและอุปกรณ์ต่อพ่วง (Hardware & Peripheral) รวมทั้งตัวระบบงาน (Software) แบบ ไคลเอนต์/เซิร์ฟเวอร์ มักจะมีราคาถูกกว่าระบบงานขนาดใหญ่ แบบ เมนเฟรม หรือ มินิคอมพิวเตอร์ (Mainframe / Mini-computer) นอกจากนี้ เรายังสามารถหาผู้ที่มีความรู้ หรือ นักพัฒนาระบบงาน ได้ง่าย และ ค่าจ้างถูกกว่า บุคลากรทางด้านระบบคอมพิวเตอร์ ขนาดใหญ่ แบบ มินิ / เมนเฟรม คอมพิวเตอร์
 - ในการพัฒนาระบบงานแบบ ไคลเอนต์ / เซิร์ฟเวอร์ เราสามารถนำอุปกรณ์คอมพิวเตอร์ โดยเฉพาะ เครื่องพีซี ที่มีอยู่มาใช้งานได้ทันที ตามความเหมาะสมของประสิทธิภาพหรือสมรรถนะของเครื่องที่มีอยู่ได้ทันที นอกจากนี้ ยังอาจพิจารณาขยายขีดความสามารถของเครื่องที่มีอยู่ได้ เช่น เปลี่ยน ซีพียู, เพิ่มหน่วยความจำ, เพิ่ม / เปลี่ยน หน่วยเก็บข้อมูลฮาร์ดดิสก์ ฯลฯ ด้วยงบประมาณที่น้อยกว่าระบบ มินิ / เมนเฟรม คอมพิวเตอร์
 - การพัฒนาระบบงานประยุกต์แบบ ไคลเอนต์ / เซิร์ฟเวอร์ สามารถแยกงาน หรือ โปรแกรม ออกเป็นโปรแกรมย่อยที่ทำงานบน ตัวลูก (Client หรือ Front - End) และโปรแกรมหลักที่ทำงานบน ตัวแม่ (Server หรือ Back - End) ได้ตามความเหมาะสมตามที่เราวิเคราะห์ และออกแบบ ซึ่งจะประหยัดค่าใช้จ่ายในการพัฒนา และบำรุงรักษาระบบ

2. เพิ่มผลผลิต หรือ ก่อให้เกิดประสิทธิผลสูง (Increase Productivity)

- การพัฒนาระบบงานแบบ ไคล์เอนด์ / เซิร์ฟเวอร์ มีการแบ่งการทำงาน และ ประมวลผล ระหว่าง เครื่องไคล์เอนด์ กับ เครื่องเซิร์ฟเวอร์ เช่น เครื่องเซิร์ฟเวอร์ จะทำหน้าที่จัดการกับงานฐานข้อมูลที่เรียกว่า ดาต้าเบสเซิร์ฟเวอร์ (DataBase Server) และ เครื่องไคล์เอนด์ จะทำหน้าที่ รับคำสั่ง และประมวลผล ซึ่งเป็นการแบ่งภาระการทำงานของ เครื่องเซิร์ฟเวอร์ ทำให้เกิดการใช้ทรัพยากรของทั้ง เครื่องเซิร์ฟเวอร์ และเครื่องไคล์เอนด์ อย่างเต็มความสามารถ และเหมาะสม ก่อให้เกิดผลผลิตมากขึ้น
- ผู้ใช้ระบบ (User) สามารถพัฒนาระบบงานได้เอง การพัฒนาระบบงานในปัจจุบัน เราสามารถใช้เครื่องมือในการพัฒนาระบบงาน / โปรแกรม ที่ใช้ง่าย ทันสมัย เรียนรู้น้อย และทำงานได้อย่างมีประสิทธิภาพ เช่น ภาษารุ่นที่ 4 (Fourth Generation Languages) , CASE Tools (Computer - Aided Software Engineering) เป็นต้น ทำให้ผู้ใช้ระบบงาน สามารถพัฒนาระบบงานบางส่วน เช่น รายงาน การวิเคราะห์ข้อมูลตัวเลข หรือ เรียกดูข้อมูล ได้เองโดยไม่จำเป็นต้องอาศัยผู้เชี่ยวชาญคอมพิวเตอร์ เช่นในอดีตที่ผ่านมา
- ปัจจุบัน การใช้โปรแกรมส่วนใหญ่มีลักษณะเอื้ออำนวยต่อ End - User มากขึ้น เช่น มีการนำระบบการใช้งานและสื่อสารด้วยภาพที่เรียกว่า GUI (Graphical User Interface) ทำให้ผู้ใช้งานเรียนรู้ได้ง่าย และสามารถใช้งานได้ด้วยตนเอง
- ในส่วนของนักพัฒนาระบบงาน การออกแบบ การเขียนโปรแกรม และการพัฒนาระบบงานประยุกต์ในรูปแบบ ไคล์เอนด์ / เซิร์ฟเวอร์ สามารถทำได้ง่าย รวดเร็ว และเพิ่มประสิทธิผลมากขึ้น เพราะมีเครื่องมือช่วยในการพัฒนา (Development Tools) ให้ใช้มากมาย และง่ายต่อการเรียนรู้
- สามารถโอนย้ายระบบงานประยุกต์จากระบบหนึ่ง (Platform) ไปใช้ภายใต้สถานะแวดล้อมของระบบอื่นได้

3. ระบบงานมีความยืดหยุ่นและขยายระบบได้ง่าย (Flexibility and Scalability)

ระบบงานแบบไคล์เอนด์ / เซิร์ฟเวอร์ สามารถออกแบบระบบให้สามารถใช้กับเครื่องไคล์เอนด์ และ เครื่องเซิร์ฟเวอร์ ได้หลากหลายชนิด หลายประเภท ตามความเหมาะสมของงาน เช่น สามารถนำเครื่องคอมพิวเตอร์แบบไหนมาต่อเชื่อมก็ได้ ไม่ว่าจะ เป็นเครื่อง ไอพีเอ็มพีซี , ไอพีเอ็มคอมแพคทีบิลด์, เครื่องแม็คคลินทอซด์ ฯลฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนั้นยังสามารถใช้กับระบบปฏิบัติการได้หลากหลายชนิด ทั้ง เอ็มเอสแอล, เอ็มเอสวินโดวส์ ไอบีเอ็มโอเอสทู, แอปเปิลซิสเต็ม 7 เป็นต้น

มีความยืดหยุ่นและสามารถขยายตัวได้ทั้งตัวโปรแกรมประยุกต์ (Application Software) และ การขยายด้านฮาร์ดแวร์

4. มีการใช้ทรัพยากรที่มีอยู่อย่างมีประสิทธิภาพ (Resource Utilization)

- ลดปริมาณการจราจรของการรับ - ส่งข้อมูลในเครือข่าย โดย เครื่องไคลเอนต์ จะส่งคำสั่งไปขอข้อมูลจากเครื่องเซิร์ฟเวอร์เฉพาะที่ต้องการเท่านั้น ตัวเซิร์ฟเวอร์เองก็จะประมวลผลโดยกรองข้อมูลเฉพาะส่วนที่ ไคลเอนต์ต้องการเท่านั้น(ไม่จำเป็นต้องส่งไปทั้งหมด) ดังนั้นข้อมูลที่จะส่งกลับ ไปให้เครื่องไคลเอนต์ผ่านระบบเครือข่าย ปริมาณข้อมูลที่วิ่งผ่านในเครือข่ายก็น้อยลง ทำให้เพิ่มประสิทธิภาพความเร็วของการติดต่อสื่อสารภายในเครือข่าย

นอกจากนี้ตัว าคาด้าเบสเซิร์ฟเวอร์ ยังสามารถเก็บบันทึกคำสั่งบางคำสั่งไว้เพื่อใช้ในบางกรณีได้ด้วย เช่น คำสั่งประเภท Store procedures, Triggers เป็นต้น ทำให้การทำงาน และการประมวลผลข้อมูลมีประสิทธิภาพสูงขึ้น ไม่ต้องเสียเวลาในการสั่งคำสั่งซ้ำ ๆ ที่ไม่จำเป็น เป็นการลดค่าใช้จ่ายของระบบอีกด้วย

5. สามารถบริหาร ควบคุมจากส่วนกลาง (Centralized Control)

การทำงานแบบไคลเอนต์ / เซิร์ฟเวอร์ สามารถออกแบบระบบงานให้ ผู้ใช้ระบบงาน ควบคุมสั่งการ การทำงาน และการประมวลผลได้จากศูนย์กลาง โดย

- การกำหนด โปรแกรมที่เครื่องเซิร์ฟเวอร์ให้ทำหน้าที่ควบคุมฐานข้อมูล และการกำหนดเงื่อนไขของการประมวลผลต่าง ๆ ได้
- รักษาความถูกต้องของข้อมูล โดย ตัวคาด้าเบสเซิร์ฟเวอร์ ส่วนใหญ่จะใช้โปรแกรมบริหารข้อมูลเชิงสัมพันธ์ (Relational Database Management System : RDBMS) ซึ่งมีคุณสมบัติ และความสามารถในการรักษาความถูกต้องและความปลอดภัยของข้อมูลในระดับต่าง ๆ ซึ่งเป็นการป้องกันผู้ที่ไม่มีความรู้เข้าไปใช้ข้อมูล อีกทั้งยังมีความสามารถในการป้องกันฐานข้อมูลโดยการเข้ารหัส (Encrypt / Encoding) เป็นต้น

6. มีคุณสมบัติที่เป็นระบบงานแบบเปิด (Open Systems)

ระบบไคลเอนต์ / เซิร์ฟเวอร์ สามารถทำงานและประมวลผลได้หลากหลาย

แพลตฟอร์ม (Platform) และหลากหลายสภาวะแวดล้อมของหลายระบบปฏิบัติการ

เอกสารนี้เป็นเอก (Operating System : OS) ที่แตกต่างกัน โดย ไม่ต้องแก้ไขโปรแกรมที่พัฒนาขึ้นใหม่ การค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 โปรแกรมเชิงวัตถุ (Object-Oriented Programming)

2.5.1 ตำนาน/ประวัติภาษาออบเจกต์โอเรียนเตด (Object-Oriented) ⁹

ภาษาที่มีแนวคิดแบบ Object-Oriented หรือที่เรียกว่า OOP นั้น เริ่มขึ้นในช่วงปี ค.ศ. 1950 - 1960 โดยภาษา LISP (List Processing) ซึ่งเป็นภาษาที่ใช้ในงานปัญญาประดิษฐ์หรือ AI (Artificial Intelligent) เนื่องจาก LISP เป็นภาษาแรกที่เริ่มรู้จักคำว่า Dynamic Binding และเห็นความสำคัญของการติดต่อผู้ใช้แบบกราฟิกที่พัฒนามาจนเป็น GUI (Graphical User Interface) ในปัจจุบัน ต่อมาราวปี ค.ศ. 1960 หลักการของคลาส (Class) และการถ่ายทอดคุณสมบัติของคลาส (Inheritance) ถูกบัญญัติขึ้นในภาษาที่มีชื่อว่า Simula ซึ่งเป็นภาษาที่ใช้ในการพัฒนาโปรแกรม Simulation ที่ใช้ในงานต่าง ๆ Simula ทำให้เกิดภาษาใหม่ขึ้นมาอีก 2 ภาษา คือ Ada และ Modula-2 ภาษาทั้งสองประสบความสำเร็จทางธุรกิจอย่างมาก และเป็นภาษาแรกที่มีชนิดข้อมูลที่เรียกว่า ข้อมูลชนิดแอ็บสแตรค (Abstract Data Type) ซึ่งยินยอมให้มีโค้ด (Code) รวมอยู่กับข้อมูลชนิดนี้ได้

ในช่วงปี 1970 ภาษา Smalltalk ที่เกิดขึ้นจากผลงานวิจัยของสำนักวิจัย Xerox Palo Alto Research Center (PARC) โดยนาย Alan Kay ภาษา Smalltalk ดึงเอาหลักการเกี่ยวกับเรื่องคลาส และการถ่ายทอดคุณสมบัติมาจากภาษา Simula-67 (ซึ่งเป็นรหัสหนึ่งของภาษา Simula) รวมเข้ากับระบบติดต่อผู้ใช้แบบกราฟิกสมบูรณ์แบบ มิสเตอร์เคย์ (Alan Kay) พยายามผลักดันจนกระทั่งภาษา Smalltalk รหัสใหม่ออกมาอีกโดยใช้ชื่อว่า Smalltalk/V ตามมาด้วย Smalltalk/V for Windows ที่ทำงานภายใต้ไมโครซอฟท์วินโดวส์ จนกล่าวได้ว่าภาษา Smalltalk เป็นภาษาแรกที่มีลักษณะของ OOP สมบูรณ์แบบที่สุด

ในขณะที่ Smalltalk เริ่มเป็นที่รู้จักนั้น เป็นช่วงเดียวกับที่ภาษายอดนิยมอย่าง ภาษาเบสิก, ภาษาซี, ภาษาปาสคาล และภาษาโคบอล เริ่มเป็นที่นิยมมากขึ้น แสดงให้เห็นว่าแนวความคิดแบบ OOP ไม่ใช่เรื่องแปลกใหม่แต่อย่างใด ภาษาในแบบโพรซีเจอร์เรอร์โปรแกรมมิงค์ (Procedural Programming) มีภาษาซี และภาษาปาสคาล เป็นหลัก ในขณะที่ทางด้าน OOP ก็มีภาษา Smalltalk และ Simula ดังนั้นเมื่อกระแสของ OOP เริ่มเป็นที่ยอมรับมากขึ้น เป็นผลให้เกิดภาษาแบบ OOP ในยุคที่สองขึ้น โดยภาษา OOP ในยุคที่สองนี้เป็นการนำเอาคุณลักษณะของ OOP ผสมเข้ากับไวยากรณ์ (Syntax) เดิมของภาษาแบบ Procedural เรียกว่า ภาษา OOP แบบไฮบริด (Hybrid Group of OOP Language)

วงการคอมพิวเตอร์จึงปรากฏภาษา OOP เป็นสองกลุ่มคือ กลุ่มที่หนึ่งเรียกว่า Pure OOP Group Language ในกลุ่มนี้ได้แก่ ภาษา Smalltalk Actor ที่พัฒนามาจาก Smalltalk ของ Whitewater Group และ Eiffel ของ Interactive Software Engineering Inc. ภาษาในกลุ่มนี้ จะมี

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไวยากรณ์ของภาษาเป็นแบบออบเจกต์ทั้งหมด ไม่ว่าจะเป็นสแตทเมนต์ของภาษาเช่น IF-ELSE หรือ DO-WHILE การดำเนินของโปรแกรมก็จะใช้วิธีส่งเมสเสจ (Message) เพียงอย่างเดียว ส่วนภาษาในกลุ่มไฮบริด OOP ได้แก่ ภาษาซี ++, ภาษาออบเจกต์ซี (Object-C), ภาษาออบเจกต์เบสิก (Object-BASIC), ภาษาออบเจกต์โคบอล (Object-COBOL), ภาษา Common Lisp Object System (CLOS) และ ภาษาออบเจกต์ปาสคาล (Object-PASCAL) ภาษาในกลุ่มนี้จะมีลักษณะทั่วไปเป็นภาษาแบบ Procedural ธรรมดา แต่เสริมความสามารถพิเศษ ของภาษาแบบ OOP เข้าไปด้วย และส่วนใหญ่ความสามารถที่เพิ่มเข้าไปนี้จะเป็นลักษณะซูเปอร์เซต (Super Set) ของภาษาเดิม

ในขณะที่ภาษาในกลุ่มแรกมีความเป็น OOP มากกว่าในด้านของตัวภาษา โครงสร้างของภาษาเป็นออบเจกต์ทั้งหมด และสร้างงานโดยการส่งเมสเสจไปยังออบเจกต์ที่ต้องการ ซึ่งส่วนใหญ่จะเก็บอยู่ในคลาสไลบรารี (Class Library) ของภาษาไว้ล่วงหน้าหมดแล้ว จึงทำให้พัฒนาโปรแกรมได้ง่าย และรวดเร็ว อีกทั้งยังใช้เป็นภาษาฝึกสอนได้เป็นอย่างดี ในขณะที่ความนิยมของภาษาคอมพิวเตอร์ยังเป็น ภาษาซี หรือ ปาสคาล เป็นหลัก โปรแกรมส่วนมากจึงถูกพัฒนาขึ้นมาโดยภาษาเหล่านี้ ความเข้ากันได้ (Compatibility) ของโค้ดจึงเป็นเรื่องสำคัญ ภาษาในกลุ่มไฮบริดได้รับความนิยมในทางปฏิบัติอย่างสูง เพราะสามารถสร้างโค้ดใหม่ลึกลับกับโค้ดเก่าได้อย่างไม่มีปัญหา

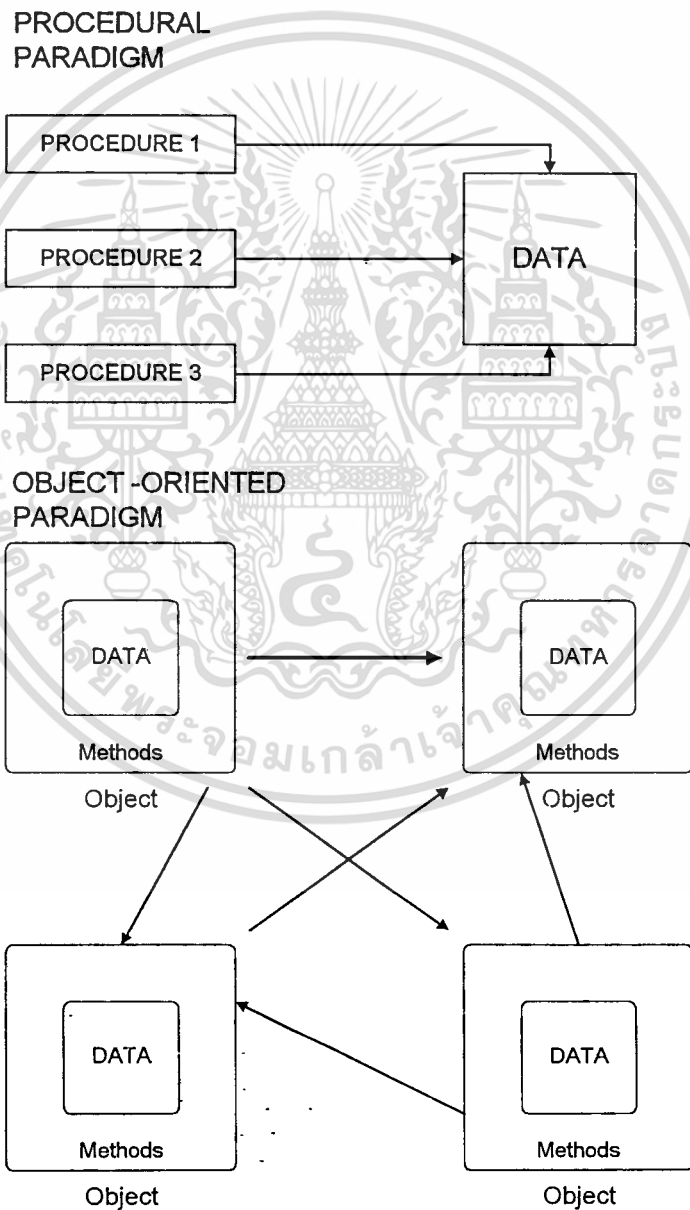
ภาษากลุ่มไฮบริดที่ได้รับความนิยมสูงสุดคือ ภาษาซีพลัสพลัส(C ++) และภาษาออบเจกต์ปาสคาล (Object-Oriented PASCAL) เนื่องด้วยภาษาเหล่านี้ โดยเฉพาะอย่างยิ่ง ภาษาซีพลัสพลัส มีความเป็น Portability สูงมากเพราะสามารถทำงานได้ในแทบทุกแพลตฟอร์ม ปัจจุบันภาษาซีพลัสพลัส มีคอมพิวเตอร์อยู่ทั้งในเครื่องระดับมินิคอมพิวเตอร์, เวิร์กสเตชัน, และไมโครคอมพิวเตอร์ทั่วไป อาจกล่าวได้ว่า ภาษาซี เป็นภาษา OOP อันดับ 1 ในปัจจุบัน

2.5.2 การเขียนโปรแกรมแบบโพรซีเยอร์โปรแกรมมิง (Procedural Programming) ⁹

การเขียนโปรแกรมแบบเดิม เราจะพยายามแบ่งงานทั้งหมดออกเป็นหน่วยเล็ก ๆ แบ่งโปรแกรมออกเป็นโพรซีเยอร์ (Procedure) หรือรูทีนย่อย (Subroutine) เพื่อจัดการส่วนต่าง ๆ ในโปรแกรม ดังนั้นคำว่าโปรแกรมในนิยามเดิมจึงหมายถึง เซตของโพรซีเยอร์ (Set of Procedure) เมื่อมองโปรแกรมเป็นเซตของโพรซีเยอร์แล้ว ขั้นตอนการออกแบบโปรแกรมจึงเป็นขั้นตอนของการจัดโครงสร้างของโพรซีเยอร์ต่าง ๆ ในโปรแกรมตามลำดับก่อนหลัง จากใหญ่ไปหาเล็ก เรียกว่า การออกแบบจากบนลงล่างหรือ Top-Down Design

2.5.3 การเขียนโปรแกรมแบบออบเจกต์โอเรียนเตด (Object-Oriented Programming) ⁹

ในการพัฒนาโปรแกรมแบบ OOP นั้น โปรแกรมมีนิยามใหม่ว่าเป็นเซตของออบเจกต์ (Set of Object) ซึ่งออบเจกต์นี้อาจถูกกำหนดให้เป็นหน่วยใหม่ที่สร้างขึ้นมาเพื่อรวมเอาทั้ง ส่วนข้อมูลและโค้ดที่จัดการกับข้อมูลนั้นไว้ด้วยกัน ดังนั้นเมื่อนิยามของโปรแกรมเปลี่ยนไป การออกแบบโปรแกรมจึงกลายเป็นการออกแบบโครงสร้างและความเกี่ยวข้องระหว่างออบเจกต์แต่ละตัวในโปรแกรม



ภาพที่ 2.5 แสดงลักษณะของการพัฒนาโปรแกรมใน 2 นิยาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 หลักการของโอโอพี (Object-Oriented Programming) ¹⁰

ออบเจกต์ คือ วัตถุ ในการเขียนโปรแกรมจะหมายถึงหน่วยสนใจ (Oriented) ของระบบ โดยระบบนี้ (OOP) จะเน้นที่ตัวปฏิบัติการ มากกว่า การปฏิบัติ โดยจะให้ความสำคัญของความเกี่ยวพันของ ออบเจกต์แต่ละตัวที่อยู่ในระบบเป็นหลัก

ออบเจกต์ แต่ละตัวจะประกอบด้วย 2 ส่วนสำคัญคือ คุณลักษณะหรือคุณสมบัติ (Property) และหน้าที่หรือกระบวนการ (Methods) ของออบเจกต์ นั้น ๆ โดยหน้าที่หรือกระบวนการนี้จะมีความสัมพันธ์กับออบเจกต์อื่น ๆ ในระบบ ผ่านทางเมสเสจหรือเหตุการณ์หรือการกระทำ (Events) ที่ออบเจกต์ต่าง ๆ กระทำต่อกัน

ตัวอย่าง พิจารณาระบบการใช้โทรศัพท์ของเรา สมมติว่าในระบบนี้มีออบเจกต์อยู่ 2 ออบเจกต์ คือ ตัวเครื่องโทรศัพท์ และ ผู้ใช้โทรศัพท์

ตัวเครื่องโทรศัพท์ ประกอบด้วย

1. คุณลักษณะ (Property) คือ
 - เป็นโทรศัพท์แบบกดปุ่ม
 - มีสีเทา
 - เป็นแบบมีสาย
 - แบบตั้งโต๊ะ
 - ฯลฯ

2. มีหน้าที่/กระบวนการ (Method) คือ - กริ่งสัญญาณจะดังได้ เมื่อ มีสายเรียกเข้ามา สายที่เรียกเข้ามานี้ เป็น เหตุการณ์ (Event) ที่จะไปกระตุ้นให้หน้าที่ในการทำสัญญาณให้ดัง ซึ่งเป็น Method ให้ทำงาน ในทางตรงกันข้าม อีกหน้าที่ (Method) หนึ่งของออบเจกต์โทรศัพท์ คือ การส่งสัญญาณสายว่างหรือไม่ว่าง เมื่อเกิดเหตุการณ์ (Event) ที่ (ออบเจกต์)ผู้ใช้ไปยกหูโทรศัพท์ เพื่อโทรออก เป็นต้น

โปรดสังเกตว่า Method ต่าง ๆ นั้นจะมีการตอบสนองต่อเหตุการณ์เฉพาะอย่างเท่านั้น เช่น Method “สัญญาณดัง” จะตอบสนองต่อเหตุการณ์ “สายเรียกเข้า” เท่านั้น เพราะหากไม่เกิดสายเรียกเข้า Method “สัญญาณดัง” ก็จะไม่ตอบสนองหรือไม่ทำงาน ในขณะที่ Method “สัญญาณสายว่าง”หรือไม่ จะทำงานเมื่อเกิดเหตุการณ์ที่ (ออบเจกต์)ผู้ใช้ยกหูฟังขึ้นเท่านั้น นั่นคือไม่ตอบสนองต่อเหตุการณ์ “สัญญาณเรียกเข้า”

2.5.5 ศัพท์พื้นฐานในแนวคิดแบบโอโอพี OOP ¹¹

1. **ออบเจกต์ (Object)** ดังได้กล่าวมาแล้วว่า ในความหมายเชิงวิชาการ ให้คำนิยามว่าคือวัตถุหรือส่วนหนึ่งในระบบที่ประกอบขึ้นด้วยองค์ประกอบ 2 ส่วน คือ ข้อมูล และ โค้ดโปรแกรม ส่วนข้อมูลใช้เก็บสถานะหรือคุณลักษณะ (Property) ของตัวมันเอง บางที่เรียกว่าค่า (DATA) และส่วนโค้ดโปรแกรมใช้ในการตอบสนองต่อ ออบเจกต์ตัวอื่นในระบบเดียวกัน เรียกว่า เมธอด (Method) ออบเจกต์ใด ๆ ในระบบจะสื่อสารกับออบเจกต์อื่นเพื่อให้บรรลุความต้องการของตน โดยการสื่อสารในลักษณะ “ร้องขอและตอบสนอง” เมื่อออบเจกต์หนึ่งขอความช่วยเหลือจากอีกออบเจกต์หนึ่ง เราเรียกว่า มันกำลังส่ง เมสเสจ (message) ไปยังออบเจกต์อื่น

เช่น ในขณะที่เรากำลังใช้งานไมโครซอฟท์วินโดวส์อยู่ เราคือออบเจกต์หนึ่งในระบบ ในขณะที่วินโดวส์ก็เป็นอีกออบเจกต์หนึ่ง เมื่อเราคลิกเมาส์..นั่นคือเรากำลังส่งเมสเสจให้กับวินโดวส์เพื่อให้ทำอะไรบางอย่าง เมื่อวินโดวส์ตอบสนอง นั้นหมายความว่า มันกำลังตอบสนองต่อเมสเสจของเรา ด้วยเมธอด (Method) ที่สร้างขึ้นเพื่อรองรับหรือตอบสนองต่อเมสเสจนั้น

2. **คลาส (Class)** กับออบเจกต์ เป็นสิ่งคู่กัน เพราะคลาสคือ ต้นแบบหรือพิมพ์เขียว (Blueprint) หรือ DNA ของออบเจกต์ เราจะดูว่าออบเจกต์นี้มีลักษณะหรือคุณสมบัติอย่างไร ให้ดูที่คลาส เช่น โทรศัพท์ในตัวอย่างข้างต้นถ้าอยากทราบว่า มีลักษณะอย่างไร ก็ต้องดูที่คลาสของโทรศัพท์ ไม่ใช่ไปดูที่คลาสของคนซึ่งเป็นผู้ใช้ เพราะคลาสของโทรศัพท์กับคลาสของคนมีลักษณะหรือคุณสมบัติที่แตกต่างกัน

2.1 **ซับคลาส (Sub-class)** เนื่องจากหลักการของ OOP เป็นหลักการที่อาศัย การถ่ายทอดคุณสมบัติ (Inheritance) เป็นหลัก คลาสแต่ละคลาสในระบบจึงสามารถให้กำเนิดลูกหลานที่สืบทอดคุณสมบัติกันได้ โดยเรียกทายาทของคลาสว่า ซับคลาส (Sub-class) ซับคลาสจะรับเอาคุณสมบัติของคลาสผู้ให้กำเนิดที่เรียกว่า คลาสพ่อแม่ (Parent Class) เช่น โทรศัพท์ไร้สาย, โทรศัพท์สาธารณะ, โทรศัพท์มือถือ ฯลฯ ต่างก็เป็นซับคลาสของคลาสเครื่องโทรศัพท์ (Parent Class) เป็นต้น บางครั้งจะพบว่า มีผู้ใช้คำว่า Derived Class หรือ Child Class แทนคำว่า ซับคลาส ก็มี

2.2 **คลาสไลบรารี (Class Library)** เนื่องจากการกำหนดคลาสโดยอาศัยการถ่ายทอดนี้เอง ทำให้เกิดคลาสขึ้นมากมายในระบบ จนเกิดสิ่งที่เรียกว่า Class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Library หรือบางทีก็เรียกว่า Class Hierarchy ซึ่งก็เปรียบเสมือนกับไลบรารีที่ใช้เก็บโปรแกรมที่จะนำมาลิงค์ต่อของโปรแกรมที่เขียนขึ้นแบบเก่า (Procedural Programming) แต่แทนที่จะเป็นชุดฟังก์ชัน (Function) ก็เป็นโครงข่ายของคลาสในระบบแทน

8. เอนแคปซูลชัน (Encapsulation) คือ กระบวนการปกปิดความสลับซับซ้อน (Complexity) ของออบเจกต์ เช่นจากกรณีเครื่องโทรศัพท์ในตัวอย่างข้างต้น เราในฐานะผู้ใช้ไม่จำเป็นต้องรู้ถึงสถาปัตยกรรมในการสร้างหรือการทำงานภายในของเครื่องโทรศัพท์ (พวกแผงวงจรไฟฟ้าภายในตัวเครื่อง) เพราะสถาปัตยกรรมหรือโครงสร้างดังกล่าวถูกปกปิดไว้ ดังนั้น การเข้าถึงค่าสถานะภายในของออบเจกต์จะกระทำได้โดยผ่านการเห็นชอบจากออบเจกต์เจ้าของเสียก่อน หมายความว่า การเปลี่ยนแปลงแก้ไขค่าตัวแปรภายในออบเจกต์จะต้องกระทำผ่านเมธอดของออบเจกต์ดังกล่าวเท่านั้น
4. Polymorphism นิยาม “One interface multiple implements” โดย Hervert Schildt ¹¹ คือ การส่งเมสเสจแบบเดียวกันสามารถตอบสนองได้หลายแบบ และไม่จำเป็นต้องได้รับการตอบสนองเหมือนกัน ขึ้นอยู่กับออบเจกต์ที่ได้รับเมสเสจเป็นสำคัญ เช่น เมสเสจเมาส์คลิก (Mouse Click) หรือ เหตุการณ์เมาส์คลิก (เป็นเมสเสจแบบเดียวกัน/ชื่อเดียวกัน คือ Mouse Click) แต่อาจได้รับการตอบสนองไม่เหมือนกัน ซึ่งก็แล้วแต่ออบเจกต์ที่ได้รับ เมสเสจนั้น เช่น ถ้าออบเจกต์ ที่ได้รับเป็น ปุ่มกด (Button) ก็อาจตอบสนองต่างจาก ออบเจกต์คอมโบบ็อกซ์ (Combo Box) แม้ว่าจะได้รับเมสเสจคลิกเหมือนกัน (แล้วแต่โค้ดที่เขียนไว้ใน เมธอดคลิกของแต่ละออบเจกต์)

2.5.6 ยูสเซอร์อินเตอร์เฟซแบบออบเจกต์โอเรียนเต็ด (Object-Oriented User Interface)¹²

การพัฒนาโปรแกรมภายใต้สภาพแวดล้อมของ ยูสเซอร์อินเตอร์เฟซแบบออบเจกต์โอเรียนเต็ดนี้จะใช้หลักการของ OOP โดยออบเจกต์มาตรฐานที่ต้องมีในแต่ละโปรแกรม เช่น หน้าต่าง (Form), ปุ่มกด (Button), เมนู (Menu), ไดอะล็อกบ็อกซ์ (Dialogue Box) และอื่น ๆ ผู้พัฒนาสามารถดึงออบเจกต์เหล่านี้จากออบเจกต์ไลบรารีมาใช้งานได้เลย โดยไม่ต้องมาเขียนโปรแกรมใหม่

Object-Oriented GUI เกือบทั้งหมดจะเป็นแบบ Event-Driven ออบเจกต์ใน GUI จะติดต่อสื่อสารกับออบเจกต์อื่น โดยการส่งเมสเสจไปยังออบเจกต์นั้น แล้วแต่ละออบเจกต์จะมีเม

ธอดของตนเองในการจัดการกับเมสเสจที่ส่งมา ตัวอย่างเช่น ถ้าใช้งานอยู่บนวินโดวส์ แล้วมีการ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โต้ตอบโดยการเลื่อนเมาส์ไปคลิกปุ่มนั้น ปุ่มนี้ นี้คือเกิดเหตุการณ์ (Event) ซึ่งนักพัฒนาจะต้องมีการเขียนโปรแกรมเพื่อจัดการเหตุการณ์ต่าง ๆ (Event Handler) ว่าเมื่อมีเมสเสจนี้ส่งมาจะจัดการอย่างไร ในวินโดวส์จะมีการกำหนดคลาสไลบรารีซึ่งบรรจุไปด้วยเมธอดมากกว่าร้อยเมธอดให้นักพัฒนาเรียกใช้ ดังนั้น การพัฒนาโปรแกรมภายใต้ Object-Oriented User Interface ถ้านักพัฒนาเข้าใจในหลักการแล้ว การพัฒนาโปรแกรมแต่ละโปรแกรมจะประหยัดเวลาไปได้มาก ถึงแม้ว่าโปรแกรมจะมีขนาดใหญ่ แต่ถ้ามีออบเจกต์ไลบรารีพื้นฐานเดิมอยู่ก็สามารถดึงมาใช้งาน และเพิ่มคุณสมบัติเฉพาะที่ผู้พัฒนาต้องการได้ทันที โดยไม่ต้องเสียเวลาเขียนโปรแกรมใหม่อีก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 เครื่องมือที่ใช้ในการพัฒนา (Development Tools)

2.6.1 วิวอลฟอกซ์โปร (Visual FoxPro)

วิวอลฟอกซ์โปรเป็นเครื่องมือที่ใช้สำหรับพัฒนาโปรแกรมประยุกต์ในแบบ PC-RDBMS รุ่นใหม่ที่มีความสามารถสูง ซึ่งสามารถนำไปใช้ในการสร้างระบบงานตั้งแต่แบบง่าย ๆ ไปจนถึงระบบที่สลับซับซ้อน ทั้งในแบบผู้ใช้คนเดียว (Stand Alone) หรือแบบผู้ใช้หลายคน (Multi-user) บนสถาปัตยกรรมแบบไคลเอนต์ / เซิร์ฟเวอร์ ให้มีประสิทธิภาพสูง เนื่องจาก วิวอลฟอกซ์โปร มีเครื่องมือต่าง ๆ ให้ผู้พัฒนาเลือกใช้ตามลักษณะงานเป็นจำนวนมาก ทั้งยังสามารถพัฒนาระบบงานในแบบโปรแกรมเชิงวัตถุ (Object Oriented Programming : OOP) ซึ่งเมื่อพัฒนาเสร็จแล้วสามารถนำกลับมาใช้งานได้อีกอย่างไม่จำกัด

วิวอลฟอกซ์โปร เป็นระบบจัดการฐานข้อมูลและระบบพัฒนาโปรแกรมประยุกต์ที่ทรงประสิทธิภาพ มีเครื่องมือที่ใช้ในการจัดการข้อมูลโดยตรงซึ่งผู้พัฒนาสามารถใช้ด้วยการคลิกเมาส์เท่านั้น ด้วยสมรรถนะและความเร็วของ วิวอลฟอกซ์โปร งานจัดการข้อมูลที่เคยทำได้ในเครื่องระดับเมนเฟรมหรือมินิคอมพิวเตอร์ก็สามารถนำมาใช้ได้บนเครื่องไมโครคอมพิวเตอร์ เครื่องมือต่าง ๆ ที่มีใน วิวอลฟอกซ์โปร ¹³

1. คลาสดีไซน์เนอร์ (Class Designer) ใช้สำหรับสร้างคลาสใหม่ ซึ่งคลาสที่สร้างขึ้นนี้ผู้พัฒนาสามารถนำไปใช้ได้กับทุก ๆ โปรแกรมไม่จำกัดเฉพาะแต่ในโปรแกรมที่กำลังพัฒนาอยู่เท่านั้น นอกจากนี้ การเปลี่ยนแปลงคุณสมบัติ (Properties) ของคลาสที่เป็นต้นแบบหรือคลาสแม่ (Parent Class) วิวอลฟอกซ์โปรก็จะเปลี่ยนแปลงคุณสมบัติในชั้นคลาส (Subclass) หรือคลาสลูก (Child Class) ให้โดยอัตโนมัติ
2. เครื่องมือพัฒนาโปรแกรมแบบ ไคลเอนต์ / เซิร์ฟเวอร์ จะช่วยให้สามารถเข้าถึงข้อมูลจากสถานที่ห่างไกลได้ (Remote Access) รวมทั้งการติดต่อเรียกใช้ข้อมูลจากฐานข้อมูลต่างฟอร์แมตหรือต่างแพลตฟอร์ม เช่น ออราเคิล (Oracle), อินฟอร์มิก (Informix), ไชร์เบส (SyBase) ฯลฯ ก็สามารถเรียกใช้ได้โดยผ่านไดรเวอร์ที่วิวอลฟอกซ์โปรเตรียมไว้ให้ที่เรียกว่า "Open DataBase Connectivity : ODBC"

นอกจากนี้ผู้พัฒนายังสามารถพัฒนาโปรแกรมต้นแบบในเครื่องที่ทำงานอยู่ เมื่อเสร็จแล้วจึงย้ายโปรแกรมไปใช้ในระบบไคลเอนต์ / เซิร์ฟเวอร์ โดยไม่ต้องเปลี่ยนแปลงโปรแกรมแต่อย่างใด

3. เครื่องมือออกแบบ แบบวิวอล "Visual" (ที่ผู้พัฒนาสามารถมองเห็นทันที) เช่น แถบเครื่องมือและหน้าต่างออกแบบต่าง ๆ (Designer Windows) จะช่วยในการสร้างและปรับแต่งส่วนประกอบต่าง ๆ ของระบบงาน เช่น ฟอร์มอินพุต / เอาต์พุต , คิวรี และ

รายงานได้อย่างรวดเร็วในแบบกราฟิกยูสเซอร์อินเตอร์เฟส (Graphical User Interface : GUI) ที่เรียกว่า จียูไอ

4. พจนานุกรมข้อมูล (Data Dictionary) ทำให้การพัฒนาโปรแกรมประยุกต์ตามทฤษฎีฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ทำได้อย่างง่ายดายโดยไม่ต้องเขียนคำสั่งให้ยุ่งยาก เช่น การสร้างกฎการบริหารข้อมูลและการตรวจสอบข้อมูล (Validity) หรือการกำหนดความถูกต้องของข้อมูล (Data Integrity) และการอ้างอิงความสัมพันธ์ระหว่างตารางในฐานข้อมูล ที่เรียกว่า Referential Integrity กฎเกณฑ์เหล่านี้ผู้พัฒนาสามารถสร้างได้โดยง่ายและไม่ต้องเขียนเป็นคำสั่งให้วุ่นวาย
5. เครื่องมืออัตโนมัติ ที่เรียกว่า วิซาร์ด (Wizard) และ บิลเดอร์ (Builder) ซึ่งจะคอยแนะนำการทำงานในระหว่างกระบวนการสร้างส่วนประกอบของโปรแกรมประยุกต์ที่กำลังพัฒนา เช่น ตาราง, ฟอรั่ม, ลิวรี และรายงาน ด้วยการถามตอบโต้ตอบกับผู้พัฒนาเพื่อให้ได้ส่วนประกอบที่ต้องการ ซึ่งวิธีการดังกล่าวนี้จะประหยัดเวลาในการพัฒนาลงไปได้มาก
6. การพัฒนาโปรแกรมประยุกต์ในวิซวลฟอกซ์โปร เป็นระบบแบบ Event-driven ผู้พัฒนาสามารถสร้างโปรแกรมประยุกต์แบบ nonmodal ขึ้นมาได้โดยไม่ต้องเขียนโปรแกรมเพิ่ม จะช่วยให้ผู้ใช้สามารถทำงานได้พร้อมกันหลายหน้าต่าง และในแต่ละหน้าต่างสามารถบรรจุได้หลายออบเจกต์ (multiple instances) และยังสามารถสลับไปมาระหว่างหน้าต่าง โดยโปรแกรมจะเก็บสถานะของแต่ละหน้าต่างไว้ก่อนได้
7. ในระบบผู้ใช้หลายคน (Multi-user) ผู้พัฒนาสามารถกำหนดค่าบัฟเฟอร์ (Buffer) ให้กับเรคอร์ด (Record) และ ตาราง (Table) สามารถล็อก (Lock) ข้อมูลเพื่อป้องกันการเปลี่ยนแปลงข้อมูลในขณะที่มีการเข้าถึงข้อมูลพร้อมกัน นอกจากนี้ยังมีคำสั่งที่ใช้กับการประมวลผลแบบ ทรานส์เซกชัน โพรเซสซิง (Transaction Processing) ซึ่งจะสามารถฟื้นคืนสภาพ (Roll Back) ข้อมูล กรณีที่เกิดความผิดพลาดขึ้นในระหว่างการประมวลผลรายการนั้น หรือ ทรานส์เซกชันนั้น ซึ่งเป็นการรักษาความถูกต้องของข้อมูลในฐานข้อมูลตลอดเวลา

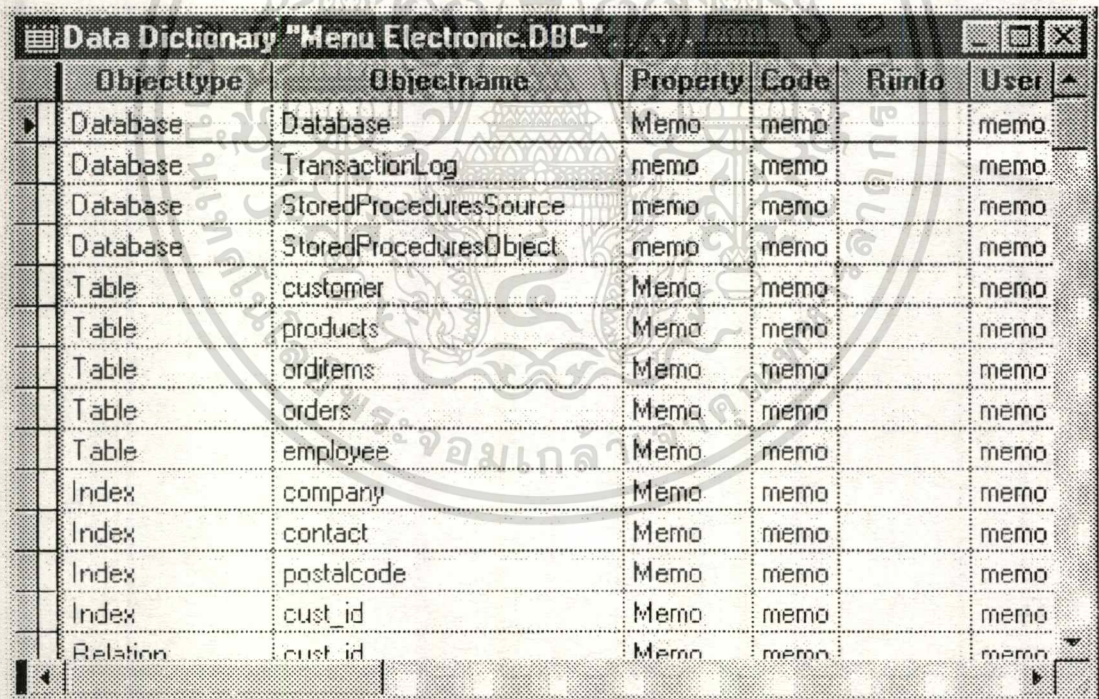
2.6.2 วิซวลฟอกซ์โปรกับการจัดการฐานข้อมูล

วิซวลฟอกซ์โปรตั้งแต่ เวอร์ชัน 3.0 เป็นต้นมา ได้เปลี่ยนแปลงแนวคิดของการจัดการดาต้าเบสไฟล์ไปจากรุ่นก่อน ๆ โดยในดาต้าเบสหนึ่ง จะประกอบไปด้วยตารางข้อมูล (Table) หลายตารางที่มีความสัมพันธ์หรืออาจมีการแชร์ข้อมูลซึ่งกันและกัน ซึ่งตารางที่กล่าวถึงนี้ก็คือ ไฟล์นาม

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์หรือการเชิงอื่นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สกุล (File Extension) .DBF เดิมในเวอร์ชัน 2.xx นั้นเอง คาด้าเบสไฟล์ใหม่ในเวอร์ชันนี้จะใช้เก็บไฟล์นามสกุล .DBC ซึ่งมีคุณสมบัติเพิ่มขึ้นหลายประการตามทฤษฎีของฐานข้อมูลแบบรีเลชันแนล เช่น

1. พจนานุกรมข้อมูล (Data Dictionary) โครงสร้างของไฟล์ .DBC เองมีลักษณะเป็นตารางเช่นเดียวกับตารางข้อมูลอื่น ๆ ที่มันเก็บอยู่ โดยฟิลด์ต่าง ๆ ที่ประกอบขึ้นเป็นไฟล์ .DBC นี้จะแสดงรายละเอียดของตารางต่าง ๆ , ลักษณะของความสัมพันธ์, กฎเกณฑ์ต่าง ๆ ที่ผู้พัฒนาสร้างขึ้นเพื่อรักษาความถูกต้องของข้อมูล (Data Integrity) ที่อยู่ในฐานข้อมูล เช่น กฎความถูกต้องของการอ้างอิง (Referential Integrity), ขอบเขตของข้อมูลในโดเมน (Domain Constraints) เช่น การตรวจสอบข้อมูล (Data Validation) เหล่านี้ เป็นต้น ซึ่งข้อมูลดังกล่าวนี้เปรียบเสมือนเป็น พจนานุกรมข้อมูล (Data Dictionary) ตามทฤษฎีของฐานข้อมูลแบบรีเลชันแนล นั้นเอง



| Objecttype | Objectname | Property | Code | Runfo | User |
|------------|------------------------|----------|------|-------|------|
| Database | Database | Memo | memo | | memo |
| Database | TransactionLog | memo | memo | | memo |
| Database | StoredProceduresSource | memo | memo | | memo |
| Database | StoredProceduresObject | memo | memo | | memo |
| Table | customer | Memo | memo | | memo |
| Table | products | Memo | memo | | memo |
| Table | orditems | Memo | memo | | memo |
| Table | orders | Memo | memo | | memo |
| Table | employee | Memo | memo | | memo |
| Index | company | Memo | memo | | memo |
| Index | contact | Memo | memo | | memo |
| Index | postalcode | Memo | memo | | memo |
| Index | cust_id | Memo | memo | | memo |
| Relation | cust_id | Memo | memo | | memo |

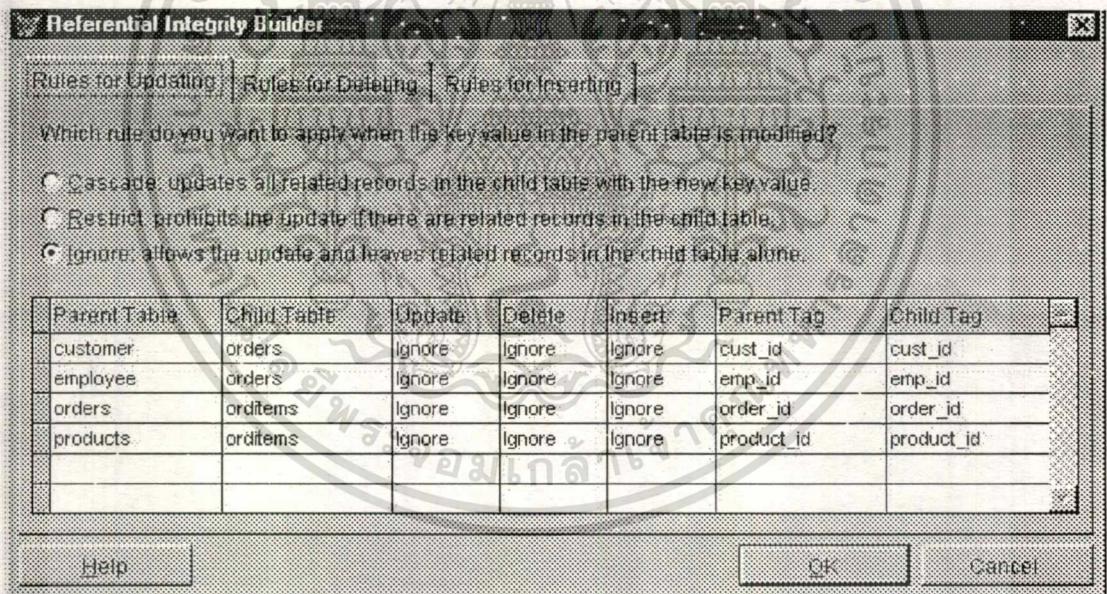
ภาพที่ 2.6 แสดงรายละเอียดในพจนานุกรมข้อมูล (Data Dictionary)

2. กฎความถูกต้องของการอ้างอิง(Referential Integrity หรือ RI)ในวิชาฟอทซ์โปรผู้พัฒนาสามารถสร้างกฎเกณฑ์ต่างเพื่อคงสภาพความถูกต้องของการอ้างอิงตามลักษณะความสัมพันธ์ต่าง ๆ ของตารางที่อยู่ในฐานข้อมูล เช่น การเพิ่มเติม,แก้ไข หรือลบข้อมูลในตารางแม่ (Parent Table) ในความสัมพันธ์แบบ 1 ต่อ กลุ่ม (One to Many)

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของกรมส่งเสริมการค้าระหว่างประเทศ กระทรวงพาณิชย์
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ย่อมมีผลกระทบต่อการอ้างอิงข้อมูลในตารางลูก (Child Table) ตัวอย่างเช่น หากเรกอร์ดใดเรกอร์ดหนึ่งในตารางแม่ที่มีความสัมพันธ์ หรือ เป็นคีย์ (Key) เพื่อเข้าถึงข้อมูลในตารางลูกถูกลบทิ้งไป การเข้าถึงหรือการอ้างอิงข้อมูลในตารางลูกด้วยคีย์ดังกล่าวย่อมเป็นไปไม่ได้ ในทางตรงข้ามเรกอร์ดดังกล่าวในตารางลูกที่มีค่าตรงกับข้อมูลที่เป็นคีย์ในตารางแม่ ย่อมไม่มีโอกาสอ้างอิงกลับไปหาข้อมูลหรือเรกอร์ดที่เกี่ยวข้องกันในตารางแม่ได้อีก (เพราะถูกลบทิ้งไปแล้ว) ลักษณะเช่นนี้ทำให้เกิดความผิดพลาดในการอ้างอิงของทั้งสองตาราง ซึ่งในทฤษฎีฐานข้อมูลแบบรีเลชันแนลจะไม่ยอมให้เกิดกรณีเช่นนี้ได้

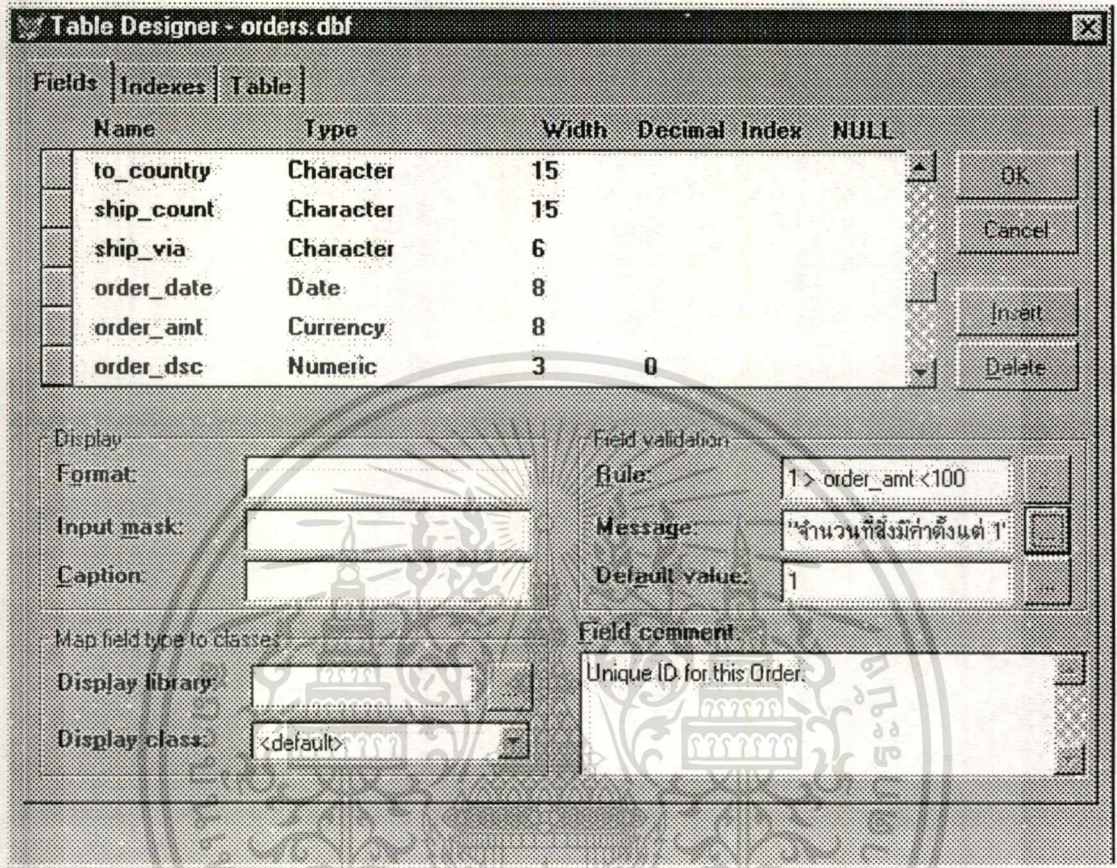
ความสามารถของวิซวลฟอกซ์โปรในตอนนี้ ได้เข้ามาช่วยนักพัฒนาโดยอนุญาตให้นักพัฒนาสามารถสร้างกฎเกณฑ์เพื่อคงสภาพความถูกต้องของการอ้างอิงดังกล่าวได้อย่างง่ายดาย โดยไม่ต้องเขียน โปรแกรมเพิ่มเติมแต่อย่างใด



ภาพที่ 2.7 แสดงการสร้าง Referential Integrity

3. การกำหนดขอบเขตของข้อมูลในโดเมน (Domain Constraint Definition) ผู้พัฒนาสามารถตรวจสอบความถูกต้องของข้อมูล โดยการสร้างขอบเขตเพื่อตรวจสอบข้อมูล (Data Validation) ก่อนมีการอัปเดต (Update) ข้อมูลลงในตาราง เช่น กำหนดว่าอายุของพนักงานจะต้องมีค่าอยู่ระหว่าง 18 - 60 ปี ($18 \leq \text{อายุ} \leq 60$) เป็นต้น เช่นเดียวกับกฎ Referential Integrity ผู้พัฒนาไม่ต้องเขียน โปรแกรมเพิ่มเพื่อคอยตรวจสอบกฎ

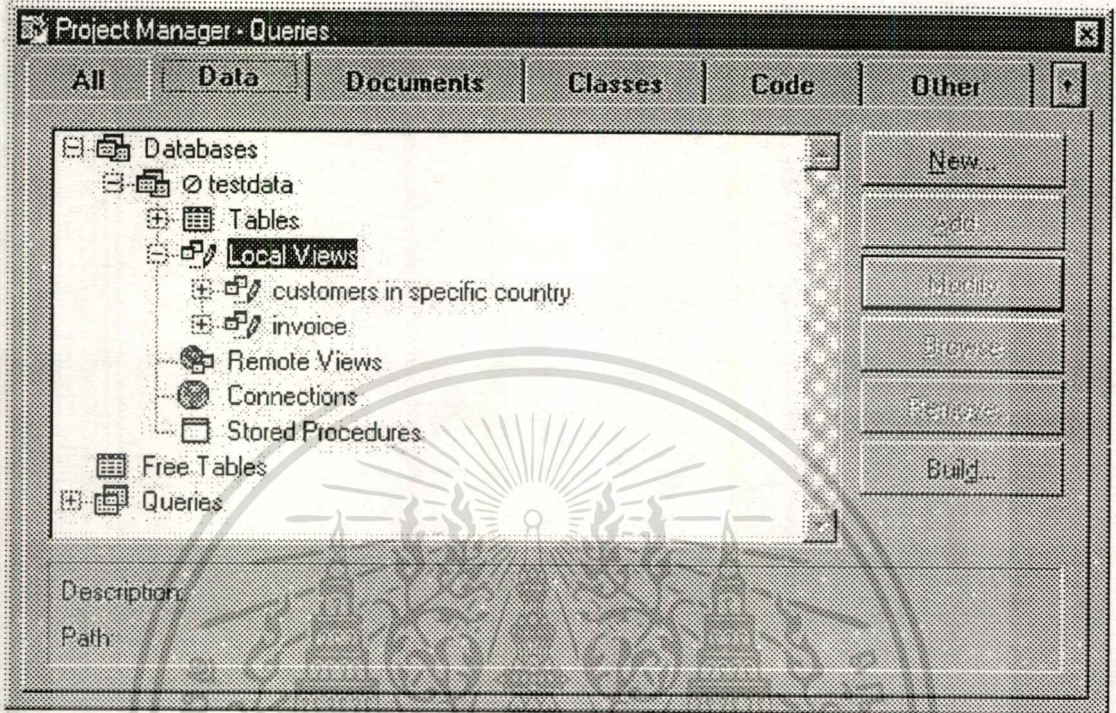
เอกสารนี้เป็นเอกสารดังกล่าว วิซวลฟอกซ์โปรจะจัดการให้เองโดยอัตโนมัติ ทำให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.8 แสดงการสร้าง Validation Rule

หมายเหตุ : Referential Integrity และ Validation Rule วิชาลฟอกซ์โปรจะสร้างให้โดยอัตโนมัติ และเก็บไว้ในส่วนที่เรียกว่า "Store Procedure" (คล้ายกับ Procedure File ในเวอร์ชันก่อน) ซึ่งจะถูกผูกติดอยู่กับค่าค่าเบสโดยอัตโนมัติ และสามารถเรียกใช้ได้ทันที โดยไม่ต้องใช้คำสั่ง SET PROCEDURE TO <Procedure Name> เช่นในเวอร์ชันก่อน ๆ

4. การสร้างวิว (View) ตามหลักการของฐานข้อมูลแบบรีเลชันนัลที่มีการแบ่งโครงสร้างของฐานข้อมูลออกเป็น 3 ระดับ คือ ระดับนอกสุด (External Level), ระดับหลักการ (Conceptual Level) และ ระดับในสุด (Internal Level) ดังที่ได้กล่าวมาแล้ว (ดูหน้า) วิชาลฟอกซ์โปรตั้งแต่เวอร์ชัน 3 เป็นต้นมา ก็มีความสามารถช่วยให้ผู้พัฒนาสร้างวิวสำหรับผู้ใ้ (Users) ในระดับต่าง ๆ ที่อาจมีการเรียกใช้ข้อมูลจากฐานข้อมูลเดียวกัน แต่ข้อมูลที่ใช้อาจแตกต่างกันได้



ภาพที่ 2.9 หน้าต่าง Project Manager แสดงการสร้างยูเออร์วิว

2.6.3 วิวลพอกซ์โปรกับภาษา SQL

นอกจากความสามารถที่เพิ่มขึ้นตามทฤษฎีฐานข้อมูลแบบรีเลชันนัลที่กล่าวมาแล้ว วิวลพอกซ์โปรยังมีความในการใช้ภาษา SQL ได้เป็นอย่างดี ทั้งการเรียกใช้แบบโต้ตอบโดยตรง (Interactive) หรือ ในแบบฝังเป็นส่วนหนึ่งของโปรแกรม (Embed) ที่พัฒนา (ดูการใช้ภาษา SQL หน้า) ทั้งนี้วิวลพอกซ์โปรยังคงยึดหลักการและรูปแบบของภาษา SQL ในแบบมาตรฐานมิได้มีการเปลี่ยนแปลง ดังนั้นวิธีการใช้และรูปแบบของภาษา SQL ใน วิวลพอกซ์โปรจึงเป็นแบบมาตรฐาน คำสั่งต่าง ๆ ในตารางที่ 2.2 จึงสามารถนำมาใช้ได้เกือบทั้งหมด

```

Object: frmpgfsam1 Procedure: update1ist

SELECT Menuitem.typenamet, Menuitem.itemserial, Menuitem.itemnamet,
      Menuitem.itemnamee, Menuitem.itemprice, Menuitem.itemunit
FROM restlmenutype, restlmenuitem ;
WHERE Menuitem.typeid = Menuitem.typeid and &mFdType = lcHigh ;
ORDER BY Menuitem.typenamet Menuitem.itemserial ;
INTO CURSOR 'Mycursor'

mFdName = iif(mLang, "itemnamet", "itemnamee")
mprice = "allt(str(itemprice))+ '/' +itemunit"

SELECT &mFdName, &mprice FROM Mycursor ;
INTO ARRAY THISFORM.aCustomers

```

ภาพที่ 2.10 ตัวอย่างการใช้ภาษา SQL ใน Visual FoxPro

2.6.4 วิชาลฟอกซ์โปรกับระบบเครือข่าย

นอกจากความสามารถพื้นฐานในการพัฒนาโปรแกรมในแบบผู้ใช้คนเดียว (Stand Alone) แล้ว วิชาลฟอกซ์โปรยังมีคำสั่งมากมายให้ผู้พัฒนาเลือกใช้ในการพัฒนาโปรแกรมประยุกต์ในแบบผู้ใช้หลายคน (Multi-user) หรือโปรแกรมที่ใช้ในระบบเครือข่าย เช่น คำสั่ง SET LOCK ON/OFF เพื่อใช้สำหรับการล็อกตาราง (Table Lock) ป้องกันการอัปเดต (UpDate) ข้อมูลซ้ำ หรือคำสั่งอื่น ๆ เช่น FLOCK(), LOCK(), RLOCK(), SET DATASESSION, SET MULTILOCKS (รายละเอียดการใช้งานโปรดศึกษาจากคู่มือวิชาลฟอกซ์โปรโดยตรง)

2.6.5 วิชาลฟอกซ์โปร กับ โคลเอนต์ / เซิร์ฟเวอร์ (Client / Server)

นอกจากคำสั่งที่ใช้ในระบบเครือข่ายแล้ว วิชาลฟอกซ์โปรยังสนับสนุนการพัฒนาโปรแกรมในแบบ โคลเอนต์ / เซิร์ฟเวอร์ เช่น ความสามารถในการสร้าง ไลคัลวิว (Local View) สำหรับติดต่อกับฐานข้อมูลที่อยู่บนเครื่องลูก (Client) ในขณะเดียวกันก็สามารถสร้างวิว ที่เรียกว่า รีโมตวิว (Remote View) สำหรับติดต่อกับฐานข้อมูลที่อยู่บนเครื่องแม่ (Server) ไม่ว่าฐานข้อมูลนั้นจะอยู่บนแพลตฟอร์มเดียวกันหรือต่างแพลตฟอร์มก็ตาม เช่น ออรัลคัล (Oracle), อินฟอรมิกซ์ (Informix), เอสคิวแอลเซิร์ฟเวอร์ (SQL Server) ของบริษัทไมโครซอฟต์ ฯลฯ โดยผ่านไคร์เวอร์ที่เรียกว่า ODBC (Open DataBase Connectivity) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการพัฒนาระบบงาน

3.1. บทนำ

ในบทที่ผ่านมาเราได้ศึกษาถึงทฤษฎีสารสนเทศต่าง ๆ ที่เกี่ยวข้องและรายละเอียดของเครื่องมือ (Development Tools) ที่จะใช้ในการสร้างโครงการงานขึ้นนี้ขึ้นมา สำหรับขั้นตอนในการพัฒนาระบบงานยังคงยึดแนวทางพื้นฐานของการพัฒนาระบบงานสารสนเทศ (Development Life Cycle) ซึ่งประกอบด้วย :

1. Strategy Planning การกำหนดแผนกลยุทธ์ และแนวทางในการพัฒนาระบบ
2. Feasibility Study ศึกษาความเป็นไปได้ ความเหมาะสมของการพัฒนา การพิจารณางบประมาณต่อประโยชน์ที่จะได้ (Cost & Benefits)
3. System Analysis การวิเคราะห์ระบบเป็นการกำหนดความต้องการของผู้ใช้ (User Requirements)
4. System Design การออกแบบระบบ หรือการกำหนดโครงร่าง (Blueprint) ของระบบที่จะพัฒนา เพื่อให้ระบบงานสามารถตอบสนองหรือทำงานได้ตามที่ผู้ใช้งานต้องการ
5. Implementation การนำระบบที่พัฒนาเสร็จแล้วไปใช้จริง ซึ่งรวมถึงการติดตั้งฮาร์ดแวร์ (Hardware) และการฝึกอบรมผู้ใช้ ให้รู้จักและเข้าใจระบบงานใหม่ที่นำมาใช้นี้
6. Maintenance การบำรุงรักษาระบบที่นำไปใช้แล้วให้สามารถทำงานได้อย่างมีประสิทธิภาพอยู่ตลอดเวลา รวมทั้งการแก้ไขข้อบกพร่องที่อาจยังมีอยู่ (Corrective Maintenance), เพิ่มเติมปรับปรุงระบบให้สมบูรณ์ยิ่งขึ้น (Perfective Maintenance) หรือ สุดท้ายการปรับปรุงระบบให้ดียิ่ง ๆ ขึ้น (Adaptive Maintenance)

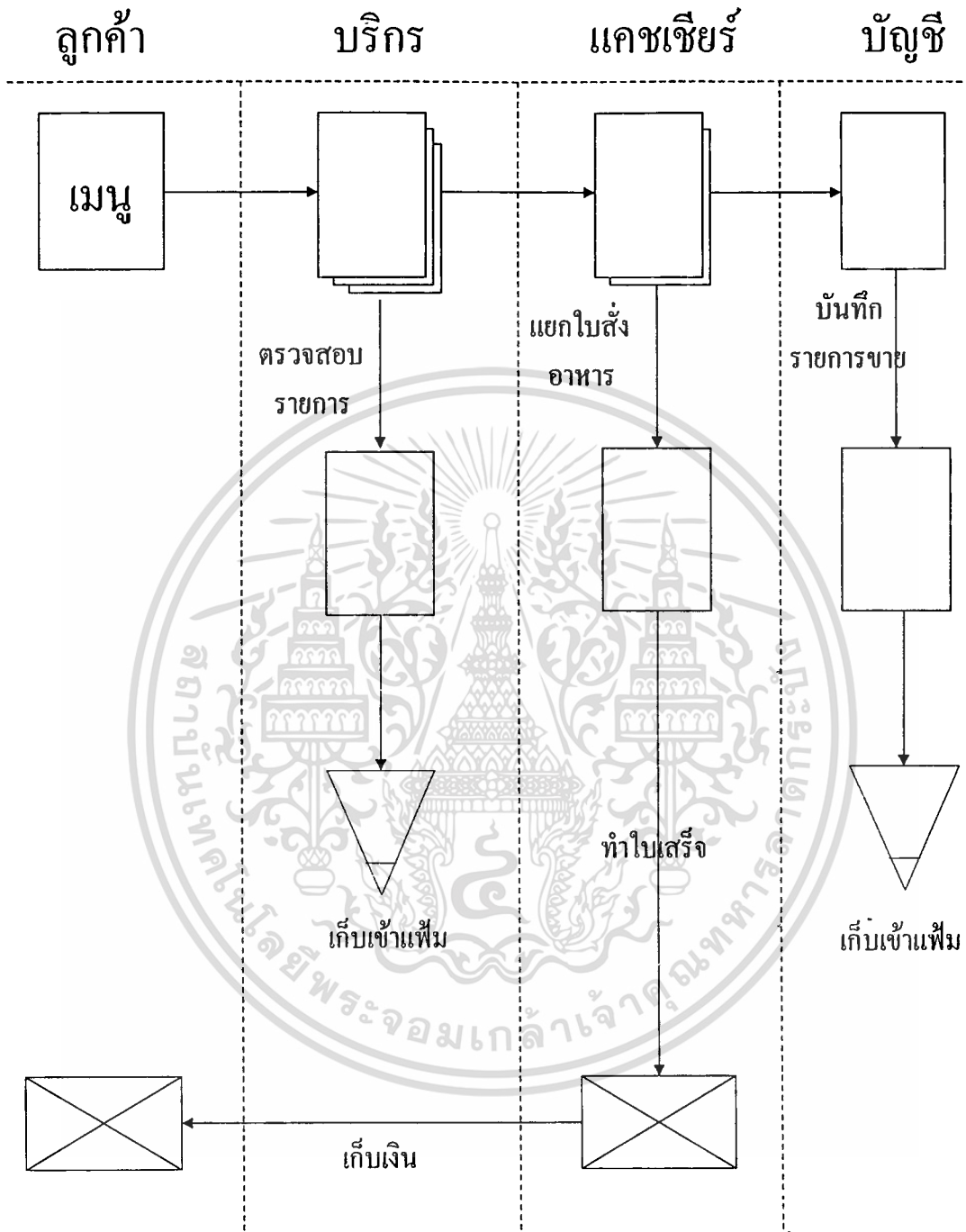
และดังได้กล่าวมาแล้วในบทที่ 1 โครงการฉบับนี้จะเน้นตั้งแต่ขั้นตอนที่ 4 คือการออกแบบระบบ (System Design) และการเขียนโปรแกรม (Coding) เนื่องจากเป็นการสร้างโปรแกรมคอมพิวเตอร์เพื่อช่วยเสริมหรือปรับปรุงการปฏิบัติงานเดิมที่เป็นอยู่ให้ดียิ่งขึ้น

3.2. ความสามารถพื้นฐานของระบบที่ต้องการ (Essential Requirements)

- เลือกรายการอาหารและเครื่องดื่มได้โดยตรงจาก หน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารหนึ่งของระบบงานสารสนเทศที่จัดทำขึ้นโดยผู้จัดทำเพื่อใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.1 แสดง Document Flow ของเอกสาร Captain Order

การทำงานของระบบจะเริ่มต้นจาก การนำรายการอาหารและเครื่องดื่มมาแสดงให้ลูกค้าเลือก โดยรูปแบบของการนำเสนอจะมีรูปแบบที่สวยงามเน้นสีสรร และภาพประกอบ พร้อมคำบรรยายชี้ชวน เช่นเดียวกับ เมนูอาหารแบบปกติในระบบเดิม หลังจากนั้นลูกค้าสามารถเลือกรายการอาหาร ๑ ที่ต้องการได้จากหน้าจอกอมพิวเตอร์โดยตรงจากระบบจอสัมผัส (Touch เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Screen) ซึ่งระบบเมนูอิเล็กทรอนิกส์ จะทำหน้าที่เก็บรวบรวมรายการอาหาร ๆ ที่ลูกค้าสั่งทั้งหมด ส่งไปเป็น Input ให้ระบบการเก็บเงิน (Cashiering System) เพื่อประมวลผลต่อไป โดยไม่ต้องผ่านพนักงานบริการ (Waiter/Waitress) และพนักงานเก็บเงิน (Cashier) โดยการใช้เอกสารใบรายการรับคำสั่งอาหาร (Captain Order) เป็น เอกสารนำเข้า (Input Document) เพื่อ Key in เข้าสู่ระบบการเก็บเงิน (Cashiering System) อีกต่อไป

นอกจากนี้ ระบบเมนูอิเล็กทรอนิกส์ ยังมีความสามารถในการเรียนแบบใบโฆษณา , แผ่นปลิว หรือ เอกสารการโฆษณาอาหารและเครื่องดื่มในแบบอื่น ๆ ที่มีจะจัดวางไว้บนโต๊ะอาหาร เพื่อดึงดูดความสนใจหรือกระตุ้นให้ลูกค้าสั่งอาหารและเครื่องดื่มชนิดนั้น ๆ โดยในระบบ ๆ นี้ผู้บริหารห้องอาหารสามารถ ทำรายการโปรโมชั่นอาหารหรือเครื่องดื่มของแต่ละวันตามต้องการได้ โดยง่าย และระบบ ๆ นี้ยังสามารถแสดงทั้งรูปภาพ และคำอธิบายหรือคำโฆษณาได้พร้อม ๆ กัน

ในส่วนการให้บริการแก่ลูกค้าผู้สั่งอาหารและเครื่องดื่ม ในระบบ ๆ นี้ ลูกค้าสามารถตรวจสอบรายการอาหารและเครื่องดื่มที่ตนได้สั่งไปแล้ว ได้ตลอดเวลา นอกจากนั้นยังสามารถตรวจสอบยอดรวมค่าใช้จ่ายของรายการทั้งหมดที่ตนเองต้องชำระได้พร้อม ๆ กันด้วย

3.4. สถาปัตยกรรมของระบบ ประกอบด้วย

1. ส่วนข้อมูล เป็นสถาปัตยกรรมแบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) กล่าวคือ ข้อมูลทั้งหมดที่ใช้ในระบบ ๆ จะถูกออกแบบให้จัดเก็บไว้ในรูปของตาราง (Table) 2 มิติ ที่ประกอบด้วย แถว (Row) หรือ เรคอร์ด (Record) และ คอลัมน์ (Column) หรือ ฟิวด์ (Field) โดยตารางเหล่านี้จะนำมารวมเข้ากันไว้เป็นฐานข้อมูล (Database) และข้อมูลที่เก็บไว้ในตารางต่าง ๆ ในฐานข้อมูลนี้จะมีความสัมพันธ์ (Relation) ซึ่งกันและกัน โดยลักษณะของความสัมพันธ์ อาจเป็นแบบ 1 ต่อ 1 (One to One), 1 ต่อ กลุ่ม (One to Many) หรือ กลุ่ม ต่อ กลุ่ม (Many to Many) ตามทฤษฎีฐานข้อมูลเชิงสัมพันธ์ที่ได้กล่าวไว้ในบทที่ 2

อย่างไรก็ตาม ฐานข้อมูลของโครงการนี้ ส่วนใหญ่จะมีความสัมพันธ์แบบ 1 ต่อ กลุ่ม หรือ 1 ต่อ 1 เท่านั้น โดยผู้พัฒนาพยายามหลีกเลี่ยงการออกแบบตารางมิให้เกิดความสัมพันธ์แบบ กลุ่มต่อกลุ่ม แม้ในขั้นตอนของการศึกษาระบบงานจะพบว่าข้อมูลบางส่วนอาจมีความสัมพันธ์แบบ กลุ่ม ต่อ กลุ่ม ผู้พัฒนา ก็จะพยายามปรับแต่ง (Tuning) โดยใช้วิธีการทำให้เป็นบรรทัดฐานที่เรียกว่า Normalization ตารางให้เกิดแต่ความสัมพันธ์ แบบ 1 ต่อ 1 หรือ 1 ต่อ กลุ่มเท่านั้น ทั้งนี้เพื่อประโยชน์ในการเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.2.2. **ItemName** ฟیلด์ชื่อรายการ เป็นฟیلด์ชนิดตัวอักษรใช้เก็บรายชื่ออาหารและเครื่องดื่ม ตามหมายเลขรหัสของรายการ (ItemCode)
- 1.2.3. **ItemPrice** ฟیلด์ราคาอาหารและเครื่องดื่ม เป็นฟیلด์ชนิดตัวเลข ใช้เก็บราคาอาหารและเครื่องดื่มแต่ละรายการ
- 1.2.4. **ItemRecipe** ฟیلด์ส่วนประกอบของอาหารและเครื่องดื่ม ชนิดตัวอักษร ใช้เก็บสูตรส่วนผสมของรายการอาหารและเครื่องดื่ม ชนิดนั้น ๆ
- 1.2.5. **ItemUnit** ฟیلด์หน่วยวัด ชนิดตัวอักษร ใช้เก็บหน่วยวัดของอาหารที่ขายชนิดนั้น ๆ เช่น จาน/ซาม, ถ้วย, โถ, แก้ว ฯลฯ

ตารางที่ 3.2 แสดงโครงสร้างของไฟล์ Menu Item

| ลำดับ | ชื่อฟیلด์ | ชนิด | ความยาว | ความยาวจุดทศนิยม |
|-------|------------|---------|-----------|------------------|
| 1 | ItemCode | String | 5 | - |
| 2 | ItemName | String | 30 | - |
| 3 | ItemPrice | Numeric | 4 | 0 |
| 4 | ItemRecipe | String | Unlimited | - |
| 5 | ItemUnit | String | 15 | - |

1.3. **Menu Item Picture Table** ตารางรูปภาพของรายการอาหารและเครื่องดื่ม ประกอบด้วยฟیلด์หลัก 2 ฟیلด์ คือ

- 1.3.1. **ItemCode** ฟیلด์รหัสของรายการอาหารและเครื่องดื่มในเมนู เป็นฟیلด์นอก (Foreign Key) ที่นำมาจากตาราง Menu Item Master Table ในหัวข้อ 1.2 เพื่อใช้เป็นตัวเชื่อมความสัมพันธ์ระหว่างตารางนี้ กับตารางอื่นที่เกี่ยวข้อง โดยใช้ฟیلด์นี้เป็นหลัก
- 1.3.2. **ItemImage** เป็นฟیلด์ชนิดตัวอักษร ใช้เก็บชื่อไฟล์รูปภาพ (Image File Name) ของรายการอาหารและเครื่องดื่มชนิดนั้น ๆ

ตารางที่ 3.3 แสดงโครงสร้างของไฟล์ Menu Item Picture

| ลำดับ | ชื่อฟิลด์ | ชนิด | ความยาว | ความยาวจุดทศนิยม |
|-------|-------------|--------|---------|------------------|
| 1 | ItemCode | String | 5 | - |
| 2 | ItemPicture | String | 15 | -- |

1.4. Captain Order Table ตารางรายการอาหารและเครื่องดื่มที่สั่ง เป็นตารางที่ใช้เก็บข้อมูลรายละเอียดของรายการอาหารและเครื่องดื่มที่ลูกค้าสั่งของแต่ละโต๊ะ ประกอบด้วยฟิลด์ย่อย 7 ฟิลด์ ดังต่อไปนี้

1.4.1. OrderNo ฟิลด์เลขที่ใบสั่งอาหาร ใช้เก็บข้อมูลเลขที่ของใบสั่งอาหารที่ลูกค้าสั่งในแต่ละโต๊ะ เป็นฟิลด์ชนิดตัวอักษร แต่บรรจุข้อมูลที่เป็นตัวเลข ความยาว 10 ตัวอักษร

1.4.2. TableNo ฟิลด์เลขที่โต๊ะ ใช้เก็บข้อมูลลำดับที่ของโต๊ะอาหารภายในห้องอาหารนั้น ๆ

1.4.3. ItemCode ฟิลด์รหัสของรายการอาหารและเครื่องดื่มในเมนู เป็นฟิลด์นอก (Foreign Key) ที่นำมาจากตาราง Menu Item Master Table ในหัวข้อ 1.2 เพื่อใช้เป็นตัวเชื่อมความสัมพันธ์ระหว่างตารางนี้ กับตารางอื่นที่เกี่ยวข้อง โดยใช้ฟิลด์นี้เป็นหลัก

1.4.4. ItemQty ฟิลด์จำนวน / ปริมาณ ใช้เก็บข้อมูลจำนวนหน่วยของรายการอาหารและเครื่องดื่มที่ลูกค้าสั่ง แต่ละรายการ เป็นฟิลด์ชนิดตัวเลข ความยาว 4 ตัวอักษร

1.4.5. OrderDate ฟิลด์วันที่ ใช้เก็บข้อมูลวันที่ของใบสั่งอาหาร (Captain Order) เป็นฟิลด์ชนิดวันที่ ความยาว 8 ตัวอักษร

1.4.6. OrderTime ฟิลด์เวลา ใช้เก็บข้อมูลเวลาของใบสั่งอาหารแต่ละใบ

1.4.7. OrderFlag เป็นฟิลด์ชนิดตรรก (Logical Type) ใช้เก็บค่า จริง กับเท็จ สำหรับระบบเป็นตัว Mark ใบสั่งอาหารที่ส่งไปยังระบบ เก็บเงิน (Cashiering System) แล้วหรือยัง

ตารางที่ 3.4 แสดงโครงสร้างของไฟล์ Captain Order

| ลำดับ | ชื่อฟิลด์ | ชนิด | ความยาว | ความยาวจุดทศนิยม |
|-------|----------------|---------|---------|------------------|
| 1 | <u>OrderNo</u> | String | 10 | - |
| 2 | <u>TableNo</u> | String | 3 | - |
| 3 | ItemCode | String | 5 | - |
| 4 | ItemQty | Numeric | 4 | 0 |
| 5 | OrderDate | Date | 8 | - |
| 6 | OrderTime | Numeric | 8 | - |
| 7 | OrderFlag | Logical | 1 | - |

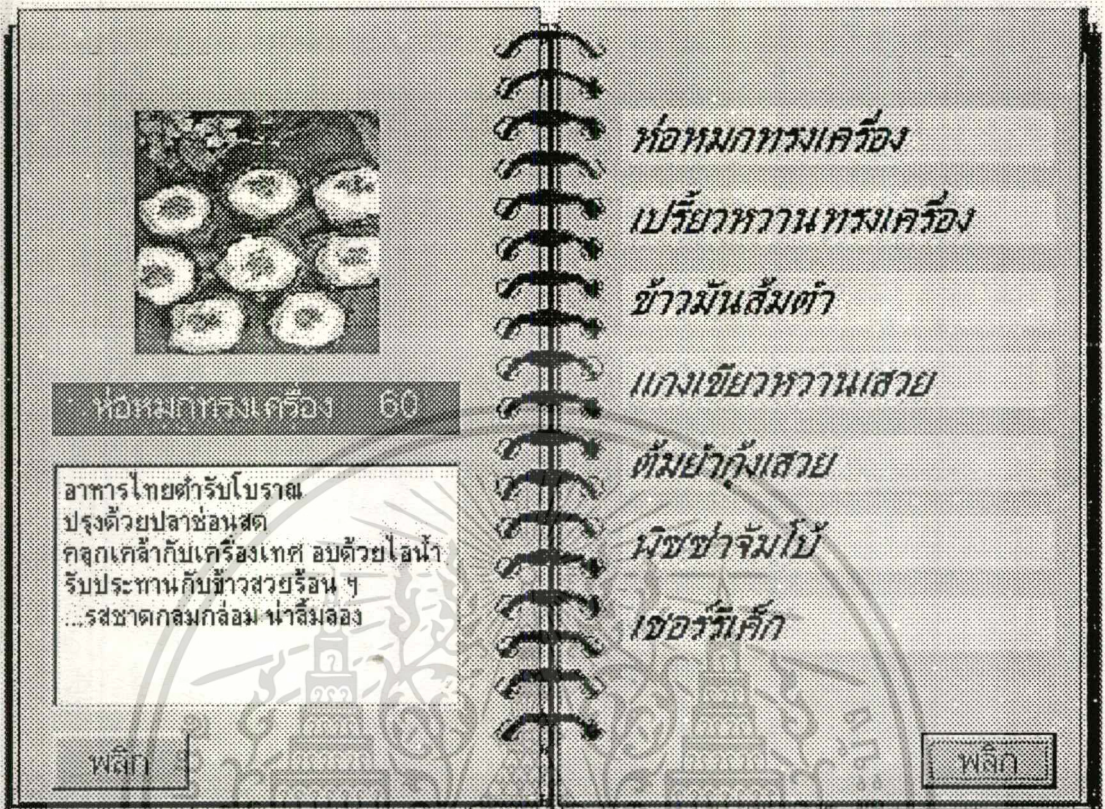
1.5. Promoted Item Table ตารางรายการอาหารและเครื่องดื่มที่ต้องการส่งเสริมการขาย (Promotion) เป็นตารางที่ใช้เก็บข้อมูลรายละเอียดของรายการอาหารและเครื่องดื่มที่ต้องการส่งเสริมการขายของแต่ละวัน ประกอบด้วยฟิลด์ย่อยเพียง 2 ฟิลด์ ดังนี้

1.5.1. ItemCode ฟิลด์รหัสของรายการอาหารและเครื่องดื่มในเมนู เป็นฟิลด์นอก (Foreign Key) ที่นำมาจากตาราง Menu Item Master Table ในหัวข้อ 1.2 เพื่อใช้เป็นตัวเชื่อมความสัมพันธ์ระหว่างตารางนี้ กับตารางอื่นที่เกี่ยวข้อง โดยใช้ฟิลด์นี้เป็นหลัก

1.5.2. PromoteDate ฟิลด์วันที่ ใช้เก็บข้อมูลวันของสัปดาห์ เช่น อาทิตย์, จันทร์, อังคาร เป็นต้น

ตารางที่ 3.5 แสดงโครงสร้างของไฟล์ Promoted Item

| ลำดับ | ชื่อฟิลด์ | ชนิด | ความยาว | ความยาวจุดทศนิยม |
|-------|-----------------|--------|---------|------------------|
| 1 | <u>ItemCode</u> | String | 5 | - |
| 2 | PromoteDate | String | 1 | - |



ภาพที่ 3.4 สมุดเมนูจากโปรแกรม

2.2. ส่วนของโมดูล หรือ ระบบย่อย (Module or Subsystem) เป็นส่วนประกอบหลักของระบบที่จะทำหน้าที่ตั้งแต่ นำรายละเอียดรายการอาหาร ๆ มาแสดงให้ลูกค้าเลือกบนจอภาพ เมื่อลูกค้าเลือกรายการที่ต้องการแล้ว ระบบจะทำหน้าที่รวบรวมรายการทั้งหมดที่ลูกค้าสั่ง ส่งเข้าไปในระบบ Cashiering System ประกอบด้วย โมดูลหลักที่สำคัญดังนี้

2.2.1. Promotion Module เป็นโมดูลที่ทำหน้าที่แสดงรายละเอียดของรายการอาหารและเครื่องดื่มที่ต้องการทำการโปรโมทของแต่ละวัน

2.2.2. Item List Module เป็นโมดูลที่ทำหน้าที่นำรายละเอียดของรายการอาหารและเครื่องดื่มจากรายชื่อข้อมูลมาแสดงให้ลูกค้าเลือกบนจอภาพ

2.2.3. Order Checking Module เป็นโมดูลที่ทำหน้าที่นำรายการอาหารและเครื่องดื่มที่ลูกค้าเลือกส่งไปแล้วมาแสดงอีกครั้งหากลูกค้าต้องการเพื่อ

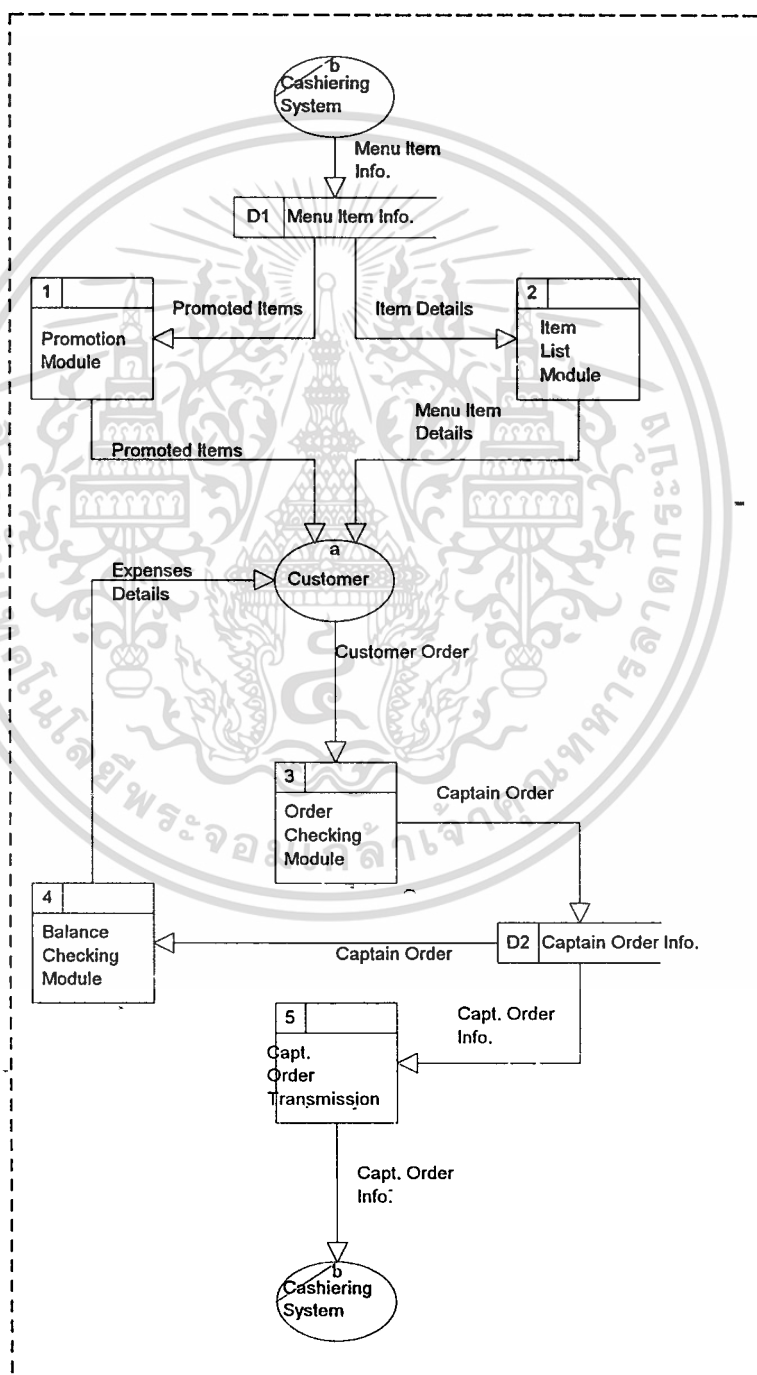
เป็นการตรวจสอบความถูกต้องของรายการที่ส่งไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในระบบของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4. Balance Checking Module เป็นโมดูลที่ทำหน้าที่คำนวณยอดค่าใช้จ่ายของค่าอาหารและเครื่องดื่มของลูกค้าแต่ละโต๊ะ

2.2.5. Captain Order Transmission เป็นโมดูลที่ทำหน้าที่เก็บรวบรวมรายการอาหารและเครื่องดื่มที่ลูกค้าสั่งทั้งหมด เพื่อส่งข้อมูลเข้าสู่ระบบ Cashiering System



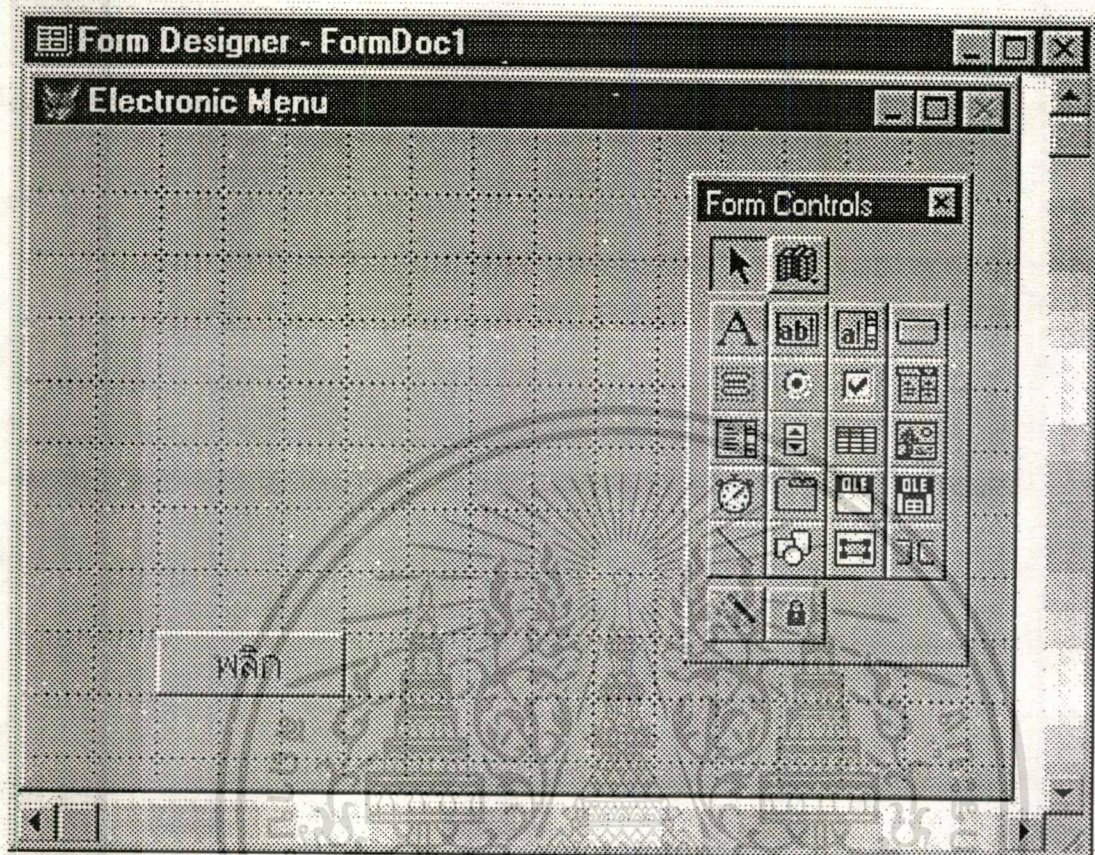
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาวิจัยเท่านั้น ไม่ใช่ออกมาให้หน้าไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5. การเขียนโปรแกรม

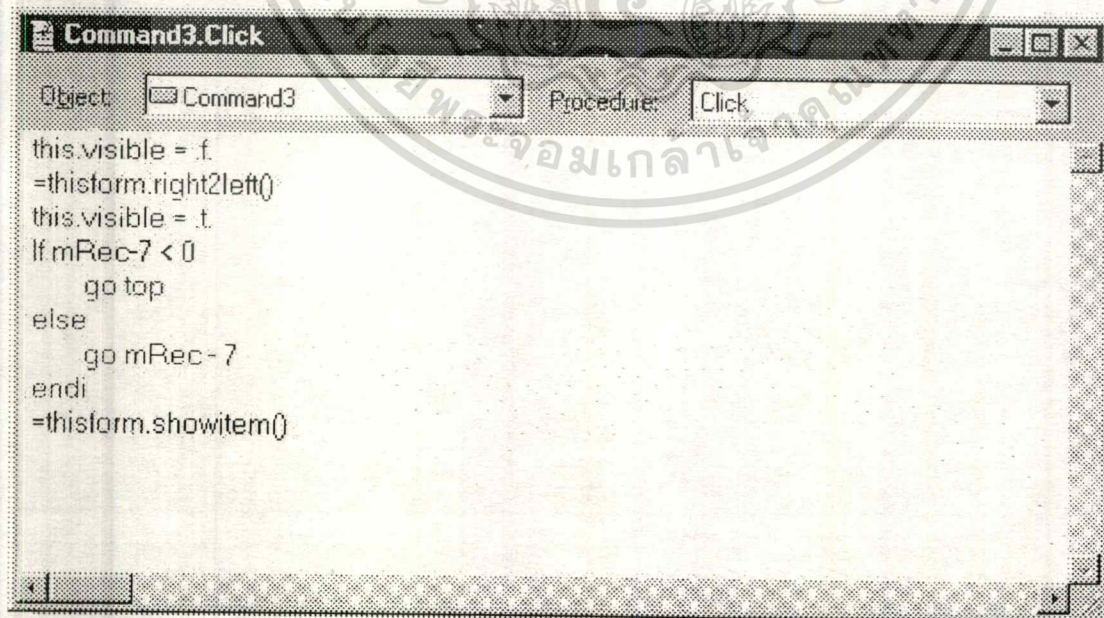
ดังได้กล่าวมาแล้วว่าเครื่องมือหลักที่ใช้พัฒนา (Development Tools) ระบบเมนูอิเล็กทรอนิกส์นี้ คือ โปรแกรม ไมโครซอฟท์ วิวอลฟอกซ์โปร เวอร์ชัน 5 ของบริษัท ไมโครซอฟท์ ซึ่งผู้พัฒนาเห็นว่ามีความเหมาะสมในหลาย ๆ ด้าน อีกทั้งตัวโปรแกรมเองก็มีขีดความสามารถ (Features) หลายประการ สำหรับการสร้างหรือพัฒนาระบบงาน ตามทฤษฎีสารสนเทศต่าง ๆ ดังที่ได้กล่าวไว้ในบทที่ 2 ในที่นี่จะอธิบายถึงการเขียนโปรแกรมและการนำคำสั่งหรือเทคนิคต่าง ๆ ของ ไมโครซอฟท์ วิวอลฟอกซ์โปร มาพัฒนาโครงการนี้ พอสังเขป สำหรับผู้สนใจนำไปใช้ในโอกาสต่อไป ส่วนรายละเอียดปลีกย่อยสามารถศึกษาได้จากคู่มือการใช้ โดยตรง

1. แนวทางการพัฒนาเป็นแบบ Object Oriented Programming (OOP) Concept

โปรแกรมระบบเมนูอิเล็กทรอนิกส์ได้รับการพัฒนามาโดยยึดหลักแนวคิดแบบ OOP ที่มีการแบ่งโปรแกรมออกเป็นส่วนย่อย ๆ ซึ่งแต่ละส่วนมีหน้าที่ (Function) เฉพาะที่จะทำงานในแต่ละด้าน เมื่อเกิดเหตุการณ์ (Event) อย่างใดอย่างหนึ่งตามที่ถูกพัฒนาเขียนโปรแกรมไว้ล่วงหน้า ที่เรียกว่า Event Driven โดยคำสั่งต่าง ๆ ที่ผู้พัฒนาเขียนไว้นี้จะผูกติดและเป็นส่วนหนึ่งของ วัตถุ (Object) ต่าง ๆ ที่ประกอบขึ้นเป็นส่วนหนึ่งของโปรแกรม เช่น การสร้าง User Interface ในวิวอลฟอกซ์โปรจะเริ่มจากการนำ หน้าต่าง (Windows) หรือที่เรียกใน วิวอลฟอกซ์โปร ว่า ฟอรั่ม (Form) ซึ่งเป็นวัตถุ (Object) ชนิดหนึ่งที่สามารถบรรจุ (Contain) ออปเจกต์อื่น ๆ เช่น ปุ่มกด (Button), กล่องแสดงข้อความ (Text Box) หรือ รูปภาพ (Image Control) ฯลฯ ที่เรียกว่า Form Controls เพื่อประกอบหรือสร้างขึ้นเป็นหน้าต่างสำหรับติดต่อกับผู้ใช้ โดยในแต่ละ Controls นี้ผู้พัฒนาสามารถกำหนดคุณลักษณะ (Property) และเขียนโปรแกรมฝังไว้ในตัว Controls นั้นเพื่อให้คำสั่งของโปรแกรมหักการทำงานเมื่อผู้ใช้ระบบกระทำการอย่างใดอย่างหนึ่งที่กำหนดไว้ล่วงหน้ากับ Controls นั้น เช่น ผู้ใช้กดปุ่มกด หรือ ชีที่ปุ่มกด นั้น โปรแกรมที่เขียนไว้ก็จะทำงานตามคำสั่งที่เขียนไว้ทันที เป็นต้น

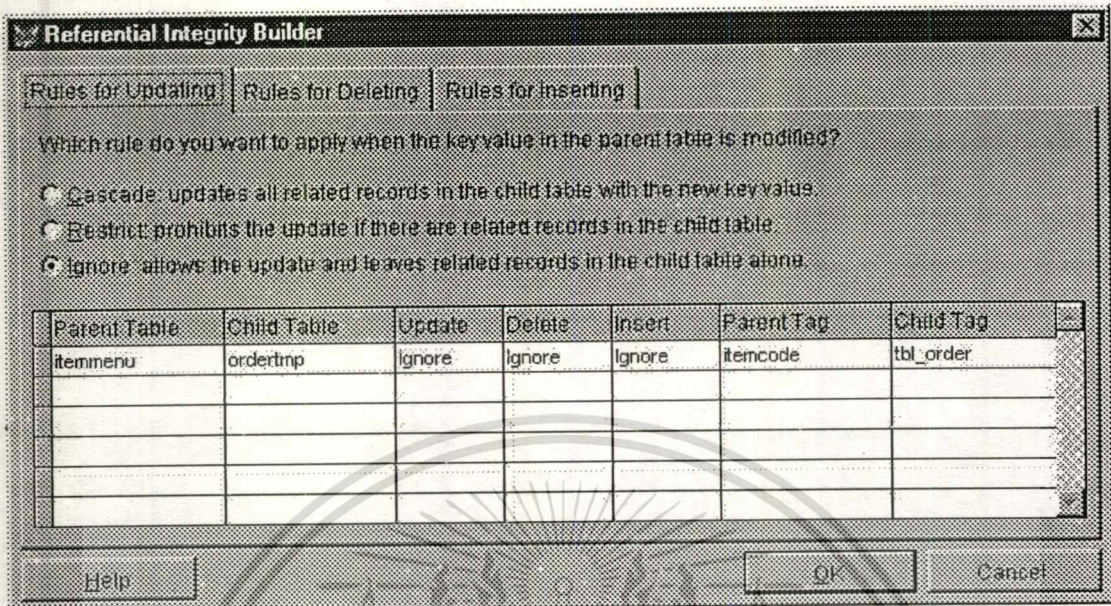


ภาพที่ 3.6 แสดงการออกแบบหน้าต่างฟอร์มและปุ่มกด "คลิก"

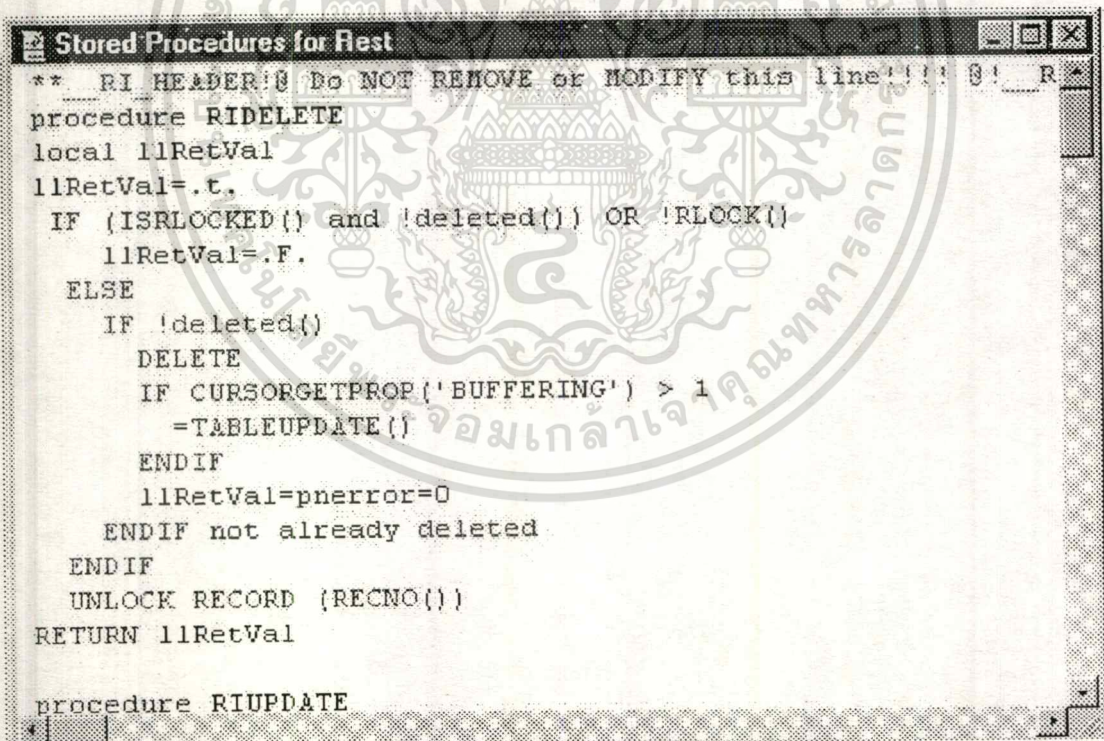


ภาพที่ 3.7 แสดงหน้าต่างการเขียนโปรแกรมสำหรับปุ่ม "คลิก"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

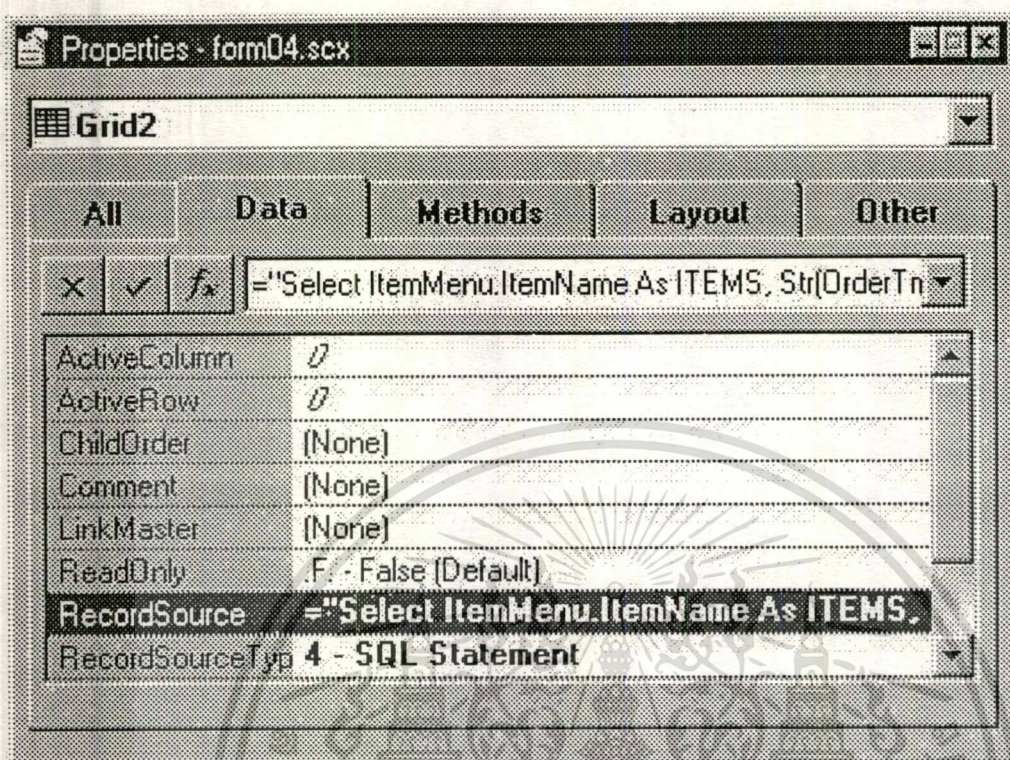


ภาพที่ 3.10 แสดงหน้าต่างการกำหนด Referential Integrity




ภาพที่ 3.11 รายละเอียดคำสั่งในไฟล์ Store Procedure

อย่างไรก็ตามคำสั่งในไฟล์ Store Procedure นี้ผู้พัฒนายังสามารถเข้าไปทำการแก้ไขคำสั่งที่วิซวลฟอกซ์เขียนให้ได้โดยตรง หรือจะใช้ไฟล์นี้เป็นที่เก็บ Function หรือ Subroutine ย่อยอื่นๆ ที่มีการเรียกใช้บ่อยๆ ในระบบงาน เช่น Error Handle ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.12 ตัวอย่างการใช้ SQL ในโมดูลหนึ่งของระบบ

5. หลีกเลี่ยงการสร้างฟิลด์ประเภท General เพื่อเก็บไฟล์รูปภาพไว้กับตาราง (Table) โดยตรง เพราะจะทำให้ไฟล์ประเภท .FPT ซึ่งเป็นไฟล์ประกอบของตารางหลัก (.DBF ไฟล์) มีขนาดใหญ่เพราะต้องเก็บข้อมูลรูปภาพไว้กับทุกเรคอร์ดในตาราง ทำให้การเรียกใช้ตาราง และการเข้าถึงข้อมูล (Access) ต้องใช้เวลานานขึ้น เทคนิคที่ผู้พัฒนานำมาใช้เพื่อแก้ปัญหานี้คือ สร้างฟิลด์ประเภทอักษร (Character) ใช้เก็บชื่อไฟล์ของรูปภาพของเรคอร์ดนั้น ๆ และเมื่อต้องการแสดงภาพ ให้ใช้ Control สำหรับแสดงรูปภาพ  Image Control ร่วมกับ Property Picture ของ Image Control เพื่อโหลด ภาพตามชื่อของไฟล์ที่อ่านได้จากฟิลด์อักษรของเรคอร์ดนั้น ๆ โดยใช้รูปแบบคำสั่ง ดังนี้

THISFORM.Image1.Picture = ชื่อฟิลด์ที่เก็บชื่อไฟล์รูปภาพ

บทที่ 4

การใช้งานระบบเมนูอิเล็กทรอนิกส์

4.1. ความต้องการของระบบ (System Requirements)

เนื่องด้วยระบบเมนูอิเล็กทรอนิกส์เป็นระบบงานที่ได้พัฒนามาในแนวทางของไคลเอนต์เซิร์ฟเวอร์ ดังนั้นเครื่องคอมพิวเตอร์ที่จะใช้ในการรัน โปรแกรมจะแยกเป็น 2 ประเภทหลักดังนี้คือ

1. เครื่องไคลเอนต์ (Client Computer) เนื่องจาก ระบบงานนี้เป็นระบบงานประเภทคอมพิวเตอร์กราฟิกด้วยต้องแสดงลักษณะของเมนูอาหารและเครื่องดื่ม ให้เหมือนจริงมากที่สุด และต้องมีสีสรรที่ดึงดูดความสนใจของผู้ใช้เพื่อใช้ทดแทนระบบเมนูเดิม ซึ่งส่วนใหญ่จะมีรูปแบบที่สวยงามอยู่แล้ว ดังนั้น ข้อกำหนด (Specification) ของเครื่องไคลเอนต์ จึงควรใช้เครื่องที่มีสมรรถสูงพอควร ควรเป็นเครื่องไคลเอนต์ประเภทที่ในทางเทคนิค เรียกว่า Fat Client ซึ่งจะมีความสามารถและสมรรถทั่ว ๆ สูงกว่า เครื่องไคลเอนต์ ประเภท Thin Client โดยควรมีข้อกำหนดอย่างน้อยที่สุด ดังนี้
 - 1.1. เป็นเครื่อง IBM คอมแพตทิเบิล (Compatible) ที่มีหน่วยประมวลผลกลาง (Central Processor Unit : CPU) ชนิด 80486DX2 หรือสูงกว่า และใช้สัญญาณนาฬิกาหรือความเร็วของซีพียู อย่างน้อย 130 เมกะเฮิร์ตส (MHz)
 - 1.2. มีหน่วยความจำหลักของระบบ (Main Memory) อย่างน้อย 8 MB (แนะนำ 16 MB)
 - 1.3. จอแสดงผลแบบ Super VGA ที่มีความละเอียดของการแสดงผล (Resolution) อย่างน้อย 640 x 480 พิกเซล (Pixels) และสนับสนุนการแสดงผลอย่างน้อย 256 สี (แนะนำให้ใช้จอภาพแบบสัมผัส สำหรับเครื่องของลูกค้า)
 - 1.4. หน่วยเก็บข้อมูล (Storage Unit) หรือ ฮาร์ดดิสก์ มีเนื้อที่ว่างอย่างน้อย 32 MB สำหรับการติดตั้งส่วนประกอบทั้งหมดของ ข้อมูลระบบและตัว โปรแกรม
 - 1.5. ติดตั้งการ์ดระบบเครือข่าย (LAN Card / LAN Adapter) สำหรับเชื่อมต่อเป็นระบบเครือข่าย
 - 1.6. ระบบปฏิบัติการ แบบ Windows 95 หรือ Windows NT
 - 1.7. เมาส์ (Mouse) หรืออุปกรณ์นำเข้า ประเภท Pointer Device อื่นที่ทำงานคล้ายคลึงหรือเหมือนกับเมาส์ ที่ใช้ทั่วไปกับ โปรแกรมวินโดวส์ (ถ้าต้องการ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เครื่องเซิร์ฟเวอร์ (Server Computer) จะมีลักษณะเป็นศูนย์กลางของระบบ ที่ทำหน้าที่ประสานการทำงานระหว่างเครื่องคอมพิวเตอร์ และอุปกรณ์รอบข้าง (Peripheral) ที่เชื่อมต่ออยู่ในระบบ รวมทั้งเป็นที่จัดเก็บข้อมูลหรือ ไฟล์หลักต่าง ๆ ประเภท Master File ทั้งข้อมูลของรายการอาหาร และ ไฟล์รูปภาพประกอบ โดยเฉพาะอย่างยิ่ง เครื่องนี้ จะทำหน้าที่ประสานการติดต่อเพื่อส่งข้อมูล ระหว่างระบบเมนูอิเล็กทรอนิกส์ กับระบบเก็บงาน (Cashiering System) ที่มีอยู่เดิม อย่างไรก็ตาม ในระยะแรกของการ Implement ระบบงานเมนูอิเล็กทรอนิกส์ หน้าที่การทำงานของเครื่องเซิร์ฟเวอร์นี้อาจนำไปฝากไว้กับเครื่อง เซิร์ฟเวอร์ เดิมของระบบ Cashiering System ก่อนก็ได้ หากปริมาณข้อมูลของระบบไม่มากนัก เช่น จำนวนรายการอาหารเครื่องดื่มในเมนู หรือจำนวนเครื่อง โคลเอนต์ ที่ติดตั้งตามโต๊ะของแต่ละห้องอาหารยังมีจำนวนน้อย อย่างไรก็ตาม สำหรับข้อกำหนดของเครื่อง เซิร์ฟเวอร์สำหรับ ห้องอาหารขนาดกลาง (จำนวนโต๊ะประมาณ 30 - 50 โต๊ะ) ควรมีข้อกำหนดอย่างน้อย ดังต่อไปนี้

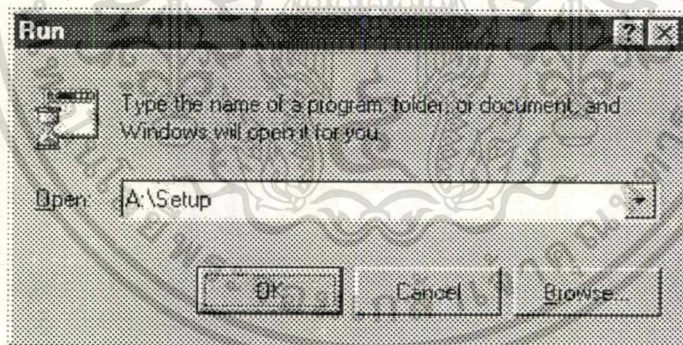
- 2.1. หน่วยประมวลผลกลาง (Central Processing Unit) เป็นแบบ เพนเทียมโปร (Pentium Pro Processor) ทำงานด้วยความเร็วของสัญญาณนาฬิกา ไม่นต่ำกว่า 200 MHz หรือเทียบเท่า
- 2.2. มีหน่วยความจำหลัก (Main Memory) ไม่น้อยกว่า 64 MB และสามารถขยายได้รวมกันแล้วไม่น้อยกว่า 1 Gbyte และจะต้องมีหน่วยตรวจสอบและแก้ไขข้อมูลที่ผิดพลาด (Error Checking and Correction)
- 2.3. มีหน่วยความจำแคชภายนอก (Secondary Cache and External Cache) ไม่น้อยกว่า 1 MB
- 2.4. มี I/O Slot Interface สำหรับเชื่อมต่อกับแผ่นควบคุมอุปกรณ์ภายนอกเพิ่มเติมได้ไม่น้อยกว่า 6 Slots และสามารถเพิ่มขยายได้สูงสุดไม่น้อยกว่า 20 Slots
- 2.5. มีหน่วยอ่านและบันทึกข้อมูลแบบจานแม่เหล็ก (Internal Disk Storage) ชนิด Fast/Wide SCSI-2 หรือดีกว่า ขนาดความจุไม่น้อยกว่า 2 GB ซึ่งมีอัตราการเข้าถึงข้อมูล (Average Seek Time) ไม่เกินกว่า 10 ms.
- 2.6. มีอุปกรณ์ควบคุมหน่วยอ่านและบันทึกข้อมูลแบบจานแม่เหล็ก (Hard Disk Controller) ชนิด Fast/Wide SCSI-2 หรือดีกว่า ซึ่งมีอัตราการถ่ายโอนข้อมูล (Data Transfer Rate) ไม่นต่ำกว่า 20MB/Sec และสามารถรองรับการเพิ่มขยายความจุของหน่วยอ่านและบันทึกข้อมูลแบบจานแม่เหล็กรวมสูงสุดได้ไม่น้อย

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


- 2.6. มีอุปกรณ์ควบคุมหน่วยอ่านและบันทึกข้อมูลแบบจานแม่เหล็ก (Hard Disk Controller) ชนิด Fast/Wide SCSI-2 หรือดีกว่า ซึ่งมีอัตราการถ่ายโอนข้อมูล (Data Transfer Rate) ไม่ต่ำกว่า 20MB/Sec และสามารถรองรับการเพิ่มขยายความจุของหน่วยอ่านและบันทึกข้อมูลแบบจานแม่เหล็กรวมสูงสุดได้ไม่น้อยกว่า 4 GB
- 2.7. มีเครื่องขับซีดีรอม (CD-ROM Drive) ซึ่งทำงานด้วยความเร็วไม่ต่ำกว่า 4 เท่าตามมาตรฐาน (4X) และมีความจุไม่ต่ำกว่า 600 MB
- 2.8. จอภาพสี ขนาดไม่น้อยกว่า 15 นิ้ว
- 2.9. มี Connector Port สำหรับการเชื่อมต่อเป็นระบบเครือข่าย
- 2.10. ใช้ได้กับระบบปฏิบัติการ ทั้ง WINDOWS NT หรือ UNIX

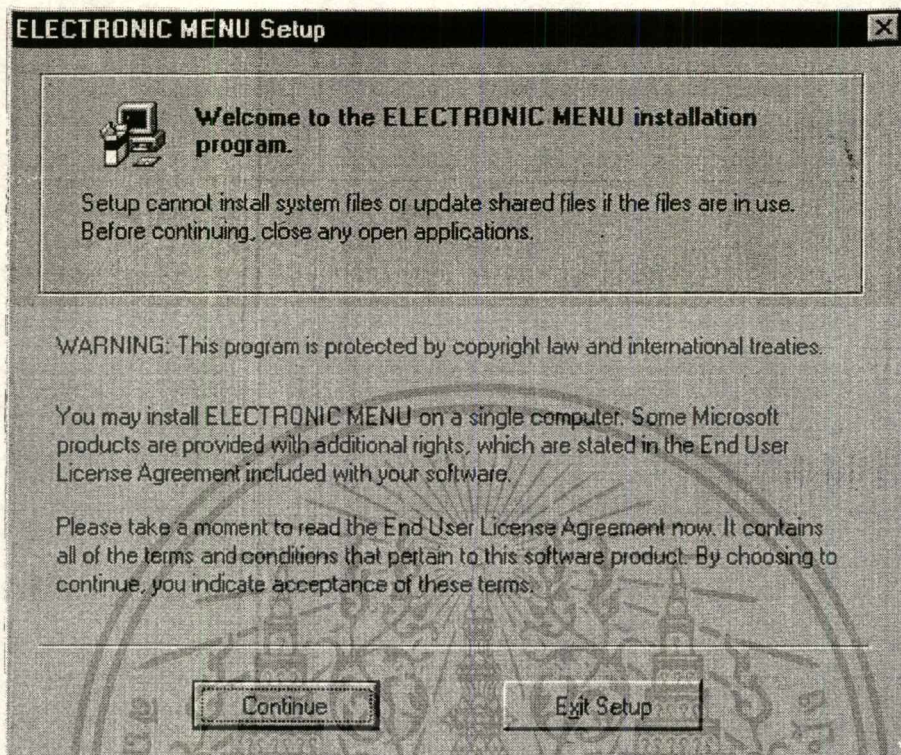
4.2. การติดตั้งโปรแกรม (Installation)

การติดตั้งโปรแกรมระบบเมนูอิเล็กทรอนิกส์ นั้นทำได้ง่ายมาก เพียงรันไฟล์ติดตั้งที่ชื่อ SETUP.EXE ในดิสก์แผ่นแรกของโปรแกรม จากหน้าต่าง Run ของ Windows ดังภาพที่ 4.1

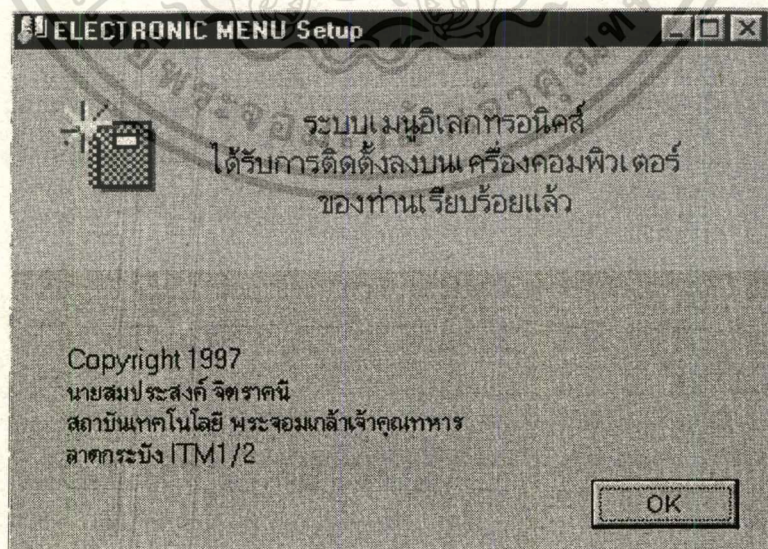


ภาพที่ 4.1 หน้าต่าง Run สำหรับเรียกโปรแกรม Setup

หลังจากกดปุ่ม  แล้วหน้าจอจะปรากฏหน้าต่างของการติดตั้ง ดังภาพที่ 4.2 ให้ดำเนินการตามคำแนะนำจนเสร็จสิ้นกระบวนการติดตั้ง จนหน้าจอแสดงหน้าต่างติดตั้งระบบเรียบร้อยแล้ว ดังภาพที่ 4.3



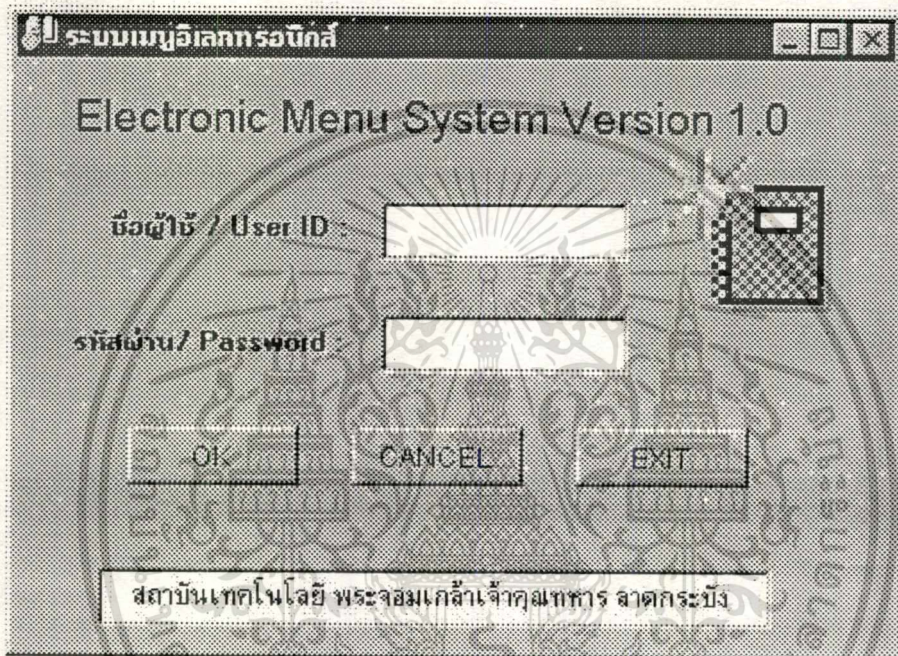
ภาพที่ 4.2 หน้าต่างแรกของการติดตั้งระบบเมนูอิเล็กทรอนิกส์



เอกสารนี้เป็นเอกสารที่ภาพที่ 4.3 หน้าต่าง เสร็จสิ้นกระบวนการติดตั้งระบบเมนูอิเล็กทรอนิกส์ ขั้นตอนการคำ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3. การใช้งาน

- 4.3.1. เริ่มต้นใช้งาน การใช้งานระบบเมนูอิเล็กทรอนิกส์ เมื่อพนักงานบริการ (Waiter) ได้นำลูกค้าเข้ามาที่นั่งที่โต๊ะอาหารเรียบร้อยแล้ว พนักงานจะเริ่มเปิดสวิตซ์ที่เครื่องคอมพิวเตอร์ หลังจากนั้นหน้าจอภาพจะปรากฏหน้าต่างแรก รอให้พนักงานผู้เปิดเครื่องใส่ชื่อและรหัส เพื่อเข้าสู่การทำงานของระบบ ฯ ต่อไป ดังภาพที่ 4.4



ภาพที่ 4.4 หน้าต่าง การ Log in เพื่อเข้าสู่ระบบเมนูอิเล็กทรอนิกส์

เมื่อพนักงานใส่รหัสเพื่อเปิดระบบได้ถูกต้องแล้ว หน้าต่างแรกของระบบเมนูอิเล็กทรอนิกส์ จะปรากฏขึ้น ดังภาพที่ 4.5

เมนูอิเล็กทรอนิกส์
วันอาทิตย์ ...เซอร์รีเด็ก

 **รายการอาหารและเครื่องดื่ม**

 **รายการที่สั่งไปแล้ว**

 **เช็คบิล**

 **ออกจากระบบ**

(02) 693-1681 จองโต๊ะล่วงหน้า



จ อ พ พง ศ ส ลา

ภาพที่ 4.5 แสดงหน้าจอแรกของระบบเมนูอิเล็กทรอนิกส์

หน้าจอแรกของระบบเมนูอิเล็กทรอนิกส์ มีรายการให้ลูกค้าเลือกอย่างใดอย่างหนึ่งดังต่อไปนี้ คือ

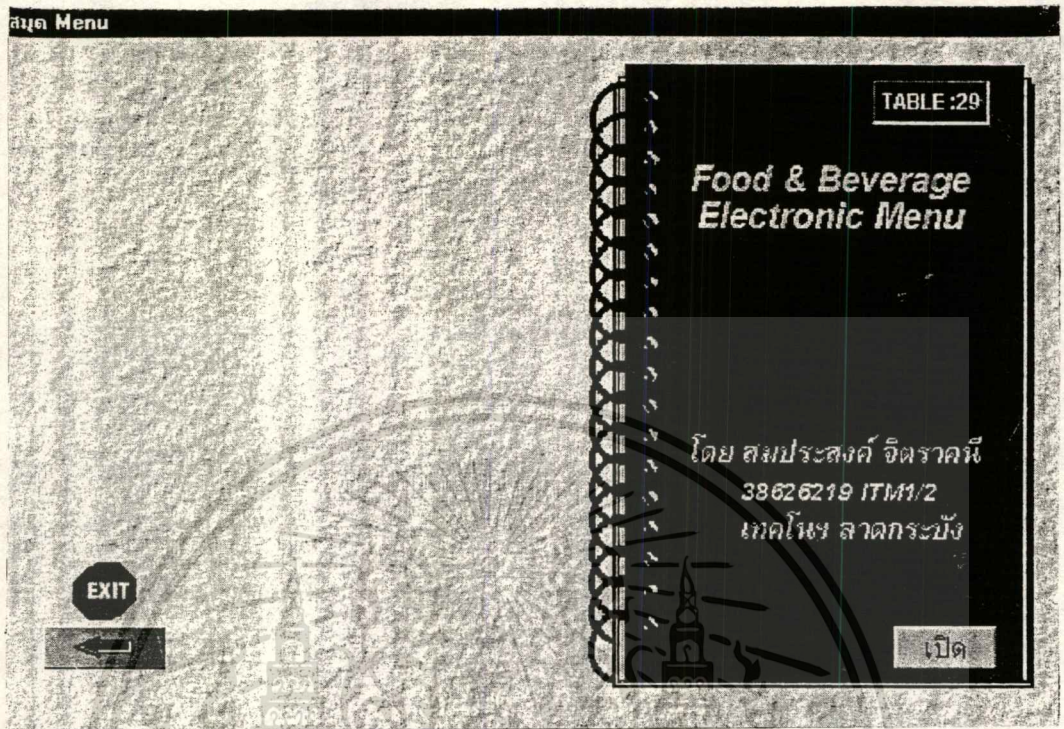
- แสดงเมนูอาหารและเครื่องดื่ม
- ตรวจสอบรายการอาหาร ฯ ที่สั่งไปแล้ว
- เช็คยอดค่าใช้จ่ายค่าอาหารและเครื่องดื่มที่ต้องชำระ
- ออกจากระบบ

4.3.2. การโฆษณาอาหาร ฯ พิเศษประจำวัน นอกจากนั้น ในหน้าจอแรกนี้จะมีภาพโฆษณาห้องอาหาร เช่น หมายเลขโทรศัพท์เพื่อโทรจองโต๊ะล่วงหน้า และรูปภาพรายการอาหารและเครื่องดื่มที่เป็นอาหารพิเศษประจำวันในสัปดาห์ โดยรูปภาพและชื่ออาหารจะเปลี่ยนไปโดยอัตโนมัติ ตามวันที่ของระบบ และหากลูกค้าต้องการทราบรายการอาหาร ฯ พิเศษของวันอื่น ๆ ก็เพียงแต่เข้าไปที่ชื่อวันของสัปดาห์บนจอภาพด้านล่างขวา ภาพของรายการอาหาร ฯ พร้อมชื่ออาหาร ก็จะเปลี่ยนไปตามวันที่ที่ลูกค้าชี้ ดังภาพที่ 4.6



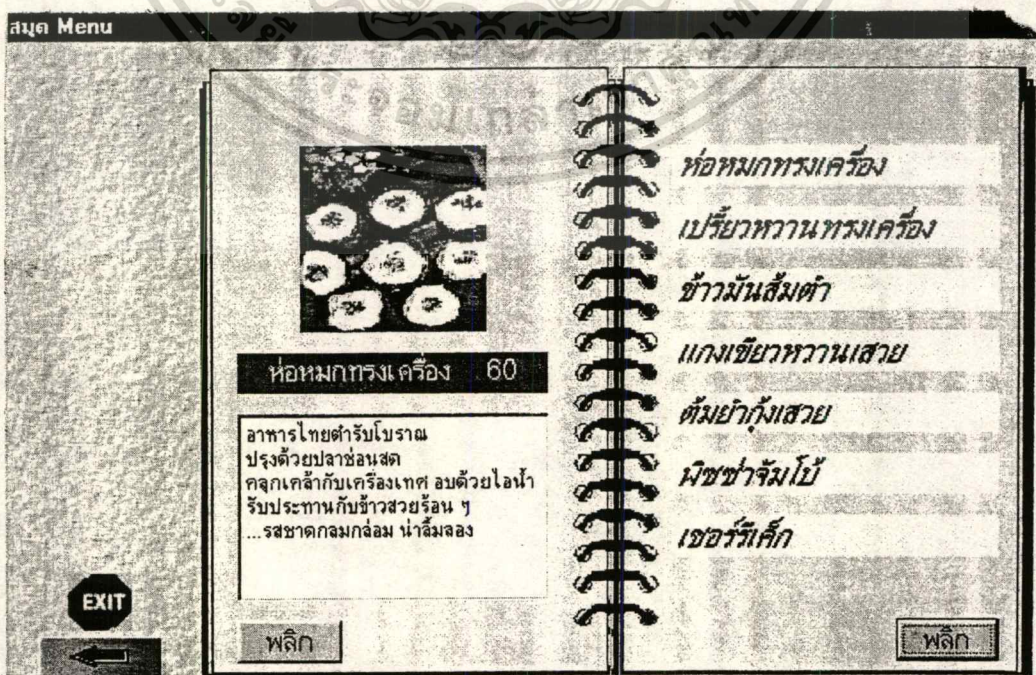
ภาพที่ 4.6 แสดงรายการอาหารพิเศษประจำวัน ด้านขวาของจอภาพ

- 4.3.2. การแสดงเมนูอาหารและเครื่องดื่ม เมื่อต้องการแสดงรายการอาหาร ฯ ในเมนู ลูกค้า ผู้ใช้ระบบฯ เพียงแต่ชี้และกดเบาๆ ไปที่แถบ “รายการอาหารและเครื่องดื่ม” จากหน้าจอแรก ตามภาพที่ 4.5 ระบบฯ จะแสดงรูปสมุดเมนูอาหารจะปรากฏดังภาพที่ 4.7



ภาพที่ 4.7 แสดงรูปสมุดเมนูอาหาร และเครื่องดื่ม

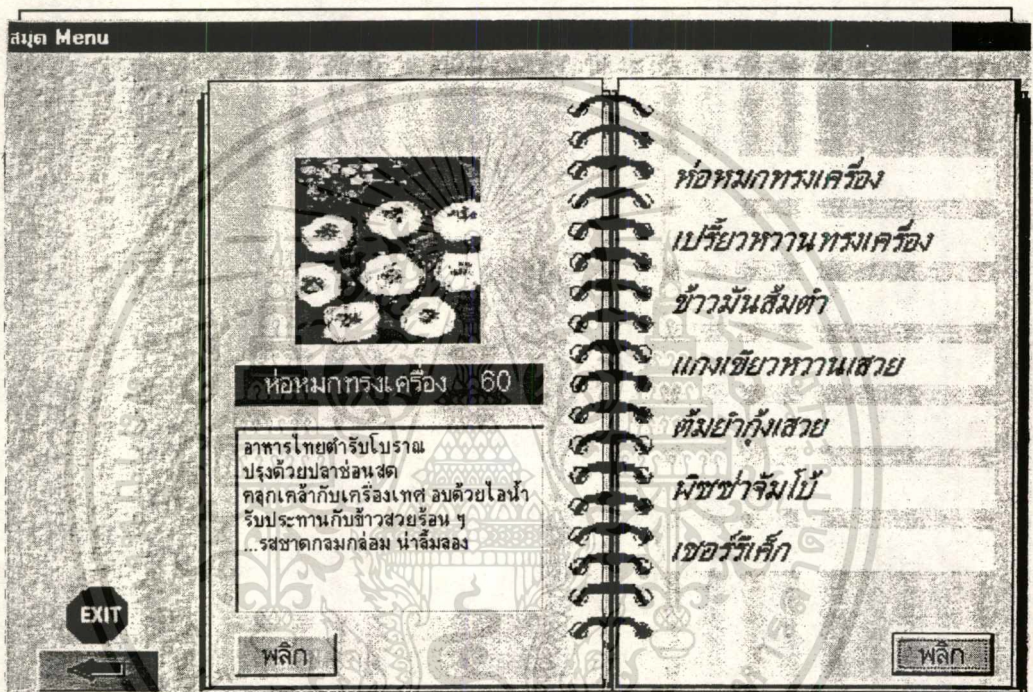
- 4.3.3. การเปิดสมุดเมนู เมื่อเข้าสู่หน้าจอของสมุดเมนูตามหัวข้อก่อนหน้าแล้ว ลูกค้าสามารถเปิดสมุดเมนูได้โดยการชี้และกดเบา ๆ ไปที่ปุ่ม “เปิด” ที่อยู่มุมขวาด้านล่างของสมุดเมนู ดังภาพที่ 4.7 หลังจากนั้นสมุดเมนูจะเปิดออกมาเป็น 2 หน้า ดังภาพที่ 4.8



ภาพที่ 4.8 แสดง เมนูที่เปิดออกเป็น 2 หน้า

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของคณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมุดเมนูเมื่อเปิดออกมาแล้วจะประกอบด้วย หน้าซ้าย และหน้าขวา โดยหน้า ด้านซ้ายจะแสดงรายละเอียดของรายการอาหารและเครื่องดื่ม เช่น รูปภาพ,ราคา,หรือ ข้อความชี้แนะชักชวนลูกค้า ซึ่งรายละเอียดในหน้าด้านซ้ายนี้จะเปลี่ยนแปลงไปตามที่ ลูกค้าชี้ผ่านรายชื่อของอาหารและเครื่องดื่มที่แสดงอยู่ในหน้าขวา ดังภาพที่ 4.9



ภาพที่ 4.9 แสดงรายละเอียดอาหารที่เปลี่ยนไปจากการชี้ชื่อรายการจากด้านขวา

นอกจากนี้ ณ มุมล่างของแต่ละหน้ายังมีปุ่ม “พลิก” ที่มีไว้ให้ลูกค้าผู้ใช้เลือก พลิก ไปหน้าถัดไป หรือพลิกย้อนกลับไปหน้าก่อนตามต้องการได้ เพียงแต่ชี้และกด เบา ๆ ไปที่ ปุ่ม “พลิก” ของแต่ละหน้า เมนูแต่ละหน้าก็จะพลิกซ้าย พลิกขวา ให้ อัตโนมัติ เหมือนกับที่ลูกค้าเคยใช้เมนูจริง ๆ นั่นเอง

4.3.2. การสั่งอาหารและเครื่องดื่มจากเมนู ลูกค้าผู้ใช้ระบบ ฯ เมื่อเลือกพิจารณารายการ อาหารและเครื่องดื่มตามที่แสดงในแต่ละหน้าแล้ว เมื่อต้องการเลือกรายการอาหาร หรือ เครื่องดื่ม ที่ต้องการก็เพียงแต่ ชี้และกดเบา ๆ ไปที่ชื่อรายการที่แสดงอยู่ในเมนู ด้านขวา (การชี้เพียงอย่างเดียวแต่ไม่ได้ออกแรงกด จะไม่ถือเป็นการเลือก เพราะการ ชี้ระบบจะตอบสนองแต่เพียงนำรายละเอียดของรายการที่ถูกชี้มาแสดงในด้านซ้าย เท่านั้น ดังนั้นหากจะเลือกต้องชี้และกดด้วยเสมอ)

เมื่อลูกค้าผู้ใช้ระบบ ฯ ชี้และกดเพื่อเลือกที่รายการใดจะปรากฏหน้าต่างแสดง
ชื่ออาหารฯที่ถูกเลือก พร้อมจำนวนที่ลูกค้าต้องการสั่ง โดยระบบจะรอให้ลูกค้าแสดง
ความจำนงค์ว่าต้องการสั่งอาหาร ฯ รายการนั้น จำนวนเท่าใด ซึ่งลูกค้าสามารถเลือก
จำนวนที่ต้องการได้โดยชี้และกดเบา ๆ ไปที่จำนวนที่ต้องการ ดังภาพที่ 4.10



ภาพที่ 4.10 แสดงการเลือกจำนวนอาหารหรือเครื่องดื่มที่ต้องการสั่ง

เมื่อลูกค้าชี้และกดเบา ๆ ไปที่จำนวนที่ต้องการสั่งแล้วระบบ จะแสดงภาพ
เคลื่อนไหว ดังนี้ เป็นสัญญาณให้ทราบว่ารายการดังกล่าวได้รับการ
บันทึกไว้แล้ว หลังจากนั้น หน้าจอจะกลับไปยังหน้าเมนูเดิม ที่แสดง
รายการอาหารและเครื่องดื่มเพื่อให้ลูกค้าเลือกสั่งรายการต่อไป

กรณีที่ต้องการยกเลิกรายการที่สั่งไว้แล้ว ให้ชี้และกดเบา ๆ ที่ปุ่ม “ยกเลิก” ใน
หน้าต่างแสดงจำนวนที่ต้องการสั่ง

- 4.3.6. การตรวจสอบรายการอาหารและเครื่องดื่มที่สั่งไปแล้ว ลูกค้าผู้ใช้ระบบสามารถ
ตรวจสอบรายการอาหารและเครื่องดื่มที่ได้สั่งไปแล้วได้ตลอดเวลา คล้ายกับการที่
เราดูสำเนาใบสั่งอาหารที่ร้านอาหารบางแห่งจะใส่แก้วไว้บนโต๊ะของลูกค้า ใน
ระบบเมนูอิเล็กทรอนิกส์นี้ ลูกค้าผู้ใช้ระบบ ฯ สามารถตรวจสอบรายการที่สั่งได้จาก
การ ชี้และกดเบา ๆ ที่แถบ “รายการที่สั่งไปแล้ว” จากหน้าจอแรกของระบบ ฯ จะ
แสดงรายละเอียดของรายการอาหารและเครื่องดื่มที่สั่งไปแล้ว ดังภาพที่ 4.11

รายการที่สั่ง

Date: 23 Oct 97 Time: 01:17:18 Order No: 2010002 โต๊ะที่: 22

| รายการ | จำนวน |
|-----------------------|-------|
| ฟลัทหมกทรงเครื่อง | 1 |
| เปรี้ยวหวานทรงเครื่อง | 2 |
| แกงเขียวหวานเสวย | 1 |
| ต้มยำกุ้งเสวย | 1 |
| ผลไม้รวม | 1 |
| ชา / กาแฟเย็น | 3 |
| | |
| | |

EXIT ส่ง Order

ภาพที่ 4.11 แสดงรายการอาหารและเครื่องดื่มที่สั่งไปแล้ว

4.3.7. การตรวจสอบค่าใช้จ่ายค่าอาหารและเครื่องดื่มที่ต้องชำระ ลูกค้ายูสเซอร์ระบบ สามารถเลือกทำการตรวจสอบรายจ่ายค่าอาหารและเครื่องดื่มที่ต้องชำระได้ด้วยตนเอง โดยเลือกแถบ “เช็คบิล” จากหน้าจอแรกของระบบ เมื่อเลือกแล้วระบบ จะแสดงรายละเอียดรายการอาหารและเครื่องดื่มที่สั่งไปแล้วพร้อมยอดค่าใช้จ่ายทั้งสิ้น ดังภาพที่ 4.12 เพื่อให้ลูกค้าเช็คสอบให้ถูกต้องก่อนชำระจริง

เช็คบิล

Date: 23 Oct 97 Time: 01:17:18 Order No: 2010002 โต๊ะที่: 22

| รายการ | จำนวน | ราคา | บาท |
|-----------------------|-------|------|-----|
| ฟลัทหมกทรงเครื่อง | 1 | 60 | 60 |
| เปรี้ยวหวานทรงเครื่อง | 2 | 50 | 100 |
| แกงเขียวหวานเสวย | 1 | 55 | 55 |
| ต้มยำกุ้งเสวย | 1 | 65 | 65 |
| ผลไม้รวม | 1 | 30 | 30 |
| ชา / กาแฟเย็น | 3 | 25 | 75 |
| | | | |
| | | | |

ภาษีมูลค่าเพิ่ม 10 % 42.35 รวม 385.00

Service Charge 10 % 38.50 รวมเงินทั้งสิ้น 465.85

EXIT

เอกสารนี้เป็นเอกสารที่ภาพที่ 4.12 แสดงรายละเอียดค่าอาหารและเครื่องดื่มที่ต้องชำระไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- เจริญพร อัญชสิทธิ์ธรรม, "CLIENT SERVER คืออะไร" นิตยสารคอมพิวเตอร์รีวิ
ปีที่ 11, ฉบับที่ 110 (ตุลาคม 2536) : 145.
- ชัยยงค์ วงศ์ชัยสุวัฒน์. "เครือข่ายคอมพิวเตอร์", นิตยสารไมโครคอมพิวเตอร์.
ฉบับที่ 88 (พฤศจิกายน 2535) : 358.
- ดวงแก้ว สวามิภักดิ์. ระบบฐานข้อมูล. กรุงเทพฯ : ซีเอ็ดยูเคชั่น, 2534.
- บทความพิเศษ "ปฐมบทสู่ SQL" นิตยสารคอมพิวเตอร์รีวิ
ปีที่ 9, ฉบับที่ 88 (ธันวาคม 2534) : 179.
- บทความพิเศษ "Object-Oriented เทคโนโลยี ปฏิวัติวงการ" นิตยสารคอมพิวเตอร์รีวิ
ปีที่ 10, ฉบับที่ 96 (สิงหาคม 2535) : 167.
- ประชา ตระการศิลป์. เอกสารประกอบการเรียนวิชา ไซส์เอนต์ / เซิร์ฟเวอร์,
หลักสูตรวิทยาการจัดการเทคโนโลยีสารสนเทศ, สถาบันเทคโนโลยีพระจอมเกล้า
เจ้าคุณทหารลาดกระบัง, ธันวาคม 2539. (อัดสำเนา)
- พงษ์ระพี เตชพาหพงษ์. "Object-Oriented User Interface" นิตยสารคอมพิวเตอร์รีวิ
ปีที่ 10, ฉบับที่ 96 (สิงหาคม 2535) : 207.
- วรชัย เขาวีระประสิทธิ์. "ภาษาคอมพิวเตอร์ในทศวรรษนี้", นิตยสารคอมพิวเตอร์รีวิ
ปีที่ 10, ฉบับที่ 96 (สิงหาคม 2535) : 186.
- สัญญาพงษ์ สายวงศ์นวล. วิชาลฟอกซ์โปร 3 สำหรับ วินโดว์. แปลจาก Microsoft Visual
FoxPro 3 for Windows Step by Step โดย Microsoft Corporation. กรุงเทพฯ ,
ซีเอ็ดยูเคชั่น, 2539.
- Date, C.J : An Introduction to Database Systems. Addison - Wesley, 1986
- Dewire, Dawna Travis : Client / Server Computing. McGRAW - HILL , 1994
- Microsoft Corporation : Developer's Guide, Microsoft Visual FoxPro version 3.0.
Microsoft Corporation , 1995
- Stamper, David A.: Business Data Communications. Benjamin/Cummings ,
Fourth Edition 1994.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

| | |
|--|---|
| ชื่อผู้เขียน | นายสมประสงค์ จิตรากณี |
| วันเดือนปีเกิด | วันที่ 21 มกราคม 2499 |
| การศึกษา | |
| ประถมศึกษา | โรงเรียน อัสสัมชัญ ธนบุรี |
| มัธยมศึกษา | วิทยาลัย บพิตรพิมุข แผนกภาษาต่างประเทศ(ฝรั่งเศส) |
| อุดมศึกษา | - บัญชีบัณฑิต มหาวิทยาลัยธรรมศาสตร์ - นิติศาสตร์บัณฑิต มหาวิทยาลัยธรรมศาสตร์ |
| ประสบการณ์การทำงาน | |
| - ผู้ตรวจสอบบัญชีภายใน | ธนาคาร ไทยพาณิชย์ จำกัด (มหาชน) กรุงเทพฯ ฯ |
| - ผู้อำนวยการฝ่ายบัญชีและการเงิน (Financial Controller) | โรงแรมคัสติร์รีสอร์ท พัทยา ชลบุรี |
| - ผู้อำนวยการฝ่ายการเงินและบริหาร (Director of Finance and Administration) | โรงแรม โนโวเทล บางนา กรุงเทพฯ ฯ |
| - ผู้ช่วยผู้จัดการทั่วไป (Resident Manager) | โรงแรมรอยัลปาร์ควิว สุขุมวิท 20 กรุงเทพฯ ฯ |
| - ผู้อำนวยการฝ่ายบัญชีและการเงิน (Financial Controller) | โรงแรมเวสต์ทิน บันยันทรี สาทร กรุงเทพฯ ฯ |
| - ผู้อำนวยการฝ่ายบัญชีและการเงิน (Financial Controller) | โรงแรมเลอคองคอร์ด เรอเนสซอง รัชดาฯ กรุงเทพฯ ฯ |
| - ที่ปรึกษาและผู้ชำนาญการ ระบบงานบัญชีและการเงิน | บริษัท NEMAC Management Consulting จำกัด |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้