



การพัฒนาระบบงานฐานข้อมูลรีเลชันแนลแบบไคลเอ็นท์เซิร์ฟเวอร์

CLIENT/SERVER RELATIONAL DATABASE SYSTEM DEVELOPMENT



โดย

นางสาว สุวรรณิ สุวรรณพงษ์ 35104503

นางสาว อรอนงค์ พัฒนาปัญญาสัตย์ 35104544

วัน เดือน ปี..... ๑/ ๑๑. ๒๕๔๐
เลขทะเบียน..... ๐๓๗๐๘๕
เลขเรียกหนังสือ..... T ๑๘๑๗๖ ค.๑๗๕ ก

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร

สาขาวิชาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา ๒๕๓๘

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารฉบับนี้ที่มีการนำไปใช้

037085

ปริญญาานิพนธ์ปีการศึกษา 2538

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาระบบงานฐานข้อมูลรีเลย์ชั้นแนลแบบโคสเอ็นทีเซิร์ฟเวอร์

ผู้จัดทำ

1. นางสาวสุวรรณี สุวรรณพงษ์ 35104503

2. นางสาวอรอนงค์ พัฒนาปัญญาสัตย์ 35104544


.....อาจารย์ที่ปรึกษา
(ดร.วรวัฒน์ ลิ้มโกศา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาระบบงานฐานข้อมูลรีเลย์ชั้นแนวแบบไคลเอ็นท์เซิร์ฟเวอร์

ศุวรรณิ ศุวรรณพงษ์

อรอนงค์ พัฒนปัญญาสัตว์

ดร.รวีวัฒน์ ลิ้มโกคา อาจารย์ที่ปรึกษา

ปีการศึกษา 2538

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เรียบเรียงขึ้นจากการศึกษาวิธีการออกแบบระบบงานประยุกต์ระบบงานที่นำมาออกแบบคือ ระบบการจัดจำหน่าย (Customer Order Processing System) เริ่มจากการศึกษาระบบงานที่มีอยู่เดิมและกำหนดขั้นตอนการออกแบบโดยอ้างอิงกับทฤษฎีเอสเอสเอดีเอ็ม (Structured Systems Analysis and Design Method - SSADM) เนื่องจากเป็นวิธีพัฒนาระบบที่ทันสมัยสำหรับผลิตภัณฑ์ที่นำมาเป็นเครื่องมือในการออกแบบและพัฒนาระบบงานประยุกต์ในครั้งนี้ได้ใช้ผลิตภัณฑ์เออร์วิน 2.0 (ERwin/ERX 2.0) ของบริษัทลอจิกเวิร์ค (Logic Works) ในการสร้างตารางเก็บข้อมูลที่มีความสัมพันธ์ของตาราง คล้ายกับอ็อร์โมเดล และ ใช้เพาเวอร์บิวเดอร์ 4.0 (PowerBuilder 4.0) ของบริษัทเพาเวอร์ซอฟท์ (PowerSoft) ที่มีระบบจัดการฐานข้อมูลวอตคอม (WatCom) ซึ่งทำงานบนเครื่องเดียว (Stand Alone) ในการสร้างระบบงานประยุกต์ต่างๆ จากนั้นจึงพัฒนาเพื่อเป็นระบบไคลเอ็นท์เซิร์ฟเวอร์ โดยใช้ระบบจัดการฐานข้อมูลไมโครซอฟท์ เอสคิวแอลเซิร์ฟเวอร์ (Microsoft SQL Server) ทำงานบนระบบปฏิบัติการไมโครซอฟท์ วินโดวส์เอ็นที (Microsoft Windows NT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLIENT/SERVER RELATIONAL DATABASE SYSTEM DEVELOPMENT

Suwanee Suwanapong

Onanong Pattanapanyasat

Prof.Ass Dr.Vorawat Limpoka

1995

Abstract

This thesis is a client/server relational database application development. The application is about "Customer Order Processing System". At first, we analyzed existing application and designed new application by reference to some part of Structure Systems Analysis and Design Method (SSADM) because of its new technology. The product that used to design and development was ERwin/ERX 2.0 from Logic Works , PowerBuilder 4.0 from PowerSoft and Watcom DBMS that attach with PowerBuilder as stand alone DBMS. Last, we ported to Microsoft SQL Server on Microsoft Windows NT operating system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทนำ.....	1
บทที่ 2.....	2
ทฤษฎีที่ใช้ในการพัฒนาระบบฐานข้อมูล	2
2.1 โมเดลเชิงสัมพันธ์ (RELATIONAL MODEL).....	2
2.1.1 ศัพท์เฉพาะของโมเดลเชิงสัมพันธ์	3
2.2 อีอาร์โมเดล (ENTITY RELATIONSHIP MODEL - ER MODEL)	4
2.2.1 การแปลงความสัมพันธ์ของข้อมูลจากอีอาร์โมเดล ไปสู่ในรูปของรีเลชัน.....	7
2.3 ทฤษฎีของระบบฐานข้อมูลเชิงสัมพันธ์.....	8
2.3.1 กฎของความคงสภาพ (integrity rule).....	8
2.3.2 ฟังก์ชันการขึ้นต่อกัน (Functional dependency).....	8
2.3.3 การนอร์มัลไลซ์ (Normalization).....	9
2.4 การเขียนแผนภาพการไหลของข้อมูล(DATA FLOW DIAGRAM)	11
2.5 สภาพแวดล้อมในการพัฒนาโปรแกรม(DEVELOPMENT ENVIRONMENT).....	12
2.5.1 สภาพแวดล้อมของการทำงานบนเครื่องเดียว (Stand Alone Environment).....	12
2.5.2 การพัฒนาระบบบนสภาพแวดล้อมของการทำงานจริง (Target Environment).....	14
2.5.2.1 ตรีโพรซีเยอร์ที่จำเป็นในการทำงานกับฐานข้อมูลบน SQL Server.....	14
2.5.2.2 การสร้างฐานข้อมูลบนไมโครซอฟท์ เอสคิวแอลเซิร์ฟเวอร์และไวยากรณ์ต่างๆที่จำเป็น	16
2.5.2.3 สภาพแวดล้อมของการทำงานกับเอสคิวแอลเซิร์ฟเวอร์.....	17
2.5.3 การย้ายแพลตฟอร์ม (Platform).....	21
2.5.3.1 : ชนิดของข้อมูล (Datatype).....	21
2.5.3.2 การใช้เอสคิวแอลแบบฝังตัว (Embedded SQL).....	24
2.5.3.3 ประโยชน์ของดาต้าวินโดวใน เพาเวอร์บิวเดอร์.....	26
2.5.3.4 การเขียนโปรแกรมส่วนที่เชื่อมต่อกับฐานข้อมูล(การ Connect Database)	26
2.5.3.5 ข้อดีข้อเสียของการพัฒนาระบบบนสภาพแวดล้อมที่แตกต่างกัน.....	27
2.5.3.6 การนำข้อมูลเข้าออกจากดาต้าเบสหนึ่งกับอีกฐานข้อมูลหนึ่ง(Import Export Data).....	28
2.5.4 โอปินดาต้าเบสคอนเน็คติวิตี (โอดีบีซี) (Open Database Connectivity, ODBC)	30
2.5.4.1 การติดต่อโดยใช้โอดีบีซี (ODBC Interface).....	30
2.5.4.2 ส่วนประกอบของโอดีบีซี.....	31
2.6 อธิบายวิธีการในการออกแบบของวิเคราะหและออกแบบโครงสร้าง(SSADM).....	32
2.7 หลักการของไคลเอ็นท์เซิร์ฟเวอร์	35

2.7.1 ชนิดของการประมวลผลไคลเอนท์เซิร์ฟเวอร์.....	36
2.7.1.1 ไคลเอนท์เซิร์ฟเวอร์ที่ทำงานบนเครื่องเดียวกัน (STAND-ALONE CLIENT-SERVER).....	36
2.7.1.2 แสทนอกโหนดไคลเอนท์เซิร์ฟเวอร์ (STAND-ALONE LAN CLIENT-SERVER).....	36
2.7.1.3 แมนนวลเอ็กแทรกต์ไคลเอนท์เซิร์ฟเวอร์ (MANUAL EXTRACT CLIENT-SERVER).....	36
2.7.1.4 ซิงเกิลไซต์อัปเดตไคลเอนท์เซิร์ฟเวอร์ (SINGLE-SITE UPDATE CLIENT-SERVER).....	37
2.7.1.5 มัลติไซต์อัปเดตไคลเอนท์เซิร์ฟเวอร์ (MULTISITE UPDATE CLIENT-SERVER).....	37
2.7.1.6 ไคลเอนท์เซิร์ฟเวอร์แบบระบบฐานข้อมูลแบบกระจาย (DISTRIBUTED DATABASE CLIENT-SERVER).....	38
2.7.2 ข้อดีข้อเสียของระบบไคลเอนท์เซิร์ฟเวอร์.....	38
2.7.2.1 ข้อดีของระบบไคลเอนท์เซิร์ฟเวอร์.....	38
2.7.2.2 ข้อเสียของระบบไคลเอนท์เซิร์ฟเวอร์.....	39
บทที่ 3.....	40
การคำนวณและการสร้าง.....	40
3.1 แผนภาพการไหลของข้อมูล (DATA FLOW DIAGRAM).....	41
3.2 แผนภาพ ER (ENTITY RELATIONSHIP DIAGRAM).....	60
3.3 การเขียนโปรแกรมด้วยเพาเวอร์บิวเดอร์.....	77
3.3.1 : การเชื่อมต่อกับฐานข้อมูล.....	77
3.3.1.1 : ตัวแปรที่ใช้ในการเชื่อมต่อกับฐานข้อมูล.....	77
3.3.1.2 : ข้อความที่ใช้เชื่อมต่อ (Connect String).....	78
3.3.1.3 การสร้างไฟล์เริ่มต้น (Initialize File).....	78
3.3.2 วิธีการใช้เมนูกับหน้าต่าง.....	79
3.3.3 การสร้างหน้าต่างลูกจากหน้าต่างแม่ (Window Inheritance).....	79
3.3.4 : การสร้างเมนูลูกจากเมนูแม่ (Menu Inheritance).....	79
3.3.6 สิ่งที่ต้องทราบในการใช้คาด้าวินโดว์.....	80
3.3.6.1 การใช้คาด้าวินโดว์กับหน้าต่าง.....	80
3.3.6.2 ฟังก์ชันของเพาเวอร์บิวเดอร์ที่จำเป็นต้องใช้กับคาด้าวินโดว์.....	80
3.3.6.3 การสร้างกฎในการใส่ค่าในคาด้าวินโดว์(Validation Rule).....	82
3.3.7 วิธีการยืนยัน(Commit)และย้อนกลับ(Roll Back).....	83
บทที่ 4.....	87
ระบบงานโปรแกรมประยุกต์ และ ผลการทดลอง.....	87
4.1 ระบบลูกค้า.....	87

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 4.1.1 จุดประสงค์ของระบบลูกค้า.....87
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น ยกเว้น ไม่ก่อให้เกิดข้อสงสัย และต้องยังต้องเป็นเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2	การทำงานของระบบลูกค้า.....	87
4.1.3	ระบบลูกค้าจะทำงานได้สำเร็จเมื่อ.....	88
4.1.4	ตัวอย่างส่วนติดต่อผู้ใช้ของระบบลูกค้า.....	89
4.2	ระบบคำสั่งซื้อ.....	92
4.2.1	จุดประสงค์ของระบบคำสั่งซื้อ.....	92
4.2.2	เงื่อนไขการทำงานของระบบคำสั่งซื้อ.....	92
4.2.3	การทำงานของระบบคำสั่งซื้อ.....	92
4.2.4	การแก้ไขคำสั่งซื้อและลำดับคำสั่งซื้อ.....	96
4.2.5	ตัวอย่างส่วนติดต่อผู้ใช้ของระบบคำสั่งซื้อ.....	98
4.3	ระบบการจอง.....	100
4.3.1	จุดประสงค์ของระบบการจอง.....	100
4.3.2	เงื่อนไขการทำงานของระบบการจอง.....	100
4.3.3	การทำงานของระบบการจอง.....	100
4.3.4	ตัวอย่างส่วนติดต่อผู้ใช้ของระบบการจอง.....	104
4.4	ระบบออกใบกำกับภาษี.....	106
4.4.1	จุดประสงค์ของระบบออกใบกำกับภาษี.....	106
4.4.2	การทำงานของระบบออกใบกำกับภาษี.....	106
4.4.2.1	ใบกำกับภาษีที่ออกจากราคาสั่งซื้อแบบปกติ.....	106
4.4.2.2	ใบกำกับภาษีที่ออกจากราคาสั่งซื้อแบบออกใบกำกับภาษีได้ทันทีที่สั่ง.....	106
4.4.2.3	การพิจารณาค่าสั่งซื้อ.....	106
4.4.2.4	การพิจารณาลำดับคำสั่งซื้อ.....	106
4.4.3	ผลของการออกใบคำสั่งซื้อ.....	107
4.4.4	ตัวอย่างส่วนติดต่อผู้ใช้ของระบบการออกใบแจ้งรายการสินค้า.....	108
4.5	ระบบการกำหนดนโยบายราคา.....	110
4.5.1	จุดประสงค์ของระบบการกำหนดนโยบายราคา.....	110
4.5.2	การทำงานของระบบการกำหนดนโยบายราคา.....	110
4.5.2.1	การกำหนดนโยบายราคา.....	110
4.5.2.2	การทำงานในระบบลูกค้า.....	112
4.5.2.3	การทำงานในระบบคำสั่งซื้อ.....	112
4.5.3	ขั้นตอนการคิดราคาในระดับลำดับคำสั่งซื้อ.....	112
4.5.4	ตัวอย่างส่วนติดต่อผู้ใช้ของระบบกำหนดนโยบายราคา.....	114
บทที่ 5	116

บทวิจารณ์สรุปและแนวทางในการพัฒนาขั้นต่อไป 116

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 5.1 ข้อจำกัดของโครงการนี้ 116
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 แนวทางการพัฒนาต่อไป	116
กิตติกรรมประกาศ.....	117
หนังสืออ้างอิง	117
ดัชนี.....	118
ภาคผนวก	119



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ 2-1 อธิบายการทำงานอย่างคร่าวๆ ของสโตร์โพรซีเยอร์ที่มีในไฟล์ PBSYB.SQL	15
ตารางที่ 2-2 อธิบายการทำงานอย่างคร่าวๆ ของสโตร์โพรซีเยอร์ที่มีในไฟล์ PBSYBRT.SQL	15
ตารางที่ 2-3 รายละเอียดของไฟล์ที่ใช้ในสภาพแวดล้อมแบบ 16 บิต	18
ตารางที่ 2-4 รายละเอียดของไฟล์ที่ใช้ในสภาพแวดล้อมแบบ 32 บิต	19
ตารางที่ 2-5 ซอฟต์แวร์ที่จำเป็นสำหรับเครือข่าย	20
ตารางที่ 2-5 เปรียบเทียบชนิดของข้อมูล	21
ตารางที่ 3-1 ตารางที่เกี่ยวข้องกับลูกค้าและคุณสมบัติต่างๆของลูกค้า	60
ตารางที่ 3-2 ตารางที่เกี่ยวข้องกับลูกค้า และ ผู้ขายError! Bookmark not defined.	
ตารางที่ 3-3 ตารางที่เกี่ยวข้องกับลูกค้า สินค้า และความสัมพันธ์ของลูกค้าด้วยการให้อนุญาตซื้อ	65
ตารางที่ 3-4 ตารางที่เกี่ยวข้องกับใบคำสั่งซื้อ ใบกำกับสินค้า/ใบกำกับภาษี	67
ตารางที่ 3-5 ตารางที่เกี่ยวข้องกับคำสั่งซื้อ ใบกำกับภาษี	70
ตารางที่ 3-6 ตารางที่เกี่ยวข้องกับสต็อก	70
ตารางที่ 3-7 ตารางที่เกี่ยวข้องกับการกำหนดนโยบายราคาระดับลำดับคำสั่งซื้อ	73
ตารางที่ 3-8 ตารางที่เกี่ยวข้องกับการกำหนดนโยบายราคาระดับคำสั่งซื้อ	73
ตารางที่ 3-9 ตัวแปรที่ใช้ในการเชื่อมต่อข้อมูล	77
ตารางที่ 3-10 ตารางการเขียนสคริปต์เพื่อใช้ดาต้าวินโดว์กับหน้าต่าง	80
ตารางที่ 3-11 ตารางแสดงฟังก์ชันที่จำเป็นต้องใช้กับดาต้าวินโดว์ ในการติดต่อกับฐานข้อมูล	80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่ 2-1 อธิบายศัพท์เฉพาะของโมเดลเชิงสัมพันธ์	3
รูปที่ 2-2 แสดงสัญลักษณ์ในการวาดภาพอีอาร์	5
รูปที่ 2-3 แสดงสัญลักษณ์ในการวาดภาพอีอาร์	6
รูปที่ 2-4 สัญลักษณ์ของแผนภาพการไหลของข้อมูล	11
รูปที่ 2-5 สถาปัตยกรรมของโปรแกรมประยุกต์และไคลเอนต์สำหรับ สภาพแวดล้อมแบบ 16 บิต17	
รูปที่ 2-6 สถาปัตยกรรมของโปรแกรมประยุกต์และไคลเอนต์สำหรับ สภาพแวดล้อมแบบ 32 บิต19	
รูปที่ 2-7 ตัวอย่างการเขียนโปรแกรมส่วนเชื่อมต่อกับฐานข้อมูล	26
รูปที่ 2-8 โครงสร้างของเอสเอสเอดีเอ็ม	33
รูปที่ 3-1 แสดงแผนภาพระบบงานระดับบนสุดของระบบการจัดจำหน่าย	44
รูปที่ 3-2 แสดงแผนภาพระบบการทำงานของลูกค้าและคำสั่งซื้อ	45
รูปที่ 3-3 แสดงแผนภาพการทำงานของการรับคำสั่งซื้อ	46
รูปที่ 3-4 แสดงแผนภาพการทำงานของระบบราคา	47
รูปที่ 3-5 แสดงแผนภาพการเพิ่มและแก้ไขคำสั่งซื้อ	48
รูปที่ 3-6 แสดงแผนภาพการทำงานการสร้างคำสั่งซื้อใหม่	49
รูปที่ 3-7 แสดงการแผนภาพการแก้ไขปรับปรุงคำสั่งซื้อ	50
รูปที่ 3-8 แสดงแผนภาพการแก้ไขรายละเอียดของคำสั่งซื้อ	51
รูปที่ 3-9 แสดงแผนภาพการเปิดใบกำกับภาษี	52
รูปที่ 3-10 แสดงแผนภาพการจองและการยกเลิกสินค้า	53
รูปที่ 3-11 แสดงแผนภาพการรวมเปิดใบกำกับภาษี	54
รูปที่ 3-12 แสดงแผนภาพการแก้ไขวงเงินในการซื้อสินค้าของลูกค้า	55
รูปที่ 3-13 แสดงแผนภาพการจัดการลูกค้า	56
รูปที่ 3-14 แสดงแผนภาพการบันทึกลูกค้าใหม่	57
รูปที่ 3-15 แสดงแผนภาพการแก้ไขข้อมูลของลูกค้า	58
รูปที่ 3-16 แสดงแผนภาพการจัดการลูกค้าและสินค้า	59
รูปที่ 3-17 แสดงแผนภาพแสดงความสัมพันธ์ของข้อมูลเกี่ยวกับลูกค้า	62

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3-17 แสดงแผนภาพแสดงความสัมพันธ์ของข้อมูลระหว่างลูกค้าและบริษัท 64
ไม่ปรากฏใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3-18 แสดงแผนภาพแสดงความสัมพันธ์ของข้อมูลระหว่างลูกค้าและสินค้า	66
รูปที่ 3-19 แสดงแผนภาพแสดงความสัมพันธ์ของข้อมูลที่เกี่ยวข้องคำสั่งซื้อ	69
รูปที่ 3-20 แสดงแผนภาพแสดงความสัมพันธ์ของข้อมูลเกี่ยวกับสินค้า	71
รูปที่ 3-21 แสดงแผนภาพแสดงความสัมพันธ์ของข้อมูลระหว่างใบกำกับภาษีและสินค้า	72
รูปที่ 3-22 แสดงความสัมพันธ์ของข้อมูลที่เกี่ยวข้องกับราคาสินค้าในระดับลำดับที่สินค้า	75
รูปที่ 3-23 แสดงความสัมพันธ์ของข้อมูลที่เกี่ยวข้องกับราคาสินค้าในระดับคำสั่งซื้อสินค้า	76
รูปที่ 4-1 แสดงลูกค้าที่มีทั้งหมดในระบบ	89
รูปที่ 4-2 แสดงรายละเอียดของลูกค้าที่มีระดับเป็น A	89
รูปที่ 4-3 แสดงรายละเอียดของสถานที่ส่งสินค้า	90
รูปที่ 4-4 แสดงรายละเอียดหมายเลขบัญชีของลูกค้า	90
รูปที่ 4-5 แสดงการกำหนดแบบนโยบายราคาของลูกค้า	91
รูปที่ 4-6 แสดงหน้าจอสร้างคำสั่งซื้อ	98
รูปที่ 4-7 ระดับลำดับคำสั่งซื้อ	98
รูปที่ 4-8 ระดับคำสั่งซื้อ	99
รูปที่ 4-9 พิมพ์คำสั่งซื้อ	99
รูปที่ 4-10 แสดงหน้าจอก่อนจะมีการหาลำดับคำสั่งซื้อที่สามารถจองได้	104
รูปที่ 4-11 แสดงผลลัพธ์ที่เกิดจากการหาลำดับคำสั่งซื้อที่สามารถจองได้	104
รูปที่ 4-12 การจองสินค้าเป็น lot/ident	105
รูปที่ 4-13 การจองสินค้าเป็น permanent/temporary	105
รูปที่ 4-14 หน้าจอแสดงเปิดใบกำกับภาษีแบบ lot / fifo	108
รูปที่ 4-15 หน้าจอแสดงเปิดใบกำกับภาษีแบบ permanent / temporary	108
รูปที่ 4-16 หน้าจอแสดงผลพิมพ์ใบกำกับภาษี	109
รูปที่ 4-17 การกำหนดนโยบายราคาในระดับลำดับคำสั่งซื้อที่เป็นนโยบายแบบ 1 และวิธีคิดราคาแบบกำหนดราคามาตรฐานซึ่งไม่คิดจากหน่วยที่เป็นราคา (มีเฉพาะที่เป็นปริมาณ)	114
รูปที่ 4-18 การเปลี่ยนแปลงแบบของนโยบาย จะต้องระบุสิ่งที่เกี่ยวข้องให้ครบจึงสามารถหาได้	114
รูปที่ 4-19 แสดงการกำหนดจำนวนที่ไม่คิดราคา ซึ่งไม่มีการคิดจากหน่วยที่เป็นราคา(มีเฉพาะที่เป็นปริมาณ)	115
รูปที่ 4-20 แสดงการกำหนดราคาบวกเพิ่มสามารถตั้งเงื่อนไขได้ทั้งจากราคาและจากจำนวนสินค้า (ปริมาณ)	115

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในการพัฒนาระบบงานขึ้นมาแทนระบบงานเดิมนั้น มีความจำเป็นมากในปัจจุบัน เนื่องจากแต่ละองค์กรต้องแข่งขันกันเพื่อความก้าวหน้าและเพิ่มประสิทธิภาพให้แก่ระบบ ดังนั้น ในโครงการนี้ จึงได้เลือกการออกแบบระบบฐานข้อมูลในรูปแบบของระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database System) ที่จะมีการทำงานภายใต้สภาพแวดล้อมแบบไคลเอ็นท์เซิร์ฟเวอร์ (Client/Server Environment) มาพัฒนาระบบการจัดจำหน่ายและส่งสินค้า ปริมาณที่บันทึกฉบับนี้จะกล่าวถึงรายละเอียดดังนี้

ในบทที่ 2. เกี่ยวกับทฤษฎีและหลักการในการพัฒนาระบบ

ในบทที่ 3. การคำนวณและการสร้าง

ในบทที่ 4. ระบบงาน โปรแกรมประยุกต์และผลการทดลอง

ในบทที่ 5. บทวิจารณ์และสรุปแนวทางในการพัฒนาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่ใช้ในการพัฒนาระบบฐานข้อมูล

2.1 โมเดลเชิงสัมพันธ์ (Relational Model)

โมเดลเชิงสัมพันธ์เป็นโมเดลที่ใช้ในการอธิบายความสัมพันธ์ของข้อมูลที่ถูกเก็บด้วยระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System : RDBMS) ซึ่งเป็นผลงานของ ดร.คอดด์ (Codd) ที่ได้เสนอผลงานวิจัยให้ชาวโลกรู้จักกันในปี พ.ศ. 2513 โดยมีบรรดานักวิชาการทางคอมพิวเตอร์ก็ได้ให้ความสนใจและทุ่มเททำการวิจัยเกี่ยวกับโมเดลนี้มากมาย จนในปัจจุบันนี้ได้แพร่หลายไปมาก มีการนำไปใช้งานกับเครื่องระดับตั้งแต่เมนเฟรมลงไปจนถึงเครื่องระดับไมโคร และเป็นที่ยอมรับโดยทั่วไปว่า บรรดาผู้ใช้ระบบฐานข้อมูล (โดยเฉพาะผู้ที่ทำงานด้วย เครื่องระดับมินิ และระดับไมโคร) จะมีความคุ้นเคยกับ โมเดลเชิงสัมพันธ์นี้มากกว่าอีก 2 โมเดล คือ โมเดลเชิงแตกสาขา (Hierarchical model) และ โมเดลเชิงโครงข่าย (Network Model) ที่มีมาก่อนหน้านี้

ข้อดีของโมเดลเชิงสัมพันธ์

1. โมเดลเชิงสัมพันธ์เป็น โมเดลที่สามารถสร้างความเข้าใจ ได้ง่ายกว่า เพราะภาพลักษณ์ของข้อมูลที่เก็บ โดยโมเดลเชิงสัมพันธ์จะมาจากมุมมองของผู้ใช้ ซึ่งจะมีความซับซ้อนน้อยกว่าภาพลักษณ์ของข้อมูลที่เก็บ โดยอีก 2 โมเดล
2. ระบบส่วนใหญ่ที่ใช้โมเดลเชิงสัมพันธ์นี้มักจะมีเครื่องมือที่ช่วยให้ผู้ใช้สามารถจัดการกับข้อมูลที่เก็บ อยู่ได้ง่ายกว่าข้อมูลที่จัดเก็บด้วย โมเดลแบบอื่น
3. โมเดลเชิงสัมพันธ์เป็น โมเดลที่มีความสอดคล้องกับหลักการของฐานข้อมูล ผู้ใช้ไม่ต้องพะวงกับรายละเอียดของการจัดเก็บข้อมูลเหมือนกับการจัดข้อมูลของ โมเดลอื่น
4. โมเดลเชิงสัมพันธ์นี้มีเครื่องมือที่ช่วยให้ผู้ใช้สามารถค้นพบปัญหาที่เกิดขึ้นในการออกแบบระบบ ฐานข้อมูล ได้โดยง่ายและยังง่ายในการแก้ไขการออกแบบที่ผิดพลาดนั้นด้วย
5. ภาษาที่ใช้ในการจัดเก็บข้อมูลที่จัดเก็บด้วยระบบจัดการฐานข้อมูลเชิงสัมพันธ์ หรือ เอสคิวแอล (Structure Query Language - SQL) เป็นภาษาแบบเซตโอเรียนเตด (set oriented) ซึ่งจะต่างกับภาษาที่ใช้ในการจัดการกับข้อมูลที่จัดเก็บด้วยระบบจัดการฐานข้อมูลของ โมเดลอื่นที่เป็นภาษาแบบเรคคอร์ด เอตอะตาม์ (record-at-a-time)

แม้ว่าโมเดลเชิงสัมพันธ์จะมีข้อดีหลายประการดังได้กล่าวไปแล้ว แต่ในปัจจุบันก็ยังมีจุดอ่อนที่มีการอ้างถึงเสมอ คือ ระบบจัดการฐานข้อมูลแบบโมเดลเชิงสัมพันธ์นี้มักจะมีประสิทธิภาพในการใช้งาน ผู้ใช้สอง โมเดล ไม่ได้ โดยเฉพาะในการประยุกต์ใช้งานขององค์กรขนาดใหญ่ จุดอ่อนนี้ก็ได้มีการแย้งกลับมาในแง่ที่ว่า โมเดลเชิงสัมพันธ์เป็นโมเดลที่มีอยู่การพัฒนาน้อยกว่าอีกสองโมเดล จึงเป็นไปได้ว่า การพัฒนาที่ผ่านมาของ โมเดลเชิงสัมพันธ์ก็ยังมีจำนวนระดับขั้นที่ได้พัฒนาไปแล้ว น้อยกว่าอีกสองโมเดล ดังนั้น หากต้องการเปรียบเทียบการ

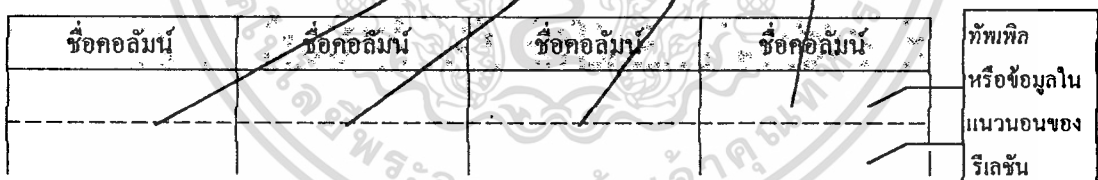
ทำงานระหว่างโมเดลเชิงสัมพันธ์กับโมเดลอื่น ก็ควรที่จะการเปรียบเทียบที่ระดับจำนวนขั้นการพัฒนาที่เท่ากันจึงจะสมเหตุสมผล

2.1.1 ศัพท์เฉพาะของโมเดลเชิงสัมพันธ์

ในหัวข้อนี้จะกล่าวถึงนิยามและคำศัพท์ต่างๆที่เกี่ยวข้องกับโมเดลนี้ จากการที่ข้อมูลที่เกี่ยวข้อง โมเดลเชิงสัมพันธ์จะถูกเก็บไว้ในตารางที่จะถูกเรียกว่า "รีเลชัน" โดยที่รีเลชันทุกรีเลชันจะอยู่ในรูปของตาราง แต่ตารางบางตารางอาจไม่เป็นรีเลชันก็ได้

ตารางที่มีลักษณะเป็นรีเลชันจะต้องมีคุณลักษณะดังนี้

1. แต่ละช่องของตารางจะบรรจุข้อมูลได้เพียงค่าเดียว
2. ชื่อหัวข้อในแต่ละคอลัมน์มีความแตกต่างกัน อันเป็นชื่อของแอตทริบิวของเอนทิตี
3. ค่าข้อมูลที่อยู่ในแต่ละคอลัมน์ คือ ค่าของแอตทริบิวตามที่ระบุหัวข้อไว้ที่หัวของคอลัมน์นั้นๆ
4. การเรียงลำดับคอลัมน์ไม่ถือว่ามีผลสำคัญ
5. ข้อมูลแต่ละแถวจะต้องแตกต่างกัน
6. การเรียงลำดับแถวไม่ถือว่ามีผลสำคัญ



รูปที่ 2-1 อธิบายศัพท์เฉพาะของ โมเดลเชิงสัมพันธ์

ตารางที่มีคุณลักษณะดังกล่าวเรียกว่า รีเลชัน

ดังนั้น เราจะได้นิยามของฐานข้อมูลเชิงสัมพันธ์คือ ฐานข้อมูลที่เกิดจากการรวบรวมรีเลชันต่างๆที่มีความสัมพันธ์ (relationship) ระหว่างกัน

เราจะเรียกข้อมูลแต่ละแถวในแนวนอนของรีเลชันว่า ทิวเพิล (tuple) และเรียกข้อมูลแต่ละแถวในแนวตั้งหรือแนวคอลัมน์ว่า แอตทริบิว (attribute) โดยที่คำว่า คีย์ (key) จะหมายถึงข้อมูลที่เกิดจากแอตทริบิว 1 ตัวหรือหลายตัวก็ได้

แต่ละรีเลชันจะต้องมีสิ่งที่เรียกว่า คีย์หลัก (primary key) คือข้อมูลของแอตทริบิว 1 ตัว หรือ มากกว่า ก็ได้ที่สามารถใช้เป็นตัวเจาะจงบอกเราได้ว่ากำลังอ้างอิงถึงข้อมูลทิวเพิลใด ส่วนคีย์ที่เป็นแอตทริบิวของรีเลชันอื่นที่ซ้ำกับแอตทริบิวที่เป็นคีย์หลักของรีเลชันหนึ่ง จะเรียกว่าคีย์นอก (foreign key) เช่น รีเลชัน A มีแอตทริบิวรหัสประจำตัวเป็นคีย์หลักแล้ว ในรีเลชัน B ว่าเป็นคีย์นอกของคีย์หลักของรีเลชัน A ในกรณีที่มีรีเลชันมีแอตทริบิว หรือ

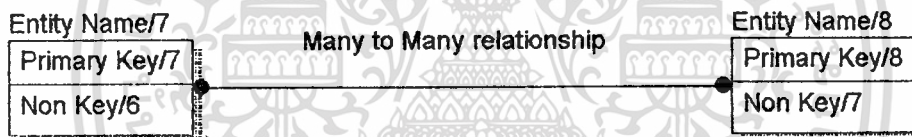
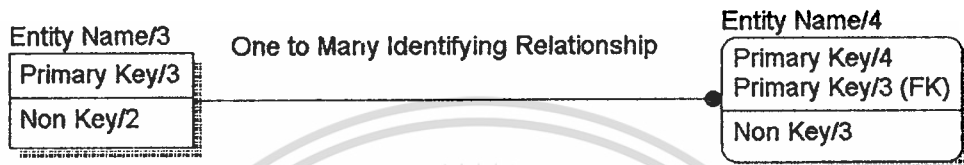
กลุ่มแอตทริบิวต์ที่มีคุณสมบัติเป็นคีย์หลักได้ อยู่หลายแอตทริบิวต์ เราจะเรียกแอตทริบิวต์หรือกลุ่มแอตทริบิวต์ที่มีได้ถูกเลือกให้เป็นคีย์หลักกว่าคีย์คู่แข่ง (candidate key) หรือคีย์สำรอง (alternate key) และแอตทริบิวต์อื่นๆที่เหลือที่มีได้เป็นคีย์หลักและไม่ได้เป็นส่วนใดส่วนหนึ่งของคีย์หลักก็จะถูกเรียกว่า นันคีย์(nonkey attribute)

คำว่า โดเมน (domain) จะหมายถึง กรอบของค่าต่างๆ ที่เป็นไปได้เช่น โดเมนของแอตทริบิวต์วันที่จะหมายถึงค่าของวันที่ที่เป็นไปได้ คือ มีค่าเท่ากับ 1 ถึง 31 ในเดือนที่ลงท้ายด้วยคำว่า "คม" มีค่าเท่ากับ 1 ถึง 30 ในเดือนที่ลงท้ายด้วยคำว่า "ชน" และในเดือนกุมภาพันธ์อาจมีค่าเท่ากับ 1 ถึง 28 หรือ 29 ก็ได้แต่ในการเก็บข้อมูลลงในรีเลชันนั้น บางกรณีที่เรามีการกำหนดโดเมนให้กับแอตทริบิวต์แล้ว แต่ข้อมูลที่จะถูกเก็บเข้าไปอาจถูกบรรจุเข้าไปในภายหลังลักษณะนี้จะทำให้เกิดค่าว่าง (Null Value) ขึ้นชั่วขณะก่อนที่จะมีการบรรจุค่าข้อมูลที่อยู่ในโดเมนที่กำหนดไว้เข้าไป ดังนั้น คำว่า "ค่าว่าง" จึงหมายถึงค่าที่ยังมีทราบชัดว่าแอตทริบิวต์นั้นจะมีค่าเป็นค่าใด หรือ ค่าของข้อมูลที่ไม่อยู่ในโดเมนที่กำหนด โดยมีข้อบังคับไว้ว่า แอตทริบิวต์ที่กำหนดให้เป็นคีย์หลักของรีเลชันจะมีค่าข้อมูลเป็นค่าว่างไม่ได้เสมอ เพราะจะทำให้การเข้าถึงข้อมูลในทึพเพิลนั้นกระทำไม่ได้

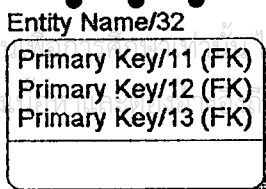
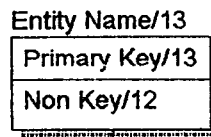
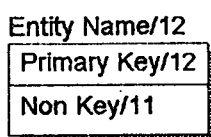
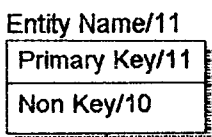
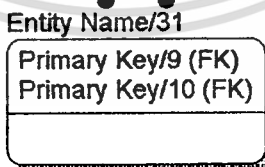
เมื่อมีการจัดเก็บข้อมูลในฐานข้อมูลใดๆ แล้ว ข้อมูลจะถูกแยกออกเป็นกลุ่มของข้อมูลเป็นชุดที่ประกอบด้วยแอตทริบิวต์ต่างๆที่มีความสัมพันธ์กัน เช่น การเก็บข้อมูลของบุคคลากรในโรงเรียนก็อาจแยกเก็บเป็นกลุ่มข้อมูลของนักเรียน , กลุ่มข้อมูลแต่ละกลุ่มนี้จะเรียกว่า เอนติตี้(entity) ซึ่งแต่ละเอนติตี้จะประกอบด้วยแอตทริบิวต์ต่างๆที่มีความสัมพันธ์กัน เช่น เอนติตี้ของนักเรียนก็จะประกอบไปด้วยชื่อ, นามสกุล, ที่อยู่, ชั้นเรียน เป็นต้น

2.2 อีอาร์โมเดล (Entity Relationship model - ER Model)

อีอาร์โมเดล เป็นแผนภาพที่ถูกรออกแบบมาเพื่อแสดงความสัมพันธ์ระหว่างเอนติตี้ต่างๆในรูปแบบที่เป็นรูปธรรมมากขึ้นเมื่อมีการใช้โมเดลอีอาร์ แสดงความสัมพันธ์ระหว่างเอนติตี้แล้วก็ไม่จำเป็นต้องมีคำอธิบายความสัมพันธ์ใดๆอีกเพราะ โมเดลอีอาร์ ประกอบไปด้วยสัญลักษณ์ต่างๆที่แสดงถึงคุณลักษณะของเอนติตี้และแอตทริบิวต์ได้เ็นตัวเองแล้ว ซึ่งสัญลักษณ์ต่างๆ ที่ใช้ในแผนภาพอีอาร์ โมเดล ของเออร์วินนี้ประกอบด้วยสัญลักษณ์ดังแสดงดังรูป



Turning Many to Many in One to Many relationship

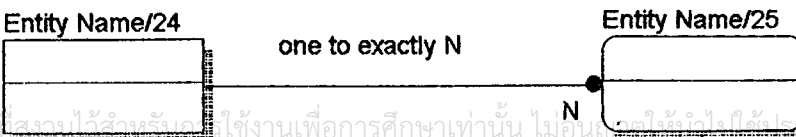
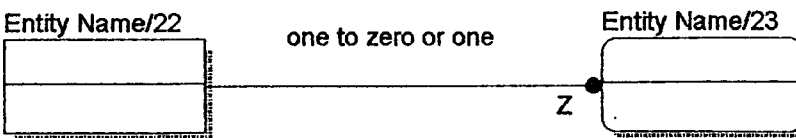
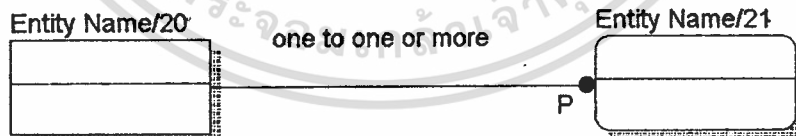
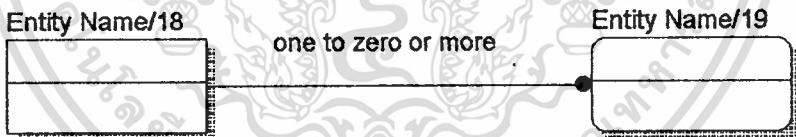
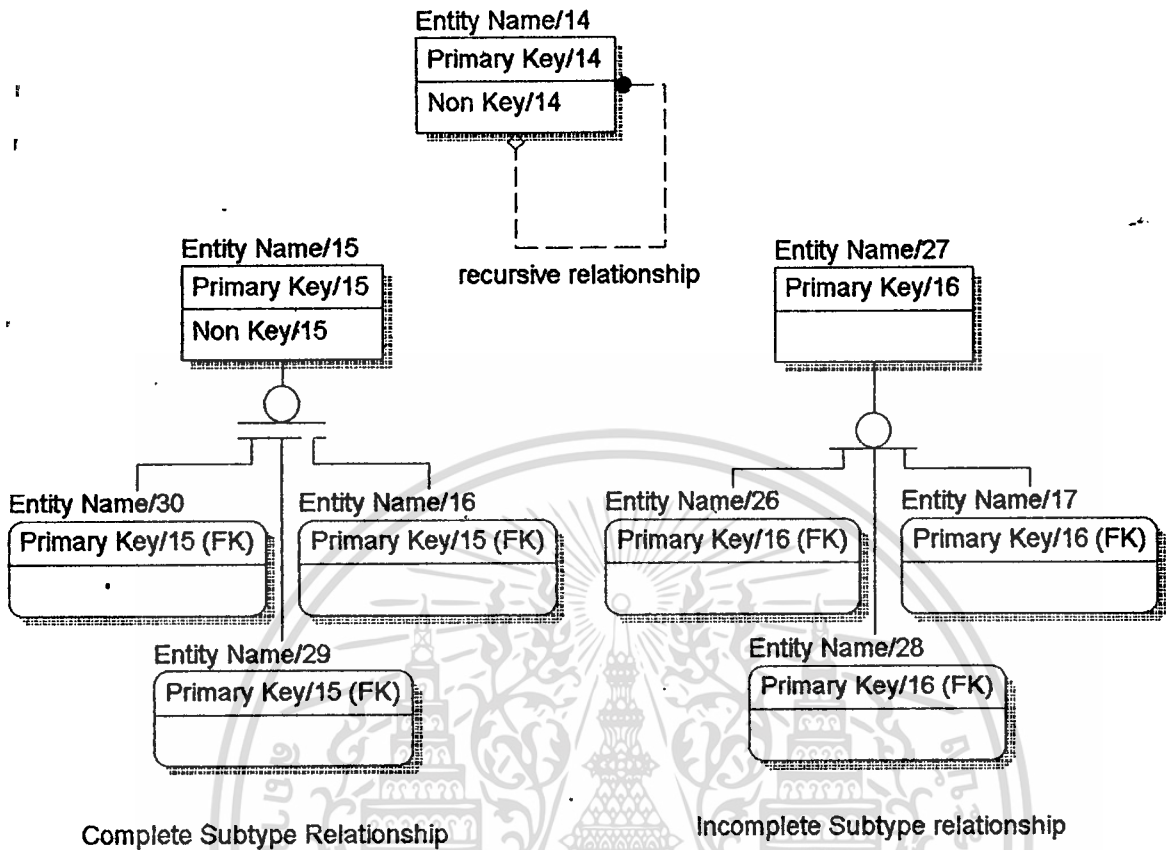


N-ary relationship

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในหน่วยงานเท่านั้น ไม่ควรเผยแพร่หรือทำซ้ำโดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงแก้ไขของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2-2 แสดงสัญลักษณ์ในการวาดแผนภาพอีอาร์



เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้เผยแพร่ไปภายนอกโดยไม่ขออนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2-3 แสดงสัญลักษณ์ในการวาดแผนภาพอีอาร์

2.2.1 การแปลงความสัมพันธ์ของข้อมูลจากอีอาร์โมเดล ไปสู่ในรูปของรีเลชัน

เนื่องจากเครื่องมือที่ใช้ในการทำแผนภาพอีอาร์ ได้แก่เออร์วิน แต่เนื่องจากเครื่องมือที่ใช้ นั้น จะทำรีเลชันไปพร้อมกัน ซึ่งสัญลักษณ์ในการแสดงความสัมพันธ์ของกลุ่มข้อมูลนั้นจะมีลักษณะที่แตกต่างจากอีอาร์โมเดล โดยทั่วไปดังรูปที่ได้กล่าวไปแล้วนั้น

การแปลงความสัมพันธ์ของข้อมูลจากอีอาร์โมเดล นี้ไปสู่ในรูปของรีเลชันชิป(relationships) ด้วยเออร์วินมีขั้นตอนดังต่อไปนี้

1. สำหรับแต่ละเอนทิตีที่ไม่ใช่เอนทิตีแบบอ่อน(weak entity) ของอีอาร์โมเดล นั้นจะสร้างรีเลชัน โดยมีแอตทริบิวต์ที่เป็นแอตทริบิวต์ธรรมดา (simple attribute) มาประกอบกัน แล้วทำการเลือกแอตทริบิวต์ใดแอตทริบิวต์หนึ่งหรือกลุ่มของแอตทริบิวต์มาทำหน้าที่เป็นคีย์หลักของรีเลชัน
2. สำหรับแต่ละเอนทิตีแบบอ่อน นั้นเราจะสร้างรีเลชันที่เรียกว่า ซับไทป์(Subtype) เกิดจากการรวมกันของแอตทริบิวต์ธรรมดาของเอนทิตีนั้น โดยที่รีเลชันนี้จะมีคีย์หลักคือคีย์รวม (combine key) ที่เกิดจากการรวมกันของคีย์หลักของเอนทิตีแบบอ่อน (partial key) กับคีย์หลักของเอนทิตีที่มันอ้างอิงถึง
3. สำหรับความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อหนึ่ง สร้างรีเลชันจากความสัมพันธ์ดังกล่าวได้ 2 ลักษณะ คือเลือกคีย์หลักของเอนทิตีใดเอนทิตีหนึ่งมาเป็นคีย์หลักของรีเลชันนี้แล้ว ให้คีย์หลักของเอนทิตีหนึ่งมาเป็นคีย์นอกของรีเลชันนี้ โดยถ้าความสัมพันธ์นี้มีแอตทริบิวต์ให้นำแอตทริบิวต์มารวมอยู่ในรีเลชันนี้
4. สำหรับความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อกลุ่ม เราจะสร้างรีเลชันจากความสัมพันธ์ดังกล่าวโดยนำเอาคีย์หลักของเอนทิตีฝั่งที่มีความสัมพันธ์แบบกลุ่มมาเป็นคีย์หลักของรีเลชันนี้แล้วให้นำเอาคีย์หลักของเอนทิตีฝั่งที่มีความสัมพันธ์แบบหนึ่งมาเป็นคีย์นอกของรีเลชันนี้ โดยถ้าความสัมพันธ์นี้มีแอตทริบิวต์ให้นำเอาแอตทริบิวต์เหล่านั้นมารวมอยู่ด้วย
5. สำหรับความสัมพันธ์ระหว่างเอนทิตีแบบกลุ่มต่อกลุ่ม เราจะสร้างรีเลชันจากความสัมพันธ์นี้โดยแบ่งเป็น 3 รีเลชันคือ 2 รีเลชันนั้นจะให้คีย์หลักของเอนทิตีนั้นเป็นคีย์หลักในแต่ละรีเลชันซึ่งประกอบด้วยแอตทริบิวต์ของเอนทิตีนั้นๆ และ อีกหนึ่งรีเลชันนั้น จะเป็นรีเลชันที่มีคีย์หลักเป็นคีย์รวมที่ประกอบด้วยคีย์หลักของทั้ง 2 เอนทิตีนั้นซึ่งจะนำแอตทริบิวต์ของทั้ง 2 เอนทิตีนี้มารวมอยู่ในรีเลชันนี้ด้วย
6. สำหรับเอนทิตีใดที่มีแอตทริบิวต์ที่มีค่าข้อมูลแบบหลายค่า (multivalued attribute หรือ repeating group) ซึ่งจะสร้างรีเลชันขึ้นมาเพิ่ม โดยมีคีย์หลักของเอนทิตีนั้น รวมกับแอตทริบิวต์ดังกล่าวมาเป็นคีย์หลักของรีเลชันใหม่นี้
7. สำหรับเอนทิตีหลายเอนทิตีที่มีความสัมพันธ์ต่อกันจะสร้างรีเลชันใหม่ที่สัมพันธ์กับเอนทิตีอื่นแบบหนึ่งต่อกลุ่ม

2.3 ทฤษฎีของระบบฐานข้อมูลเชิงสัมพันธ์

2.3.1 กฎของความคงสภาพ (integrity rule)

กฎของความคงสภาพของ โมเดลเชิงสัมพันธ์ เป็นทฤษฎีที่ช่วยยืนยันความถูกต้องของความสัมพันธ์ระหว่างข้อมูลว่า รีเลชันใดที่เป็นไปตามกฎของความคงสภาพนี้แล้ว่อมจะมีความสัมพันธ์ระหว่างข้อมูลอย่างถูกต้องอยู่ตลอดเวลา ไม่ว่ารีเลชันนั้นจะมีการเปลี่ยนแปลงข้อแก้ไขข้อมูลไปในแบบใดก็ตาม

กฎของความคงสภาพมีความหมายอยู่ 2 ลักษณะ คือ กฎความคงสภาพของเอนทิตี (entity integrity rule) และ กฎความคงสภาพของการอ้างอิง (referential integrity rule) ดังอธิบายได้ดังนี้

1. กฎความคงสภาพของเอนทิตี กล่าวว่

"แอตทริบิวต์ทุกตัวที่เป็นส่วนของคีย์หลัก จะ ไม่นอนุญาตให้มีค่าว่าง"

หมายความว่า คีย์หลักของทุกรีเลชันจะไม่สามารถเก็บค่าข้อมูลที่เป็นค่าว่างได้ เหตุผลของข้อกำหนดนี้ก็คือ เพื่อให้การเข้าถึงข้อมูลในทัพเพิลใด ๆ ของรีเลชันมีความเป็นไปได้เสมอ เพราะถ้าคีย์หลักของทัพเพิลใดมีค่าข้อมูลเป็นค่าว่างแล้ว ก็จะส่งผลให้การเข้าถึงข้อมูลในทัพเพิลนั้น ไม่สามารถกระทำได้อย่างแน่นอน

2. กฎความคงสภาพของการอ้างอิง กล่าวว่

"ถ้าเรามีรีเลชัน R2 ซึ่งมี FK เป็นคีย์นอกที่อ้างอิงถึงคีย์หลัก PK ใน รีเลชัน R1 สำหรับทุกค่าของ FK ใน R2 จะต้อง

ก. มีค่าเท่ากับค่า PK ในทัพเพิลใดทัพเพิลหนึ่งในรีเลชัน R1 หรือ

ข. มีค่าของแอตทริบิวต์ทุกตัวใน FK เป็นค่าว่าง"

หมายความว่า แอตทริบิวต์ใด ๆ ที่เป็นคีย์หลักของรีเลชันหนึ่ง เมื่อมีการนำแอตทริบิวต์นั้นไปเป็นคีย์นอกของอีกรีเลชันหนึ่ง การเป็นคีย์นอกของแอตทริบิวต์นั้นจะต้องมีโดเมนเป็น โดเมน เดียวกับแอตทริบิวต์ที่เป็นคีย์หลัก ทั้งนี้ ก็เพื่อให้การนำรีเลชันมาใช้งานร่วมกัน (การนำรีเลชันมา join กัน) กระทำได้อย่างถูกต้อง คือ ทุกแอตทริบิวต์ที่เป็นคีย์นอกจะต้องมีข้อมูลซ้ำกับข้อมูลของแอตทริบิวต์ที่เป็นคีย์หลักอย่างแน่นอน แต่อาจมีบางค่าข้อมูลของแอตทริบิวต์ที่เป็นคีย์หลักเป็นข้อมูลไม่อยู่ในโดเมนของแอตทริบิวต์ที่เป็นคีย์นอกก็ได้ นั่นคือ โดเมนของคีย์นอกจะต้องเล็กกว่า หรือเท่ากับ โดเมนของคีย์หลักเสมอ

2.3.2 ฟังก์ชันการขึ้นต่อกัน (Functional dependency)

ฟังก์ชันการขึ้นต่อกันเป็นข้อกำหนดที่ช่วยให้เราเห็นถึงความสัมพันธ์ ของแอตทริบิวต์ต่าง ๆ ที่อยู่ภายในรีเลชัน ทั้งเพราะแอตทริบิวต์ต่าง ๆ ที่อยู่ภายในเอนทิตีเดียวกันก็เป็นไปได้ที่แอตทริบิวต์เหล่านั้น จะมีความสัมพันธ์กันเอง โดยที่ความสัมพันธ์นี้อาจเกี่ยวข้องหรือไม่เกี่ยวข้องกับความสัมพันธ์ที่มันมีต่อกีย์หลักของเอนทิตีนั้นก็เป็นได้ ซึ่งการ

ที่แอตทริบิวเหล่านั้นมีความสัมพันธ์กันเองจะเป็นสิ่งที่เราต้องพิจารณาแยกออกเป็นรีเลชันย่อย ๆ เพราะแอตทริบิวของแต่ละรีเลชันของตนเองเท่านั้น

กำหนดรีเลชัน R ของ Y ของ R เป็นฟังก์ชันที่ขึ้นต่อแอตทริบิว X ของรีเลชัน เราสามารถเขียนแทนได้ด้วยสัญลักษณ์

$$R.X \longrightarrow R.Y$$

อ่านว่า R.X มีฟังก์ชันการขึ้นอยู่กับ R.Y

หรือ R.X มีฟังก์ชันในการเลือก R.Y

หรือ R.Y ขึ้นอยู่กับ R.X

นิยาม R.X มีฟังก์ชันการขึ้นอยู่กับ R.Y ก็ต่อเมื่อ ทุกค่าข้อมูลของแอตทริบิว X ใน R จะมีค่าข้อมูลของแอตทริบิว Y ใน R ได้เพียงค่าเดียวเสมอ โดยที่แอตทริบิว X และ Y อาจจะเป็นคีย์แบบรวม(composite key) ก็ได้

นิยาม R.X มีฟังก์ชันการขึ้นอยู่กับ R.Y อย่างเต็มที่ (R.Y fully functionally dependent on R.X) ก็ต่อเมื่อ R.X มีฟังก์ชันการขึ้นอยู่กับ R.X และไม่ขึ้นอยู่กับข้อมูลเพียงบางส่วนของ R.X โดยที่แอตทริบิว X และ Y อาจจะเป็นคีย์แบบรวมก็ได้

2.3.3 การนอร์มัลไลซ์ (Normalization)

การนอร์มัลไลซ์เป็นการออกแบบฐานข้อมูลแบบที่เป็นมาตรฐานที่สุด ออกแบบโดยคอดด์ โดยมีจุดประสงค์ของการออกแบบก็เพื่อลดความซ้ำซ้อนของความสัมพันธ์ของข้อมูลให้เหลือน้อยที่สุด (decomposition method)

ต่อมาได้มีการออกแบบเพิ่มเติมขึ้นอีก 2 ระดับ คือ

นอร์มัลระดับที่ 4 (Fourth Normal Form - 4NF)

นอร์มัลระดับที่ 5 (Fifth Normal Form - 5NF)

หากรีเลชันใดมีมาตรฐานถึงรูปแบบนอร์มัลระดับที่ 5 แล้วก็จะมั่นใจได้ว่ารีเลชันนั้นจะไม่มี ความซ้ำซ้อนของความสัมพันธ์ของข้อมูลอย่างแน่นอน

นอกจากนี้ยังมีการออกแบบรูปแบบนอร์มัลเพิ่มเติมระหว่างรูปแบบนอร์มัลระดับที่ 3 (Third Normal Form - 3NF) และ รูปแบบนอร์มัลระดับที่ 4 โดยบอยซ์(Boyce) และคอดด์ ซึ่งมีชื่อว่าบอยซ์คอดด์นอร์มัลฟอร์ม (Boyce Codd Normal Form - BCNF) อีกด้วย

รูปแบบนอร์มัลระดับที่ 1 (First Normal Form - 1NF)

การปรับรีเลชันให้อยู่ในรูปแบบนอร์มัลระดับที่ 1 คือ การปรับจากรีเลชันที่ไม่นอร์มัล (unnormalized relation) เช่น รีเลชันที่มีข้อมูลของแอตทริบิวบางช่องมีมากกว่า 1 ค่า (มีแอตทริบิว ที่มีข้อมูลเป็น repeating group)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นิยาม รีเลชันจะอยู่ในรูปแบบนอร์มัลระดับที่ 1 ก็ต่อเมื่อ โดเมนของแต่ละแอตทริบิวต์ประกอบด้วยข้อมูลที่เป็นหน่วยย่อยที่สุด “A relation is in first normal form (1NF) if and only if all underlying simple domains contain atomic values only”

สิ่งที่ได้จากการที่รีเลชันอยู่ในรูปแบบนอร์มัลระดับที่ 1 ก็คือ รีเลชันยังคงมีความซ้ำซ้อนของความสัมพันธ์ระหว่างข้อมูลอยู่มากมาย เพราะนิยามของรูปแบบนอร์มัลระดับที่ 1 นี้กำหนดเพียงเฉพาะว่า แต่ละแอตทริบิวต์ของรีเลชันจะมีโดเมนที่มีสมาชิกเป็นหน่วยที่เล็กที่สุดเท่านั้น มิได้เป็นการลดความซ้ำซ้อนของความสัมพันธระหว่างข้อมูลแต่ประการใด

รูปแบบนอร์มัลระดับที่ 2 (Second Normal Form - 2NF)

นิยาม รีเลชันจะอยู่ในรูปแบบนอร์มัลระดับที่ 2 ก็ต่อเมื่อรีเลชันนั้นอยู่ในรูปแบบนอร์มัลระดับที่ 1 แล้ว และทุกแอตทริบิวต์ที่ไม่เป็นส่วนใดส่วนหนึ่งของคีย์หลัก (non-key attribute) จะต้องขึ้นอยู่กับคีย์หลักของรีเลชันนั้นอย่างเต็มที่ “A relation is in 2NF if and only if it is in 1NF and every nonkey attribute fully depends on the primary key”

สิ่งที่ได้จากการที่รีเลชันอยู่ในรูปแบบนอร์มัลระดับที่ 2 คือ ข้อมูลของบางแอตทริบิวต์ไม่ใช่คีย์หลักอาจมีความสัมพันธ์กันเอง โดยที่ไม่มีความสัมพันธ์กับคีย์หลักเลย ซึ่งความสัมพันธ์ดังกล่าวนี้ถือว่าเป็นความซ้ำซ้อนประการหนึ่งของรีเลชันนั้น ๆ ที่จะต้องทำการลดด้วยรูปแบบนอร์มัลในระดับต่อไป

รูปแบบนอร์มัลระดับที่ 3 (Third Normal Form - 3NF)

นิยาม รีเลชันจะอยู่ในรูปแบบนอร์มัลระดับที่ 3 ก็ต่อเมื่อรีเลชันนั้นอยู่ในรูปแบบนอร์มัลระดับที่ 2 แล้ว และทุกแอตทริบิวต์ที่ไม่เป็นส่วนใดส่วนหนึ่งของคีย์หลัก จะต้องไม่เป็นฟังก์ชันที่ขึ้นต่อกันเอง “A relation is in third normal form if and only if it is in 2NF and every nonkey attribute is nontransitively dependent on the primary key”

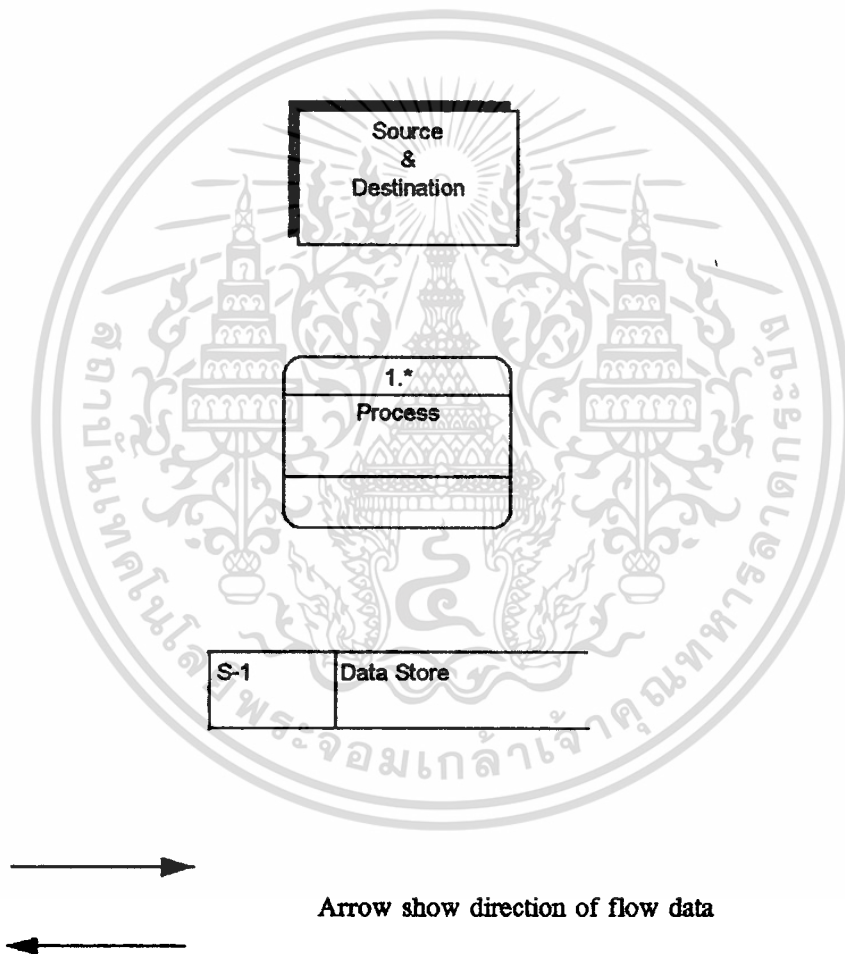
โดยปกติแล้ว สิ่งที่ได้จากการที่รีเลชันอยู่ในรูปแบบนอร์มัลระดับที่ 3 คือ รีเลชันจะไม่มีซ้ำซ้อนอีกต่อไป โดยที่จะสอดคล้องกับรูปแบบนอร์มัลระดับที่ 4 และ 5 ต่อไปเอง รีเลชันดังกล่าวจะมีลักษณะดังต่อไปนี้

- 1) เป็นรีเลชันที่มีหลายคีย์คู่แข่ง
- 2) เป็นคีย์คู่แข่งที่เกิดจากการรวมกันของคีย์ย่อยๆ
- 3) เป็นคีย์คู่แข่งที่มีการเหลื่อมซ้อนกัน (overlap กัน)

เนื่องจากรีเลชัน ณ ที่นี้ไม่มีลักษณะดังกล่าวจึงไม่ขอกล่าวถึงรูปแบบ บอยซ์คอตต์นอร์มัลฟอร์ม รูปแบบนอร์มัลระดับที่ 4 และ ระดับที่ 5

2.4 การเขียนแผนภาพการไหลของข้อมูล(Data Flow Diagram)

เครื่องมือที่ใช้ในการเขียนแผนภาพการไหลของข้อมูล ได้แก่ ซิลเวอร์รัน (Silverrun) มีสัญลักษณ์ที่ใช้ในการแสดงดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่บนสื่อโซเชียลมีเดียและต้องยกย่องเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2 4 สัญลักษณ์ของแผนภาพการไหลของข้อมูล

2.5 สภาพแวดล้อมในการพัฒนาโปรแกรม(Development Environment)

2.5.1 สภาพแวดล้อมของการทำงานบนเครื่องเดียว (Stand Alone Environment)

ขั้นตอนที่ต้องใช้เพื่อให้โปรแกรมประยุกต์(Application) และ ฐานข้อมูล(Database) สามารถทำงานบนเครื่องเดียวกันได้โดยไม่มีโปรแกรมที่ใช้พัฒนาหรือเพาเวอร์บีวเคอร์ จะต้องศึกษาสิ่งต่อไปนี้

การนำไฟล์ .EXE ที่สร้างได้จากโปรแกรมเพาเวอร์บีวเคอร์ไปใช้งานนั้น เนื่องจากไฟล์ .EXE ที่ได้นั้นไม่ใช่เป็นโปรแกรมที่สามารถเรียกมาใช้งานได้ทันทีที่เรียกว่าพีโค้ด (P-code) ซึ่งอยู่ระหว่างซอสโค้ดและภาษาเครื่องที่เอ็กซ์คิวต์ได้บนวินโดวส์ (Windows) โดยที่ไฟล์ .EXE จะมีการเก็บโค้ด(Code) ที่จะไปเรียกใช้ไฟล์.DLL ที่มีการเก็บโค้ดที่สามารถทำงานได้ ขึ้นมาทำงาน ดังนั้นไฟล์ .DLL จึงมีความจำเป็นสำหรับ โปรแกรมประยุกต์ ไฟล์ .DLLเหล่านี้ประกอบไปด้วย

PBBGR040.DLL	PBECT040.DLL	PBRTE040.DLL
PBCMP040.DLL	PBIDBF40.DLL	PBRTF040.DLL
PBDBI040.DLL	PBITXT40.DLL	PBSHR040.DLL
PBDEC040.DLL	PBLMI040.DLL	PBTYP040.DLL
PBDWE040.DLL	PBOUI040.DLL	PVBX040.DLL
PBDWO040.DLL	PBPRT040.DLL	

ในการใช้ระบบจัดการฐานข้อมูล จะมีความต้องการในการใช้ ไฟล์ .DLL เพิ่มเติมอีก โดยไฟล์ .DLL เหล่านี้จะรวมอยู่ในเครื่องมือพัฒนาการใช้ฐานข้อมูลของเพาเวอร์บีวเคอร์ (Database Development และ Deployment Kit) ซึ่งจะอยู่ในเพาเวอร์บีวเคอร์ชุดเอ็นเตอร์ไพรส์ (Enterprise Edition) ในกรณีที่ใช้ระบบจัดการฐานข้อมูลวอตคอม (Watcom DBMS) ที่เป็นรุ่น ผู้ใช้คนเดียว ในขณะที่ใช้งานจะต้องมีไฟล์เหล่านี้ด้วย

DB32W.EXE

DBL40W.DLL

WL40EN.DLL

และ ไดรเวอร์(Driver)

WOD40W.DLL

ถ้าโปรแกรมประยุกต์ใช้ โอคิบีซีไดรเวอร์ ต้องการไฟล์ดังนี้

ODBC.DLL ไฟล์นี้จะเก็บไว้ใน ไดรเรกทอรี (Directory) windows\system

PBODB040.DLL ไฟล์ไดรเวอร์โอคิบีซีของเพาเวอร์บีวเคอร์

PBODB040.INI ไฟล์เริ่มต้นของ PBODB040.DLL

และต้องสร้างหรือปรับปรุงไฟล์ของโอคิบีซี 2 ไฟล์ ในไดเรกทอรี windows

เอกสารนี้เป็นเอกสารของ **ODBCINST.INI** สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใด **ODBC.INI** ก็ทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่1 สร้างพีบีดี (PowerBuilder Dynamic Libraries - PBD) เพื่อช่วยลดขนาดของไฟล์เอ็กซีคิวต์โดยสามารถแบ่งโปรแกรมประยุกต์ที่เราสร้างขึ้นมานั้นเป็นหลายๆ ไฟล์ เช่น ในกรณีที่มีออบเจกต์(object) บางตัวที่ถูกใช้งานมากกว่า 1 โปรแกรมประยุกต์ เราสามารถนำส่วนประกอบนั้นมาทำพีบีดีได้เพื่อใช้ร่วมกันได้ ตัวอย่าง ได้แก่ หน้าต่างที่อนุญาตการเข้าถึงของผู้ใช้ (login window) และ คาด้าวินโดว์(DataWindow) เป็นต้น ซึ่งจะทำให้ขนาดของไฟล์เอ็กซีคิวต์ ของทุกโปรแกรมประยุกต์รวมกันลดลงและรวมพีบีอาร์ (PowerBuilder Resource File - PBR) ด้วย

ข้อควรคำนึงเมื่อทำพีบีดี

1. ถ้าพีบีดีมากเกินไปจะทำให้เปลืองเนื้อที่พอๆกับไฟล์เอ็กซีคิวต์ ขนาดใหญ่หนึ่งไฟล์
2. พยายามเลือกออบเจกต์ที่ใช้มากที่สุดในทุกโปรแกรมประยุกต์
3. เก็บแต่ออบเจกต์ที่ถูกใช้บ่อยๆ
4. ในการทำพีบีดีนั้นต้องรวมทุกไลบรารี(.PBL ไฟล์) ของโปรแกรมประยุกต์นั้น เข้าไว้ในพีบีดี หมด เพราะเวลาเอ็กซีคิวท์นั้นเพาเวอร์บิวเดอร์จะไม่ว่าออบเจกต์ใดถูกใช้หรือไม่

ออบเจกต์ที่ใช้ในโปรแกรมประยุกต์ประกอบด้วย

1. ภาษาที่ใช้ในการ โปรแกรมของเพาเวอร์บิวเดอร์ ที่เรียกว่าเพาเวอร์บิวเดอร์สคริปต์โค้ด (PowerScript Code)
2. ไฟล์รีซอร์ส (Resource file) อันได้แก่ ไฟล์รูปภาพ(BMP และ RLE files) , ไอคอน(ICO files) , และ พอยน์เตอร์(CUR files) ซึ่งในการทำไฟล์เอ็กซีคิวต์นี้ ไม่ควรรวมไฟล์รีซอร์สพวกนี้เข้าไปด้วยควรแยกไว้ต่างหาก

ขั้นตอนที่2 สร้างพีบีอาร์ เพื่อบ่งถึงไฟล์ทรัพยากรที่ต้องใช้ในเวลาที่ไฟล์เอ็กซีคิวต์ หรือ ไฟล์พีบีดีทำงาน โดยเขียนในลักษณะต่อไปนีลงใน เพาเวอร์บิวเดอร์ อิดิเตอร์ เช่นสมมติว่า เราจะเอาคาด้าวินโดว์ได้แก่ d_allocate_details ซึ่งเป็นรีซอร์สจากไลบรารีชื่อ allocate และรวมไฟล์นามสกุล PBL เช่น allocate.pbl (d_allocate_details) เป็นต้น

ขั้นตอนที่3 สร้างไฟล์เอ็กซีคิวต์ จากไลบรารีที่ไม่ได้รวมอยู่ในพีบีดีและกำหนดพีบีอาร์ที่ต้องใช้ในไฟล์เอ็กซีคิวต์

ขั้นตอนที่4 นำไฟล์เอ็กซีคิวต์ , ไฟล์พีบีอาร์ และ ไฟล์พีบีดีมารวมกันในโปรเจกต์เพนเตอร์(project painter)ที่เป็นขั้นตอนการทำงานใน เพาเวอร์บิวเดอร์

ขั้นตอนที่5 เมื่อทำการสร้าง เอ็กซีคิวไฟล์แล้วต้องการที่จะต่อกับเครื่องปลายทาง(Target Machine) ได้นั้นเพาเวอร์บิวเดอร์มีเครื่องมือในการติดต่อกับแต่ละชนิดของตัวจัดการฐานข้อมูล อันได้แก่เครื่องมือพัฒนาการใช้ฐานข้อมูลที่ประกอบด้วยไฟล์ดีแอลแอล ที่จำเป็นในเวลาเอ็กซีคิวต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 การพัฒนาระบบบนสภาพแวดล้อมของการใช้งานจริง (Target Environment)

การใช้งานจริงในที่นี้คือจะถูกใช้งานกับสิ่งต่อไปนี้

- ระบบจัดการฐานข้อมูล ที่ใช้คือ ไมโครซอฟท์ เอสคิวแอลเซิร์ฟเวอร์ 6.0 บน วินโดวส์เอ็นที 3.5
- เครื่องมือการพัฒนาระบบ ที่ใช้คือ เพาเวอร์ซอฟท์ เพาเวอร์บิวเดอร์ 4.0 บน วินโดวส์ฟอร์เวอร์ครูป
(ในที่นี้จะไม่กล่าวถึงการติดตั้ง วินโดวส์ฟอร์เวอร์ครูป , วินโดวส์เอ็นที ,เอสคิวแอลเซิร์ฟเวอร์และ เพาเวอร์บิวเดอร์ ถือว่าจะต้องติดตั้งให้เรียบร้อยก่อน จึงจะสามารถทำงานร่วมกันได้อย่างถูกต้อง)

ก่อนที่ระบบการจัดจำหน่าย จะถูกนำไปพัฒนาต่อในสภาพแวดล้อมของการใช้งานจริง และแจกจ่ายให้ ผู้นำไปใช้งาน จำเป็นจะต้องคำนึงถึงสิ่งต่อไปนี้

1. ต้องนำสโตร์โพรซีเยอร์ของเพาเวอร์ซอฟท์ ที่จำเป็นสำหรับ การใช้งานกับฐานข้อมูลบนเอสคิวแอลเซิร์ฟเวอร์ไปติดตั้งที่เครื่องเซิร์ฟเวอร์ก่อน
2. การสร้างฐานข้อมูลบนไมโครซอฟท์ เอสคิวแอลเซิร์ฟเวอร์ และ ไวยากรณ์ต่างๆที่จำเป็น
3. สภาพแวดล้อมของการใช้งานกับเอสคิวแอลเซิร์ฟเวอร์

2.5.2.1 สโตร์โพรซีเยอร์ที่จำเป็นในการใช้งานกับฐานข้อมูลบน SQL Server

เมื่อจะใช้ เพาเวอร์บิวเดอร์ เชื่อมต่อกับฐานข้อมูลบนเอสคิวแอลเซิร์ฟเวอร์เป็นครั้งแรกจำเป็นต้องนำสโตร์โพรซีเยอร์ไปติดตั้งที่เครื่องเซิร์ฟเวอร์ ด้วยการสั่งให้ทำงานภายนอกเพาเวอร์บิวเดอร์ โดยจะต้องมีส่วนเชื่อมต่อกับฐานข้อมูลของเพาเวอร์ซอฟท์ (Powersoft database Interface)คือ PBSYB040.DLL สโตร์โพรซีเยอร์ คือ กลุ่มของคำสั่งเอสคิวแอลที่ได้ถูกคอมไพล์และออปติไมซ์ไประดับหนึ่ง เพื่อที่จะทำงานด้านฐานข้อมูล (database operation) จะถูกทิ้งให้ทำงานอยู่บนเครื่องเซิร์ฟเวอร์เพื่อที่จะได้สามารถเรียกใช้ได้เสมอเมื่อต้องการ

สโตร์โพรซีเยอร์ที่จำเป็นเหล่านี้ใช้เพื่อจะหาข้อมูลของตารางและคอลัมน์จากตารางระบบของเอสคิวแอล เซิร์ฟเวอร์ (SQL Server system catalog) จะถูกเก็บรวมอยู่ในไฟล์ที่จำเป็น 2 ไฟล์ คือ PBSYB.SQL และ PBSYBRT.SQL ซึ่งไฟล์ทั้งสองสามารถหาได้จาก แผ่นที่ 1 ของไดรเวอร์ฐานข้อมูล (1 active Database Driver) จะต้องติดตั้งเองเพราะสโตร์โพรซีเยอร์จะไม่ถูกติดตั้งเมื่อติดตั้ง เพาเวอร์บิวเดอร์

PBSYB.SQL จะมี คำสั่งให้ทำลายสโตร์โพรซีเยอร์ของเพาเวอร์บิวเดอร์ ที่ได้มีอยู่ก่อนฐานข้อมูลหลัก (master database) บนเครื่องเซิร์ฟเวอร์ (ถ้าไม่เคยมี จะแสดงข้อความความผิดพลาด ซึ่งกรณีนี้ไม่ต้องสนใจ) หลังจากนั้นจะสร้างสโตร์โพรซีเยอร์เหล่านี้ขึ้นมาใหม่

จะต้องสั่งให้ PBSYB.SQL ทำงานหนึ่งครั้งต่อเครื่องเซิร์ฟเวอร์ที่ให้บริการฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2-1 อธิบายการทำงานอย่างคร่าวๆ ของสโตร์โพรซีเจอร์ที่มีในไฟล์ PBSYB.SQL

ชื่อสโตร์โพรซีเจอร์	การทำงาน
sp_pbcolumn	ข้อมูลของคอลัมน์ จากแคตตาล็อก
sp_pbdb	รายชื่อของฐานข้อมูล จากแคตตาล็อก
sp_pbindex	ข้อมูลของดัชนี จากแคตตาล็อก
sp_pbproc	ข้อมูลของสโตร์โพรซีเจอร์จากแคตตาล็อก
sp_pbhelpprotect	ข้อมูลความลับ จากแคตตาล็อก
sp_phtable	ข้อมูลตาราง จากแคตตาล็อก
sp_pbttext	ข้อมูลที่เป็นข้อความ จากแคตตาล็อก
sp_pbprimarykey	ข้อมูลของคีย์หลัก จากแคตตาล็อก
sp_pbforeignkey	ข้อมูลของคีย์นอก จากแคตตาล็อก
sp_pbfktable	ข้อมูลของตารางจากคีย์ อ้างถึง โดยตารางปัจจุบัน

PBSYBRT.SQL จะเป็นส่วนย่อยของ PBSYB.SQL จะมี คำสั่งให้ทำลายสโตร์โพรซีเจอร์ของเพาเวอร์วิวเคอร์ บางส่วนที่ได้มีอยู่ก่อนฐานข้อมูลหลัก (master database) บนเครื่องเซิร์ฟเวอร์ (ถ้าไม่เคยมี จะแสดงข้อความความคิดพลาด ซึ่งกรณีนี้ไม่ต้องสนใจ) หลังจากนั้นจะสร้างสโตร์โพรซีเจอร์เหล่านี้ขึ้นมาใหม่

กรณีที่จะต้องสั่งให้ PBSYBRT.SQL ทำงานก็คือก่อนที่จะทำงานกับ โปรแกรมที่สร้างโดยเพาเวอร์วิวเคอร์ แต่จะใช้ PBSYBRT.SQL ก็ต่อเมื่อสิ่งต่อไปนี้ไม่เป็นจริงทั้งคู่

1. วางแผนจะใช้เฮสติวแอลเซิร์ฟเวอร์ร่วมกับโปรแกรม
2. ไม่เคยทำการพัฒนาโปรแกรมด้วยเพาเวอร์วิวเคอร์บนเครื่องเซิร์ฟเวอร์นี้มาก่อน

ข้อยกเว้น ไม่จำเป็นต้องสั่งให้ PBSYBRT.SQL ทำงาน ถ้าเคยสั่งให้ PBSYB.SQL ทำงานไปแล้ว จะต้องสั่งให้ PBSYBRT.SQL ทำงานหนึ่งครั้งต่อเครื่องลูกข่ายเข้าสู่ฐานข้อมูล

ตารางที่ 2-2 อธิบายการทำงานอย่างคร่าวๆ ของสโตร์โพรซีเจอร์ที่มีในไฟล์ PBSYBRT.SQL

ชื่อสโตร์โพรซีเจอร์	การทำงาน
sp_pbindex	ข้อมูลของอินเด็กซ์ จากแคตตาล็อก
sp_phtable	ข้อมูลตาราง จากแคตตาล็อก
sp_pbprimarykey	ข้อมูลของคีย์หลัก จากแคตตาล็อก

วิธีการสั่งให้สโตร์โพรซีเจอร์ทำงาน

สามารถสั่งให้สโตร์โพรซีเจอร์ทำงาน โดยเปิดไฟล์ PBSYB.SQL หรือ PBSYBRT.SQL ใน WISQL หรือ ISQL แต่จะต้องแก้ไข ไฟล์ที่บรรทัดแรกของไฟล์

```
use master
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 go
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น มิได้ดัดแปลงเนื้อหาและองค์ความรู้เดิมของเอกสารฉบับนี้ที่มีการนำไปใช้
 ให้เปลี่ยน master เป็น ชื่อฐานข้อมูลที่จะทำงานด้วย หรือลบทั้งสองบรรทัดทิ้งก็ได้

2.5.2.2 การสร้างฐานข้อมูลบนไมโครซอฟท์ เอสคิวแอลเซิร์ฟเวอร์และไวยากรณ์ต่างๆที่จำเป็น
ฐานข้อมูลจะอยู่ในอุปกรณ์ที่เก็บฐานข้อมูล (database device) ได้ตั้งแต่ 1 อุปกรณ์ขึ้นไป และในแต่ละอุปกรณ์สามารถเก็บฐานข้อมูล ได้มากกว่า 1 ฐานข้อมูล แต่ละฐานข้อมูลจำเป็นจะต้องมี ล็อกของการเดินรายการ (transaction log) เพื่อบันทึกถึงสิ่งที่เกิดขึ้นกับฐานข้อมูล

ล็อกของการเดินรายการ ของเอสคิวแอลเซิร์ฟเวอร์เป็นแบบเขียนก่อน (write ahead) ซึ่งหมายความว่าคำสั่งที่ส่ง ไปยังเอสคิวแอลเซิร์ฟเวอร์จะต้องถูกบันทึกลงล็อกก่อน จึงจะทำให้ฐานข้อมูลเปลี่ยนแปลงได้ ล็อกของการเดินรายการนี้ความจริงก็เป็นตารางระบบ (system table) ตารางหนึ่ง ที่มีชื่อว่า syslogs ซึ่งไม่ควรที่จะเปลี่ยนแปลงหรือขออุโดยตรง ล็อกของการเดินรายการนี้จะใช้เนื้อที่ดิสก์ เป็นจำนวนมากดังนั้นในการที่จะเพิ่มประสิทธิภาพ ควรจะสร้างล็อก เอาไว้คนละอุปกรณ์ (device) กับ อุปกรณ์ที่เก็บฐานข้อมูลและการใส่ฐานข้อมูล และล็อกเอาไว้ในอุปกรณ์เดียวกัน เป็นการลดความสามารถในการกู้ข้อมูลอีกด้วย

สิ่งที่สำคัญ อุปกรณ์จะต้องถูกสร้างก่อนที่ฐานข้อมูลถูกสร้าง เพราะเมื่อสร้างฐานข้อมูล จะเก็บฐานข้อมูลและล็อกลงไปในอุปกรณ์

อุปกรณ์คือไฟล์ของระบบปฏิบัติการ แบ่งเป็น 2 ประเภท ได้แก่

- อุปกรณ์ที่เก็บฐานข้อมูล และ ล็อกของการเดินรายการ เรียกว่า database device จะเก็บอยู่ในไฟล์บนดิสก์
- อุปกรณ์ที่ใช้เก็บสำรองฐานข้อมูล และ ล็อกการเดินรายการ (การสำรองข้อมูลที่เกิดจากการคัมพ์) เรียกว่า อุปกรณ์คัมพ์ (dump device) จะสามารถเก็บได้ทั้งบนดิสก์, ดิสเก็ตต์ หรือ เทป

ผู้ที่สามารถสร้างอุปกรณ์ได้ คือ ผู้ดูแลระบบ (system administrator) เท่านั้น

เมื่อมีการติดตั้งเอสคิวแอลเซิร์ฟเวอร์เป็นที่เรียบร้อย จะมีอุปกรณ์ที่ถูกสร้าง โดยอัตโนมัติ คือ

1. อุปกรณ์ฐานข้อมูลชื่อ MASTER --- อุปกรณ์นี้ใช้เก็บฐานข้อมูลที่ชื่อ master , model , tempdb และ pubs
2. อุปกรณ์ฐานข้อมูลชื่อ MSDBDATA และ MSDBLOG --- อุปกรณ์ที่ชื่อ MSDBDATA ใช้โดย ฐานข้อมูล msdb ที่ใช้ในการจัดเวลา (scheduling) และ อุปกรณ์ที่ชื่อ MSDBLOG ใช้เก็บล็อกของ msdb
3. อุปกรณ์ที่ใช้เก็บสำรองฐานข้อมูล 3 อุปกรณ์ ได้แก่ DISKDUMP , DISKETTEDUMPA และ DISKETTEDUMPB

เมื่อผู้ดูแลระบบสร้างอุปกรณ์เพื่อเก็บฐานข้อมูล และ ล็อกการเดินรายการ หรือสร้างอุปกรณ์ที่ใช้เก็บสำรองฐานข้อมูล ควรจะต้องคัมพ์ฐานข้อมูล master เพราะว่าการสร้างอุปกรณ์ จะทำให้ฐานข้อมูล master เปลี่ยนแปลง

การทำงานกับฐานข้อมูล มีคำสั่งที่สำคัญ ได้แก่

เอกสารนี้เป็นเอกสาร **CREATE DATABASE** เป็นการสร้างฐานข้อมูลขึ้นมาใหม่ จะต้องมียุโรปกรณ์ที่จะใช้เก็บก่อน ดังที่ รศ. ได้กล่าวมาแล้ว ไม่ว่า

อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ALTER DATABASE เป็นการเพิ่มขนาดของฐานข้อมูลที่ได้สร้างไปแล้ว

ขนาดของฐานข้อมูลมีได้ตั้งแต่ 1 เมกะไบต์ (เพียงขนาด 2 กิโลไบต์ จำนวน 512 เพจ) จนถึง 2^{24} เมกะไบต์ ถ้าไม่สามารถของขนาดได้มากพอ จะจงเป็นจำนวนเท่าของ 0.5 เมกะไบต์

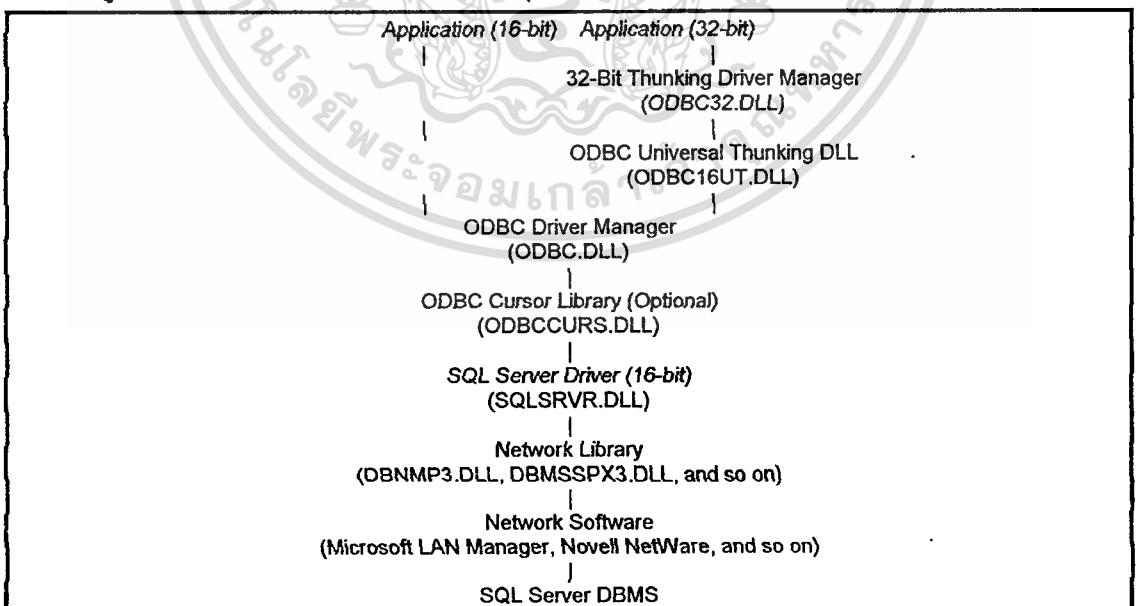
การทำงานกับฐานข้อมูลสามารถทำได้ทั้งบนเครื่องเซิร์ฟเวอร์และที่ไคลเอ็นท์ (ควรทำที่เซิร์ฟเวอร์ เพราะไม่จำเป็นต้องรู้คำสั่ง แต่ถ้าไม่ได้เป็นผู้ดูแลระบบ อาจต้องทำที่เครื่องไคลเอ็นท์ถ้า ไม่มีสิทธิใช้เครื่องที่เป็นเซิร์ฟเวอร์)

2.5.2.3 สภาพแวดล้อมของการใช้งานกับเอสคิวแอลเซิร์ฟเวอร์

เอสคิวแอล เซิร์ฟเวอร์เป็นระบบจัดการฐานข้อมูลแบบรีเลชันแนล ชนิดหลายผู้ใช้ (multi-user relational database management system) ซึ่งทำงานบนเครือข่ายท้องถิ่น (local area network) ไมโครซอฟท์ เอสคิวแอลเซิร์ฟเวอร์ทำงานบนระบบปฏิบัติการ OS/2 หรือ วินโดวส์เอ็นที จะสามารถหาไดรเวอร์ทั้งที่เป็นแบบ 16 บิต และเป็นแบบ 32 บิตได้ ใช้ภาษาเอสคิวแอลในการเข้าถึงข้อมูลในฐานข้อมูลของเอสคิวแอลเซิร์ฟเวอร์เครื่องไคลเอ็นท์สามารถสื่อสารกับเอสคิวแอลเซิร์ฟเวอร์ข้ามเครือข่าย ตัวอย่างเช่น ไมโครซอฟท์ แลนแมนเนเจอร์ (Microsoft LAN Manager), โนวेल เน็ตแวร์ (Novell Netware), (Banyan VINES) หรือเครือข่ายทีซีพี/ไอพี (TCP/IP)

ไดรเวอร์ของเอสคิวแอลเซิร์ฟเวอร์ทำให้โปรแกรมประยุกต์สามารถเข้าถึงข้อมูลในฐานข้อมูลของไมโครซอฟท์ เอสคิวแอลเซิร์ฟเวอร์ผ่านทาง การเชื่อมต่อ (Interface) ของโอดีบีซี

รูปที่ 2-5 สถาปัตยกรรมของโปรแกรมประยุกต์และไดรเวอร์สำหรับ สภาพแวดล้อมแบบ 16 บิต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

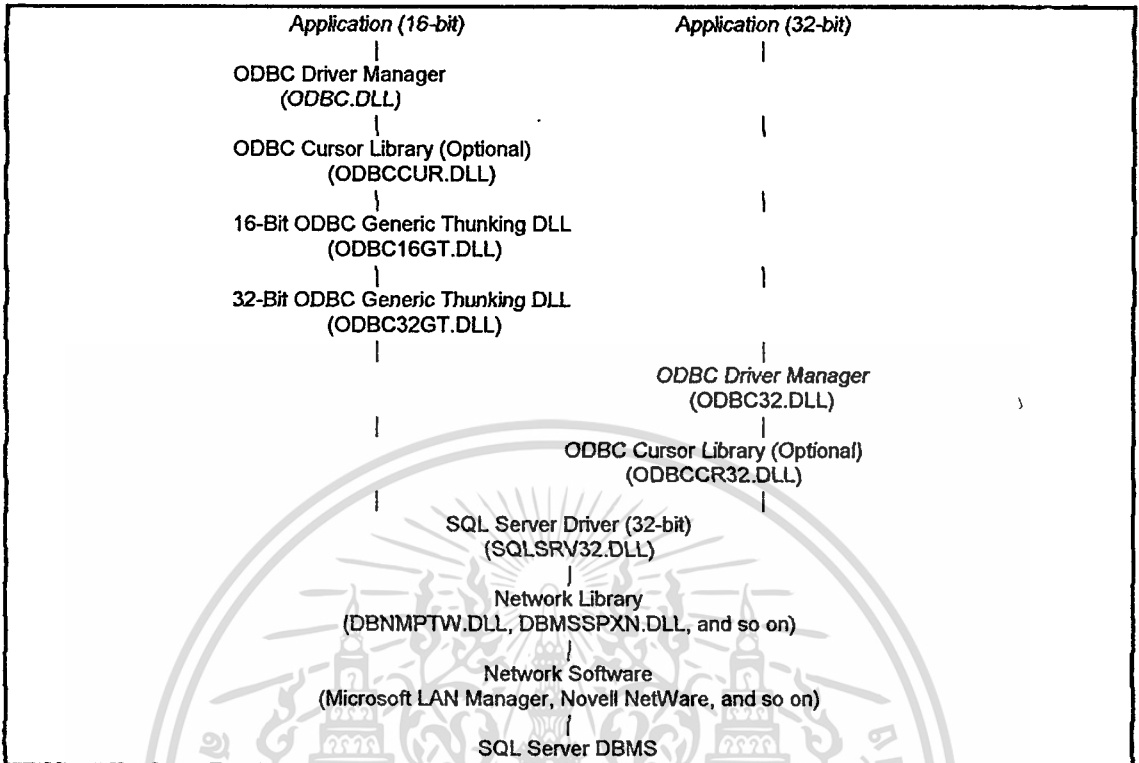
ตารางที่ 2-3 รายละเอียดของไฟล์ที่ใช้ในสภาพแวดล้อมแบบ 16 บิต

ชื่อไฟล์	พาท	ขนาดไฟล์(ไบต์)	วันที่(วัน/เดือน/ปี)
ODBC32.DLL	c:\windows\system	67,584	10/03/1994
ODBC16UT.DLL	c:\windows\system	5,792	17/08/1994
ODBC.DLL	c:\windows\system	55,792	17/08/1994
ODBCCURS.DLL	c:\windows\system	88,896	17/08/1994
SQLSRVR.DLL	c:\windows\system	161,392	12/07/1994
DBNMP3.DLL	c:\sql\bin	10,944	26/01/1994
(เวอร์ชัน 4.21.0.0)	c:\windows\system		
DBMSSPX3.DLL	c:\sql\bin	11,536	26/01/1994
(เวอร์ชัน 4.21.08)			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-6 สถาปัตยกรรมของ โปรแกรมประยุกต์และ ไครเวอร์สำหรับ สถาปัตยกรรมแบบ 32 บิต



ตารางที่ 2-4 รายละเอียดของไฟล์ที่ใช้ในสถาปัตยกรรมแบบ 32 บิต

ชื่อไฟล์	พาท	ขนาดไฟล์(ไบต์)	วันที่(วัน/เดือน/ปี)
ODBC32.DLL	c:\windows\system	67,584	10/03/1994
ODBC.DLL	c:\windows\system	55,792	17/08/1994

ความต้องการทางด้านฮาร์ดแวร์และซอฟต์แวร์

ในการจะเข้าถึงข้อมูลของเอสคิวแอลเซิร์ฟเวอร์ได้จะต้องมี

- ไครเวอร์ของเอสคิวแอลเซิร์ฟเวอร์
- ระบบจัดการฐานข้อมูลของ SQL Server
- เครื่องข่ายการติดต่อของระบบคอมพิวเตอร์

ไครเวอร์ของเอสคิวแอลเซิร์ฟเวอร์

เอสคิวแอลเซิร์ฟเวอร์จะต้องการฮาร์ดแวร์ดังต่อไปนี้

- เป็นคอมพิวเตอร์ที่ใช้สถาปัตยกรรมแบบมาตรฐานอุตสาหกรรม (Industry Standard Architecture -ISA) เช่น IBM PC/AT หรือ รุ่นเทียบเคียง
- เป็นคอมพิวเตอร์ที่ใช้สถาปัตยกรรมแบบไมโครช่องทาง (Micro Channel Architecture - MCA) เช่น IBM PS/2 หรือ รุ่นเทียบเคียง

- เป็นคอมพิวเตอร์ที่ใช้สถาปัตยกรรมแบบมาตรฐานอุตสาหกรรมที่ขยาย (Extended Industry Standard Architecture - EISA) ใช้ไมโครคอมพิวเตอร์ 80286,80386,หรือ 80486

- มีหน่วยความจำ (Random-access memory) อย่างน้อย 2 เมกกะไบต์ ; แนะนำให้มี 4 เมกกะไบต์

- มีฮาร์ดดิสก์ ประมาณ 200 กิโลไบต์สำหรับ ไดรเวอร์ของเอสคิวแอลเซิร์ฟเวอร์และ โอทีบีซีไครเวอร์ แมนเนเจอร์(ประมาณ 300 กิโลไบต์ในการติดตั้ง เคอร์เซอร์ไลบรารี(Cursor Library))

ไดรเวอร์ของเอสคิวแอลเซิร์ฟเวอร์ต้องการซอฟต์แวร์ในการทำงานดังนี้

- MS-DOS เวอร์ชัน 3.3 หรือสูงกว่า
- ไมโครซอฟท์ วินโดวส์ เวอร์ชัน 3.1 หรือ สูงกว่า (ไดรเวอร์ 16 บิต) หรือ ไมโครซอฟท์ วินโดวส์เอ็นที เวอร์ชัน 3.1 หรือ สูงกว่า (ไดรเวอร์ 32 บิต)
- ไครเวอร์ โอทีบีซี แมนเนเจอร์ เวอร์ชัน 2.0 หรือ สูงกว่า (ODBC.DLL สำหรับ 16 บิต หรือ ODBC32.DLL สำหรับ 32 บิต)

SQL Server

ในการเข้าถึงข้อมูลในเอสคิวแอลเซิร์ฟเวอร์ด้วยไดรเวอร์ของเอสคิวแอลเซิร์ฟเวอร์จะต้องมี ไมโครซอฟท์เอสคิวแอลเซิร์ฟเวอร์เวอร์ชัน 1.2 หรือ สูงกว่า จะต้องติดตั้ง แคตตาล็อก สตอร์โปรซีเจอร์ (Catalog Stored Procedure) สำหรับข้อมูลเกี่ยวกับฮาร์ดแวร์และซอฟต์แวร์ที่ต้องการสำหรับเอสคิวแอลเซิร์ฟเวอร์สามารถดูได้จากคู่มือการตั้งค่าระบบของ ไมโครซอฟท์ เอสคิวแอลเซิร์ฟเวอร์ (Configuration Guide) (สำหรับวินโดวส์ เอ็นที) หรือ คู่มือการติดตั้งไมโครซอฟท์ เอสคิวแอลเซิร์ฟเวอร์ (Installation Guide) (สำหรับ OS/2)

ซอฟต์แวร์สำหรับเครือข่าย

เครือข่ายที่ต้องการติดต่อกับแพคฟอร์มานเอสคิวแอลเซิร์ฟเวอร์และที่มีไดรเวอร์ของเอสคิวแอลเซิร์ฟเวอร์อยู่ด้วย ในการติดต่อกับไมโครซอฟท์ เอสคิวแอลเซิร์ฟเวอร์ จะสามารถใช้ ไมโครซอฟท์ วินโดวส์ฟอร์เวอร์กรุป, ไมโครซอฟท์ แลนแมนเนเจอร์หรือรุ่นเทียบเคียง เช่น IBM LAN Server หรือ DEC PathWorks, โนวเนตแควร์ , ทีซีพี/ไอพี (สำหรับ วินโดวส์เอ็นที) หรือ Banyan VINES (สำหรับ OS/2) สำหรับข้อมูลเกี่ยวกับฮาร์ดแวร์ และซอฟต์แวร์ ที่ต้องการสำหรับแต่ละเครือข่าย สามารถดูได้จากคู่มือของเครือข่าย

ไดรเวอร์ของเอสคิวแอลเซิร์ฟเวอร์ ติดต่อกับซอฟต์แวร์ของเน็ตเวิร์คโดยผ่านเอสคิวแอลเซิร์ฟเวอร์เน็ต-ไลบรารี อินเตอร์เฟส และต้องการ เนต-ไลบรารี ไดนามิก-ลิงค์ ไลบรารี ข้างล่างจะแสดงเน็ตเวิร์คไดนามิกลิงค์ไลบรารี ที่สามารถจะใช้กับทุกเครือข่ายสำหรับ ไมโครซอฟท์ เอสคิวแอลเซิร์ฟเวอร์

ตารางที่ 2-5 ซอฟต์แวร์ที่จำเป็นสำหรับเครือข่าย

สำหรับเครือข่าย	ไฟล์ DLL	แพคเกจ
Microsoft Windows NT; Microsoft Windows for Workgroups; Microsoft LAN Manager และรุ่นเทียบเท่า,เช่น IBM LAN Server หรือ DEC Pathworks Novell NetWare	DBNMP3.DLL (16 บิต) DBNMPTW.DLL (32 บิต) DBMSSPX3.DLL (16 บิต) DBMSSPXN.DLL (32 บิต)	ไดรเวอร์ของเอสคิวแอลเซิร์ฟเวอร์, Microsoft SQL Server Microsoftเอสคิวแอลเซิร์ฟเวอร์ สำหรับ Windows NT ; Network Integration Kit for Novell Netware

Banyan VINES	DBMSVIN3.DLL (16 บิต) DBMSVINN.DLL (32 บิต)	Microsoft เน็ตคิวแอลเซิร์ฟเวอร์ for Window NT; Network Integration Kit for Banyan VINES (สำหรับ OS/2)
TCP/IP network	DBMSSOC3.DLL (16 บิต) DBMSSOCN.DLL (32 บิต)	Microsoft เน็ตคิวแอลเซิร์ฟเวอร์ 4.21 (สำหรับ Window NT)

2.5.3 การย้ายแพลตฟอร์ม (Platform)

ในการย้ายแพลตฟอร์ม จากสภาพแวดล้อมของการพัฒนา ไปยังสภาพแวดล้อมของการทำงานจริง ต้องคำนึงถึงสิ่งต่อไปนี้

1. ชนิดของข้อมูลที่ใช้ควรจะใช้ชนิดของข้อมูลที่มีรองรับบนระบบจัดการฐานข้อมูลทั้งแบบทำงานบนเครื่องเดียวกัน และ ระบบจัดการฐานข้อมูลแบบบนเครื่องเซิร์ฟเวอร์
2. การใช้ เอสคิวแอลแบบฝังตัว ในโปรแกรมจะต้องคำนึงถึงไวยากรณ์ของภาษา
3. ประโยชน์ของคาค้าวิน โคว์โน เพาเวอร์บิวเคอร์ เพื่อการทำงานบนสภาพแวดล้อมที่แตกต่างกัน
4. การเขียน โปรแกรมส่วนที่เชื่อมต่อกับฐานข้อมูล (การ Connect Database) ที่ทำงานบนคนละสภาพแวดล้อมอาจแตกต่างกัน ในช่วงที่จำเป็นต้องพัฒนาควบคู่กันไป ควรจะหาวิธี ให้เลือก ได้สะดวกว่าจะเชื่อมต่อกับระบบจัดการฐานข้อมูลใด

2.5.3.1 : ชนิดของข้อมูล (Datatype)

จากการที่สภาพแวดล้อมที่เราสนใจในที่นี้แบ่งเป็น 3 อย่าง เราจะพิจารณาชนิดของข้อมูลที่มีให้ใช้ทั้ง 3 สภาพแวดล้อมซึ่งได้แก่

- ชนิดของข้อมูลที่เพาเวอร์บิวเคอร์รองรับ
- ชนิดของข้อมูลที่ระบบจัดการฐานข้อมูลวอดคอมมีให้ใช้
- ชนิดของข้อมูลที่เอสคิวแอลเซิร์ฟเวอร์มีให้ใช้
- ชนิดของข้อมูลทั้ง 3 แบบ จะอ้างอิงกับชนิดของข้อมูลของ ไอดีบีซี

ตารางที่ 2-6 เปรียบเทียบชนิดของข้อมูล

ODBC	SQL server	Watcom	เพาเวอร์บิวเคอร์	คำอธิบายของ ODBC
SQL_BINARY	binary, timestamp	BINARY(n) (1 ≤ n ≤ 32,767)		ข้อมูลฐานสองขนาด n (1 ≤ n ≤ 255)
SQL_BIT	bit	BINARY	Boolean	ข้อมูลฐานสองขนาด 1 บิต
SQL_BIGINT				ข้อมูลตัวเลข 19 หลัก (ไม่มีเครื่องหมาย) 20 หลัก (มีเครื่องหมาย) และ ทศนิยม 0 ตำแหน่ง (มีเครื่องหมาย $-2^{63} \leq n \leq 2^{63}-1$, ไม่มีเครื่องหมาย $0 \leq n \leq 2^{64}-1$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของลิขสิทธิ์
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารหรือลิขสิทธิ์ที่ปรากฏ

ODBC	SQL server	Watcom	เพาเวอร์วิวเดอร์	คำอธิบายของ ODBC
SQL_CHAR	char, character	CHAR[n], CHARACTER[n], VARCHAR[n], CHARACTER VARING[n] (1 <= n <= 32,767)	Char หรือ character (ความยาว 1 อักขระ), String (ความยาว 1 - 60000)	อักขระความยาวคงที่ n (1 <= n <= 256)
SQL_DATE		DATE (4 ไบต์)	Date	ข้อมูลวันที่
SQL_DECIMAL	decimal, dec, money, smallmoney	DECIMAL[(p,s)] <i>default p=30, s=6</i>	Decimal หรือ Dec ตัวเลขมีเครื่องหมาย ความยาว 18 หลัก	ตัวเลขมีเครื่องหมายขนาด p หลัก และ ทศนิยม s ตำแหน่ง (1 <= p <= 15; 0 <= s <= p)
SQL_FLOAT	float, double precision, float(n) n=8-15	FLOAT, DOUBLE (8 ไบต์ มีค่าตั้งแต่ 2.22507385850720 160e-308 ถึง 1.79769313486231 560e+308)	Double ตัวเลขมีเครื่องหมาย ความยาว 15 หลัก 2.2e-308 ถึง 1.7e308	ตัวเลขมีเครื่องหมาย ค่าประมาณ มี mantissa 15 หลัก (ศูนย์หรือค่า absolute 10^{-308} ถึง 10^{308})
SQL_INTEGER	int, integer	INTEGER, INT (1 <= n <= 2,147,483,647)		ตัวเลขขนาด 10 หลัก และ ทศนิยม 0 ตำแหน่ง (มีเครื่องหมาย $-2^{31} <= n <= 2^{31}-1$, ไม่มีเครื่องหมาย $0 <= n <= 2^{32}-1$)
SQL_LONGVAR BINARY	image	LONG BINARY		ข้อมูลฐานสองขนาดเปลี่ยนแปลงได้ ขนาดที่มากที่สุดขึ้นอยู่กับขนาดไฟล์
SQL_LONGVAR CHAR	text	LONG VARCHAR		ข้อมูลอักขระขนาดเปลี่ยนแปลงได้ ขนาดที่มากที่สุดขึ้นอยู่กับขนาดไฟล์
SQL_NUMERIC	numeric	NUMERIC[(p,s)]		ตัวเลขมีเครื่องหมายขนาด p หลัก และ ทศนิยม s ตำแหน่ง (1 <= p <= 15; 0 <= s <= p)
SQL_REAL	real, float(n) n=1-7	REAL (4 ไบต์ มีค่าตั้งแต่ 1.175494351e-38 ถึง 3.402823466 e+38)	Real (ตัวเลขมีเครื่องหมาย ความยาว 6 หลัก 1.17e-38 ถึง 3.4e+38)	ตัวเลขมีเครื่องหมาย ค่าประมาณ มี mantissa 7 หลัก (ศูนย์หรือค่า absolute 10^{-38} ถึง 10^{38})
SQL_SMALLINT	smallint	SMALLINT (มีเครื่องหมาย - 32768 <= n <= 32767 หรือ 2 ไบต์)	Integer หรือ Int (16 บิต -32768 - 32767), UnsignedInteger, U nsignedInt, UInt	ตัวเลขขนาด 5 หลักและทศนิยม 0 ตำแหน่ง (มีเครื่องหมาย -32768 <= n <= 32767, ไม่มีเครื่องหมาย $0 <= n <= 65535$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาและการวิจัยเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ODBC	SQL server	Watcom	เพาเวอร์บิวเดอร์	คำอธิบายของ ODBC
SQL_TIME		TIME (8 ไบต์)	Time (16 บิต 0 - 65,535)	ข้อมูลเวลา
SQL_TIMESTAMP	datetime, smalldatetime	TIMESTAMP (8 ไบต์)	DateTime	ข้อมูลวันที่ และ เวลา
SQL_TINYINT	tinyint			ตัวเลขมี 3หลักและทศนิยม 0 ตำแหน่ง แบบมีเครื่องหมาย $-128 \leq n \leq 127$ แบบไม่มีเครื่องหมาย $0 \leq n \leq 255$
SQL_VARBINARY	varbinary, binary varying			ข้อมูลฐานสองขนาดมากที่สุด n $1 \leq n \leq 255$
SQL_VARCHAR	varchar, character varying, char varying, sysname		Long (32 บิต มีเครื่องหมาย มีค่า -2,147,483,648 ถึง 2,147,483,647)	อักขระขนาดเปลี่ยนแปลงได้มีขนาด มากที่สุด n ($1 \leq n \leq 254$)
			UnsignedLong, Ulong (32 บิต ไม่มีเครื่องหมาย มีค่า 0- 4,294,967,295)	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3.2 การใช้เอสคิวแอลแบบฝังตัว (Embedded SQL)

ในหัวข้อนี้จะแสดงตัวอย่างการใช้เอสคิวแอลแบบฝังตัว ที่มักใช้ใน โปรแกรมได้แก่คำสั่ง

SELECT

INSERT

UPDATE

และ DELETE

ในเอสคิวแอล เพนเตอร์ ของเพาเวอร์บิวเคอร์ นั้นจะสร้างคำสั่งเอสคิวแอล ที่มีไวยากรณ์

ลักษณะนี้

1. ชื่อตัวแปร เมื่อนำไปใช้ในคำสั่งเอสคิวแอล ต้องมี : (colon) นำหน้า
2. เมื่อจบคำสั่งเอสคิวแอล จะต้องมี ; (semicolon) ปิดท้าย
3. ชื่อของตาราง และ ชื่อคอลัมน์ จะต้องอยู่ในเครื่องหมาย “...”
4. การอ้างถึงชื่อคอลัมน์ ต้องอ้างชื่อตารางก่อนเสมอ และชั้นระหว่างชื่อตารางกับชื่อคอลัมน์ด้วย .
5. ชื่อของตาราง และ ชื่อคอลัมน์ที่จะอ้างถึง Watcom ไม่สนใจว่าเป็นตัวใหญ่หรือเล็ก (ไม่ case sensitive)
6. การแทนค่าคงที่ในคำสั่ง ถ้าเกี่ยวข้องกับตัวแปรที่เป็นตัวอักษร (string) ให้ใช้เครื่องหมาย ‘’ เช่น ‘BANGKOK’ และถ้าเป็นตัวแปรที่เป็นตัวเลข (numeric) ไม่ต้องใช้เครื่องหมายใดๆ เช่น 20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไวยากรณ์ที่สร้างโดยเอสคิวแอล เพนเตอร์
<pre>SELECT "customer"."customer_name", "customer"."customer_address" INTO :cust_name ,:cust_address FROM "customer" WHERE "customer"."customer_id" = :cust_id ;</pre>

ไวยากรณ์ของเอสคิวแอลเซิร์ฟเวอร์
<pre>SELECT customer_name, customer_address INTO :cust_name ,:cust_address FROM customer WHERE customer_id = :cust_id ;</pre>

คำสั่งข้างบน เป็นการหาชื่อลูกค้า (customer_name) และ ที่อยู่ลูกค้า (customer_address) ที่มีหมายเลข (customer_id) เหมือนกับ cust_id แล้วเก็บค่านั้นไว้ใน cust_name และ cust_address ตามลำดับ

ไวยากรณ์ที่สร้างโดยเอสคิวแอล เพนเตอร์
<pre>INSERT INTO "customer" ("customer_id", "salesman_id", "customer_name", "customer_abbrev", "credit_receive") VALUES ('00001', '0001', :cust_name, null, 20) ;</pre>

ไวยากรณ์ของเอสคิวแอลเซิร์ฟเวอร์
<pre>INSERT INTO customer (customer_id , salesman_id , customer_name , customer_abbrev , credit_receive) VALUES ('00001', '0001', :cust_name, null, 20) ;</pre>

คำสั่งข้างบนเป็นการสร้างค่าใหม่ลงในตารางลูกค้า (customer) โดยกำหนดหมายเลขลูกค้าเป็น 00001 , กำหนดให้ชื่อลูกค้า เหมือนกับค่าในตัวแปร cust_name , ไม่ได้กำหนดค่าชื่อย่อของลูกค้า จึงให้เป็นค่าว่างและค่าสถิติการได้รับเครดิต เป็น 20

ไวยากรณ์ที่สร้างโดยเอสคิวแอล เพนเตอร์
<pre>UPDATE "customer" SET "customer_id" = '00001', "customer_name" = :cust_name, "customer_address" = null ;</pre>

ไวยากรณ์ของเอสคิวแอลเซิร์ฟเวอร์
<pre>UPDATE customer SET customer_id = '00001', customer_name = :cust_name, customer_address = null ;</pre>

คำสั่งข้างบนเป็นการปรับปรุงค่าในตารางลูกค้า โดยกำหนดค่าหมายเลขลูกค้า เป็น 00001 , ชื่อลูกค้า เป็นเหมือนกับในตัวแปร cust_name และให้ค่าที่อยู่ลูกค้าเป็นค่าว่าง

ไวยากรณ์ที่สร้างโดยเอสคิวแอล เพนเตอร์
<pre>DELETE FROM "customer" WHERE "customer"."customer_id" = :cust_id ;</pre>

ไวยากรณ์ของเอสคิวแอลเซิร์ฟเวอร์
<pre>DELETE FROM customer WHERE customer_id = :cust_id ;</pre>

คำสั่งข้างบนเป็นการลบลูกค้าที่มีหมายเลขลูกค้าเหมือนกับในตัวแปร cust_id

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3.3 ประโยชน์ของดาต้าวินโดวใน เพาเวอร์บิวคอร์ด

ในการทำงานบนสภาพแวดล้อมที่แตกต่างกัน ในช่วงที่ทำการย้ายระบบจากเครื่องที่ใช้พัฒนา (Development Machine) ไปยังเครื่องที่ใช้งานจริงหรือเครื่องเป้าหมาย (Target Machine) จะมีช่วงหนึ่งของการพัฒนาที่ต้องทำความเข้าใจกันไปที่ทั้งสองเครื่อง ในช่วงนี้ ถ้าเราได้วางแผนให้ฐานข้อมูลที่อยู่บนสองสภาพแวดล้อมที่แตกต่างกัน มีโครงสร้าง(schema) ที่เหมือนกันทุกอย่าง ได้แก่ มีตารางที่เหมือนกัน ในตารางมีชื่อคอลัมน์เหมือนกัน และ คอลัมน์ที่มีชื่อเดียวกันของตารางชื่อเดียวกัน จะต้องใช้ชนิดของข้อมูลเดียวกัน หรือ พยายามให้เหมือนกันให้มากที่สุด (เรื่องการเปรียบเทียบชนิดของข้อมูลมีในหัวข้อที่ 2.5.2.3) และ มีการสร้างสโตร์โพรซีเจอร์ และ เคอร์เซอร์ ที่ทำงานได้ผลลัพธ์เหมือนกันทุกประการ การวางแผนแบบนี้ จะเป็นประโยชน์มาก เมื่อใช้งานร่วมกับ ดาต้าวินโดวของเพาเวอร์บิวคอร์ด เพราะว่าดาต้าวินโดวเป็นการทำส่วนเชื่อมต่อกับผู้ใช้ (user interface) สำหรับคำสั่งเอสคิวแอล ซึ่งจากหัวข้อที่แล้ว จะเป็นได้ว่า ไวยากรณ์ที่เป็นคำสั่งเอสคิวแอลอาจไม่เหมือนกัน แต่ถ้าเป็นไวยากรณ์ที่สร้างโดยดาต้าวินโดวเพนเตอร์ จะไม่ต้องมาแก้ไข เมื่อย้ายการทำงานจากฐานข้อมูลหนึ่ง ไปยังอีกฐานข้อมูลหนึ่ง ที่มีโครงสร้างเหมือนกันดังได้กล่าวมาแล้ว ดาต้าวินโดวเมื่อถูกเรียกขึ้นมาแก้ไข จะพบว่าคำสั่งเอสคิวแอล ได้ถูกแก้ไขให้เรียบร้อยแล้ว

2.5.3.4 การเขียนโปรแกรมส่วนที่เชื่อมต่อกับฐานข้อมูล(การ Connect Database)

ในช่วงที่ทำการย้ายเครื่องพัฒนา (เหมือนในหัวข้อที่แล้ว) นอกจากจะต้องคำนึงถึงคำสั่ง เอสคิวแอลที่มีไวยากรณ์แตกต่างกันแล้ว วิธีการเชื่อมต่อกับฐานข้อมูล ก็ยังแตกต่างกันด้วย ที่เห็นได้ชัดเจนคือ ตัวอักษรที่ใช้ในการเชื่อมต่อ (Connect String) ตัวอย่างเช่น

ตัวอักษรที่ใช้ในการเชื่อมต่อกับบวอดคอมคือ

```
'DSN=cops;uid=[sqlca.logid];pwd=[sqlca.logpass];'
```

ตัวอักษรที่ใช้ในการเชื่อมต่อกับเอสคิวแอลเซิร์ฟเวอร์คือ

```
'ODBC;uid=[sqlca.logid];pwd=[sqlca.logpass];DSN=[data source name];'
```

โดย [sqlca.logid] เป็นชื่อผู้ใช้ของระบบ (login identification) และ [sqlca.logpass] เป็นรหัสผ่านของผู้ใช้คนเดียวกัน และ [data source name] เป็นชื่อเรียกแหล่งฐานข้อมูล จะอ้างมาจากการตั้งค่าด้วยไอดีบีซี ดังนั้นอาจใช้ปุ่มตัวเลือกในการเลือกฐานข้อมูล ขณะที่ทำการเข้าสู่ระบบ เช่น

รูปที่ 2-7 ตัวอย่างการเขียนโปรแกรมส่วนเชื่อมต่อกับฐานข้อมูล

```
IF sqlca.dbms = 'ODBC' THEN
  IF rb_sql_server.Checked THEN
    sqlca.Dbparm = "connectstring='ODBC;uid=" + sqlca.logid + ";pwd=" + &
      sqlca.logpass + ";DSN=Cops SQL Server,'"
  ELSE // rb_watcom.Checked
    sqlca.Dbparm = "connectstring='DSN=cops;uid=" + &
```

```
sqlca.logid + ";pwd=" + sqlca.logpass + ""
```

```
END IF
```

```
END IF
```

2.5.3.5 ข้อดีข้อเสียของการพัฒนาระบบบนสภาพแวดล้อมที่แตกต่างกัน

ข้อดี การพัฒนาระบบบนสภาพแวดล้อมที่แตกต่างกันมีข้อดีคือ

1. การพัฒนาระบบสามารถเกิดควบคู่ไปกับการใช้งานจริง โดยไม่รบกวนกัน

เมื่อระบบถูกนำไปใช้งานจริง ย่อมต้องมีความต้องการที่จะแก้ไขปรับปรุงเป็นระยะๆ แต่เมื่อมีผู้ใช้ที่ต้องการใช้งานอยู่ตลอดเวลา ทำให้ไม่สามารถหยุดการใช้งานเพื่อที่จะแก้ไขปรับปรุง โปรแกรม และการหยุดอาจทำให้เกิดผลเสียมูลค่ามหาศาล ดังนั้นทางแก้ก็คือ ต้องหาทางพัฒนาหรือปรับปรุงระบบให้ได้เป็นไปตามต้องการให้มากที่สุดเสียก่อน แล้วใช้เวลาในการย้ายการท่งระบบใหม่เข้ามาสู่การใช้งานจริงให้เร็วที่สุด และกระทบกระเทือนกับการทำงานจริงน้อยที่สุด ซึ่งสภาพแวดล้อมของการพัฒนานั้น ไม่จำเป็นต้องเหมือนกันทุกประการกับสภาวะการใช้งานจริง เช่นสามารถทำบนเครื่องเดียวกันได้ ในขณะที่การใช้งานจริงเป็นแบบไคลเอ็นท์เซิร์ฟเวอร์

2. ทำให้จำกัดความเสียหาย

ในกรณีที่ระหว่างขั้นตอนการพัฒนาเกิดความผิดพลาด เช่น ความผิดพลาดจากการทดลองเขียนโปรแกรม เมื่อฐานข้อมูลที่ทำมาทดสอบนั้นเป็นคนละฐานข้อมูลกับที่ผู้ใช้ใช้งานอยู่ ย่อมเกิดความเสียหายน้อยหรืออาจไม่นับเป็นความเสียหายเลยก็ได้

ข้อเสีย การพัฒนาระบบบนสภาพแวดล้อมที่แตกต่างกันมีข้อเสียคือ

1. การพัฒนาแบบสองสภาพแวดล้อมจะทำได้ก็ต่อเมื่อ ทั้งสองสภาวะมีความแตกต่างกันที่จำกัด ถ้าไม่สามารถหาข้อสรุปของความเหมือนกันได้ ก็ไม่สามารถทำงานแบบนี้ได้ เช่น ควรจะมีมาตรฐานในการเชื่อมต่อกับฐานข้อมูลแบบเดียวกัน เช่นการใช้โอเคบีซี
2. ไม่ได้เป็นการรับประกันว่า โปรแกรมที่พัฒนาและตรวจสอบจนทำงานได้ถูกต้องแล้ว บนสภาพแวดล้อมของการพัฒนา จะทำงานได้ถูกต้องทุกประการเมื่อนำไปใช้งานบนสภาพแวดล้อมของการใช้งานจริง อาจต้องแก้ไขบางสิ่งที่อาจมาจากความแตกต่างกันของสภาพแวดล้อม
3. การจะสามารถพัฒนาระบบบนสองสภาพแวดล้อมได้ค่านั้น จะต้องเข้าใจระบบทั้งสองเป็นอย่างดี และสามารถเปรียบเทียบหาความแตกต่างกัน เพื่อที่จะได้ทราบว่าจะงานที่ต้องทำหลังจากที่นำไปพัฒนาต่อบนสภาพแวดล้อมของการใช้งานจริง มีอะไรบ้าง และต้องให้ความสำคัญเป็นพิเศษกับสิ่งไหน ถ้ามีสิ่งใดที่สามารถป้องกันไม่ให้เกิดปัญหาได้ ตั้งแต่การพัฒนาระบบบนสภาพแวดล้อมของการพัฒนา ก็ควรจะทำ เพื่อลดงานที่จะตามมาในอนาคต

ตัวอย่าง การใช้เอสคิวแอลแบบฝังตัว ในสภาพแวดล้อมของการพัฒนาเอสคิวแอลของวอดคอม มีไอเอกสารนี้ เป็นเอกสารที่สงวนไว้สำหรับบุคคลากรที่ควรจะศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ขorman ได้ 2 แบบ แต่แบบเดียวเท่านั้น ที่นำไปใช้ได้กับสภาพแวดล้อมของการใช้งานจริงเอสคิวแอลของ เอสคิวแอลเซิร์ฟเวอร์ และทำให้ไม่ต้องแก้สคริปต์ใน เพาเวอร์วิวเคอร์

4. การพัฒนาระบบบนสองสภาพแวดล้อมควบคู่กัน ไปนั้น จะได้ผลดีก็ต่อเมื่อการออกแบบได้ถูกระบุลงไปแน่ชัดแล้ว เช่น โครงสร้างของฐานข้อมูลจะไม่ถูกแก้ไขอีก เพราะการแก้ไขจะต้องทำทั้งสองสภาพแวดล้อม (เนื่องจากมีฐานข้อมูลสองฐานข้อมูล ที่มีโครงสร้างเหมือนกัน) ถ้าต้องแก้ไขบ่อยครั้ง เป็นการเพิ่มงานที่ต้องทำเป็นสองเท่า และทำให้เสียเวลามากขึ้น
5. จะต้องคำนึงถึงข้อจำกัดที่มีมากขึ้น เพราะต้องพิจารณาของทั้งสองสภาพแวดล้อม ตั้งแต่เริ่มต้นการพัฒนา ระบบบนสภาพแวดล้อมการพัฒนา ถ้าทำไม่ครบถ้วนเพียงพอ ในที่สุด ข้อจำกัดของสภาพแวดล้อมที่ใช้งานจริง จะต้องถูกพบและนำไปหามาให้ได้ในภายหลัง

2.5.3.6 การนำข้อมูลเข้าออกจากค่าเบสหนึ่งกับอีกฐานข้อมูลหนึ่ง(Import Export Data)

ในเพาเวอร์วิวเคอร์รี่ได้สร้างออปเจกต์ตัวหนึ่งได้แก่ ค่าไปป์ไลน์เพนเตอร์(Data Pipeline Painter) ประโยชน์ของค่าไปป์ไลน์เพนเตอร์มีดังนี้

1. ส่งผ่านข้อมูล และแอตทริบิวต์ที่เพิ่มขึ้นมาตั้งแต่ 1 ตารางขึ้นไปในคิวจัดการฐานข้อมูลเดียวกันหรือแตกต่างกันก็ได้
2. การส่งฐานข้อมูลทั้งหมดนั้นจะทำการส่งทีละตาราง
3. สามารถสร้างตารางที่การออกแบบตารางเหมือนกันแต่ไม่มีข้อมูลก็ได้
4. สามารถส่งข้อมูลจากค่าเบสเซิร์ฟเวอร์มาที่ ระบบจัดการฐานข้อมูลวอดคอม และเราสามารถทำงานกับ ข้อมูลนั้นออกมาเป็นรายงาน โดยไม่ต้องผ่านเครือข่าย
5. สามารถส่งข้อมูลที่มีการเปลี่ยนแปลงในฐานข้อมูลท้องถิ่นไปที่ส่วนกลางได้บ่อยครั้ง
6. สร้างตารางใหม่เมื่อมีการเปลี่ยนแปลงจะไม่อนุญาตให้เปลี่ยนใน ตาราง โคอะลิสก์ บอกร์ เราสามารถส่งข้อมูลได้มากกว่า 1 ตารางในฐานข้อมูลแหล่งกำเนิด (Source Database) ไปยังฐานข้อมูลปลายทาง (Destination database)

สิ่งที่ต้องกำหนด(define)ในการสร้างค่าไปป์ไลน์

1. ฐานข้อมูลแหล่งกำเนิด(Source Database)
2. ฐานข้อมูลปลายทาง(Destination Database)
3. ตารางของแหล่งกำเนิดที่ต้องถูกดึงข้อมูลออกมา
4. ไปป์ไลน์โอเปอเรชั่น(Pipeline Operation)
5. ตารางปลายทาง(Destination Table)

และเมื่อเราได้กำหนดการทำงานทุกอย่างเรียบร้อยแล้วต้องใส่รายละเอียดของข้อมูลนั้นๆด้วยจึงจะเรียกว่าเป็นการทำ ค่าไปป์ไลน์หรือ ไปป์ไลน์ นอกจากนี้ยังจะเก็บไปป์ไลน์นี้ในรูปออปเจกต์ที่สามารถนำมาใช้ได้อีกในการพัฒนาและ โปรแกรมประยุกต์ที่ทำขึ้นมา

ไปป์ไลน์กับชนิดของข้อมูล (Data types)

เมื่อมีการส่งข้อมูลจากฐานข้อมูลหนึ่งไปยังฐานข้อมูลนั้นเพาเวอร์วิวเคอร์รี่ จะทำการแปลงชนิดของข้อมูลจากฐานข้อมูลหนึ่ง ไปสู่ชนิดของข้อมูล ที่เหมาะสมกับฐานข้อมูลนั้นให้โดยอัตโนมัติ ซึ่งการสนับสนุนไม่ว่าชนิดของข้อมูลของจะสนับสนุนทุกชนิด ที่ใช้เป็นส่วนมากยกเว้นประเภท blob ของเอกสารทุกครั้งที่มีการนำไปใช้