

# สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

## ระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลแบบรีเลชันแนล AN AUTOMATIC RELATIONAL DATABASE DESIGN ENVIRONMENT



นายทวีชัย สุทธิทศธรรม  
MR.TWATCHAI SUTITOSATHAM

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2538

ISBN 974-621-332-6

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ไว้สำหรับใช้เพื่อการศึกษาค้นคว้าเท่านั้น มิอนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างถึงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# AN AUTOMATIC RELATIONAL DATABASE DESIGN ENVIRONMENT



A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE  
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

1995

ISBN 974-621-332-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	ระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลแบบรีเลชันแนล
นักศึกษา	นายธวัชชัย สุทธิพิศธรรม
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.ดร. ศุภมิตร จิตตะยโสธร
ระดับการศึกษา	วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.	2538

### บทคัดย่อ

การออกแบบฐานข้อมูลโดยทั่วไปจะทำได้โดยไม่ต้องอาศัยเครื่องมือช่วยอาศัยเพียงความรู้ความสามารถของผู้ออกแบบ แต่เมื่องานมีขนาดใหญ่การทำด้วยมือเริ่มมีความลำบากโดยเฉพาะอย่างยิ่งในช่วงของการวิเคราะห์ความสัมพันธ์ระหว่างข้อมูลซึ่งจำเป็นต้องมีการปรับปรุงแบบจำลองฐานข้อมูลเพื่อให้ถูกต้องยิ่งขึ้นเสมอจึงควรมีเครื่องมือช่วยในการออกแบบเพื่อช่วยให้สะดวกในการทำงานยิ่งขึ้น

วิทยานิพนธ์นี้นำเสนอการวิจัยและพัฒนาเครื่องมือช่วยในการออกแบบฐานข้อมูลแบบรีเลชันแนล ชื่อ ER/5 ซึ่งใช้แบบจำลองอีอาร์ (Entity Relationship Model) [2] เป็นแบบจำลองฐานข้อมูล โดยปกติแล้วฐานข้อมูลที่ได้จากแบบจำลองอีอาร์ จะเป็นตารางฐานข้อมูลในรูปบรรทัดฐานขั้นแรก (First Normal Form) ระบบ ER/5 จะใช้แบบจำลองอีอาร์ร่วมกับวิธีการทำให้เป็นบรรทัดฐานขั้นสูง (Further Normalization) ทำให้ได้ตารางฐานข้อมูลในรูปบรรทัดฐานขั้นที่ห้า (Fifth Normal Form) ซึ่งเป็นตารางฐานข้อมูลที่ปราศจากความซ้ำซ้อนโดยสิ้นเชิง

ผลลัพธ์ที่ได้จากเครื่องมือ ER/5 นี้จะเป็นคำสั่งสร้างโครงสร้างฐานข้อมูลในภาษาสอบถามเชิงโครงสร้าง (Data Definition Language of SQL) [4] ของระบบจัดการฐานข้อมูลออรากเคิล (Oracle Database Management System) [5] และผลลัพธ์เดียวกันนี้ยังสามารถนำไปใช้ได้กับระบบจัดการฐานข้อมูลอื่นๆได้เช่นเดียวกัน

<b>Thesis Title</b>	An Automatic Relational Database Design Environment
<b>Student</b>	Mr. Twatchai Suttitosatham
<b>Thesis Advisor</b>	Asst.Prof.Dr. Suphamit Chittayasothorn
<b>Level of Study</b>	Master of Engineering in Electrical Engineering
<b>Department</b>	Computer Engineering Faculty of Engineering King Mongkut's Institute of Technology Ladkrabang
<b>Year</b>	1995

## Abstract

In general, the database system design process can be done by human designers without any specialized tool, but in a large system, the design are more complicated. The data relationship analysis phase requires frequent data model reevaluation. Therefore designers need to have specialized tools.

This paper presents the research and development of a relational database design tool named ER/5 which uses the Entity Relationship Model. The model normally generates first normal form relational schema. However, ER/5 with further normalization generates fifth normal form relational schema. This method guarantees data to be redundant free.

The results from the ER/5 design tool are the SQL data definition language statements on the ORACLE Relational Database Management System (RDBMS). This result can be used with other database management systems as well.

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้จะไม่สามารถทำให้เสร็จลงได้ถ้าไม่ได้รับการอนุเคราะห์ช่วยเหลือกำลังทรัพย์ กำลังสติปัญญา กำลังงาน และกำลังใจจากท่านผู้มีพระคุณดังต่อไปนี้

1. นายสุทธิ สุทธิทศธรรม (บิดา)
2. นางกัญญารัตน์ สุทธิทศธรรม (มารดา)
3. ผศ.ดร.ศุภมิตร จิตตะยโสธร (อาจารย์ที่ปรึกษา)
4. นายสุเทพ ตันนิรันดร์
5. นายสมชิต ศิริผลหลาย
6. นายธนา หงษ์สุวรรณ
7. นายณภพล ใจดี

และเจ้าหน้าที่ฝ่ายระบบสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง รุ่นปี 36 ทุกท่าน

ข้าพเจ้าผู้ทำวิจัยขอกราบขอบพระคุณเป็นอย่างสูง

**“ความดีของวิทยานิพนธ์ฉบับนี้ขอมอบให้แต่บิดามารดาและครูอาจารย์ซึ่งเป็นปูชนียบุคคลที่ได้ให้ปัจจุบันและอนาคตที่ดีแก่กระผม”**

นายธวัชชัย สุทธิทศธรรม

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความรู้ทั่วไป	1
1.2 ส่วนประกอบของวิทยานิพนธ์	2
บทที่ 2 การออกแบบฐานข้อมูลในรูปบรรทัดฐาน (Normal Form)	3
2.1 โครงสร้างของตารางฐานข้อมูลแบบรีเลชันแนล	3
2.2 ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานชั้นแรก	4
2.3 ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานชั้นที่สอง	5
2.4 ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานชั้นที่สาม	7
2.5 ฐานข้อมูลที่อยู่ในรูปบรรทัดฐาน Boyce/Codd	8
2.6 ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานชั้นที่สี่	9
2.7 ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานชั้นที่ห้า	10
บทที่ 3 การออกแบบและสร้างตารางฐานข้อมูลด้วยแบบจำลองอีอาร์	13
3.1 สัญลักษณ์ที่ใช้ในแบบจำลองอีอาร์	13
3.1.1 เอ็นตีตี้	13
3.1.2 ความเกี่ยวพัน	14
3.1.3 แอททริบิวต์	15
3.1.4 การกำหนดตัวควบคุม (Constraint) ลงบนแอททริบิวต์	18
3.1.5 ไรล	19
3.1.6 อัตราส่วนคาคินอลิตี้	20
3.1.7 สรุปสัญลักษณ์และตัวอย่างแบบจำลองอีอาร์ ที่สร้างด้วยระบบ ER/5	21
3.2 หลักการสร้างตารางฐานข้อมูลจากแบบจำลองอีอาร์	22

บทที่ 4 สถาปัตยกรรมของระบบ ER/5	26
---------------------------------	----

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1	สถาปัตยกรรมของระบบ ER/5 โดยรวม	26
4.2	ฐานข้อมูลวินโดวเมทต้าไฟล์ (Windows Meta File)	27
4.2.1	วินโดวเมทต้าไฟล์เฮดเดอร์	28
4.2.2	วินโดวเมทต้าไฟล์ข้อมูล	28
4.3	โครงสร้างข้อมูลวัตุดับเบิลลิงคิสต์ (Object Double Linklist)	34
4.4	ฐานข้อมูลอีอาร์เมทต้า (ER Meta Database)	36
4.5	ฐานข้อมูลเอฟดีเมทต้า (FD Meta Database)	39
4.6	ฐานข้อมูลเจดีเมทต้า (JD Meta Database)	41
4.7	ฐานข้อมูลตารางและอินเด็กซ์เมทต้า (Table and Index Meta Database)	41
4.8	หน่วยติดต่อผู้ใช้ด้านกราฟฟิก (Graphic User Interface Unit)	44
4.9	ขบวนการสร้างตารางฐานข้อมูลในรูปบรรทัดฐานขั้นแรกจากแบบจำลองอีอาร์ (ER to First NF. Mapping)	44
4.10	หน่วยติดต่อผู้ใช้ด้านความสัมพันธ์ระหว่างแอฟทริบิวต์ (Functional Dependency Interface Unit)	44
4.11	ขบวนการสร้างตารางฐานข้อมูลในรูปบรรทัดฐานขั้น Boyce/Codd	44
4.12	หน่วยติดต่อผู้ใช้ด้านความสัมพันธ์เจดี (Join Dependency Interface Unit)	45
4.13	ขบวนการสร้างตารางฐานข้อมูลในรูปบรรทัดฐานขั้นที่ห้า	45
4.14	หน่วยสร้างภาษาสอบถามเชิงโครงสร้าง (Structure Query Language Generator)	45
<b>บทที่ 5 การพัฒนาระบบ ER/5</b>		47
5.1	อุปกรณ์ที่ใช้ในการพัฒนาระบบ ER/5	47
5.2	การเขียนโปรแกรมระบบ ER/5 บนไมโครซอฟท์วินโดว	48
5.2.1	ความแตกต่างของการเขียนโปรแกรมบนดอสกับไมโครซอฟท์วินโดว	48
5.2.2	รูปแบบของแอฟพลิเคชันบนไมโครซอฟท์วินโดว	50
5.2.3	แอฟพลิเคชัน ER/5 บนไมโครซอฟท์วินโดว	55
5.3	หน่วยโปรแกรมด้านกราฟฟิกของระบบ ER/5	58
5.4	หน่วยโปรแกรมประมวลผลข้อมูลสร้างตารางในบรรทัดฐานขั้นแรก	68
5.4.1	ส่วนตรวจสอบความถูกต้องของแบบจำลองอีอาร์	68
5.4.2	ส่วนสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นแรก	70
5.5	หน่วยโปรแกรมประมวลผลข้อมูลสร้างตารางในบรรทัดฐานขั้น	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญูให้เ้าไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Boyce/Codd	72
5.5.1 ส่วนติดต่อเอฟดี (Functional Dependency)	72
5.5.2 ส่วนตรวจสอบความถูกต้องของความสัมพันธ์ระหว่าง แอททริบิวต์	73
5.5.3 ส่วนสร้างตารางฐานข้อมูลในบรรทัดฐานชั้น Boyce/Codd	75
5.6 หน่วยโปรแกรมประมวลผลข้อมูลสร้างตารางในบรรทัดฐานชั้นที่ห้า	76
5.6.1 ส่วนติดต่อเจดี (Join Dependency)	76
5.6.2 ส่วนตรวจสอบความถูกต้องของความสัมพันธ์เจดี	77
5.6.3 ส่วนสร้างตารางฐานข้อมูลในบรรทัดฐานชั้นที่ห้า	79
5.7 หน่วยสร้างภาษาสอบถามเชิงโครงสร้างของระบบ ER/5	80
5.7.1 ส่วนสร้างตาราง	80
5.7.2 ส่วนสร้างอินเด็กซ์	81
5.7.3 ส่วนสร้างวิว	82
<b>บทที่ 6 การทดสอบและประเมินผลระบบ ER/5</b>	<b>83</b>
6.1 คุณสมบัติทั่วไปของระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูล	83
6.2 ระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลของตัวจัดการฐานข้อมูล ORACLE	84
6.3 การเปรียบเทียบระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูล ของตัวจัดการฐานข้อมูล ORACLE กับระบบ ER/5	91
6.4 ผลการทดลองใช้ระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูล ER/5	93
6.4.1 ผลการทดลองระดับฐานข้อมูลในบรรทัดฐานชั้นแรก	93
6.4.2 ผลการทดลองระดับฐานข้อมูลในบรรทัดฐานชั้น Boyce/Codd	98
6.4.3 ผลการทดลองระดับฐานข้อมูลในบรรทัดฐานชั้นที่ห้า	102
<b>บทที่ 7 สรุปและแนวทางการวิจัย</b>	<b>105</b>
7.1 สรุป	105
7.2 แนวทางการวิจัย	108
<b>เอกสารอ้างอิง</b>	<b>109</b>
<b>ภาคผนวก ก การติดตั้ง การเรียกโปรแกรม และการใช้งานระบบ ER/5</b>	<b>110</b>
<b>ภาคผนวก ข โปรแกรมระบบ ER/5</b>	<b>161</b>
<b>ประวัติผู้เขียน</b>	<b>268</b>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

	หน้า
ตารางที่ 2.1 ตารางฐานข้อมูลแบบรีเลชันแนล	3
ตารางที่ 2.2 ตารางฐานข้อมูลที่ไม่อยู่ในรูปบรรทัดฐาน	5
ตารางที่ 2.3 ตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นแรก	5
ตารางที่ 2.4 ตารางฐานข้อมูลที่ไม่อยู่ในรูปบรรทัดฐานขั้นที่สอง	6
ตารางที่ 2.5 ตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สอง	6
ตารางที่ 2.6 ตารางฐานข้อมูลที่ไม่อยู่ในรูปบรรทัดฐานขั้นที่สาม	7
ตารางที่ 2.7 ตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สาม	7
ตารางที่ 2.8 ตารางฐานข้อมูลที่ไม่อยู่ในรูปบรรทัดฐานขั้น Boyce/Codd	8
ตารางที่ 2.9 ตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้น Boyce/Codd	9
ตารางที่ 2.10 ตารางฐานข้อมูลที่ไม่อยู่ในรูปบรรทัดฐานขั้นที่สี่	9
ตารางที่ 2.11 ตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สี่	10
ตารางที่ 2.12 ตารางฐานข้อมูลที่ไม่อยู่ในรูปบรรทัดฐานขั้นที่ห้า	10
ตารางที่ 2.13 ตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่ห้า	11

# สารบัญภาพ

	หน้า
รูปที่ 3.1 แสดงการกำหนดตัวควบคุมให้กับแอฟทรีวิวิตซ์ชนิดตัวเลข	18
รูปที่ 3.2 แสดงการกำหนดตัวควบคุมให้กับแอฟทรีวิวิตซ์ชนิดตัวอักษรแบบในเซต	19
รูปที่ 3.3 แสดงการกำหนดตัวควบคุมให้กับแอฟทรีวิวิตซ์ชนิดตัวอักษรแบบนอกเซต	19
รูปที่ 3.4 แสดงสัญลักษณ์ต่างๆที่ใช้บนแบบจำลองอีอาร์	21
รูปที่ 3.5 แสดงแบบจำลองอีอาร์ที่สร้างด้วยระบบ ER/5	22
รูปที่ 3.6 แสดงโครงสร้างตารางฐานข้อมูลที่ได้จากแบบจำลองอีอาร์ในรูปที่ 3.5	24
รูปที่ 3.7 แสดงความเกี่ยวพันชนิด 3-ary บนแบบจำลองอีอาร์	25
รูปที่ 3.8 แสดงโครงสร้างตารางฐานข้อมูลที่ได้จากแบบจำลองอีอาร์ในรูปที่ 3.7	25
รูปที่ 4.1 แสดงสถาปัตยกรรมของระบบช่วยออกแบบฐานข้อมูล ER/5	26
รูปที่ 4.2 แสดงโครงสร้างของฐานข้อมูลวินโดว์เมทต้าไฟล์	27
รูปที่ 4.3 แสดงโครงสร้างข้อมูลวัตถุแบบเบิ้ลลิงคิลิสต์	34
รูปที่ 4.4 แสดงข้อมูลที่อยู่ภายในโครงสร้างข้อมูลวัตถุแบบเบิ้ลลิงคิลิสต์	35
รูปที่ 4.5 แสดงแบบจำลอง NIAM ที่ออกแบบสำหรับเก็บข้อมูลอีอาร์	36
รูปที่ 4.6 แสดงโครงสร้างตารางฐานข้อมูลที่ได้จากแบบจำลอง NIAM ในรูปที่ 4.5	37
รูปที่ 4.7 แสดงแบบจำลอง NIAM ที่ออกแบบสำหรับเก็บข้อมูลเอพีดี	39
รูปที่ 4.8 แสดงโครงสร้างตารางฐานข้อมูลที่ได้จากแบบจำลอง NIAM ในรูปที่ 4.7	39
รูปที่ 4.9 แสดงแบบจำลอง NIAM ที่ออกแบบสำหรับเก็บข้อมูลเจดี	41
รูปที่ 4.10 แสดงโครงสร้างตารางฐานข้อมูลที่ได้จากแบบจำลอง NIAM ในรูปที่ 4.9	41
รูปที่ 4.11 แสดงแบบจำลอง NIAM ที่ออกแบบสำหรับเก็บข้อมูลตารางและอินเด็กซ์	42
รูปที่ 4.12 แสดงโครงสร้างตารางฐานข้อมูลที่ได้จากแบบจำลอง NIAM ในรูปที่ 4.11	42
รูปที่ 4.13 แสดงผลลัพธ์ที่ได้จากหน่วยสร้างภาษาสอบถามเชิงโครงสร้าง	46
รูปที่ 5.1 แสดงส่วนประกอบต่างๆของวินโดว์	51
รูปที่ 5.2 แสดงตัวอย่างกรอบข้อความบนวินโดว์	51
รูปที่ 5.3 แสดงการจัดการเมสเสจจากคีย์บอร์ด	53
รูปที่ 5.4 แสดงการจัดการเมสเสจสำหรับแอฟพลีเคชันสองตัว	54
รูปที่ 5.5 แสดงการจัดการเมสเสจของวินโดว์	54
รูปที่ 5.6 แสดงการจัดการเมสเสจของแอฟพลีเคชัน ER/5	57
รูปที่ 5.7 แสดงวินโดว์ทั้งหมดของแอฟพลีเคชัน ER/5	58
รูปที่ 5.8 แสดงผังการทำงานในการวาดเอ็นดีดี	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.9 แสดงผังการทำงานในการวาดความเกี่ยวพัน	61
รูปที่ 5.10 แสดงผังการทำงานในการวาดแอททริบิวต์	62
รูปที่ 5.11 แสดงผังการทำงานในการวาดการเชื่อมต่อแอททริบิวต์	63
รูปที่ 5.12 แสดงผังการทำงานในการวาดการเชื่อมต่อความเกี่ยวพัน	64
รูปที่ 5.13 แสดงผังการทำงานในการวาดข้อความ	65
รูปที่ 5.14 แสดงผังการทำงานในการลบวัตถุออกจากแบบจำลองอ็อบเจกต์ (1)	66
รูปที่ 5.15 แสดงผังการทำงานในการลบวัตถุออกจากแบบจำลองอ็อบเจกต์ (2)	67
รูปที่ 5.16 แสดงผังการทำงานในการตรวจสอบความถูกต้องของแอททริบิวต์	68
รูปที่ 5.17 แสดงผังการทำงานในการตรวจสอบความถูกต้องของความเกี่ยวพัน	69
รูปที่ 5.18 แสดงผังการทำงานในการตรวจสอบความถูกต้องของเอ็นติตี้	70
รูปที่ 5.19 แสดงผังการทำงานในการสร้างตารางฐานข้อมูลจากแบบจำลองอ็อบเจกต์	71
รูปที่ 5.20 แสดงผังการทำงานในส่วนติดต่อเฟรม	72
รูปที่ 5.21 แสดงผังการทำงานในการตรวจสอบความถูกต้องของเฟรม	74
รูปที่ 5.22 แสดงผังการทำงานในการสร้างตารางฐานข้อมูลชั้น Boyce/Codd จากเฟรม	75
รูปที่ 5.23 แสดงผังการทำงานในส่วนติดต่อเจดี	76
รูปที่ 5.24 แสดงผังการทำงานในการตรวจสอบความถูกต้องของเจดี	78
รูปที่ 5.25 แสดงผังการทำงานในการสร้างตารางฐานข้อมูลชั้นที่ห้าจากเจดี	79
รูปที่ 5.26 แสดงผังการทำงานในส่วนสร้างตาราง	80
รูปที่ 5.27 แสดงผังการทำงานในส่วนสร้างอินเด็กซ์	81
รูปที่ 5.28 แสดงผังการทำงานในส่วนสร้างวิว	82
รูปที่ 6.1 แสดงระบบช่วยออกแบบฐานข้อมูลของตัวจัดการฐานข้อมูลออรากเคิล	84
รูปที่ 6.2 แสดงแบบจำลองอ็อบเจกต์ที่สร้างขึ้นจากตัวจัดการฐานข้อมูลออรากเคิล	85
รูปที่ 6.3 แสดงเมนูสำหรับกำหนดแอททริบิวต์ให้กับเอ็นติตี้	86
รูปที่ 6.4 แสดงการเลือกกำหนดแอททริบิวต์ให้กับเอ็นติตี้ EMPLOYEE	86
รูปที่ 6.5 แสดงแอททริบิวต์บางส่วนของเอ็นติตี้ EMPLOYEE	87
รูปที่ 6.6 แสดงเมนูสำหรับกำหนดตัวควบคุมให้กับแอททริบิวต์	87
รูปที่ 6.7 แสดงเมนูสำหรับกำหนดตัวควบคุมชนิดตัวอักษรให้กับแอททริบิวต์ SEX	88
รูปที่ 6.8 แสดงเมนูสำหรับกำหนดตัวควบคุมชนิดตัวเลขให้กับแอททริบิวต์ SALARY	88
รูปที่ 6.9 แสดงเมนูสำหรับสร้างตารางฐานข้อมูล	89
รูปที่ 6.10 แสดงผลลัพธ์ของตารางฐานข้อมูลในรูปแบบ Table Definition	89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
รูปที่ 6.11 แสดงผลลัพธ์ของตารางฐานข้อมูลในรูปแบบ Data Definition Language	90
รูปที่ 6.12 แสดงแบบจำลองอีอาร์ที่ใช้เก็บข้อมูลของบริษัทแห่งหนึ่ง	94
รูปที่ 6.13 แสดงคำสั่งสร้างตารางฐานข้อมูลที่ได้จากแบบจำลองอีอาร์ในรูปที่ 6.12	95
รูปที่ 6.14 แสดงแบบจำลองอีอาร์ที่ใช้เก็บข้อมูลของสนามบินแห่งหนึ่ง	95
รูปที่ 6.15 แสดงคำสั่งสร้างตารางฐานข้อมูลที่ได้จากแบบจำลองอีอาร์ในรูปที่ 6.14	96
รูปที่ 6.16 แสดงแบบจำลองอีอาร์ที่ใช้เก็บข้อมูลของมหาวิทยาลัยแห่งหนึ่ง	97
รูปที่ 6.17 แสดงคำสั่งสร้างตารางฐานข้อมูลที่ได้จากแบบจำลองอีอาร์ในรูปที่ 6.16	98
รูปที่ 6.18 แสดงการกำหนดเอพีดีให้กับตาราง DEPENDENT	100
รูปที่ 6.19 แสดงการกำหนดเอพีดีให้กับตาราง WORKS_ON	100
รูปที่ 6.20 แสดงคำสั่งสร้างตารางฐานข้อมูลที่ได้ในรูปแบบรทฐานชั้น Boyce/Codd	101
รูปที่ 6.21 แสดงการกำหนดเจดีให้กับตาราง WORKS_ON	102
รูปที่ 6.22 แสดงคำสั่งสร้างตารางฐานข้อมูลที่ได้ในรูปแบบรทฐานชั้นที่ห้า	104
รูปที่ 7.1 แสดงตารางฐานข้อมูลที่มีความซ้ำซ้อน	105
รูปที่ 7.2 แสดงตารางฐานข้อมูลที่ไม่มีความซ้ำซ้อน	106
รูปที่ 7.3 แสดงตารางฐานข้อมูลที่ใส่ข้อมูลผิดพลาดจากผู้ใช้งาน	106
รูปที่ 7.4 แสดงการกำหนดตัวควบคุมให้กับแอททริบิวต์ชนิดตัวเลข	107
รูปที่ 7.5 แสดงการกำหนดตัวควบคุมให้กับแอททริบิวต์ชนิดตัวอักษรแบบในเซต	107
รูปที่ 7.6 แสดงการกำหนดตัวควบคุมให้กับแอททริบิวต์ชนิดตัวอักษรแบบนอกเซต	107

# บทที่ 1

## บทนำ

### 1.1 ความรู้ทั่วไป

ปัจจุบันนี้เทคโนโลยีด้านการใช้คอมพิวเตอร์ช่วยออกแบบระบบซอฟต์แวร์ (Computer-Aided Software Engineering (CASE)) [5] มีการพัฒนากันขึ้นมากทำให้ผู้พัฒนาซอฟต์แวร์เฉพาะด้านสามารถเลือกใช้ CASE ที่สนับสนุนงานด้านนั้นมาผลิตสินค้าได้สะดวก ดังนั้นการออกแบบสร้าง CASE นั้นจึงเป็นการสร้างขึ้นมาใช้เฉพาะงานประเภทเดียวแต่มีประสิทธิภาพสูง ยกตัวอย่างเช่น CASE ที่ใช้ในระบบฐานข้อมูล ก็จะมีหน้าที่ตั้งแต่รับการออกแบบฐานข้อมูลของผู้พัฒนาระบบ ตรวจสอบความถูกต้องของข้อมูลตลอดจนสร้างระบบงานสารสนเทศ เป็นต้น

ปัจจุบันนี้ CASE ที่ใช้ในระบบฐานข้อมูลนั้นมีด้วยกันหลายชนิดแต่ละชนิดก็จะใช้วิธีการที่แตกต่างกันไปซึ่งไม่ว่าจะใช้วิธีการใดก็ตามจุดประสงค์ก็เพื่อที่จะช่วยลดความซ้ำซ้อนของข้อมูลเพื่อให้ได้ฐานข้อมูลที่มีประสิทธิภาพสูงสุด

ระบบ ER/5 ที่นำเสนอในวิทยานิพนธ์นี้เป็นเพียงส่วนหนึ่งของระบบ CASE ที่ใช้ในระบบฐานข้อมูล ทำหน้าที่เป็นเครื่องมือช่วยออกแบบฐานข้อมูลโดยใช้แบบจำลองอ็อยาร์ (Entity Relationship Model) [2] สำหรับการออกแบบฐานข้อมูลด้วยแบบจำลองอ็อยาร์นั้นเป็นวิธีที่เข้าใจง่ายและนิยมใช้กันอย่างแพร่หลายแต่อย่างไรก็ตามการออกแบบฐานข้อมูลด้วยแบบจำลองอ็อยาร์เพียงอย่างเดียวยังไม่เพียงพอที่จะทำให้เกิดผลลัพธ์ของตารางฐานข้อมูลที่ได้เป็นตารางฐานข้อมูลในรูปบรรทัดฐานชั้นที่ห้าเนื่องจากจำกัดอยู่ที่ความสามารถของตัวแบบจำลองอ็อยาร์เองทำให้ผลลัพธ์ของตารางฐานข้อมูลยังมีความซ้ำซ้อนเกิดขึ้นคือ ยังอยู่ในรูปบรรทัดฐานชั้นแรก ดังนั้นในวิทยานิพนธ์นี้จึงได้นำเสนอการเพิ่มความสามารถในการทำบรรทัดฐานชั้นสูงเพิ่มเติมเพื่อให้ได้ตารางฐานข้อมูลในรูปบรรทัดฐานชั้นที่ห้าซึ่งเป็นตารางฐานข้อมูลที่ปราศจากความซ้ำซ้อนโดยสิ้นเชิงด้วยการกำหนดความสัมพันธ์ระหว่างแอททริบิวต์ (Functional Dependency) [1] และความสัมพันธ์เจต (Join Dependency) [4] เพิ่มเติมลงในตารางฐานข้อมูลที่ได้มาจากแบบจำลองอ็อยาร์ ดังนั้นเมื่อเราทราบค่าความสัมพันธ์ระหว่างแอททริบิวต์และความสัมพันธ์เจตแล้วทำให้สามารถสร้างตารางฐานข้อมูลในรูปบรรทัดฐานชั้นที่ห้าได้ และยังเพิ่มความสามารถพิเศษให้สามารถกำหนดตัวควบคุม (Constraint) ลงบนแบบจำลองอ็อยาร์ได้อีกด้วยทำให้ผลลัพธ์ของตารางฐานข้อมูลที่สร้างขึ้นด้วยระบบ ER/5 นี้สามารถตรวจสอบความถูกต้องตามตัวควบคุมที่กำหนดและป้องกันไม่ให้เกิดการผิดพลาดของข้อมูลขณะที่นำไปใช้งานได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2 ส่วนประกอบของวิทยานิพนธ์

วิทยานิพนธ์นี้ประกอบด้วยเนื้อหาจำนวน 7 บท และภาคผนวก 2 บท เริ่มจากบทที่ 1 จะกล่าวถึงที่มาของวิทยานิพนธ์และประโยชน์ที่จะได้รับเมื่อวิทยานิพนธ์นี้เสร็จสิ้นลงรวมทั้งส่วนประกอบของวิทยานิพนธ์ บทที่ 2 จะกล่าวถึงฐานข้อมูลที่ใช้ได้กับระบบจัดการฐานข้อมูลแบบรีเลชันแนลและฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นต่างๆ (Further Normalization) [4] ทำให้ทราบถึงผลเสียที่เกิดขึ้นของฐานข้อมูลที่ไม่เข้าเงื่อนไขและแนวทางการแก้ไขว่าทำอย่างไรจึงจะได้ตารางฐานข้อมูลที่ไม่ซ้ำซ้อน บทที่ 3 จะกล่าวถึงการออกแบบและสร้างตารางฐานข้อมูลด้วยแบบจำลองอ็อบเจกต์ซึ่งจะได้รู้จักกับสัญลักษณ์ต่างๆของแบบจำลองอ็อบเจกต์ที่ใช้ในวิทยานิพนธ์นี้และความหมายของสัญลักษณ์นั้นๆ พร้อมทั้งตัวอย่างของแบบจำลองอ็อบเจกต์ที่ใช้งานได้จริงทำให้สามารถที่จะออกแบบและสร้างตารางฐานข้อมูลด้วยแบบจำลองอ็อบเจกต์ได้ด้วยตนเองซึ่งความรู้ที่ได้จากบทที่ 2 และ 3 นี้ มีความจำเป็นอย่างยิ่งสำหรับผู้ที่ต้องการใช้งานระบบ ER/5 บทที่ 4 จะกล่าวถึงสถาปัตยกรรมโดยรวมของระบบ ER/5 ซึ่งประกอบด้วยตัวโครงสร้างฐานข้อมูลและหน่วยประมวลผลต่างๆที่ใช้ในระบบ ER/5 ทำให้ทราบถึงลักษณะและวิธีการเก็บข้อมูลของระบบ ER/5 และหลักการทำงานที่สำคัญทั้งหมดของระบบ ER/5 บทที่ 5 จะกล่าวถึงการพัฒนาระบบ ER/5 ซึ่งจะอธิบายถึงอุปกรณ์และวิธีการที่ใช้ในการพัฒนาระบบ ER/5 รวมทั้งหลักการทำงานของโปรแกรมในโมดูลต่างๆ บทที่ 6 จะกล่าวถึงการทดสอบและประเมินผลระบบ ER/5 ว่ามีประสิทธิภาพและข้อจำกัดอย่างไรโดยเปรียบเทียบกับระบบที่มีอยู่เดิมตามท้องตลาด บทที่ 7 ซึ่งเป็นบทสุดท้ายจะกล่าวสรุปและเสนอแนะแนวทางในการวิจัยเพิ่มเติมในอนาคต ส่วนในภาคผนวก ก จะเป็นการแนะนำการติดตั้ง, การเรียกโปรแกรมและการใช้งานระบบ ER/5 ทำให้สามารถนำระบบ ER/5 ไปติดตั้งและใช้งานได้ถูกต้องและมีประสิทธิภาพ ภาคผนวก ข จะเป็นตัวโปรแกรมของระบบ ER/5 ที่เขียนขึ้นทั้งหมดเพื่อผู้ที่สนใจในวิทยานิพนธ์นี้สามารถที่จะนำไปวิจัยและพัฒนาต่อได้โดยสะดวก

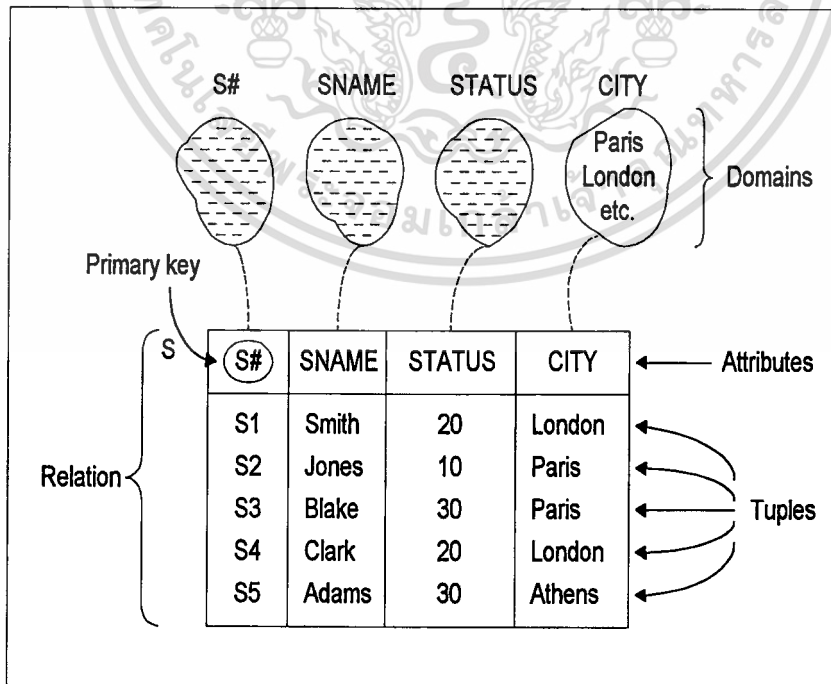
## บทที่ 2

### การออกแบบฐานข้อมูลในรูปบรรทัดฐาน

การออกแบบฐานข้อมูลในรูปบรรทัดฐาน [4] เป็นการออกแบบตารางฐานข้อมูลเพื่อใช้งานกับระบบจัดการฐานข้อมูลแบบรีเลชันแนล (ฐานข้อมูลที่ไม่อยู่ในรูปบรรทัดฐานนั้นไม่สามารถที่จะนำไปใช้ได้กับระบบจัดการฐานข้อมูลแบบรีเลชันแนล) โดยมีจุดประสงค์ในการออกแบบก็เพื่อที่จะให้ได้ตารางฐานข้อมูลที่ไม่มีความซ้ำซ้อนเกิดขึ้น ซึ่งฐานข้อมูลใดก็ตามที่ยังมีความซ้ำซ้อนอยู่อาจเป็นสาเหตุทำให้เกิดการประมวลผลข้อมูลที่ไม่ตรงกับความเป็นจริง (database inconsistency and modification anomalies) [3] ในบทนี้จะกล่าวถึงข้อเสียของตารางฐานข้อมูลที่ซ้ำซ้อนและแนวทางแก้ไขโดยใช้วิธีการทำให้ตารางฐานข้อมูลอยู่ในรูปบรรทัดฐานขั้นสูงตั้งแต่ขั้นแรกจนถึงขั้นที่ห้า (Further Normalization) [4] ซึ่งถือว่าเป็นขั้นสูงสุดที่สามารถกำจัดความซ้ำซ้อนของข้อมูลได้อย่างสิ้นเชิง

#### 2.1 โครงสร้างของตารางฐานข้อมูลแบบรีเลชันแนล

ในหัวข้อนี้จะกล่าวถึงศัพท์เฉพาะต่างๆที่ใช้เรียกส่วนประกอบของตารางฐานข้อมูลแบบรีเลชันแนลที่ใช้ในวิทยานิพนธ์นี้ จากรูปที่ 2.1



รูปที่ 2.1 แสดงโครงสร้างของฐานข้อมูลแบบรีเลชันแนล [4]

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **รีเลชัน (Relation)** หมายถึง ตารางฐานข้อมูล จากรูปที่ 2.1 ตารางฐานข้อมูลนี้มีชื่อเรียกว่า S ซึ่งตารางฐานข้อมูลที่สามารถเรียกได้ว่าเป็นรีเลชันนั้นจะต้องเป็นตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานเท่านั้น

- **แอททริบิวต์ (Attributes)** หมายถึง คุณลักษณะต่างๆที่มีความสัมพันธ์กันแล้วประกอบขึ้นเป็นตารางฐานข้อมูลเช่น แอททริบิวต์ SNAME หมายถึง ชื่อพนักงาน แอททริบิวต์ CITY หมายถึง ชื่อเมือง มีความสัมพันธ์กันคือ พนักงานชื่อ (SNAME) ..... อาศัยอยู่ที่เมือง (CITY) ..... เป็นต้น (ลักษณะของความสัมพันธ์ระหว่างแอททริบิวต์จะไม่สามารถมองเห็นได้จากตารางฐานข้อมูลแต่จะมองเห็นได้จากแบบจำลองฐานข้อมูลเช่น Fd Diagram, แผนภาพ NIAM หรือ แบบจำลองอีอาร์ เป็นต้น)

- **โดเมน (Domains)** หมายถึง ขอบเขตของข้อมูลในแต่ละแอททริบิวต์เช่น ข้อมูลที่เก็บลงบนแอททริบิวต์ CITY จะต้องเป็นชื่อของเมืองต่างๆในทวีปยุโรปเท่านั้น หรือ ข้อมูลที่เก็บลงบนแอททริบิวต์ SNAME จะต้องเป็นชื่อคนเท่านั้นจะเป็นอย่างอื่นไม่ได้ เป็นต้น

- **ทับเปิล (Tuples)** — หมายถึง แถวของตารางฐานข้อมูล

- **กุญแจหลัก (Primary key)** หมายถึง กลุ่มของแอททริบิวต์ที่ทำหน้าที่เป็นตัวแยกความแตกต่างของแต่ละทับเปิลในตารางฐานข้อมูล ดังนั้นกลุ่มของแอททริบิวต์ที่เป็นกุญแจหลักของตารางฐานข้อมูลจะต้องมีข้อมูลที่ประกอบเข้าด้วยกันแล้วไม่ซ้ำกัน (unique)

## 2.2 ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นแรก

ตารางฐานข้อมูลที่สามารถใช้ได้กับระบบจัดการฐานข้อมูลแบบรีเลชันแนลนั้นจะต้องมีคุณสมบัติอย่างน้อยที่สุดเป็นตารางฐานข้อมูลที่อยู่ในรูปของบรรทัดฐานขั้นแรกซึ่งต้องมีคุณสมบัติดังต่อไปนี้

- ไม่มีทับเปิลที่ซ้ำกัน คุณสมบัตินี้นิยามจากความจริงที่ว่าข้อมูลที่อยู่ในรีเลชันก็คือเซตของทับเปิลซึ่งในทางคณิตศาสตร์แล้วสมาชิกในเซตต้องไม่ซ้ำกัน เหตุผลที่ตามมาก็คือว่าทุกรีเลชันจะต้องมีกุญแจหลักเพื่อแยกความแตกต่างระหว่างทับเปิล

- ลำดับของทับเปิลไม่มีความสำคัญ คุณสมบัตินี้นิยามจากความจริงที่ว่าข้อมูลที่อยู่ในรีเลชันก็คือเซตของทับเปิลซึ่งในทางคณิตศาสตร์แล้วลำดับของสมาชิกในเซตไม่มีความสำคัญ

- ลำดับของแอททริบิวต์ไม่มีความสำคัญ คุณสมบัตินี้นิยามจากความจริงที่ว่าตารางฐานข้อมูลประกอบขึ้นจากเซตของแอททริบิวต์ซึ่งในทางคณิตศาสตร์แล้วลำดับของสมาชิกในเซตไม่มีความสำคัญ

- เมื่อกำหนดแถวและหลักของตารางจะต้องได้ค่าของข้อมูลเพียงค่าเดียวเท่านั้น

S#	P#	QTY
S1	P1	300
S1	P4	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

(a) Duplicate tuples

S#	PQ	
	P#	QTY
S1	P1	300
	P2	200
	P3	400
	P4	200
	P5	100
	P6	100
S2	P1	300
	P2	400
S3	P2	200
S4	P2	200
	P4	300
	P5	400

(b) Repeating groups

รูปที่ 2.2 แสดงตารางฐานข้อมูลที่ไม่อยู่ในรูปบรรทัดฐานหรือเรียกอีกอย่างหนึ่งว่าไม่เป็นรีเลชั่น [4]

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

รูปที่ 2.3 แสดงตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นแรก [4]

### 2.3 ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สอง

ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สองต้องมีคุณสมบัติดังต่อไปนี้

- ต้องเป็นฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นแรก
- ทุกแอททริบิวต์ที่ไม่ใช่กุญแจหลักของตารางต้องขึ้นอยู่กับกุญแจหลักเท่านั้นจะขึ้นอยู่กับสับเซตของกุญแจหลักไม่ได้ หรือกล่าวอีกนัยหนึ่งว่าทุกแอททริบิวต์ที่ไม่ใช่กุญแจหลักของตารางต้อง Fully Depen on กุญแจหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## PROJECT-DATA

PERSON_ID	PROJ#	PROJ_BUDGET	TOTAL_TIME_SPENT_BY_PERSON_ON_PROJECT
P1	PROJ1	20	20
P3	PROJ1	20	16
P2	PROJ2	17	35
P2	PROJ3	84	42
P3	PROJ2	17	17
P2	PROJ1	20	83
P4	PROJ3	84	41
-	PROJ4	90	-

รูปที่ 2.4 แสดงตารางฐานข้อมูลที่ไม่อยู่ในรูปบรรทัดฐานขั้นที่สอง [1]

จากตารางฐานข้อมูลในรูปที่ 2.4 เป็นตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นแรกที่มีกุญแจหลักของตารางคือแอททริบิวต์ PERSON\_ID รวมกับ แอททริบิวต์ PROJ# ตารางฐานข้อมูลนี้ยังไม่เป็นตารางฐานข้อมูลในรูปบรรทัดฐานขั้นที่สองเนื่องจากแอททริบิวต์ PROJECT\_BUDGET สามารถขึ้นอยู่กับแอททริบิวต์ที่เป็นสับเซตของกุญแจหลักคือ PROJ# เท่านั้น ทำให้เกิดความซ้ำซ้อนของตารางฐานข้อมูลขึ้นได้แก่ เมื่อไหร่ก็ตามที่เรากำหนด PROJ# จะได้ผลลัพธ์ของ PROJECT\_BUDGET ซึ่งเป็นค่าเดียวกันจำนวนหลายแถว ข้อเสียคือถ้าต้องการแก้ไขข้อมูลของ PROJ# จะต้องแก้ไขข้อมูลของ PROJECT\_BUDGET ด้วยค่าที่สอดคล้องกันจะแก้ไขเพียงแอททริบิวต์ใดแอททริบิวต์หนึ่งไม่ได้ จากเหตุผลดังกล่าวทำให้ข้อมูลที่ถูกเก็บลงในตารางฐานข้อมูลนี้อาจจะไม่ตรงกับความจริงได้ถ้าเมื่อไรก็ตามที่มีการแก้ไข PROJ# หรือ PROJECT\_BUDGET เพียงแอททริบิวต์เดียว วิธีการกำจัดความซ้ำซ้อนของตารางฐานข้อมูลนี้เราต้องแยกตารางฐานข้อมูลนี้ออกเป็นสองตารางดังแสดงในรูปที่ 2.5 ซึ่งตารางฐานข้อมูลทั้งสองที่เกิดขึ้นใหม่นี้จัดอยู่ในตารางฐานข้อมูลในรูปบรรทัดฐานขั้นที่สอง

PROJECTS		WORK		
PROJ#	PROJ_BUDGET	PERSON_ID	PROJ#	TOTAL_TIME_SPENT_BY_PERSON_ON_PROJECT
PROJ1	20	P1	PROJ1	20
PROJ2	17	P3	PROJ1	16
PROJ3	84	P2	PROJ2	35
PROJ4	90	P2	PROJ3	42
		P3	PROJ2	17
		P3	PROJ1	83
		P4	PROJ3	41

รูปที่ 2.5 แสดงตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สอง

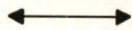
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเฉพาะเท่านั้น เมื่ออนุญาตให้เข้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สาม

ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สามต้องมีคุณสมบัติดังต่อไปนี้

- ต้องเป็นฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สอง
- ทุกแอททริบิวต์ที่ไม่ใช่กุญแจหลักของตารางต้องไม่ขึ้นอยู่กับแอททริบิวต์ที่ไม่ใช่กุญแจหลักของตารางด้วยตัวเอง

### PROJECTS



PROJ#	MANAGER	DATE_OF_BIRTH
P1	Joe	Jan 63
P3	Vicki	March 57
P2	Joe	Jan 63
P4	Marilyn	July 53

รูปที่ 2.6 แสดงตารางฐานข้อมูลที่ไม่อยู่ในรูปบรรทัดฐานขั้นที่สาม [1]

จากตารางฐานข้อมูลในรูปที่ 2.6 เป็นตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สอง เนื่องจากกุญแจหลักของตารางประกอบด้วยแอททริบิวต์เพียงแอททริบิวต์เดียวคือ PROJ# ตารางฐานข้อมูลนี้ยังไม่เป็นตารางฐานข้อมูลในรูปบรรทัดฐานขั้นที่สามเนื่องจากแอททริบิวต์ที่ไม่ได้เป็นกุญแจหลักของตารางคือ DATE\_OF\_BIRTH สามารถขึ้นอยู่กับแอททริบิวต์ MANAGER ซึ่งเป็นแอททริบิวต์ที่ไม่เป็นกุญแจหลักของตารางเช่นเดียวกัน ทำให้เกิดความซ้ำซ้อนของตารางฐานข้อมูลขึ้นได้แก่ เมื่อไหร่ก็ตามที่เรากำหนด MANAGER จะได้ผลลัพธ์ของ DATE\_OF\_BIRTH ซึ่งเป็นค่าเดียวกันจำนวนหลายแถว ข้อเสียคือถ้าต้องการแก้ไขข้อมูลของ MANAGER จะต้องแก้ไขข้อมูลของ DATE\_OF\_BIRTH ด้วยค่าที่สอดคล้องกันจะแก้ไขเพียงแอททริบิวต์ใดแอททริบิวต์หนึ่งไม่ได้ จากเหตุผลดังกล่าวทำให้ข้อมูลที่ถูกเก็บลงในตารางฐานข้อมูลนี้อาจจะไม่ตรงกับความจริงได้ถ้าเมื่อไหร่ก็ตามที่มีการแก้ไข MANAGER หรือ DATE\_OF\_BIRTH เพียงแอททริบิวต์เดียว วิธีการกำจัดความซ้ำซ้อนของตารางฐานข้อมูลนี้เราต้องแยกตารางฐานข้อมูลนี้ออกเป็นสองตารางดังแสดงในรูปที่ 2.7 ซึ่งตารางฐานข้อมูลทั้งสองที่เกิดขึ้นใหม่นี้จัดอยู่ในตารางฐานข้อมูลในรูปบรรทัดฐานขั้นที่สาม

PROJECTS		MANAGERS	
PROJ#	MANAGER	MANAGER	DATE_OF_BIRTH
P1	Joe	Joe	Jan 63
P2	Joe	Vicki	March 57
P3	Vicki	Marilyn	July 53
P4	Marilyn		

รูปที่ 2.7 แสดงตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สาม

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการเรียนการสอนเท่านั้น เมื่อผู้ญาติหน้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

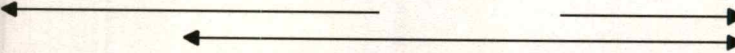
## 2.5 ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานชั้น Boyce/Codd

ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานชั้น Boyce/Codd ต้องมีคุณสมบัติดังต่อไปนี้

- ต้องเป็นฐานข้อมูลที่อยู่ในรูปบรรทัดฐานชั้นที่สาม
- ทุกแอททริบิวต์ที่เป็นกุญแจหลักของตารางจะต้องไม่ขึ้นอยู่กับสับเซตของกุญแจ

หลักอันอื่นหรือกล่าวอีกนัยหนึ่งว่าทุก determinant ต้องเป็น candidate key

### ATTENDANCE



TEACHER	SEMESTER	SUBJECT	SECTION	ATTENDANCES
Joe	1/88	COBOL	SECTION1	35
Jenny	1/88	MATHS	SECTION1	40
Gregory	2/88	UNIX	SECTION2	33
Jenny	1/88	MATHS	SECTION2	42
Gregory	2/88	UNIX	SECTION1	47
Joe	1/88	COBOL	SECTION2	50
Joe	1/88	COBOL	SECTION3	12

รูปที่ 2.8 แสดงตารางฐานข้อมูลที่ไม่อยู่ในรูปบรรทัดฐานชั้น Boyce/Codd [1]

จากตารางฐานข้อมูลในรูปที่ 2.8 เป็นตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานชั้นที่สามเนื่องจากแอททริบิวต์ที่ไม่เป็นกุญแจหลักของตารางมีเพียงแอททริบิวต์เดียวคือ ATTENDANCES (ไม่มีแอททริบิวต์ที่ไม่ใช่กุญแจหลักของตารางตัวอื่นมาขึ้นกับแอททริบิวต์ ATTENDANCES) โดยมีกุญแจหลักของตารางจำนวนสองตัวได้แก่ กลุ่มแอททริบิวต์ (TEACHER, SEMESTER, SECTION) กับ (SUBJECT, SEMESTER, SECTION) ตารางฐานข้อมูลนี้ยังไม่เป็นตารางฐานข้อมูลในรูปบรรทัดฐานชั้น Boyce/Codd เนื่องจากแอททริบิวต์ที่เป็นส่วนหนึ่งในกุญแจหลักของตารางตัวที่หนึ่งคือ TEACHER ขึ้นอยู่กับสับเซตของกุญแจหลักตัวที่สองคือ แอททริบิวต์ SEMESTER รวมกับ SUBJECT ทำให้เกิดความซ้ำซ้อนของตารางฐานข้อมูลขึ้นได้แก่ เมื่อไหร่ก็ตามที่เรากำหนด SEMESTER กับ SUBJECT จะได้ผลลัพธ์ของ TEACHER ซึ่งเป็นค่าเดียวกันจำนวนหลายแถว ข้อเสียคือถ้าต้องการแก้ไขข้อมูลของ SEMESTER กับ SUBJECT จะต้องแก้ไขข้อมูลของ TEACHER ด้วยค่าที่สอดคล้องกันจะแก้ไขเพียงแอททริบิวต์ใดแอททริบิวต์หนึ่งไม่ได้ จากเหตุผลดังกล่าวทำให้ข้อมูลที่ถูกเก็บลงในตารางฐานข้อมูลนี้อาจจะไม่ตรงกับความจริงได้ถ้าเมื่อไหร่ก็ตามที่มีการแก้ไข SEMESTER กับ SUBJECT หรือ TEACHER เพียงแอททริบิวต์เดียว วิธีการกำจัดความซ้ำซ้อนของตารางฐานข้อมูลนี้เราต้องแยกตารางฐานข้อมูลนี้ออกเป็นสองตารางดังแสดงในรูปที่ 2.9 ซึ่งตารางฐานข้อมูลทั้งสองที่เกิดขึ้นใหม่นี้จัดอยู่ในตารางฐานข้อมูลในรูปบรรทัดฐานชั้น Boyce Codd

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**TEACHERS**

TEACHER	SEMESTER	SUBJECT
Joe	1/88	COBOL
Jenny	1/88	MATHS
Gregory	2/88	UNIX

**ATTENDANCE**

SEMESTER	SUBJECT	SECTION	ATTENDANCES
1/88	COBOL	SECTION1	35
1/88	MATHS	SECTION1	40
2/88	UNIX	SECTION2	33
1/88	MATHS	SECTION2	42
2/88	UNIX	SECTION1	47
1/88	COBOL	SECTION2	50
1/88	COBOL	SECTION3	12

รูปที่ 2.9 แสดงตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้น Boyce/Codd

**2.6 ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สี่**

ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สี่ต้องมีคุณสมบัติดังต่อไปนี้

- ต้องเป็นฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้น Boyce/Codd
- ต้องไม่มี multivalued dependency (MVD) [4] มากเกินหนึ่งตัว

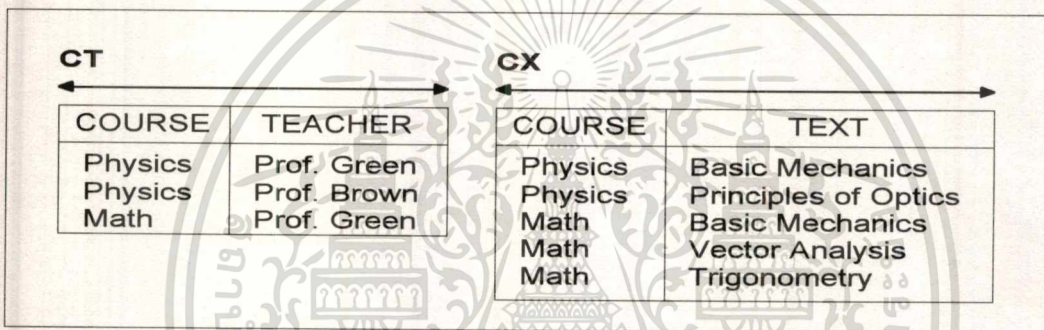
**CTX**

COURSE	TEACHER	TEXT
Physics	Prof. Green	Basic Mechanics
Physics	Prof. Green	Principles of Optics
Physics	Prof. Brown	Basic Mechanics
Physics	Prof. Brown	Principles of Optics
Math	Prof. Green	Basic Mechanics
Math	Prof. Green	Vector Analysis
Math	Prof. Green	Trigonometry

รูปที่ 2.10 แสดงตารางฐานข้อมูลที่ไม่อยู่ในรูปบรรทัดฐานขั้นที่สี่ [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางฐานข้อมูลในรูปที่ 2.10 เป็นตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้น Boyce/Codd เนื่องจาก มีกฎแฉหลักเพียงตัวเดียว ตารางฐานข้อมูลนี้ยังไม่เป็นตารางฐานข้อมูลในรูปบรรทัดฐานขั้นที่สี่ เนื่องจากมี MVD มากกว่าหนึ่งตัว ได้แก่ เมื่อเรากำหนด COURSE จะได้เซตของ TEACHER และเซตของ TEXT ข้อเสียของความซ้ำซ้อนที่เกิดขึ้นได้แก่ เมื่อต้องการเพิ่มข้อมูลเพียงว่า วิชา MATH สามารถสอนด้วย Prof. White เราจำเป็นต้อง INSERT ข้อมูลนี้ถึง 3 Tuple เนื่องจาก TEXT ที่ใช้ในการสอนวิชา MATH มีด้วยกันจำนวนสามเล่ม ดังนั้นจะเห็นว่าเราไม่สามารถที่จะ INSERT ข้อมูลทีละ 1 Tuple ได้ถ้ามี MVD เกิดขึ้นมากกว่าหนึ่งตัวในตารางเดียวกัน วิธีการกำจัดความซ้ำซ้อนของตารางฐานข้อมูลนี้เราต้องแยกตารางฐานข้อมูลนี้ออกเป็นสองตารางดังแสดงในรูปที่ 2.11 ซึ่งตารางฐานข้อมูลทั้งสองที่เกิดขึ้นใหม่นี้จัดอยู่ในตารางฐานข้อมูลในรูปบรรทัดฐานขั้นที่สี่



รูปที่ 2.11 แสดงตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นสี่

## 2.7 ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่ห้า

ฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่ห้าต้องมีคุณสมบัติดังต่อไปนี้

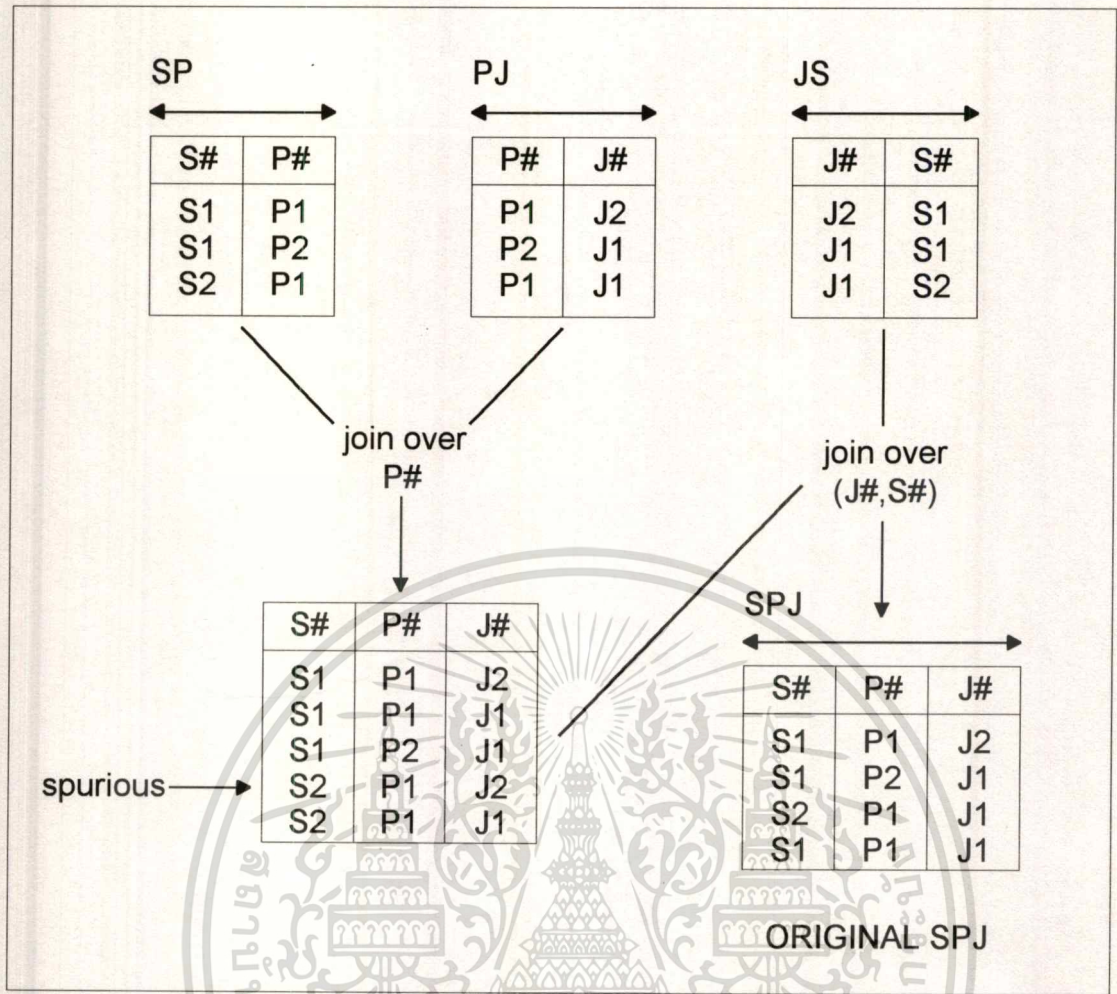
- ต้องเป็นฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สี่
- ผลลัพธ์ของการ Join ของ JD ใน R ต้องเป็น Lossless Join

SPJ

S#	P#	J#
S1	P1	J2
S1	P2	J1
S2	P1	J1
S1	P1	J1

รูปที่ 2.12 แสดงตารางฐานข้อมูลที่ไม่อยู่ในรูปบรรทัดฐานขั้นห้า [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 แสดงตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นห้า [4]

จากตารางฐานข้อมูลในรูปที่ 2.12 เป็นตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สี่ เนื่องจากทุกแอททริบิวต์รวมเป็นกุญแจหลักของตาราง และ involves no nontrivial FDs or MVDs และจากตารางฐานข้อมูลในรูปที่ 2.13 เราแยกตาราง SPJ ออกเป็น 3 ตารางคือ (SP,PJ,JS) จะเห็นว่าเมื่อทำการ join ตารางทั้งสามจะทำให้ผลลัพธ์ของข้อมูลในตารางมีค่าเท่ากับตารางเดิมทุกประการ แต่ถ้าทำการ join ตารางแค่เพียงสองตารางจะทำให้ข้อมูลของตารางที่เกิดขึ้นใหม่มีจำนวนเพิ่มมากขึ้นซึ่งไม่ถูกต้อง แสดงให้เห็นว่าเราไม่สามารถที่จะแยกตารางออกเป็นสองตารางได้แต่แยกเป็นสามตารางได้ ซึ่งการ join ของตารางทั้งสามตารางสามารถอธิบายได้ดังนี้

- ถ้ามีข้อมูล (s1,p1) ปรากฏในตาราง SP
- และมีข้อมูล (p1,j1) ปรากฏในตาราง PJ
- และมีข้อมูล (j1,s1) ปรากฏในตาราง JS
- แล้วจะต้องมีข้อมูล (s1,p1,j1) ปรากฏในตาราง SPJ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

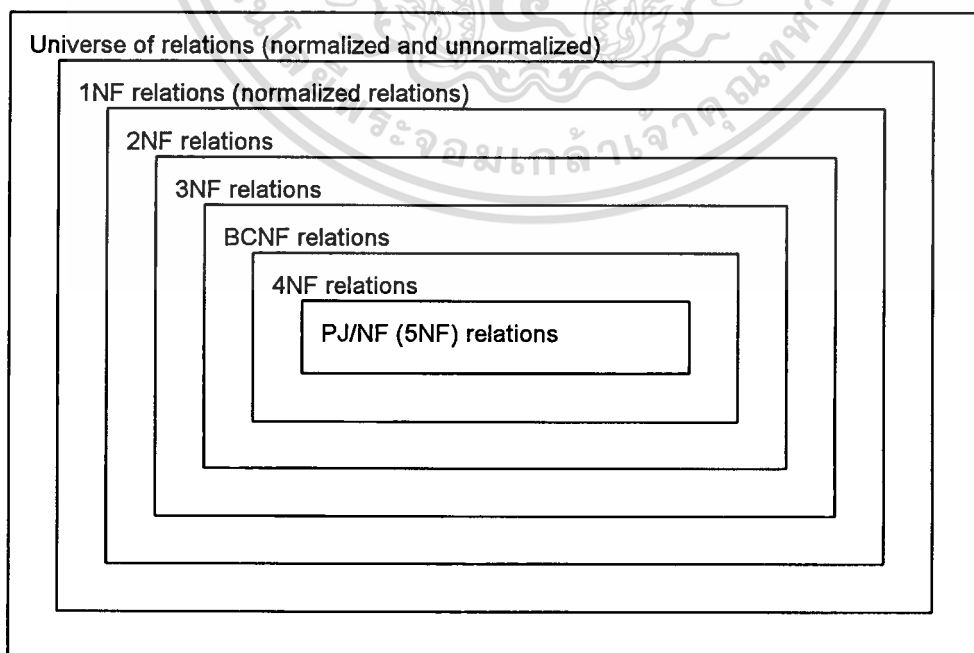
หรือกล่าวอีกนัยหนึ่งว่า

ถ้ามีข้อมูล  $(s_1, p_1, j_2), (s_2, p_1, j_1), (s_1, p_2, j_1)$  ปรากฏในตาราง SPJ

แล้วจะต้องมีข้อมูล  $(s_1, p_1, j_1)$  ปรากฏในตาราง SPJ ด้วยเช่นกัน

ดังนั้นลักษณะความซ้ำซ้อนของตาราง SPJ คือ ถ้าเราต้องการแทรกข้อมูล  $(s_2, p_1, j_1)$  เราจะต้องแทรกข้อมูล  $(s_1, p_1, j_1)$  ด้วยเช่นเดียวกัน แต่ในทางกลับกันไม่จำเป็น และถ้าเราต้องการลบข้อมูล  $(s_2, p_1, j_1)$  จะไม่มีผลกระทบใดๆ กับตาราง SPJ แต่ถ้าเราต้องการลบข้อมูล  $(s_1, p_1, j_1)$  ข้อมูลอื่นจะต้องถูกลบทิ้งด้วยอีกหนึ่งแถว จะเห็นว่าในการแทรกและลบข้อมูลของตาราง SPJ มีความยุ่งยากมากและเกิดความผิดพลาดได้ง่าย ดังนั้นถ้าเราแยกตาราง SPJ ออกเป็นสามตารางแล้วปัญหาต่างๆที่เกิดขึ้นจะหมดไปซึ่งตารางฐานข้อมูลทั้งสามเกิดขึ้นใหม่นี้จัดอยู่ในตารางฐานข้อมูลในรูปบรรทัดฐานขั้นที่ห้าเพราะการ join กันของตารางทั้งสามเป็น Lossless join [3]

ที่ได้กล่าวมาทั้งหมดนี้ทำให้สามารถแสดงความสัมพันธ์ระหว่างฐานข้อมูลในรูปบรรทัดฐานขั้นต่างๆในรูปของเซตได้ดังรูปที่ 2.14 คือ ตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่ห้าซึ่งเป็นขั้นสูงสุดนั้นเป็นตารางฐานข้อมูลในรูปบรรทัดฐานขั้นที่สี่ด้วยแต่ตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่สี่ไม่จำเป็นที่จะต้องเป็นตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่ห้าหรือกล่าวโดยสรุปว่าตารางฐานข้อมูลที่อยู่ในรูปบรรทัดฐานขั้นที่ห้าจะต้องผ่านคุณสมบัติของการเป็นตารางฐานข้อมูลรูปบรรทัดฐานขั้นที่หนึ่งจนถึงขั้นที่สี่มาก่อนแล้วทั้งสิ้น และยังต้องมีคุณสมบัติของการเป็นตารางฐานข้อมูลรูปบรรทัดฐานขั้นที่ห้าอีกด้วย



รูปที่ 2.14 แสดงความสัมพันธ์ของตารางฐานข้อมูลในรูปบรรทัดฐานขั้นต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ญาติเห็นไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### การออกแบบและสร้างตารางฐานข้อมูลด้วยแบบจำลองอีอาร์

การออกแบบสร้างตารางฐานข้อมูลด้วยแบบจำลองอีอาร์ [2] แบ่งออกเป็นสองขั้นตอนหลักคือ ขั้นตอนแรก เป็นการสร้างแบบจำลองอีอาร์ขั้นตอนนี้จะกล่าวถึงความหมายของสัญลักษณ์ต่างๆที่ใช้ในแบบจำลองอีอาร์เพื่อที่จะทำให้สามารถสร้างแบบจำลองอีอาร์ขึ้นใช้งานได้อย่างถูกต้องและมีประสิทธิภาพ ขั้นตอนที่สอง เป็นการแปลงแบบจำลองอีอาร์ให้เป็นตารางฐานข้อมูลเพื่อใช้งานกับตัวจัดการฐานข้อมูลแบบรีเลชันแนลขั้นตอนนี้จะกล่าวถึงหลักการสร้างตารางฐานข้อมูลจากแบบจำลองอีอาร์ เมื่อเข้าใจขั้นตอนดังกล่าวทั้งสองนี้ดีแล้วจะทำให้สามารถออกแบบฐานข้อมูลและสร้างตารางฐานข้อมูลด้วยแบบจำลองอีอาร์ขึ้นใช้งานได้ด้วยตนเอง

#### 3.1 สัญลักษณ์ที่ใช้ในแบบจำลองอีอาร์

การออกแบบฐานข้อมูลด้วยแบบจำลองอีอาร์นั้นก่อนอื่นผู้ออกแบบจำเป็นต้องรู้จักกับความหมายของสัญลักษณ์ต่างๆที่ใช้ในแบบจำลองอีอาร์เพื่อที่จะทำให้สามารถสร้างแบบจำลองอีอาร์ได้อย่างถูกต้อง ดังนั้นในหัวข้อนี้จึงจะกล่าวถึงความหมายและการใช้งานของสัญลักษณ์ต่างๆของแบบจำลองอีอาร์พร้อมทั้งตัวอย่างการใช้งานในแต่ละสัญลักษณ์เพื่อให้เข้าใจในการทำความเข้าใจตามลำดับ

3.1.1 เอ็นติตี้ ที่ใช้บนแบบจำลองอีอาร์มีอยู่ด้วยกัน 2 ชนิดได้แก่

- เอ็นติตี้ชนิดปกติ

ENTITY NAME

**สัญลักษณ์** จะใช้สัญลักษณ์รูปกรอบสี่เหลี่ยมโดยที่ภายในจะเป็นตัวหนังสือบอกถึงชื่อของเอ็นติตี้ นั้น ซึ่งโดยทั่วไปการตั้งชื่อเอ็นติตี้จะต้องสื่อความหมายได้ชัดเจนในสิ่งที่เราสนใจและต้องเป็นชื่อที่สามารถใช้เป็นชื่อของตารางฐานข้อมูลได้

**ความหมาย** หมายถึง สิ่งต่างๆที่เราให้ความสนใจในองค์กร เช่นพนักงาน (EMPLOYEE) หน่วยงาน (DEPARTMENT) หรือ โครงการงาน (PROJECT) เป็นต้น

**ตัวอย่าง**

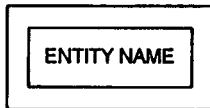
EMPLOYEE

DEPARTMENT

PROJECT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

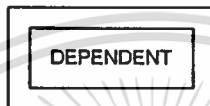
- เอ็นตีตีชนิดอ่อน



**สัญลักษณ์** จะใช้สัญลักษณ์รูปกรอบสี่เหลี่ยมซ้อนกันสองรูปโดยที่ภายในจะเป็นตัวหนังสือบอกถึงชื่อของเอ็นตีตีชนิดอ่อนนั้น

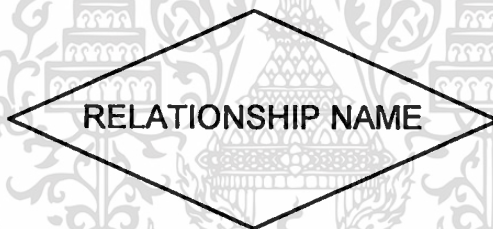
**ความหมาย** หมายถึง เอ็นตีตีที่ไม่สามารถมีกฎแจกหลักเป็นของตัวเอง

**ตัวอย่าง**



3.1.2 ความเกี่ยวพัน ที่ใช้บนแบบจำลองอาร์มีอยู่ด้วยกัน 2 ชนิดได้แก่

- ความเกี่ยวพันชนิดปกติ



**สัญลักษณ์** จะใช้สัญลักษณ์รูปสี่เหลี่ยมข้าวหลามตัดโดยที่ภายในจะเป็นตัวหนังสือบอกถึงชื่อของความสัมพันธ์นั้น ซึ่งโดยทั่วไปการตั้งชื่อความสัมพันธ์จะต้องสื่อความหมายได้ชัดเจนว่าแต่ละเอ็นตีตีมีความเกี่ยวพันกันอย่างไร

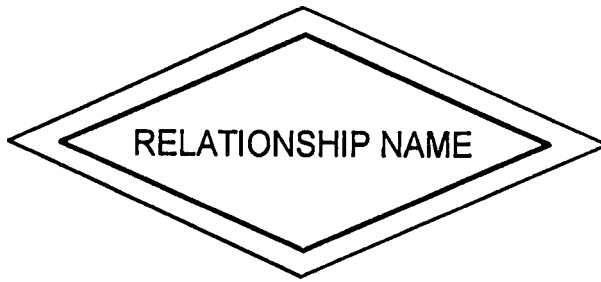
**ความหมาย** หมายถึง ความเกี่ยวพันระหว่างเอ็นตีตีตั้งแต่สองเอ็นตีตีขึ้นไป เช่น เอ็นตีตี DEPARTMENT กับเอ็นตีตี EMPLOYEE มีความเกี่ยวพันกันคือ ทุก DEPARTMENT ต้องมีผู้บริหาร (ผู้บริหารก็คือพนักงาน (EMPLOYEE) คนหนึ่ง) มาบริหารงาน (MANAGES) และมีได้เพียงคนเดียวเท่านั้น

**ตัวอย่าง**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

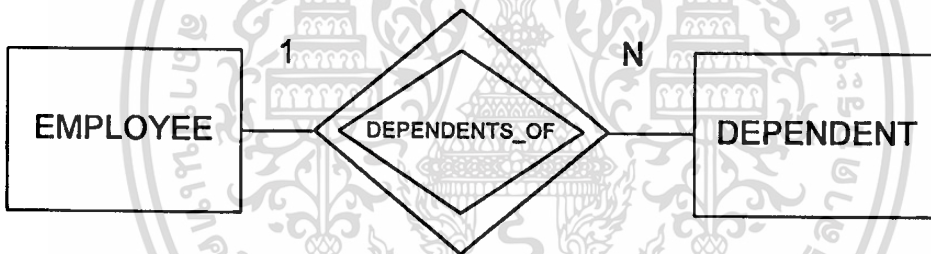
- ความเกี่ยวพันชนิดอ่อน



**สัญลักษณ์** จะใช้สัญลักษณ์รูปสี่เหลี่ยมข้าวหลามตัดซ้อนกันสองรูป โดยที่ภายในจะเป็นตัวหนังสือบอกถึงชื่อของความเกี่ยวพันชนิดอ่อนนั้น

**ความหมาย** หมายถึงความเกี่ยวพันที่เชื่อมระหว่างเอนิตีชนิดปกติกับเอนิตีชนิดอ่อนเช่น ลูกของพนักงานอาจจะมีชื่อซ้ำกันได้แต่จะไม่มีกรซ้ำถ้ารวมเอาเลขประจำตัวของพนักงานที่เป็นผู้ปกครองเข้าไปด้วย

**ตัวอย่าง**



3.1.3 แอททริบิวต์ ที่ใช้บนแบบจำลองอีอาร์มีอยู่ด้วยกัน 5 ชนิดได้แก่

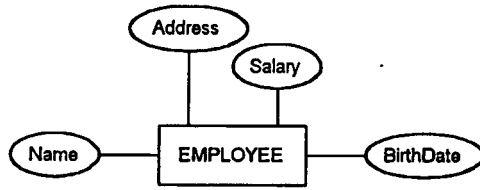
- แอททริบิวต์ชนิดปกติ



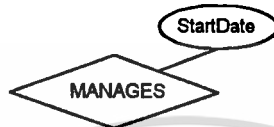
**สัญลักษณ์** จะใช้สัญลักษณ์รูปวงรี โดยที่ภายในจะเป็นตัวหนังสือบอกถึงชื่อของแอททริบิวต์นั้น ซึ่งโดยทั่วไปการตั้งชื่อแอททริบิวต์จะต้องสื่อความหมายได้ชัดเจนถึงคุณสมบัติของเอนิตีหรือความเกี่ยวพันที่มีแอททริบิวต์นั้นเชื่อมโยงอยู่และชื่อของแอททริบิวต์ต้องเป็นชื่อที่สามารถใช้เป็นชื่อของแอททริบิวต์ของตารางฐานข้อมูลได้

**ความหมาย** หมายถึง ลักษณะประกอบของแต่ละเอนิตีหรือความเกี่ยวพันเช่น เอนิตีพนักงาน (EMPLOYEE) ประกอบด้วย ชื่อ ที่อยู่ เงินเดือน และ วันเกิด หรือ ความเกี่ยวพันบริหาร (MANAGES) ประกอบด้วย วันที่เริ่มรับตำแหน่ง เป็นต้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง



แอททริบิวต์ของเอนทิตี



แอททริบิวต์ของความสัมพันธ์

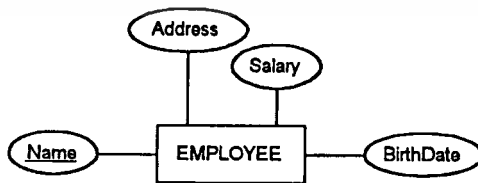
- แอททริบิวต์ชนิดกุญแจ



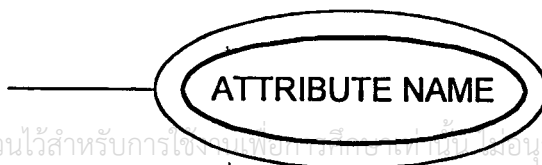
สัญลักษณ์ จะใช้สัญลักษณ์รูปวงรี โดยที่ภายในจะเป็นตัวหนังสือบอกถึงชื่อของแอททริบิวต์นั้น และชื่อแอททริบิวต์จะต้องมีการขีดเส้นใต้

ความหมาย หมายถึงแอททริบิวต์ที่มีคุณสมบัติพิเศษดังนี้คือ ข้อมูลภายในแอททริบิวต์ชนิดกุญแจนั้นต้องไม่ซ้ำและไม่เป็นค่าเป็น NULL จากตัวอย่างแอททริบิวต์ชนิดกุญแจคือแอททริบิวต์ Name

ตัวอย่าง



- แอททริบิวต์ชนิดหลายค่า



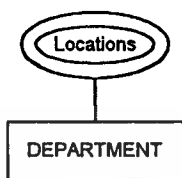
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สถานหอสมุดกลาง พระจอมเกล้าลาดกระบัง

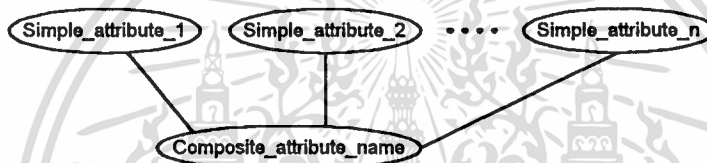
**สัญลักษณ์** จะใช้สัญลักษณ์รูปวงรีซ้อนกันสองวง โดยที่ภายในจะเป็นตัวหนังสือบอกถึงชื่อของแอททริบิวต์นั้น

**ความหมาย** หมายถึง แอททริบิวต์ที่สามารถมีค่าได้หลายค่าเช่น ในหนึ่งหน่วยงาน (DEPARTMENT) สามารถมีสถานที่ทำการ (LOCATIONS) ได้หลายแห่ง

**ตัวอย่าง**



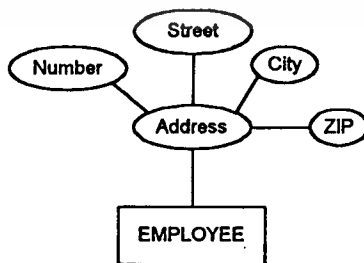
- แอททริบิวต์ชนิดกลุ่ม



**สัญลักษณ์** จะใช้สัญลักษณ์รูปวงรีซ้อนกันสองวงแยกกันโดยมีเส้นเชื่อมโยงวงรีทั้งหลายวงนั้นเข้ากับวงรีหลักเพียงวงเดียว โดยที่ภายในวงรีทั้งหลายจะเป็นตัวหนังสือบอกถึงชื่อของแอททริบิวต์ย่อยนั้นและภายในวงรีหลักจะเป็นตัวหนังสือบอกถึงความหมายโดยรวมของวงรีย่อยนั้นทั้งหมด

**ความหมาย** หมายถึง แอททริบิวต์ที่ประกอบด้วย Simple แอททริบิวต์หลายๆ แอททริบิวต์เช่น แอททริบิวต์ชนิดกลุ่ม Address ประกอบด้วย Simple แอททริบิวต์ เลขที่บ้าน (NUMBER) ถนน (STREET) จังหวัด (CITY) และรหัสไปรษณีย์ (ZIP) เป็นต้น

**ตัวอย่าง**



- แอททริบิวต์ชนิดวิเคราะห์ได้

ATTRIBUTE NAME

**สัญลักษณ์** จะใช้สัญลักษณ์รูปวงรีที่วาดด้วยเส้นไขว่ปลา โดยที่ภายในจะเป็นตัวหนังสือบอกถึงชื่อของแอททริบิวต์นั้น

**ความหมาย** หมายถึง แอททริบิวต์ที่สามารถคำนวณหาค่าได้ตามฟังก์ชันทางคณิตศาสตร์ เช่น จำนวนพนักงาน (Number Of Employee) ที่ทำงานในแต่ละ DEPARTMENT สามารถหาได้จากผลรวมของพนักงานที่ทำงานใน DEPARTMENT นั้น

ตัวอย่าง



### 3.1.4 การกำหนดตัวควบคุม (Constraint) ลงบนแอททริบิวต์

การกำหนดตัวควบคุมลงบนแอททริบิวต์นี้เป็นส่วนเพิ่มเติมขึ้นในวิทยานิพนธ์ฉบับนี้ เพื่อให้สามารถสร้างวิวัฒนาการควบคุมความถูกต้องของข้อมูลในตารางฐานข้อมูลที่สร้างขึ้นอีกชั้นหนึ่งมีอยู่ด้วยกัน 2 ประเภทคือ

- การกำหนดตัวควบคุมเป็นตัวเลข

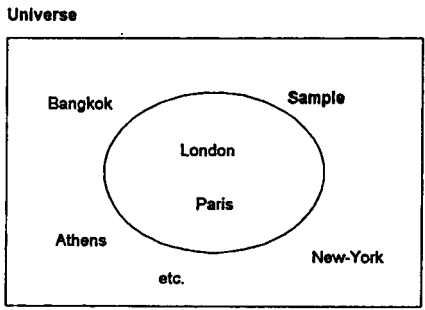


รูปที่ 3.1 แสดงข้อมูลที่อยู่ในเส้นจำนวน

จะใช้กับแอททริบิวต์ที่มีลักษณะข้อมูล (Datatype) เป็นตัวเลข โดยจะกำหนดเป็นลักษณะของเส้นจำนวน เช่นถ้าต้องการให้แอททริบิวต์ Sample มีข้อมูลที่อยู่บนเส้นจำนวนดังรูปที่ 3.1 จะต้องกำหนดตัวควบคุมที่มีรูปแบบดังนี้ { (<,-15),(-10,-5],[5,10],[15,>) } เป็นต้น

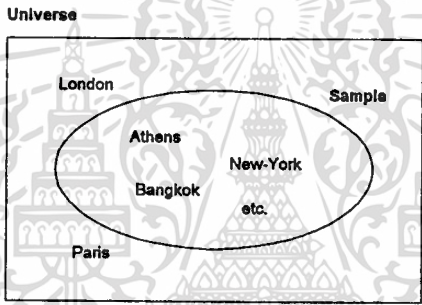
- การกำหนดตัวควบคุมเป็นตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงข้อมูลที่อยู่ในเซต Sample

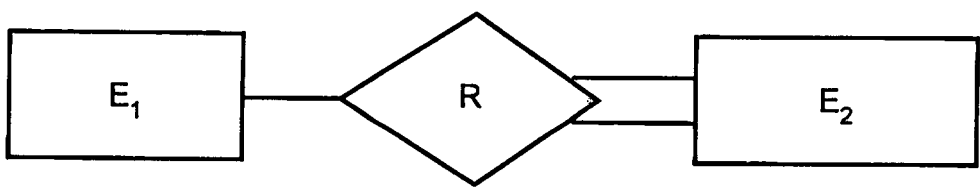
จะใช้กับแอททริบิวต์ที่มีลักษณะข้อมูล (Datatype) เป็นตัวอักษรโดยจะกำหนดในรูปของเซต เช่นถ้าต้องการให้แอททริบิวต์ Sample มีข้อมูลที่อยู่ในเซตดังรูปที่ 3.2 จะต้องกำหนดตัวควบคุมที่มีรูปแบบดังนี้ { 'London','Paris' }



รูปที่ 3.3 แสดงข้อมูลไม่ที่อยู่ในเซต Sample

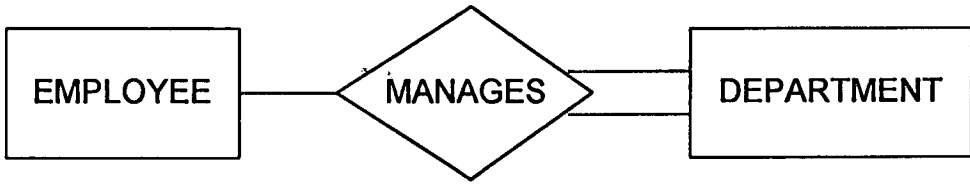
แต่ถ้าไม่ต้องการให้แอททริบิวต์ Sample มีข้อมูลอยู่ในเซตดังรูปที่ 3.3 ต้องกำหนดตัวควบคุมที่มีรูปแบบดังนี้ ! { 'London','Paris' }

3.1.5 ไรล ที่ใช้บนแบบจำลองอ็อบเจกต์มีอยู่ด้วยกัน 2 ชนิดได้แก่  
- ไทฮอล

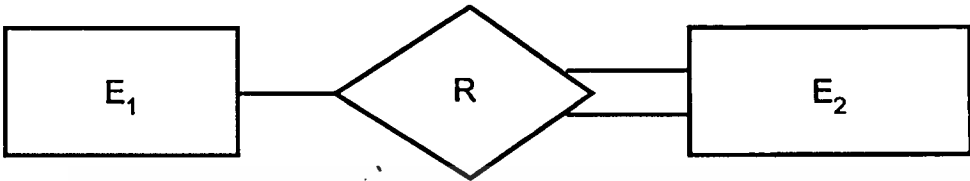


**สัญลักษณ์** จะใช้สัญลักษณ์เส้นคู่เชื่อมโยงระหว่างความเกี่ยวพันและเอ็นติตี้  
**ความหมาย** หมายถึงข้อมูล ภายในเอ็นติตี้ E2 ต้องมีค่าทุกค่าจะไม่ทราบค่า (NULL) ไม่ได้ เช่น DEPARTMENT ต้องมีผู้จัดการจะไม่มีไม่ได้

เอกสารนี้ ตัวอย่าง ารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- พาเซ็ล



**สัญลักษณ์** จะใช้สัญลักษณ์เส้นเดี่ยวเชื่อมโยงระหว่างความเกี่ยวพันและเอนิตตี้  
**ความหมาย** หมายถึงข้อมูล ภายในเอนิตตี้ E<sub>1</sub> สามารถที่จะไม่ทราบค่า (NULL) ได้ เช่น EMPLOYEE ที่ไม่ได้เป็นผู้จัดการของ DEPARTMENT ใดเลยสามารถกำหนดค่าเป็น NULL ได้

ตัวอย่าง



### 3.1.6 อัตราส่วนคาคินออลลิตี้



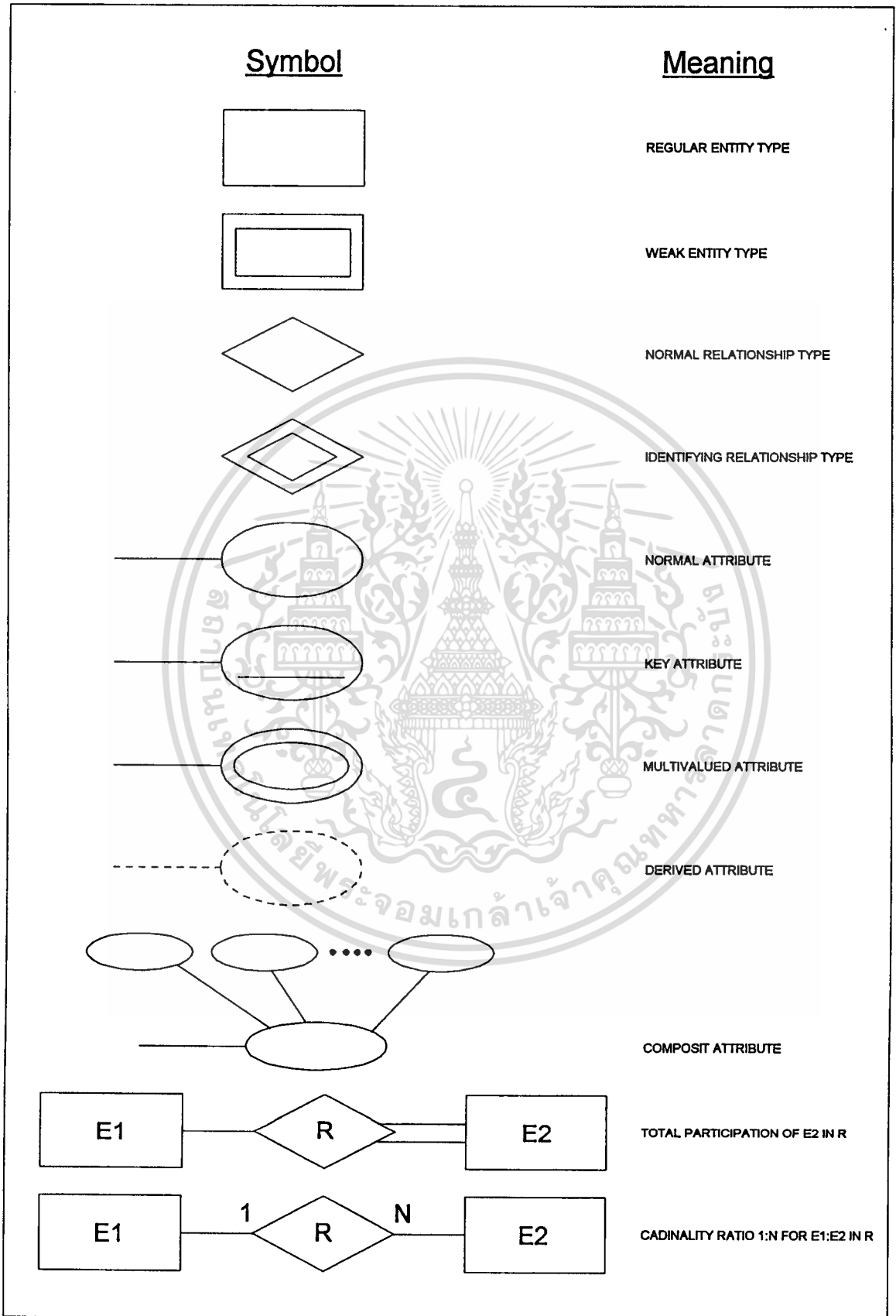
**สัญลักษณ์** จะใช้สัญลักษณ์ตัวอักษร 1,N หรือ M วางอยู่ที่เส้นที่เชื่อมกับ  
**ความเกี่ยวพัน**

**ความหมาย** หมายถึง เอนิตตี้ E<sub>1</sub> สามารถมีข้อมูลของเอนิตตี้ E<sub>2</sub> ได้หลายค่า เช่น DEPARTMENT สามารถมีพนักงานได้หลายคนแต่พนักงานแต่ละคนอยู่ได้ DEPARTMENT เดียว

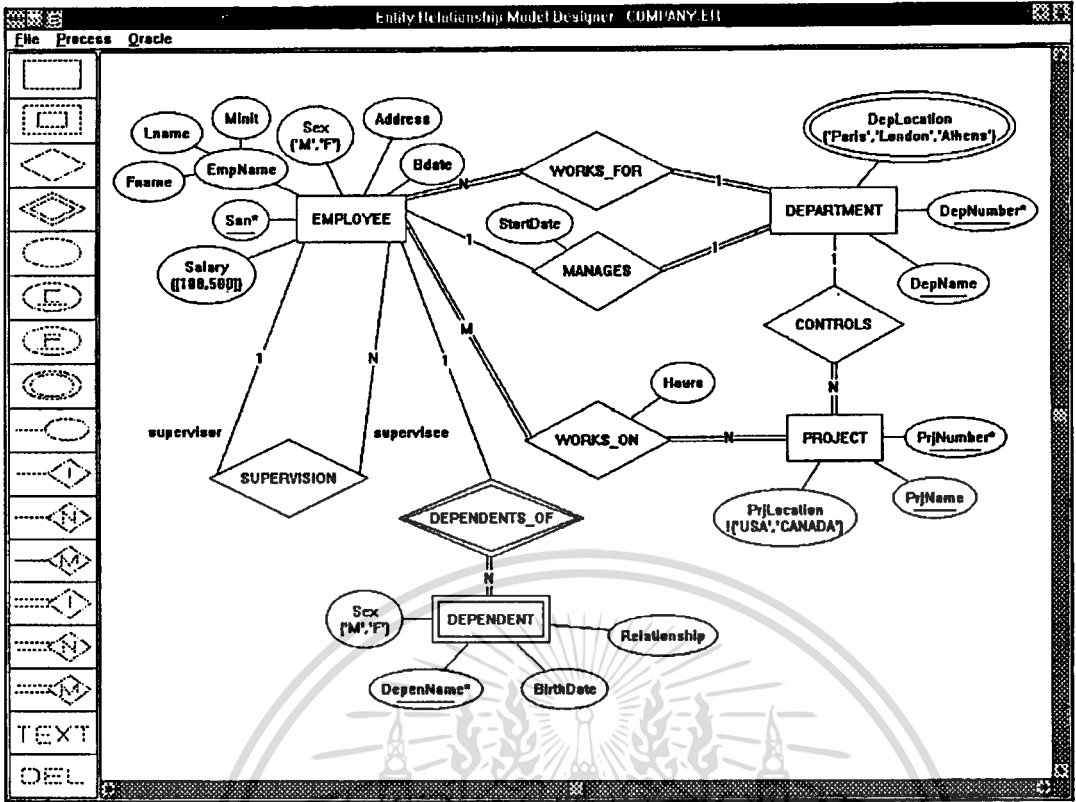
ตัวอย่าง



### 3.1.7 สัญลักษณ์และตัวอย่างแบบจำลองอ็อบเจกต์ที่สร้างด้วยระบบER/5



เอกสารนี้เป็นเอกสารที่สงวนรูปที่ 3.4 แสดงสัญลักษณ์ที่ใช้ในแบบจำลองอ็อบเจกต์ที่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงแบบจำลองอ็อบเจกต์ที่สร้างด้วยระบบ ER/5

### 3.2 หลักการสร้างตารางฐานข้อมูลจากแบบจำลองอ็อบเจกต์

การแปลงแบบจำลองอ็อบเจกต์ให้เป็นตารางข้อมูลในรูปบรรทัดฐานขั้นแรกมีขั้นตอนดังต่อไปนี้

- สำหรับแต่ละเอนทิตีชนิดปกติ (Regular entity type) บนแบบจำลองอ็อบเจกต์สามารถสร้างตารางได้ 1 ตาราง ซึ่งประกอบด้วยแอททริบิวต์ชนิดเดี่ยว (Simple attribute) ของเอนทิตีนั้น แต่สำหรับแอททริบิวต์ชนิดกลุ่ม (Composite attribute) เราจะรวมเอาเฉพาะแอททริบิวต์ที่ประกอบขึ้นเป็นกลุ่ม (Simple component attribute) เท่านั้นและจะเลือกหนึ่งในจำนวนแอททริบิวต์ชนิดกุญแจ (Key attribute) มาเป็นกุญแจหลัก (Primary Key) ของตาราง แต่ถ้าแอททริบิวต์ชนิดกุญแจของเอนทิตีที่เลือกนั้นเป็นแอททริบิวต์ชนิดกลุ่มแล้วแอททริบิวต์ที่ประกอบขึ้นเป็นกลุ่มจะรวมกันเป็นกุญแจหลักชนิดกลุ่ม (Combine Primary Key)

จากรูปที่ 3.5 เราสามารถสร้างตารางตามขั้นตอนนี้ได้ 3 ตาราง คือ EMPLOYEE DEPARTMENT และ PROJECT (สำหรับแอททริบิวต์ที่เป็นกุญแจชนิดกุญแจหลักของตารางอื่น (foreign key (f.k.)) และแอททริบิวต์ของความสัมพันธ์ (attribute of relationship (\*)) จะยังไม่ถูกรวมอยู่ในตารางแต่จะกล่าวถึงในขั้นตอนต่อไป) โดยที่แอททริบิวต์ SSN DNUMBER และ PNUMBER เป็นกุญแจหลักของตาราง EMPLOYEE DEPARTMENT และ PROJECT ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะวิธีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สำหรับแต่ละเอ็นติตี้ชนิดอ่อน (Weak entity type) บนแบบจำลองอ็อบเจกต์ซึ่งขึ้นกับเอ็นติตี้ชนิดปกติสามารถสร้างตารางได้ 1 ตาราง ซึ่งประกอบด้วยแอททริบิวต์ชนิดเดี่ยว (หรือแอททริบิวต์ที่ประกอบขึ้นเป็นกลุ่มของแอททริบิวต์ชนิดกลุ่ม) และรวมแอททริบิวต์ที่เป็นกุญแจหลักของเอ็นติตี้ชนิดปกติเข้าไปด้วยทำให้กุญแจหลักของตารางคือกุญแจหลักชนิดกลุ่มที่ประกอบขึ้นจากแอททริบิวต์ที่เป็นกุญแจหลักของเอ็นติตี้ชนิดปกติกับแอททริบิวต์ที่เป็นกุญแจชนิดย่อย (Partial key แอททริบิวต์) ของเอ็นติตี้ชนิดอ่อนหลังจากนั้นให้แปลงเอ็นติตี้ชนิดอ่อนให้เป็นเอ็นติตี้ชนิดปกติเพื่อนำไปประมวลผลในขั้นตอนต่อไป

จากรูปที่ 3.5 เราสามารถสร้างตารางตามขั้นตอนนี้ได้ 1 ตาราง คือ DEPENDENT ซึ่งประกอบด้วยแอททริบิวต์ DEPARTMENT\_NAME SEX BirthDate และแอททริบิวต์ที่เป็นกุญแจหลักของเอ็นติตี้ EMPLOYEE คือ SSN (เพื่อป้องกันการสับสนจึงเปลี่ยนชื่อแอททริบิวต์นี้ใหม่เป็น ESSN) ซึ่งมีกุญแจหลัก คือ {ESSN,DEPARTMENT\_NAME}

- สำหรับแต่ละความสัมพันธ์ชนิด Binary 1:1 บนแบบจำลองอ็อบเจกต์ที่เชื่อมต่อระหว่างเอ็นติตี้ S และ T เราจะต้องเพิ่มแอททริบิวต์ที่เป็นกุญแจหลักของ S ให้กับตาราง T หรือกุญแจหลักของ T ให้กับตาราง S อย่างใดอย่างหนึ่งและต้องกำหนดเป็นแอททริบิวต์ชนิดกุญแจหลักของตารางอื่นโดยมีหลักการเลือกคือ เราจะเลือกตารางของเอ็นติตี้ที่มีตัวควบคุม (constraint) ชนิดโททอลกำกับอยู่เพื่อกำจัดข้อมูลที่ไม่ทราบค่า (NULL VALUE) หลังจากนั้นก็จะรวมเอาแอททริบิวต์ของความสัมพันธ์เข้าไปด้วยโดยกุญแจหลักของตารางยังคงเดิม

จากรูปที่ 3.5 จะเห็นว่าตารางที่ถูกเลือกคือ DEPARTMENT เนื่องจากมีตัวควบคุมชนิดโททอลกำกับอยู่และแอททริบิวต์ที่เพิ่มขึ้นคือ MGRSSN (กุญแจหลักของเอ็นติตี้ EMPLOYEE) และ MGRSTART เป็นแอททริบิวต์ของความสัมพันธ์)

- สำหรับแต่ละความสัมพันธ์ชนิด Binary 1:N บนแบบจำลองอ็อบเจกต์ที่เชื่อมต่อระหว่างเอ็นติตี้ S และ T โดยที่ S เป็นเอ็นติตี้ที่อยู่ทางด้าน N และ T เป็นเอ็นติตี้ที่อยู่ทางด้าน 1 เราจะต้องเพิ่มแอททริบิวต์ที่เป็นกุญแจหลักของ T ให้กับตาราง S โดยที่กุญแจหลักของตารางยังคงเดิม

จากรูปที่ 3.5 จะเห็นว่าเอ็นติตี้ EMPLOYEE อยู่ทางด้าน N ของความสัมพันธ์ WORK\_FOR และ SUPERVISION ดังนั้นแอททริบิวต์ที่เพิ่มขึ้นก็คือ SUPERSSN (กุญแจหลักของเอ็นติตี้ EMPLOYEE ที่อยู่ทางด้าน 1 โดยเปลี่ยนชื่อมาจาก SSN) และ DNO (กุญแจหลักของเอ็นติตี้ DEPARTMENT ที่อยู่ทางด้าน 1) โดยกุญแจหลักของตารางยังคงเดิมและในเอ็นติตี้ PROJECT ก็ทำนองเดียวกัน

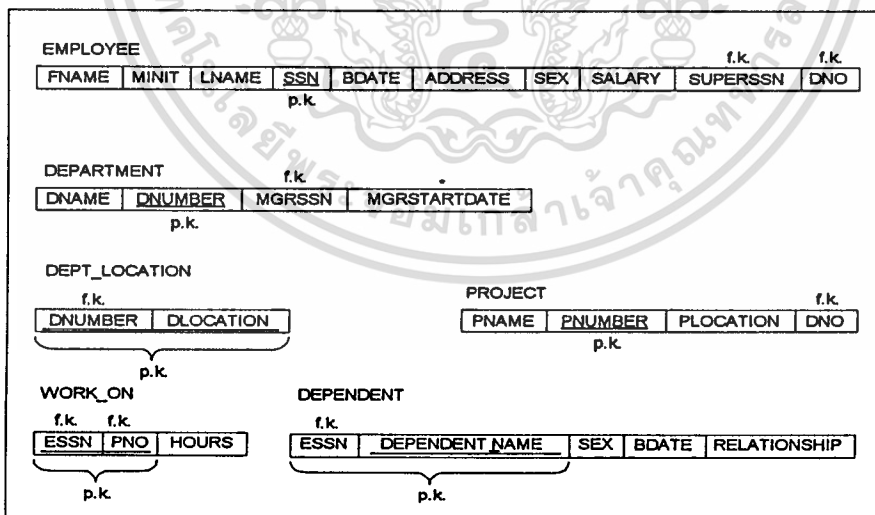
- สำหรับแต่ละความสัมพันธ์ชนิด Binary M:N บนแบบจำลองอ็อบเจกต์ที่เชื่อมต่อระหว่างเอ็นติตี้ S และ T เราจะสร้างตารางได้ 1 ตาราง ซึ่งประกอบด้วยแอททริบิวต์หรือ Simple  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอททริบิวต์ที่ประกอบขึ้นเป็นกลุ่มของแอททริบิวต์ชนิดกลุ่มของความเกี่ยวพันนั้นรวมกับแอททริบิวต์ที่เป็นกุญแจหลักของเอ็นติตี้ S และ T โดยกุญแจหลักของตารางที่ได้คือแอททริบิวต์ชนิดกลุ่มที่ประกอบขึ้นจากกุญแจหลักของเอ็นติตี้ S กับ T

จากรูปที่ 3.5 สามารถสร้างตารางตามขั้นตอนนี้ได้ 1 ตารางคือ WORK\_ON ซึ่งประกอบด้วยแอททริบิวต์ Hours (แอททริบิวต์ของความเกี่ยวพัน) ESSN (กุญแจหลักของเอ็นติตี้ EMPLOYEE) และ PNO (กุญแจหลักของเอ็นติตี้ PROJECT) โดยมีกุญแจหลักของตารางคือ (ESSN,PNO)

- สำหรับแต่ละแอททริบิวต์ชนิดหลายค่า (Multivalued attribute) A เราต้องสร้างตารางใหม่ ซึ่งประกอบด้วยแอททริบิวต์ A รวมกับแอททริบิวต์ที่เป็นกุญแจหลัก K ของเอ็นติตี้หรือความเกี่ยวพันที่เชื่อมต่อกับ A และกุญแจหลักของตารางก็คือ กุญแจหลักชนิดกลุ่มที่ประกอบขึ้นจากแอททริบิวต์ A กับ K (ถ้าแอททริบิวต์ชนิดหลายค่านั้นเป็นแอททริบิวต์ชนิดกลุ่มด้วยแล้วเราจะรวมเฉพาะแอททริบิวต์ที่ประกอบขึ้นเป็นกลุ่มของแอททริบิวต์ชนิดกลุ่มนั้น)

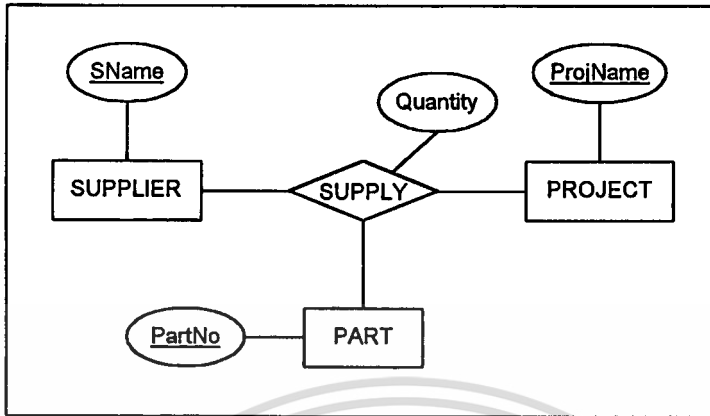
จากรูปที่ 3.5 สามารถสร้างตารางตามขั้นตอนนี้ได้ 1 ตารางคือ DEPT\_LOCATION โดยมีแอททริบิวต์ชนิดหลายค่าคือ Location ของ DEPARTMENT (ใช้ชื่อย่อเป็น DLOCATION) ขณะที่ NUMBER ของ DEPARTMENT (ใช้ชื่อย่อเป็น DNUMBER) เป็นกุญแจหลักของเอ็นติตี้ DEPARTMENT ซึ่งเชื่อมต่อกับ DLOCATION ดังนั้นตารางที่ได้จึงประกอบด้วยแอททริบิวต์ DNUMBER และ DLOCATION โดยที่กุญแจหลักของตารางคือ (DNUMBER,DLOCATION)



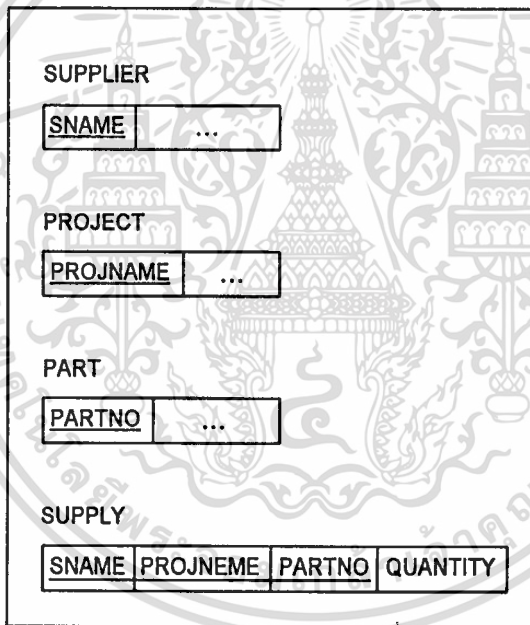
รูปที่ 3.6 แสดงตารางที่ได้ทั้งหมดจากการแปลงแบบจำลองอ็อบเจกต์ในรูปที่ 3.5

- สำหรับแต่ละความเกี่ยวพันชนิด n-ary บนแบบจำลองอ็อบเจกต์โดยที่  $n > 2$  เราสามารถสร้างตารางได้ 1 ตาราง ซึ่งประกอบด้วยแอททริบิวต์ที่เป็นกุญแจหลักของเอ็นติตี้ทุกเอ็นติตี้ที่เชื่อมต่อกับความเกี่ยวพันชนิด n-ary นั้นรวมกับแอททริบิวต์ของความเกี่ยวพันชนิด n-ary ของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวเองโดยที่กุญแจหลักของตารางคือแอททริบิวต์ชนิดกลุ่มที่ประกอบขึ้นจากแอททริบิวต์ที่เป็นกุญแจหลักของทุกเ็นิตี้นั้นๆ



รูปที่ 3.7 แสดงความสัมพันธ์ชนิด 3-ary บนแบบจำลองอริอาร์



รูปที่ 3.8 แสดงตารางที่สร้างจากความสัมพันธ์ชนิด 3-ary บนแบบจำลองอริอาร์ในรูปที่ 3.7

จากรูปที่ 3.7 จะเห็นว่าความสัมพันธ์ SUPPLY เป็นชนิด 3-ary ดังนั้นตารางที่สร้างใหม่จึงประกอบด้วยแอททริบิวต์ SNAME PROJNAME PARTNO (กุญแจหลักของเ็นิตี้นั้นๆ SUPPLIER PROJECT และ PART ตามลำดับ) และ QUANTITY (แอททริบิวต์ของความสัมพันธ์ชนิด 3-ary) โดยที่กุญแจหลักของตารางคือ {SNAME,PROJNAME,PARTNO} ดังแสดงในรูปที่ 3.8

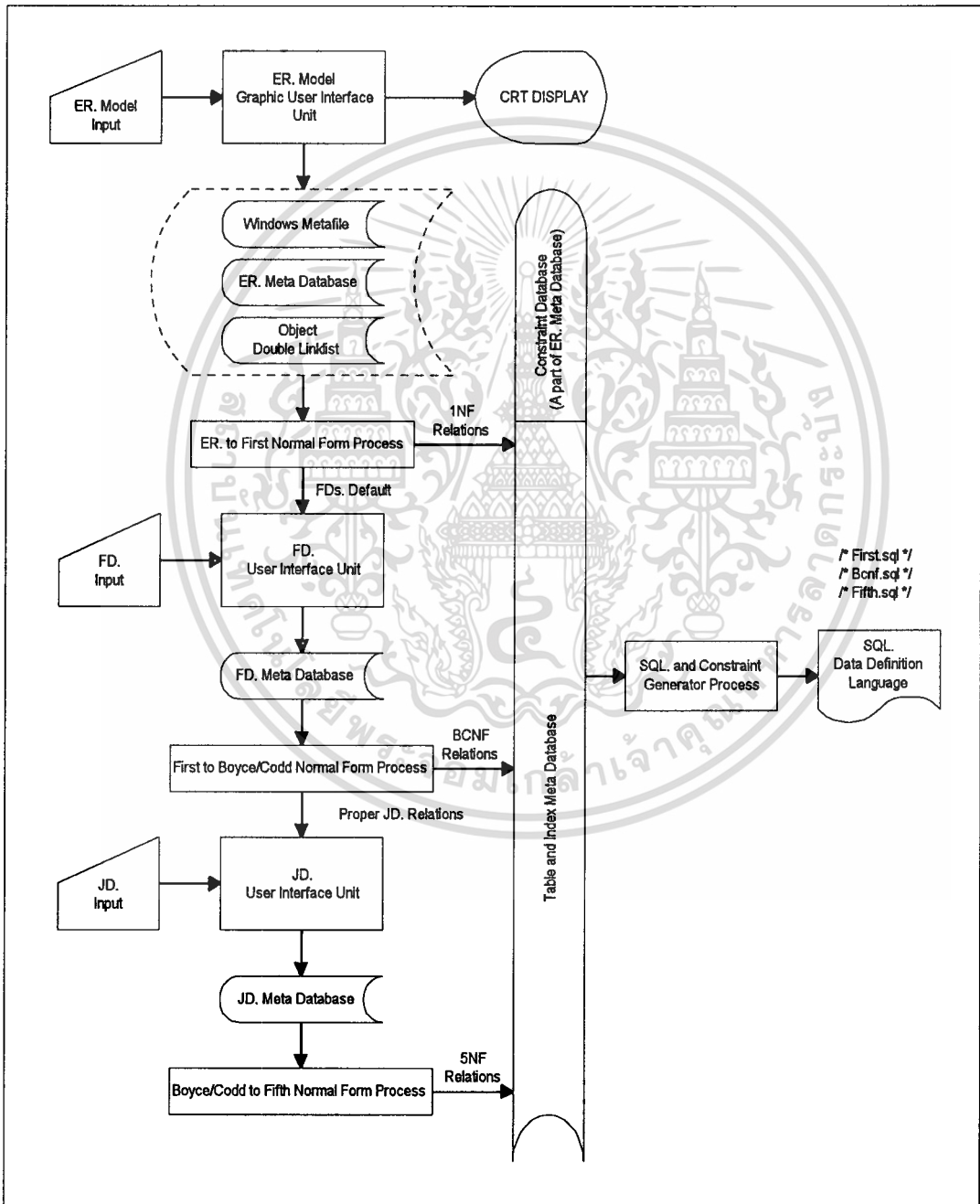
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

## สถาปัตยกรรมของระบบ ER/5

### 4.1 สถาปัตยกรรมของระบบ ER/5 โดยรวม

จากรูปที่ 4.1 แสดงสถาปัตยกรรมของระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลแบบรีเลชันแนลชื่อ ER/5 ซึ่งประกอบด้วยส่วนการทำงาน 13 ส่วนหลักดังต่อไปนี้คือ



รูปที่ 4.1 แสดงสถาปัตยกรรมของระบบช่วยออกแบบฐานข้อมูล ER/5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.ฐานข้อมูลวินโดวเมทต้าไฟล์ (Windows Meta File)
2. โครงสร้างข้อมูลวัตถุดับเบิลลิงคิลิสต์ (Object Double Linklist)
- 3.ฐานข้อมูลอีอาร์เมทต้า (ER Meta Database)
- 4.ฐานข้อมูลเอฟดีเมทต้า (FD Meta Database)
- 5.ฐานข้อมูลเจดีเมทต้า (JD Meta Database)
- 6.ฐานข้อมูลตารางและอินเด็กซ์เมทต้า (Table and Index Meta Database)
- 7.หน่วยติดต่อผู้ใช้ด้านกราฟฟิก (Graphic User Interface Unit)
8. ขบวนการสร้างตารางฐานข้อมูลในรูปบรรทัดฐานขั้นแรกจากแบบจำลองอีอาร์ (ER to First NF. Mapping)
9. หน่วยติดต่อผู้ใช้ด้านความสัมพันธ์ระหว่างแอททริบิวต์ (Functional Dependency Interface Unit)
10. ขบวนการสร้างตารางฐานข้อมูลในรูปบรรทัดฐานขั้น Boyce/Codd จาก FD
11. หน่วยติดต่อผู้ใช้ด้านความสัพันธ์เจดี (Join Dependency Interface Unit)
12. ขบวนการสร้างตารางฐานข้อมูลในรูปบรรทัดฐานขั้นที่ห้าจาก JD
13. หน่วยสร้างภาษาสอบถามเชิงโครงสร้าง (Structure Query Language Generator)

## 4.2 ฐานข้อมูลวินโดวเมทต้าไฟล์

WMF file header		
Offset	Size	Description
0	2	Metafile type, always 1 for a disk file
2	2	Header size in words, generally 9.
4	2	Version number, hex 0300 for Windows 3
6	4	File size in words
10	2	Number of object handles used
12	4	Size in words of largest record
16	2	unused

WMF record format		
Offset	Size	Description
0	4	Size in words of the record
4	2	Function code
6	2*N	Function parameters

รูปที่ 4.2 แสดงโครงสร้างของฐานข้อมูลวินโดวเมทต้าไฟล์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.2 เป็นโครงสร้างฐานข้อมูลวินโดว์เมทต้าไฟล์ [9] ทำหน้าที่เก็บภาพกราฟฟิกทั้งหมดที่ประกอบขึ้นเป็นแบบจำลองฮิวริสติกที่วาดด้วยระบบ ER/5 ประกอบด้วยส่วนสำคัญสองส่วนดังต่อไปนี้คือ

#### 4.2.1 วินโดว์เมทต้าไฟล์เฮดเดอร์

จะถูกเก็บลงในตอนต้นของวินโดว์เมทต้าไฟล์ ซึ่งจะเป็นตัวข้อมูลที่บ่งบอกว่าเป็นไฟล์ที่เก็บรูปภาพที่วาดขึ้นด้วยโปรแกรมไมโครซอฟท์วินโดว์นั้นถูกเก็บลงบนแผ่นดิสก์หรืออยู่ในหน่วยความจำ จำนวนวัตถุที่ใช้กับรูปภาพนั้นโดยไม่รวมถึงวัตถุที่ถูกกำหนดให้ตอนเริ่มต้น (ตัวอย่างของวัตถุที่ใช้ในการวาดภาพบนวินโดว์ได้แก่ แปรงทาสี ปากกา หรือลักษณะตัวอักษรที่ใช้ เป็นต้น ซึ่งไม่เกี่ยวข้องกับคำสั่งในการวาดภาพ) ขนาดของไฟล์ ขนาดข้อมูลทีมากที่สุดของคำสั่งวาดภาพและรุ่นของตัวโปรแกรมไมโครซอฟท์วินโดว์ที่ใช้ในการวาดภาพเพื่อที่จะสามารถนำกลับมาวาดใหม่ได้อย่างถูกต้อง ดังนั้นข้อมูลในเมทต้าไฟล์เฮดเดอร์ของแบบจำลองฮิวริสติกในรูปที่ 3.5 เป็นดังนี้

Metafile type	1
Object	0
Filesize	2672
Max Record	20
Version	300

#### 4.2.2 วินโดว์เมทต้าไฟล์ข้อมูล

จะถูกเก็บลงหลังจากที่เก็บเฮดเดอร์แล้วซึ่งทั้งเฮดเดอร์และข้อมูลจะถูกเก็บลงในไฟล์เดียวกัน ลักษณะของข้อมูลจะเป็นรหัสคำสั่งที่ใช้ในการวาดภาพกราฟฟิกพร้อมด้วยพารามิเตอร์ที่เกี่ยวข้องกับคำสั่งนั้นและขนาดของเรคคอร์ดที่ใช้ในการวาดโดยแต่ละคำสั่งจะนับเป็นหนึ่งเรคคอร์ด เรียงลำดับตามการวาดก่อนและหลังจนจบ และในขณะที่มีการเปลี่ยนแปลงแก้ไขภาพแบบจำลองฮิวริสติกข้อมูลนี้ก็จะถูกเปลี่ยนแปลงแก้ไขให้สอดคล้องกันด้วยในลักษณะตอบสนองทันที (Interactive) ส่วนการแสดงผลทางจอภาพนั้นจะใช้ฟังก์ชันของวินโดว์อ่านไฟล์นี้ขึ้นมา ดังนั้นข้อมูลในเมทต้าไฟล์ข้อมูลของแบบจำลองฮิวริสติกในรูปที่ 3.5 เป็นดังนี้

Rec.#	Fn_Name	Fn_Code	Size	Parameters
1	MoveTo	02 14	5	6A 02 43 02
2	LineTo	02 13	5	69 02 4E 02
3	Ellipse	04 18	7	84 02 45 02 5C 02 01 02
4	TextOut	05 21	6	00 00 70 02 23 02
5	TextOut	05 21	9	05 00 46 6E 61 6D 65 00 68 02 0D 02 /* Fname */
6	MoveTo	02 14	5	4D 02 4D 02
7	LineTo	02 13	5	54 02 5C 02

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นโดยสำนักงานส่งเสริมการค้าในต่างประเทศ ณ นครโฮจิมินห์ เมืองโฮจิมินห์ ประเทศเวียดนาม ไม่สามารถนำออกนอกราชอาณาจักรได้

Rec #	Fn_name	Fn_Code	Size	Parameters
8	Ellipse	04 18	7	55 02 56 02 2D 02 12 02
9	TextOut	05 21	6	00 00 41 02 34 02
10	TextOut	05 21	9	05 00 4C 6E 61 6D 65 00 39 02 1E 02 /* Lname */
11	MoveTo	02 14	5	46 02 7A 02
12	LineTo	02 13	5	4F 02 7A 02
13	Ellipse	04 18	7	47 02 97 02 1F 02 5F 02
14	TextOut	05 21	6	00 00 33 02 7B 02
15	TextOut	05 21	9	05 00 4D 69 6E 69 74 00 2B 02 6B 02 /* Minit */
16	MoveTo	02 14	5	72 02 9A 02
17	LineTo	02 13	5	7D 02 B2 02
18	Ellipse	04 18	7	78 02 A8 02 50 02 4E 02
19	TextOut	05 21	6	00 00 64 02 7B 02
20	TextOut	05 21	10	07 00 45 6D 70 4E 61 6D 65 00 5C 02 5A 02 /* EmpName */
21	MoveTo	02 14	5	95 02 95 02
22	LineTo	02 13	5	95 02 AF 02
23	Ellipse	04 18	7	AA 02 96 02 82 02 60 02
24	TextOut	05 21	6	00 00 96 02 7B 02
25	TextOut	05 21	8	04 00 53 73 6E 2A 8E 02 6C 02 /* Ssn* */
26	MoveTo	02 14	5	A2 02 6D 02
27	LineTo	02 13	5	A2 02 88 02
28	MoveTo	02 14	5	BB 02 80 02
29	LineTo	02 13	5	A9 02 AF 02
30	Ellipse	04 18	7	E7 02 89 02 AF 02 2B 02
31	TextOut	05 21	12	0B 00 7B 5B 31 30 30 2C 35 30 30 5D 7D 00 CB 02 37 02 /* [(100,500) */
32	TextOut	05 21	9	06 00 53 61 6C 61 72 79 BB 02 45 02 /* Salary */
33	Rectangle	04 1B	7	AC 02 1A 03 7E 02 B0 02
34	TextOut	05 21	10	08 00 45 4D 50 4C 4F 59 45 45 8D 02 BF 02 /* EMPLOYEE */
35	MoveTo	02 14	5	5E 02 CD 02
36	LineTo	02 13	5	7D 02 DB 02
37	Ellipse	04 18	7	60 02 E7 02 28 02 9D 02
38	TextOut	05 21	11	09 00 7B 27 4D 27 2C 27 46 27 7D 00 44 02 A9 02 /* ('M','F') */
39	TextOut	05 21	8	03 00 53 05 78 00 34 02 B6 02 /* Sex */
40	MoveTo	02 14	5	47 02 0B 03
41	LineTo	02 13	5	7D 02 F1 02
42	Ellipse	04 18	7	48 02 3C 03 20 02 F0 02
43	TextOut	05 21	6	00 00 34 02 16 03
44	TextOut	05 21	10	07 00 41 64 72 65 73 73 00 2C 02 FC 02 /* Address */
45	MoveTo	02 14	5	6E 02 1D 03
46	LineTo	02 13	5	7D 02 07 03

เอกสารนี้เป็นเอกสารที่ส่งไปสำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Rec #	Fn Name	Fn Code	Size	Parameters
47	Ellipse	04 18	7	74 02 52 03 4C 02 14 03
48	TextOut	05 21	6	00 00 60 02 33 03
49	TextOut	05 21	9	05 00 42 64 61 74 65 00 58 02 20 03 /* Bdate */
50	MoveTo	02 14	5	8F 02 0F 05
51	LineTo	02 13	5	8F 02 F3 04
52	Ellipse	04 18	7	A4 02 7A 05 7C 02 10 05
53	TextOut	05 21	6	00 00 90 02 45 05
54	TextOut	05 21	11	0A 00 44 65 70 4E 75 6D 62 65 72 2A 88 02 1C 05 /* DepNumber* */
55	MoveTo	02 14	5	9C 02 2A 05
56	LineTo	02 13	5	9C 02 5F 05
57	Rectangle	04 1B	7	A6 02 F3 04 78 02 79 04
58	TextOut	05 21	11	0A 00 44 45 50 41 52 54 4D 45 4E 54 87 02 88 04 /* DEPARTMENT */
59	MoveTo	02 14	5	5F 02 E1 04
60	LineTo	02 13	5	77 02 CC 04
61	Ellipse	04 18	7	61 02 65 05 1D 02 99 04
62	Ellipse	04 18	7	5B 02 5F 05 23 02 9F 04
63	TextOut	05 21	20	1B 00 7B 27 50 61 72 69 73 27 2C 27 4C 6F 6E 64 6F 6E 27 2C 27 41 74 68 65 6E 73 27 7D 00 3F 02 4B 04 /* ('Paris','London','Athens') */
64	TextOut	05 21	12	0B 00 44 65 70 4C 6F 63 61 74 69 6F 6E 00 2F 02 D7 04 /* DepLocation */
65	MoveTo	02 14	5	C4 02 06 05
66	LineTo	02 13	5	A6 02 D8 04
67	Ellipse	04 18	7	EA 02 4C 05 C2 02 F4 04
68	TextOut	05 21	6	00 00 D6 02 20 05
69	TextOut	05 21	10	07 00 44 65 70 4E 61 6D 65 00 CE 02 00 05 /* DepName */
70	MoveTo	02 14	5	E2 02 0A 05
71	LineTo	02 13	5	E2 02 36 05
72	Polygon	03 24	12	04 00 87 03 67 02 D1 03 41 02 1B 04 67 02 D1 03 8D 02
73	TextOut	05 21	11	09 00 57 4F 52 4B 53 5F 46 4F 52 00 5F 02 A5 03 /* WORKS_FOR */
74	Polygon	03 24	12	04 00 93 03 C8 02 D2 03 A2 02 11 04 C8 02 D2 03 EE 02
75	TextOut	05 21	10	07 00 4D 41 4E 41 47 45 53 00 C0 02 B1 03 /* MANAGES */
76	MoveTo	02 14	5	7E 02 1A 03
77	LineTo	02 13	5	64 02 8B 03
78	MoveTo	02 14	5	82 02 1A 03
79	LineTo	02 13	5	69 02 8A 03
80	TextOut	05 21	7	01 00 4E 00 6B 02 4C 03 /* N */
81	MoveTo	02 14	5	79 02 78 04
82	LineTo	02 13	5	64 02 16 04
83	MoveTo	02 14	5	7D 02 78 04
84	LineTo	02 13	5	69 02 18 04

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Rec.#	Fn_Name	Fn_Code	Size	Parameters
85	TextOut	05 21	7	01 00 31 00 69 01 44 04 /* 1 */
86	MoveTo	02 14	5	8D 02 1A 03
87	LineTo	02 13	5	C6 02 95 03
88	TextOut	05 21	7	01 00 31 00 A1 02 53 03 /* 1 */
89	MoveTo	02 14	5	9C 02 78 04
90	LineTo	02 13	5	C5 02 0E 04
91	MoveTo	02 14	5	A0 02 78 04
92	LineTo	02 13	5	CA 02 0D 04
93	TextOut	05 21	7	01 00 31 00 AA 02 40 04 /* 1 */
94	MoveTo	02 14	5	AA 02 A8 03
95	LineTo	02 13	5	B3 02 B4 03
96	Ellipse	04 18	7	AE 02 BB 03 86 02 67 03
97	TextOut	05 21	6	00 00 9A 02 91 03
98	TextOut	05 21	11	09 00 53 74 61 72 74 44 61 74 65 00 92 02 73 03 /* StartDate */
99	MoveTo	02 14	5	6A 03 F9 04
100	LineTo	02 13	5	6A 03 E5 04
101	Ellipse	04 18	7	7F 03 5C 05 57 03 FA 04
102	TextOut	05 21	6	00 00 6B 03 2B 05
103	TextOut	05 21	11	0A 00 50 72 6A 4E 75 6D 62 65 72 2A 63 03 06 05 /* PrjNumber */
104	MoveTo	02 14	5	77 03 12 05
105	LineTo	02 13	5	77 03 43 05
106	Rectangle	04 1B	7	82 03 E5 04 54 03 89 04
107	TextOut	05 21	10	07 00 50 52 4F 4A 45 43 54 00 63 03 98 04 /* PROJECT */
108	MoveTo	02 14	5	95 03 FC 04
109	LineTo	02 13	5	82 03 DC 04
110	Ellipse	04 18	7	BA 03 3F 05 92 03 EF 04
111	TextOut	05 21	6	00 00 A6 03 17 05
112	TextOut	05 21	10	07 00 50 72 6A 4E 61 6D 65 00 9E 03 FB 04 /* PrjName */
113	MoveTo	02 14	5	B2 03 03 05
114	LineTo	02 13	5	B2 03 2B 05
115	MoveTo	02 14	5	9D 03 97 04
116	LineTo	02 13	5	82 03 A8 04
117	Ellipse	04 18	7	D5 03 CC 04 9D 03 42 04
118	TextOut	05 21	15	11 00 21 7B 27 55 53 41 27 2C 27 43 41 4E 41 44 41 27 7D 00 B9 03 4E 04 /* !("USA";"CANADA" */
119	TextOut	05 21	12	0B 00 50 72 6A 4C 6F 63 61 74 69 6F 6E 00 49 03 63 04 /* PrjLocation */
120	Polygon	03 24	12	04 00 73 04 FD 02 B6 04 D7 02 F9 04 FD 02 B6 04 23 03
121	TextOut	05 21	10	08 00 43 4F 4E 54 52 4F 4C 53 F5 02 91 04 /* CONTROL */
122	MoveTo	02 14	5	53 03 B4 04

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้เรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Row #	Fn Name	Fn Code	Size	Parameters
123	LineTo	02 13	5	22 03 B4 04
124	MoveTo	02 14	5	53 03 B8 04
125	LineTo	02 13	5	22 03 B8 04
126	TextOut	05 21	7	01 00 4E 00 32 03 B1 04 /* N */
127	MoveTo	02 14	5	AB 02 B7 04
128	LineTo	02 13	5	D7 02 B7 04
129	TextOut	05 21	7	01 00 31 00 B6 02 B3 04 /* 1 */
130	Polygon	03 24	12	04 00 8C 03 6C 03 D2 03 46 03 18 04 6C 03 D2 03 92 03
131	TextOut	05 21	10	08 00 57 4F 52 4B 53 5F 4F 4E 64 03 AA 03 /* WORKS_ON */
132	MoveTo	02 14	5	68 03 88 04
133	LineTo	02 13	5	69 03 15 04
134	MoveTo	02 14	5	6C 03 88 04
135	LineTo	02 13	5	6D 03 16 04
136	TextOut	05 21	7	01 00 4E 00 62 03 4C 04 /* N */
137	MoveTo	02 14	5	9D 02 1A 03
138	LineTo	02 13	5	69 03 8E 03
139	MoveTo	02 14	5	96 02 1A 03
140	LineTo	02 13	5	68 03 92 03
141	TextOut	05 21	7	01 00 4D 00 F8 02 4E 03 /* M */
142	MoveTo	02 14	5	43 03 0E 04
143	LineTo	02 13	5	56 03 F2 03
144	Ellipse	04 18	7	49 03 44 04 21 03 06 04
145	TextOut	05 21	6	00 00 35 03 25 04
146	TextOut	05 21	9	05 00 48 6F 75 72 73 00 2D 03 12 04 /* Hours */
147	Polygon	03 24	12	04 00 12 03 B8 03 6B 03 92 03 C4 03 B8 03 6B 03 DE 03
148	Polygon	03 24	12	04 00 18 03 B8 03 6B 03 98 03 BE 03 B8 03 6B 03 D8 03
149	TextOut	05 21	13	0D 00 44 45 50 45 4E 44 45 4E 54 53 5F 4F 46 00 B0 03 30 03 /* DEPENDENTS_OF */
150	Rectangle	04 1B	7	32 04 A4 03 04 04 32 03
151	Rectangle	04 1B	7	2C 04 9E 03 0A 04 38 03
152	TextOut	05 21	11	09 00 44 45 50 45 4C 44 45 4E 54 00 13 04 41 03 /* DEPENDENT */
153	MoveTo	02 14	5	03 04 67 03
154	LineTo	02 13	5	DD 03 67 03
155	MoveTo	02 14	5	03 04 6B 03
156	LineTo	02 13	5	DC 03 6B 03
157	TextOut	05 21	7	01 00 4C 00 EB 03 64 03 /* N */
158	MoveTo	02 14	5	AC 02 15 03
159	LineTo	02 13	5	92 03 6B 03
160	TextOut	05 21	7	01 00 31 00 17 03 3C 03 /* 1 */

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในหน่วยงานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

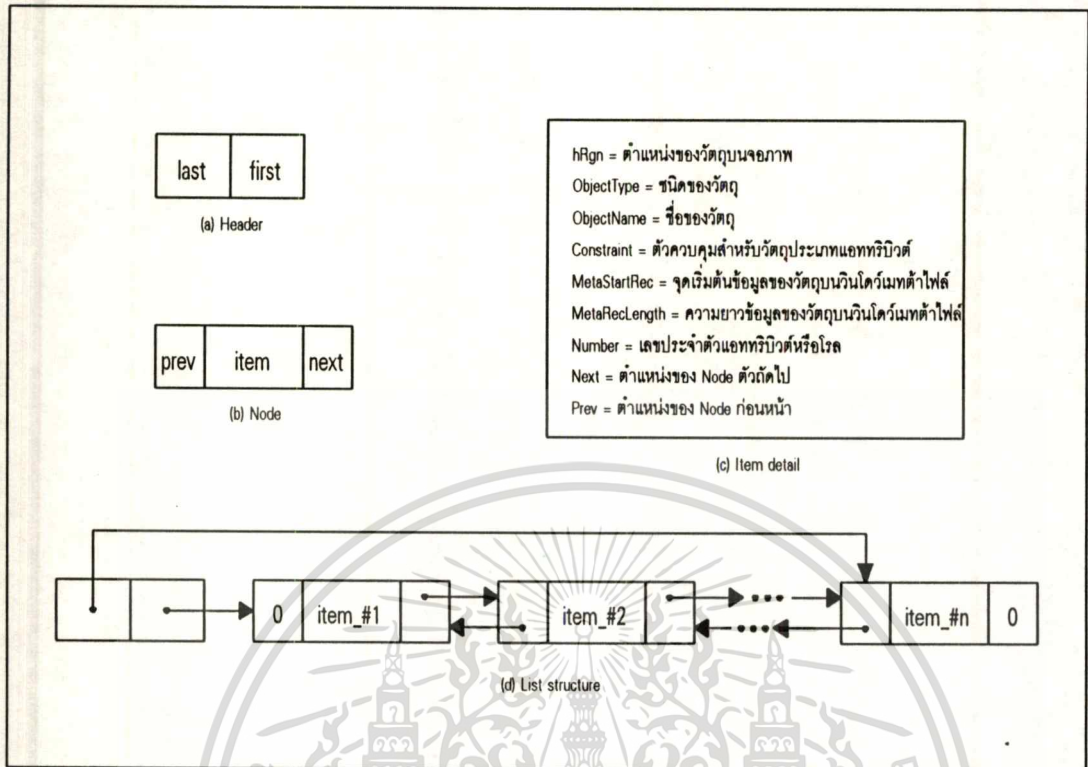
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Row #	Fn. Name	Fn. Code	Size	Parameters
161	MoveTo	02 14	5	1A 04 15 03
162	LineTo	02 13	5	1A 04 31 03
163	Ellipse	04 18	7	37 04 16 03 FF 03 CC 02
164	TextOut	05 21	11	09 00 7B 27 4D 27 2C 27 46 27 7D 00 1B 04 D8 02 /* ('M','F') */
165	TextOut	05 21	8	03 00 53 65 78 00 0B 04 E5 02 /* Sex */
166	MoveTo	02 14	5	4B 04 3D 03
167	LineTo	02 13	5	32 04 55 03
168	Ellipse	04 18	7	73 04 63 03 4B 04 F5 02
169	TextOut	05 21	6	00 00 5F 04 2C 03
170	TextOut	05 21	11	0A 00 44 65 6E 4E 61 6D 65 2A 57 04 01 03 /* DepenName* */
171	MoveTo	02 14	5	6B 04 10 03
172	LineTo	02 13	5	6B 04 47 03
173	MoveTo	02 14	5	4C 04 9E 03
174	LineTo	02 13	5	32 04 83 03
175	Ellipse	04 18	7	73 04 DC 03 4B 04 88 03
176	TextOut	05 21	6	00 00 5F 04 B2 03
177	TextOut	05 21	11	09 00 42 89 72 74 68 44 61 74 65 00 57 04 94 03 /* BirthDate */
178	MoveTo	02 14	5	25 04 DD 03
179	LineTo	02 13	5	20 04 A4 03
180	Ellipse	04 18	7	3F 04 46 04 17 04 DC 03
181	TextOut	05 21	6	00 00 2B 04 11 04
182	TextOut	05 21	12	0C 00 52 65 6C 61 74 69 6F 6E 73 68 69 70 23 04 E8 03 /* Relationship */
183	Polygon	03 24	12	04 00 5C 02 91 03 A8 02 6B 03 F4 02 91 03 A8 02 B7 03
184	TextOut	05 21	12	0B 00 53 55 50 45 52 56 49 53 49 4F 4E 00 89 03 7A 02 /* SUPERVISION */
185	MoveTo	02 14	5	AC 02 B7 02
186	LineTo	02 13	5	8D 03 63 02
187	TextOut	05 21	7	01 00 31 00 14 03 89 02 /* 1 */
188	MoveTo	02 14	5	AC 02 09 03
189	LineTo	02 13	5	8D 03 EC 02
190	TextOut	05 21	7	01 00 4E 00 14 03 F5 02 /* N */
191	TextOut	05 21	11	0A 00 73 75 70 65 72 76 69 73 6F 72 5B 03 22 02 /* supervisor */
192	TextOut	05 21	11	0A 00 73 75 70 65 72 76 69 73 65 65 5C 03 FA 02 /* supervisee */

จากข้อมูลดังกล่าวแสดงให้เห็นว่าในการวาดเส้นติดจะใช้ฟังก์ชัน Rectangle (รูปสี่เหลี่ยมผืนผ้า) การวาดความเกี่ยวพันจะใช้ฟังก์ชัน Polygon (รูปสี่เหลี่ยมขนมเปียกปูน) การวาดแอททริบิวต์จะใช้ฟังก์ชัน Ellipse (รูปวงรี) การวาดโรลจะใช้ฟังก์ชัน MoveTo ตามด้วย LineTo (เส้นตรง) และการเขียนตัวหนังสือจะใช้ฟังก์ชัน TextOut

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3 โครงสร้างข้อมูลวัตถุดับเบิลลิงคิลิสต์ (Object Double Linklist)



รูปที่ 4.3 แสดงโครงสร้างข้อมูลวัตถุดับเบิลลิงคิลิสต์

จากรูปที่ 4.3 เป็นโครงสร้างข้อมูลวัตถุดับเบิลลิงคิลิสต์ (Object Double Linklist) [13] ทำหน้าที่สำคัญอยู่ 2 ประการดังต่อไปนี้คือ

1. ใช้ในการป้องกันไม่ให้วัตถุที่ถูกวาดขึ้นทีหลังนั้นวางทับกับวัตถุที่มีอยู่เดิม โดยในการวาดวัตถุทุกครั้งจะมีการเก็บขอบเขตของวัตถุไว้ (hRgn) และเมื่อมีการวาดวัตถุใหม่ขึ้นทีหลังจะสามารถตรวจสอบได้ว่าตำแหน่งที่ต้องการวางวัตถุนั้นไม่ได้ทับกับวัตถุที่มีอยู่เดิม

2. ใช้ในการลบวัตถุออกจากแบบจำลองฮิวริสติก โดยการตรวจสอบว่าตำแหน่งเม้าท์อยู่บนขอบเขตของวัตถุที่ต้องการลบหรือไม่และเป็นวัตถุชนิดใด จะได้นำข้อมูลภายใน Item ของวัตถุนั้นไปใช้ในการค้นหาวัตถุที่สัมพันธ์กันทั้งหมดแล้วนำไปลบทิ้งจากฐานข้อมูลวินโดว์เมตาไฟล์ (ลบบรรูปภาพ) ฐานข้อมูลฮิวริสติกเมตาไฟล์ (ลบข้อมูล) และโครงสร้างข้อมูลวัตถุดับเบิลลิงคิลิสต์ (ตัวกำกับข้อมูล) ได้อย่างถูกต้องเสมือนหนึ่งว่าไม่เคยวาดวัตถุเหล่านั้นมาก่อน

จากรูปที่ 4.4 แสดงข้อมูลภายใน Item วัตถุดับเบิลลิงคิลิสต์ของแบบจำลองฮิวริสติกในรูปที่ 3.5 ทั้งหมด จะเห็นว่าในการวาดวัตถุทุกครั้งจะมีการเก็บขอบเขตของวัตถุ (hRgn) ที่เป็นตัวเลขซึ่งตัวเลขนี้เกิดขึ้นจากการสร้างขอบเขตวัตถุของวินโดว์ โดยค่าตัวเลขที่ได้นี้จะใช้เป็นพารามิเตอร์หลักในฟังก์ชันการประมวลผลเกี่ยวกับขอบเขตของวินโดว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

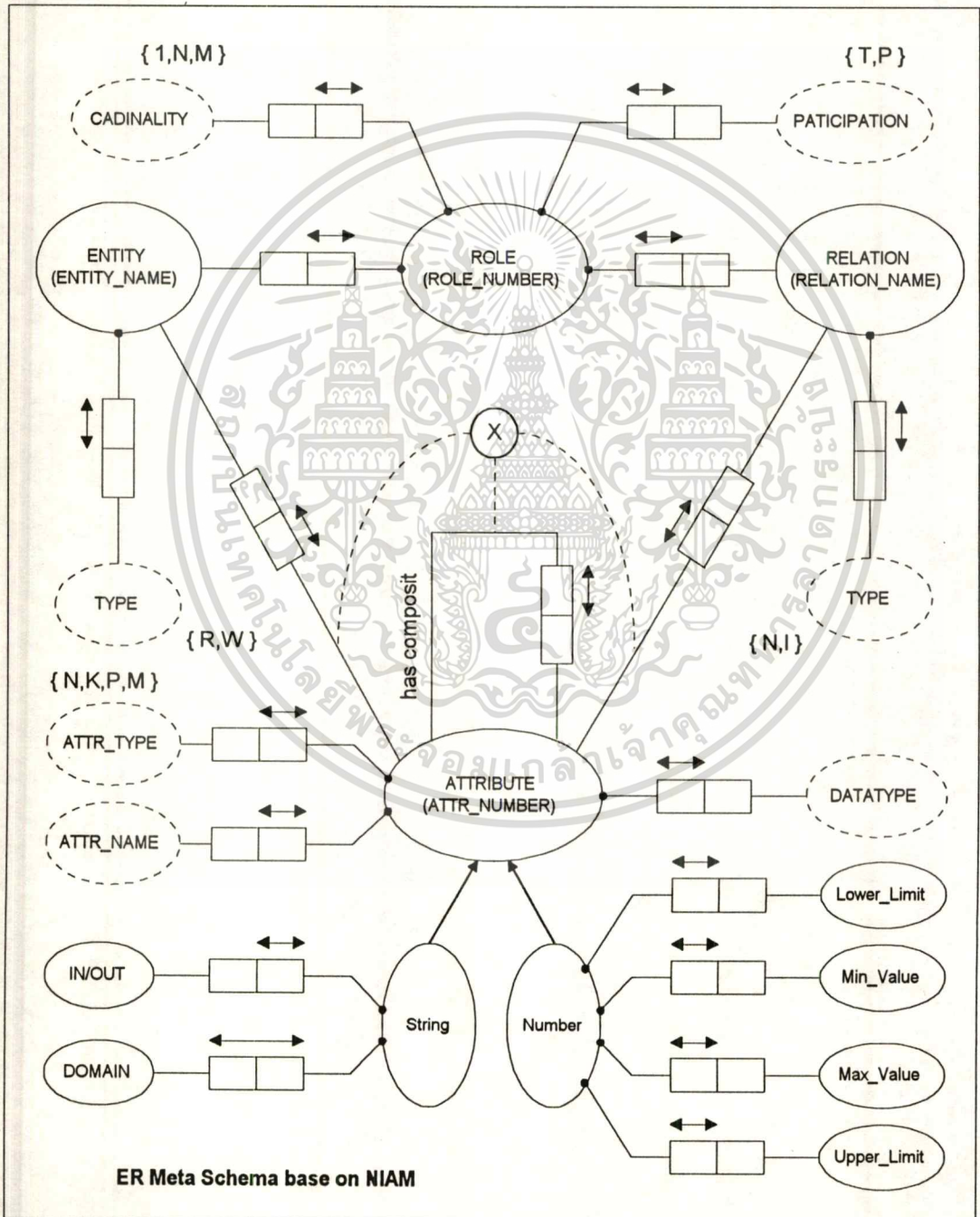
Item#	IRgn	ObjectType	ObjectName	Constraint	MetaStartRec	MetaRecLength	Number	Next	Prev
1	7278	N_ATTRIBUTE	Fname		1	5	1	3694	0
2	7302	N_ATTRIBUTE	Lname		6	5	2	3702	3710
3	7290	N_ATTRIBUTE	Minit		11	5	3	3590	3694
4	7306	N_ATTRIBUTE	EmpName		16	5	4	3678	3702
5	7310	P_ATTRIBUTE	Ssn*		21	7	5	3598	3590
6	7282	N_ATTRIBUTE	Salary	{(100,500)}	28	5	6	3686	3678
7	7146	R_ENTITY	EMPLOYEE		33	2	1	4382	3598
8	7314	N_ATTRIBUTE	Sex	{'M','F'}	35	5	7	4446	3686
9	7318	N_ATTRIBUTE	Address		40	5	8	4438	4382
10	7322	N_ATTRIBUTE	Bdate		45	5	9	4390	4446
11	7326	P_ATTRIBUTE	DepNumber*		50	7	10	4430	4438
12	7330	R_ENTITY	DEPARTMENT		57	2	1	4422	4390
13	7334	MV_ATTRIBUTE	DepLocation	{'Paris','London','Athens'}	59	6	11	4454	4430
14	7346	K_ATTRIBUTE	DepName		65	7	12	4462	4422
15	7350	N_RELATION	WORKS_FOR		72	2	1	4470	4454
16	7354	N_RELATION	MANAGES		74	2	1	4478	4462
17	7358	TOTAL	N		76	5	1	4486	4470
18	7362	TOTAL	1		81	5	2	4494	4478
19	7366	PARTIAL	1		86	3	3	4502	4486
20	7370	TOTAL	1		89	5	4	4510	4494
21	7374	N_ATTRIBUTE	StartDate		94	5	13	4518	4502
22	7378	P_ATTRIBUTE	PrjNumber*		99	7	14	4526	4510
23	1	R_ENTITY	PROJECT		106	2	1	4534	4518
24	7386	K_ATTRIBUTE	PrjName		108	7	15	4542	4526
25	7390	N_ATTRIBUTE	PrjLocation	{'USA','CANADA'}	115	5	16	4550	4534
26	7394	N_RELATION	CONTROLS		120	2	-1	4558	4542
27	7398	TOTAL	N		122	5	5	4566	4550
28	7402	PARTIAL	1		127	3	6	4574	4558
29	1	N_RELATION	WORKS_ON		130	2	1	4582	4566
30	7410	TOTAL	N		132	5	7	4590	4574
31	7414	TOTAL	M		137	5	8	4598	4582
32	7418	N_ATTRIBUTE	Hours		142	5	17	4606	4590
33	7422	I_RELATION	DEPENDENTS_OF		147	3	-1	4614	4598
34	7426	W_ENTITY	DEPENDENT		150	3	-1	4622	4606
35	7430	TOTAL	N		153	5	9	4630	4614
36	7434	PARTIAL	1		158	3	10	4638	4622
37	7438	N_ATTRIBUTE	Sex	{'M','F'}	161	5	18	4646	4630
38	7442	P_ATTRIBUTE	DepenName*		166	7	19	4654	4638
39	7446	N_ATTRIBUTE	BirthDate		173	5	20	4662	4646
40	7450	N_ATTRIBUTE	Relationship		178	5	21	4670	4654
41	7454	N_RELATION	SUPERVISION		183	2	1	4678	4662
42	7458	PARTIAL	1		185	3	11	4686	4670
43	7462	PARTIAL	N		188	3	12	4694	4678
44	7466	TEXT	supervisor		191	1	1	4702	4686
45	7470	TEXT	supervisee		192	1	1	0	4694

รูปที่ 4.4 แสดงข้อมูลภายใน Item วัตถุประสงค์ลิขสิทธิ์ของแบบจำลองอ็อบเจกต์ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

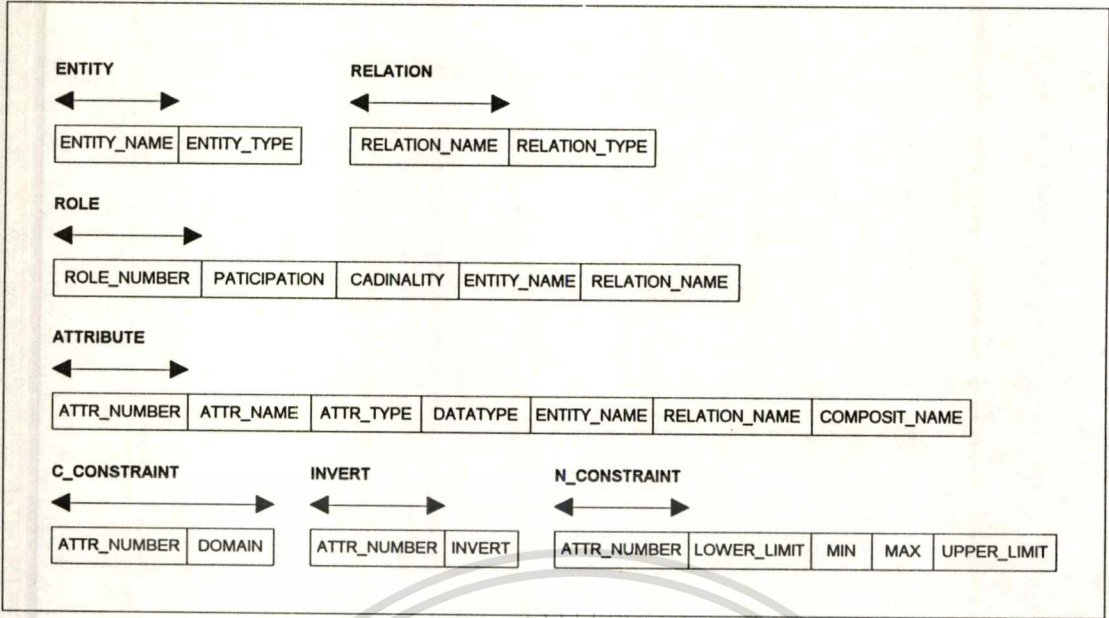
#### 4.4 ฐานข้อมูลอ็อบเจกต์

ทำหน้าที่เป็นโครงสร้างฐานข้อมูลของระบบสำหรับเก็บข้อมูลที่ได้จากแบบจำลองอ็อบเจกต์และตัวควบคุม (Constraint) ต่างๆ ดังนั้นในการวาดภาพแบบจำลองอ็อบเจกต์แต่ละครั้งรูปภาพจะถูกเก็บลงบนฐานข้อมูลวินโดวเมตาไฟล์แต่ตัวข้อมูลของแผนภาพอ็อบเจกต์จะถูกเก็บลงบนฐานข้อมูลอ็อบเจกต์เมตา จากรูปที่ 4.5 แสดงแบบจำลอง NIAM [6] ที่ออกแบบสำหรับเก็บข้อมูลอ็อบเจกต์และรูปที่ 4.6 แสดงตารางฐานข้อมูลที่ได้จากแบบจำลอง NIAM



รูปที่ 4.5 แสดงแบบจำลอง NIAM ที่ออกแบบสำหรับเก็บข้อมูลอ็อบเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษายกเว้นกรณีอื่น มิฉะนั้นผู้ใดที่นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงตารางฐานข้อมูลที่ได้จากแบบจำลอง NIAM

ดังนั้นข้อมูลของแบบจำลองอ็อบเจกต์ในรูปที่ 3.5 จะถูกเก็บลงบนฐานข้อมูลอ็อบเจกต์เมทาด้า

ดังนั้น

ENTITY_NAME	ENTITY_TYPE
DEPARTMENT	0
DEPENDENT	1
EMPLOYEE	0
PROJECT	0

RELATION_NAME	RELATION_TYPE
CONTROLS	2
DEPENDENTS_OF	3
MANAGES	2
SUPERVISION	2
WORKS_FOR	2
WORKS_ON	2

ROLE_NUMBER	PATICIPATION	CADINALITY	ENTITY_NAME	RELATION_NAME
1	1	1	EMPLOYEE	WORKS_FOR
2	1	0	DEPARTMENT	WORKS_FOR
3	0	0	EMPLOYEE	MANAGES
4	1	0	DEPARTMENT	MANAGES
5	1	1	PROJECT	CONTROLS
6	0	0	DEPARTMENT	CONTROLS
7	1	1	PROJECT	WORKS_ON
8	1	1	EMPLOYEE	WORKS_ON
9	1	1	DEPENDENT	DEPENDENTS_OF
10	0	0	EMPLOYEE	DEPENDENTS_OF
11	0	0	EMPLOYEE	SUPERVISION
12	0	1	EMPLOYEE	SUPERVISION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ATTR_ NUMBER	ATTR_ NAME	ATTR_ TYPE	ATTR_ DATATYPE	ENTITY_ NAME	RELATION_ NAME	COMPOSIT_ NUMBER
1	Fname	4	CHAR(1)	NULL	NULL	4
2	Lname	4	CHAR(10)	NULL	NULL	4
3	Minit	4	CHAR(1)	NULL	NULL	4
4	EmpName	4	CHAR(30)	EMPLOYEE	NULL	-1
5	Ssn	6	NUMBER	EMPLOYEE	NULL	-1
6	Salary	4	NUMBER	EMPLOYEE	NULL	-1
7	Sex	4	CHAR(1)	EMPLOYEE	NULL	-1
8	Address	4	CHAR(30)	EMPLOYEE	NULL	-1
9	Bdate	4	DATE	EMPLOYEE	NULL	-1
10	DepNumber	6	NUMBER	DEPARTMENT	NULL	-1
11	DepLocation	7	CHAR(30)	DEPARTMENT	NULL	-1
12	DepName	5	CHAR(30)	DEPARTMENT	NULL	-1
13	StartDate	4	DATE	NULL	MANAGES	-1
14	PrjNumber	6	NUMBER	PROJECT	NULL	-1
15	PrjName	5	CHAR(30)	PROJECT	NULL	-1
16	PrjLocation	4	CHAR(30)	PROJECT	NULL	-1
17	Hours	4	DATE	NULL	WORKS_ON	-1
18	Sex	4	CHAR(1)	DEPENDENT	NULL	-1
19	DepenName	6	CHAR(30)	DEPENDENT	NULL	-1
20	BirthDate	4	DATE	DEPENDENT	NULL	-1
21	Relationship	4	CHAR(30)	DEPENDENT	NULL	-1

ATTR_ NUMBER	DOMAIN_ NAME
7	M
7	F
11	Paris
11	London
11	Athens
16	USA
16	CANADA
18	M
18	F

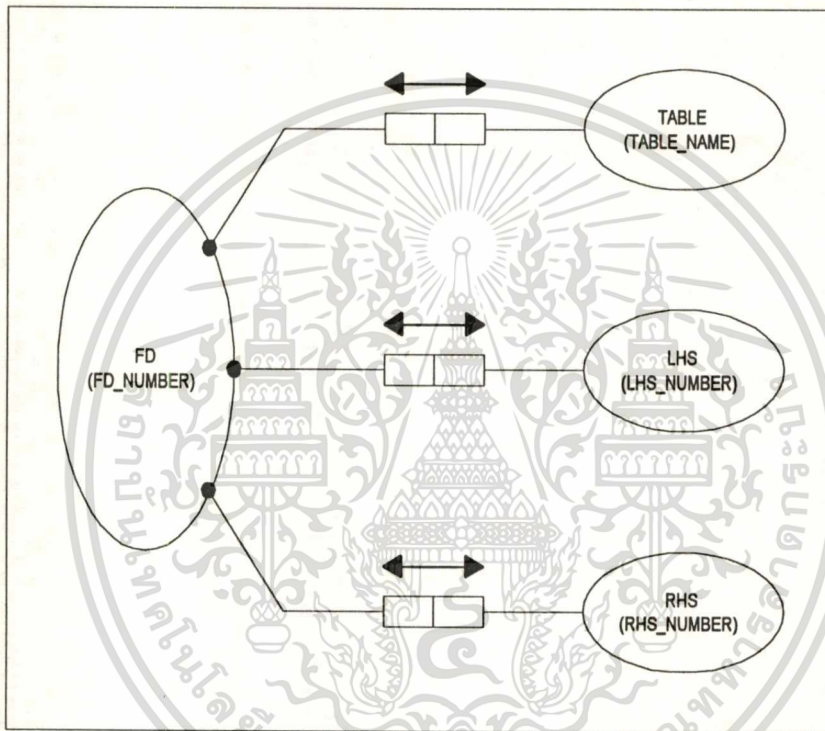
ATTR_ NUMBER	INVERT
7	N
11	N
16	Y
18	N

ATTR_ NUMBER	LOWER_ LIMIT	MIN	MAX	UPPER_ LIMIT
6	[	100	500	]

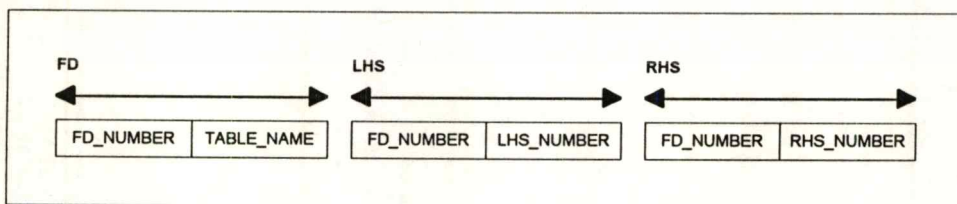
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.5 ฐานข้อมูลเฟดดิเมทต้า

ทำหน้าที่เป็นโครงสร้างฐานข้อมูลของระบบสำหรับเก็บข้อมูลเฟดดิ ซึ่งข้อมูลเฟดดิจะ  
ได้จากขบวนการสร้างตารางฐานข้อมูลในรูปบรรทัดฐานขั้นแรกจากแบบจำลองอีอาร์ในตอน  
ต้นและผู้ใช้งานสามารถที่จะเปลี่ยนแปลงหรือแก้ไขเฟดดิดังกล่าวได้อีกเพื่อนำไปใช้เป็นข้อมูล  
ในการประมวลผลให้ได้ตารางฐานข้อมูลในรูปบรรทัดฐานขั้น Boyce/Codd ต่อไป จากรูปที่  
4.7 แสดงแบบจำลอง NIAM ที่ออกแบบสำหรับเก็บข้อมูลเฟดดิและรูปที่ 4.8 แสดงตารางฐาน  
ข้อมูลที่ได้จากแบบจำลอง NIAM



รูปที่ 4.7 แสดงแบบจำลอง NIAM ที่ออกแบบสำหรับเก็บข้อมูลเฟดดิ



รูปที่ 4.8 แสดงตารางฐานข้อมูลที่ได้จากแบบจำลอง NIAM

ดังนั้นข้อมูลของเฟดดิเริ่มต้นที่ได้จากขบวนการสร้างตารางฐานข้อมูลในรูป  
บรรทัดฐานขั้นแรกจากแบบจำลองอีอาร์ในรูปที่ 3.5 (ข้อมูลนี้ยังไม่ถูกเปลี่ยนแปลงแก้ไขจากผู้  
ใช้งาน) จะถูกเก็บลงบนฐานข้อมูลเฟดดิเมทต้าดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FD_NUMBER	TABLE_NAME
1	EMPLOYEE
2	EMPLOYEE
3	EMPLOYEE
4	EMPLOYEE
5	EMPLOYEE
6	EMPLOYEE
7	EMPLOYEE
8	EMPLOYEE
9	EMPLOYEE
10	EMPLOYEE
11	DEPARTMENT
12	DEPARTMENT
13	DEPARTMENT
14	DEPARTMENT
15	PROJECT
16	PROJECT
17	PROJECT
18	PROJECT
19	DEPENDENT
20	DEPENDENT
21	DEPENDENT
22	DEPENDENT
23	DEPENDENT
24	DepLocation
25	DepLocation
26	WORKS_ON
27	WORKS_ON
28	WORKS_ON

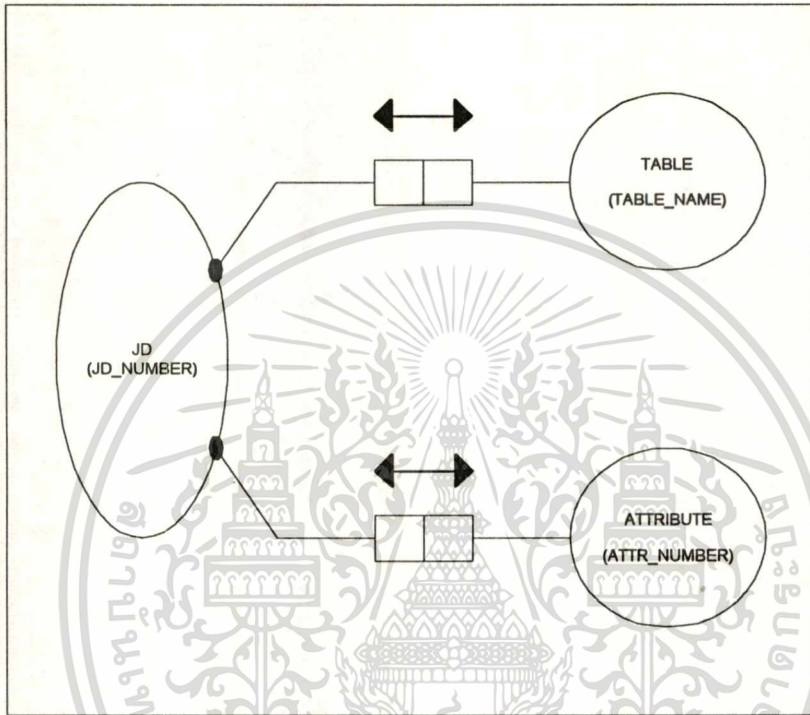
FD_NUMBER	LHS_NUMBER
1	5
2	5
3	5
4	5
5	5
6	5
7	5
8	5
9	5
10	5
11	10
12	10
13	10
14	10
15	14
16	14
17	14
18	14
19	19
19	22
20	19
20	22
21	19
21	22
22	19
22	22
23	19
23	22
24	10
24	11
25	10
25	11
26	5
26	14
27	5
27	14
28	5
28	14

FD_NUMBER	RHS_NUMBER
1	1
2	2
3	3
4	5
5	6
6	7
7	8
8	9
9	10
10	23
11	5
12	10
13	12
14	13
15	10
16	14
17	15
18	16
19	18
20	19
21	20
22	21
23	22
24	10
25	11
26	5
27	14
28	17

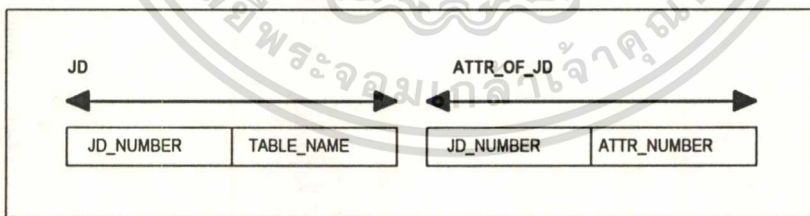
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.6 ฐานข้อมูลเจตีมेटต้า

ทำหน้าที่เป็นโครงสร้างฐานข้อมูลของระบบสำหรับเก็บข้อมูลที่ได้จากการกำหนดความสัมพันธ์เจตีมेटต้า (Join Dependency) จากผู้ใช้งาน เพื่อนำไปใช้ในการประมวลผลให้ได้ฐานข้อมูลขั้นที่ห้า จากรูปที่ 4.9 แสดง แบบจำลอง NIAM ที่ออกแบบสำหรับเก็บข้อมูลเจตีมेटต้าและรูปที่ 4.10 แสดงตารางฐานข้อมูลที่ได้จากแบบจำลอง NIAM



รูปที่ 4.9 แสดงแบบจำลอง NIAM ที่ออกแบบสำหรับเก็บข้อมูลเจตีมेटต้า

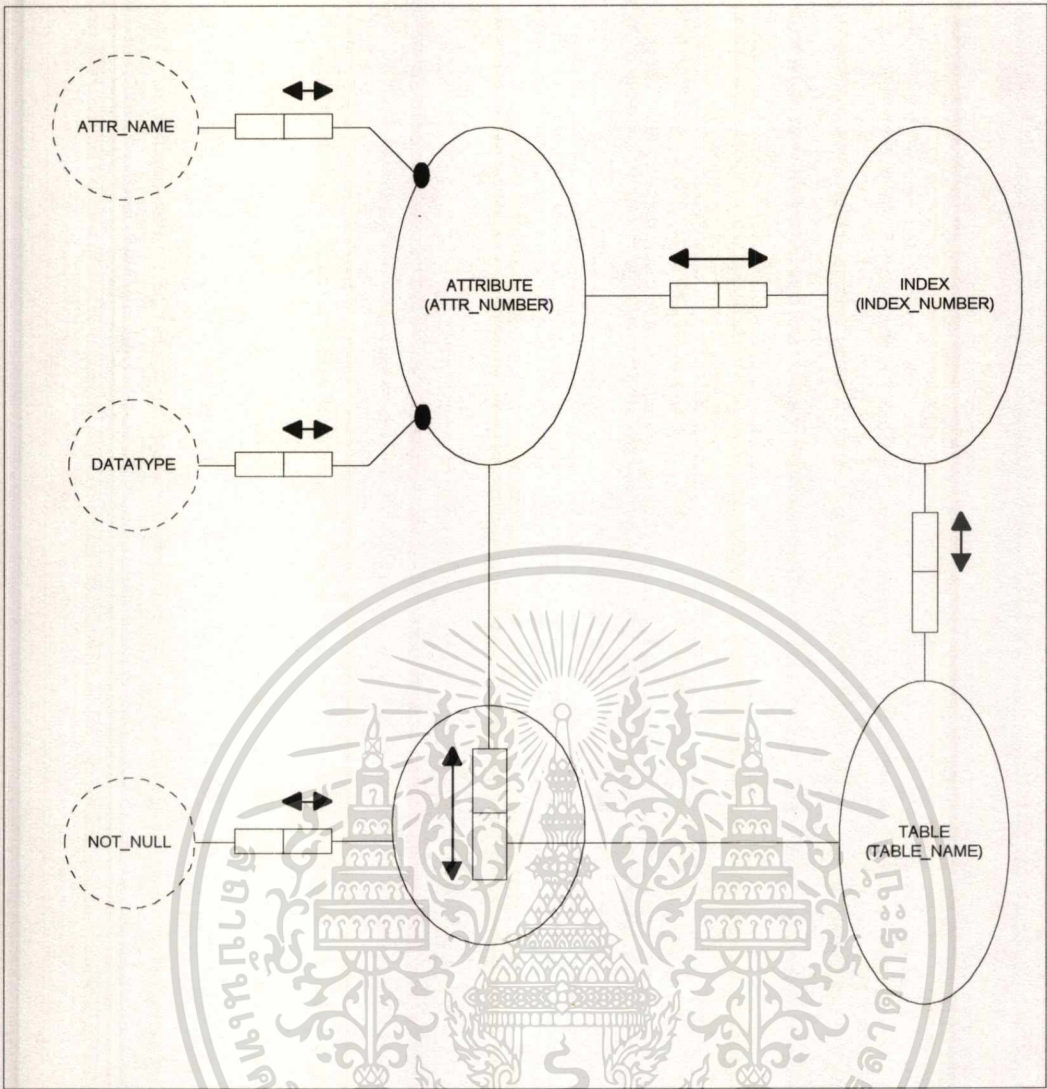


รูปที่ 4.10 แสดงตารางฐานข้อมูลที่ได้จากแบบจำลอง NIAM

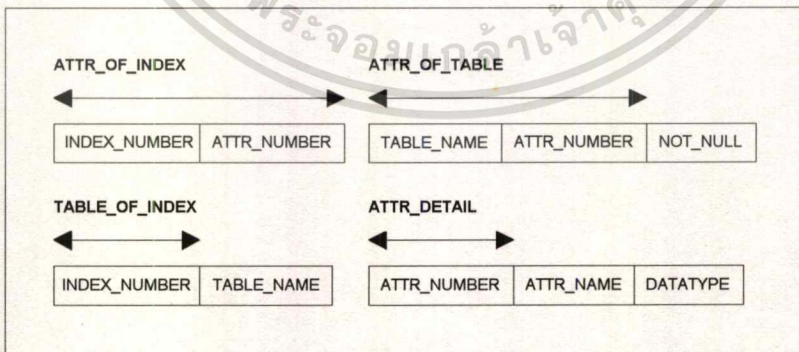
#### 4.7 ฐานข้อมูลตารางและอินเด็กซ์เมทต้า

ทำหน้าที่เป็นโครงสร้างฐานข้อมูลของระบบสำหรับเก็บข้อมูลที่ได้จากการขั้นตอนการสร้างตารางต่างๆ เพื่อนำไปใช้ในขั้นตอนการสร้างเป็นภาษาสอบถามเชิงโครงสร้างต่อไป จากรูปที่ 4.11 แสดงแบบจำลอง NIAM ที่ออกแบบสำหรับเก็บข้อมูลตารางและอินเด็กซ์เมทต้าซึ่งจะสามารถเก็บโครงสร้างของตารางไว้ได้ทั้งหมดและรูปที่ 4.12 แสดงตารางฐานข้อมูลที่ได้จากแบบจำลอง NIAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 แสดงแบบจำลอง NIAM ที่ออกแบบสำหรับเก็บข้อมูลตารางและอินเด็กซ์



รูปที่ 4.12 แสดงตารางฐานข้อมูลที่ได้จากแบบจำลอง NIAM

ดังนั้นข้อมูลในฐานข้อมูลตารางและอินเด็กซ์เมทาด้าซึ่งได้จากขั้นตอนการจากขบวนการสร้างตารางฐานข้อมูลในรูปแบบรหัสฐานขั้นแรกจากแบบจำลองอ็อบเจกต์ในรูปที่ 3.5 เป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE_NAME	ATTR_NUMBER	NOT_NULL
DEPARTMENT	10	Y
DEPARTMENT	12	Y
DEPARTMENT	5	Y
DEPARTMENT	13	N
DEPENDENT	22	Y
DEPENDENT	18	N
DEPENDENT	21	N
DEPENDENT	20	N
DEPENDENT	19	Y
DepLocation	11	Y
DepLocation	10	Y
EMPLOYEE	5	Y
EMPLOYEE	2	N
EMPLOYEE	1	N
EMPLOYEE	23	N
EMPLOYEE	10	Y
EMPLOYEE	6	N
EMPLOYEE	7	N
EMPLOYEE	8	N
EMPLOYEE	9	N
EMPLOYEE	3	N
PROJECT	14	Y
PROJECT	15	Y
PROJECT	10	N
PROJECT	16	N
WORKS_ON	14	Y
WORKS_ON	5	Y
WORKS_ON	17	N

ATTR_NUMBER	ATTR_NAME	DATATYPE
1	Fname	CHAR(1)
2	Lname	CHAR(10)
3	Minit	CHAR(1)
4	EmpName	CHAR(30)
5	Ssn	NUMBER
6	Salary	NUMBER
7	Sex	CHAR(1)
8	Address	CHAR(30)
9	Bdate	DATE
10	DepNumber	NUMBER
11	DepLocation	CHAR(30)
12	DepName	CHAR(30)
13	StartDate	DATE
14	PrijNumber	NUMBER
15	PrijName	CHAR(30)
16	PrijLocation	CHAR(30)
17	Hours	DATE
18	Sex	CHAR(1)
19	DepenName	CHAR(30)
20	BirthDate	DATE
21	Relationship	CHAR(30)
22	Ssn	NUMBER
23	Ssn23	NUMBER

INDEX_NUMBER	ATTR_NUMBER
1	5
2	10
3	14
4	22
4	19
5	11
5	10
6	5
6	14

INDEX_NUMBER	TABLE_NAME
1	EMPLOYEE
2	DEPARTMENT
3	PROJECT
4	DEPENDENT
5	DepLocation
6	WORKS_ON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.8 หน่วยติดต่อผู้ใช้ด้านกราฟฟิก

ทำหน้าที่รับข้อมูลเป็นกราฟฟิกในรูปของแบบจำลองของอีอาร์จากผู้ใช้งานผ่านทางไมโครซอฟท์วินโดวส์ ซึ่งในหน่วยนี้จะมีเครื่องมือที่แทนสัญลักษณ์ต่างๆที่ใช้ในแบบจำลองอีอาร์สำหรับช่วยในการวาดจำนวนมากให้ใช้ทำให้สามารถวาดภาพแบบจำลองอีอาร์ได้โดยสะดวกและยังสามารถที่จะพิมพ์ภาพแบบจำลองอีอาร์ออกทางเครื่องพิมพ์หรือจัดเก็บภาพแบบจำลองอีอาร์ลงบนแผ่นดิสก์เพื่อที่จะนำมาแก้ไขปรับปรุงได้ในภายหลัง ข้อมูลรูปภาพจะถูกเก็บลงในฐานข้อมูลวินโดวส์เมทาดาไฟล์และข้อมูลของแผนภาพอีอาร์เก็บลงในฐานข้อมูลอีอาร์เมทาดาในลักษณะตอบสนองทันที (Interactive)

## 4.9 ขบวนการสร้างตารางฐานข้อมูลในรูปบรรทัดฐานขั้นแรกจากแบบจำลองอีอาร์

ทำหน้าที่แปลงข้อมูลในฐานข้อมูลอีอาร์เมทาดาให้เป็นตารางฐานข้อมูลในรูปบรรทัดฐานขั้นแรก โดยวิธีประมวลผลจะทำการประมวลผลตามหลักการสร้างตารางจากแผนภาพอีอาร์ที่กล่าวมาแล้วในบทที่ 3 ซึ่งขั้นตอนการทำงานโดยส่วนใหญ่จะเป็นการประมวลผลข้อมูลด้วยคำสั่ง SQL และได้เพิ่มเติมในส่วนของ การตรวจสอบความถูกต้องของแผนภาพอีอาร์และส่วนเชื่อมต่อตัวควบคุม ผลลัพธ์ที่ได้จะถูกเก็บลงในฐานข้อมูลตารางและอินเด็กซ์เมทาดาซึ่งจะถูกส่งไปยังหน่วยสร้างภาษาสอบถามเชิงโครงสร้างเพื่อให้ได้ผลลัพธ์ในภาษา SQL ซึ่งสามารถนำไปใช้งานได้ นอกจากนี้ในขบวนการนี้ยังสร้างความสัมพันธ์ระหว่างแอททริบิวต์เบื้องต้นเพื่อเตรียมนำไปใช้งานในหน่วยติดต่อผู้ใช้ด้านความสัมพันธ์ระหว่างแอททริบิวต์

## 4.10 หน่วยติดต่อผู้ใช้ด้านความสัมพันธ์ระหว่างแอททริบิวต์

ทำหน้าที่ติดต่อกับผู้ใช้งานเพื่อรับข้อมูลเพิ่มเติมและแก้ไขในส่วนของความสัมพันธ์ระหว่างแอททริบิวต์ (FD.) ที่ถูกสร้างขึ้นจากขบวนการสร้างตารางฐานข้อมูลในรูปบรรทัดฐานขั้นแรกจากแบบจำลองอีอาร์ แล้วนำไปประมวลผลเพื่อสร้างตารางฐานข้อมูลในรูปบรรทัดฐานขั้น Boyce/Codd

## 4.11 ขบวนการสร้างตารางฐานข้อมูลในรูปบรรทัดฐานขั้น Boyce/Codd

ทำหน้าที่ประมวลผลข้อมูลในฐานข้อมูลเอพดีเมทาดาให้เป็นตารางฐานข้อมูลในรูปบรรทัดฐานขั้น Boyce/Codd ซึ่งขั้นตอนการทำงานโดยส่วนใหญ่จะเป็นการประมวลผลด้วยคำสั่ง SQL โดยจะทำหน้าที่กำจัดความซ้ำซ้อนของความสัมพันธ์ระหว่างแอททริบิวต์ที่เกิดขึ้นทั้งหมดผลลัพธ์ที่ได้จะถูกเก็บลงในฐานข้อมูลตารางและอินเด็กซ์เมทาดาซึ่งจะถูกส่งไปยังหน่วยสร้างภาษาสอบถามเชิงโครงสร้างเพื่อให้ได้ผลลัพธ์ในภาษา SQL ซึ่งสามารถนำไปใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.12 หน่วยติดต่อผู้ใช้ด้านความสัมพันธ์เจดี

ทำหน้าที่ค้นหาว่าตารางฐานข้อมูลในชั้น Boyce/Codd ที่ได้มานั้นตารางใดที่ยังมีความเป็นไปได้ที่จะมีความสัมพันธ์เจดีแอบแฝงอยู่ จากนั้นก็จะติดต่อกับผู้ใช้งานเพื่อรับข้อมูลเพิ่มเติมในส่วนของความสัมพันธ์เจดีแล้วเก็บข้อมูลลงบนฐานข้อมูลเจดีเมทต้า

#### 4.13 ขบวนการสร้างตารางฐานข้อมูลในรูปบรรทัดฐานขั้นที่ห้า

ทำหน้าที่ประมวลผลข้อมูลในฐานข้อมูลเจดีเมทต้าให้เป็นตารางฐานข้อมูลในรูปบรรทัดฐานขั้นที่ห้า ซึ่งขั้นตอนการทำงานโดยส่วนใหญ่จะเป็นการประมวลผลด้วยคำสั่ง SQL โดยจะทำหน้าที่ตรวจสอบความถูกต้องของข้อมูลเจดีและผลลัพธ์ที่ได้จะถูกเก็บลงในฐานข้อมูลตารางและอินเด็กซ์เมทต้าซึ่งจะถูกส่งไปยังหน่วยสร้างภาษาสอบถามเชิงโครงสร้างเพื่อให้ได้ผลลัพธ์ในภาษา SQL ซึ่งสามารถนำไปใช้งานได้ต่อไป

#### 4.14 หน่วยสร้างภาษาสอบถามเชิงโครงสร้าง

ทำหน้าที่สร้างภาษาสอบถามเชิงโครงสร้างในส่วนของ การสร้างโครงสร้างฐานข้อมูล (Data Definition Language) [7] โดยนำข้อมูลจากฐานข้อมูลตารางและอินเด็กซ์เมทต้าร่วมกับฐานข้อมูลตัวควบคุมในอ็อบเจกต์เมทต้ามาประมวลผล ซึ่งการประมวลผลโดยส่วนใหญ่จะเป็นคำสั่ง SQL ผลลัพธ์ที่ได้ คือคำสั่งภาษา SQL ที่ใช้ในการสร้างตารางและวิว (ในกรณีที่ตารางนั้นมีตัวควบคุมกำกับอยู่) พร้อมนำไปใช้งานกับระบบจัดการฐานข้อมูลออบเจกต์หรือตัวจัดการฐานข้อมูลแบบรีเลชันแนลอื่นๆ ดังแสดงในรูปที่ 4.13

```
CREATE TABLE DEPARTMENT
(
    Ssn          NUMBER          NOT NULL,
    DepNumber    NUMBER          NOT NULL,
    DepName      CHAR(30)       NOT NULL,
    StartDate    DATE
);
CREATE TABLE DEPENDENT
(
    Sex          CHAR(1),
    DepenName    CHAR(30)       NOT NULL,
    BirthDate    DATE,
    Relationship  CHAR(30),
    Ssn          NUMBER          NOT NULL
);
CREATE TABLE DepLocation
(
    DepNumber    NUMBER          NOT NULL,
    DepLocation  CHAR(30)       NOT NULL
);
CREATE TABLE EMPLOYEE
(
    Fname        CHAR(1),
    Lname        CHAR(10),
    Minit        CHAR(1),
    Ssn          NUMBER          NOT NULL,
    Salary       NUMBER,
    Sex          CHAR(1),
    Address      CHAR(30),
```

```

        Bdate          DATE,
        DepNumber     NUMBER      NOT NULL,
        Ssn23         NUMBER
    );
CREATE TABLE PROJECT
(
        DepNumber     NUMBER,
        PrjNumber     NUMBER      NOT NULL,
        PrjName       CHAR(30)    NOT NULL,
        PrjLocation   CHAR(30)
    );
CREATE TABLE WORKS_ON
(
        Ssn           NUMBER      NOT NULL,
        PrjNumber     NUMBER      NOT NULL,
        Hours        DATE
    );
CREATE UNIQUE INDEX INDEX_#1 ON EMPLOYEE (Ssn) ;
CREATE UNIQUE INDEX INDEX_#2 ON DEPARTMENT (DepNumber) ;
CREATE UNIQUE INDEX INDEX_#3 ON PROJECT (PrjNumber) ;
CREATE UNIQUE INDEX INDEX_#4 ON DEPENDENT (DepenName,Ssn) ;
CREATE UNIQUE INDEX INDEX_#5 ON DepLocation (DepNumber,DepLocation) ;
CREATE UNIQUE INDEX INDEX_#6 ON WORKS_ON (Ssn,PrjNumber) ;

CREATE VIEW DEPENDENT_VIEW AS SELECT *
FROM DEPENDENT
WHERE Sex IN ('F','M')
WITH CHECK OPTION ;

CREATE VIEW DepLocation_VIEW AS SELECT *
FROM DepLocation
WHERE DepLocation IN ('Athens','London','Paris')
WITH CHECK OPTION ;

CREATE VIEW EMPLOYEE_VIEW AS SELECT *
FROM EMPLOYEE
WHERE Sex IN ('F','M') AND ((Salary >= 100 AND Salary <= 500))
WITH CHECK OPTION ;

CREATE VIEW PROJECT_VIEW AS SELECT *
FROM PROJECT
WHERE PrjLocation NOT IN ('CANADA','USA')
WITH CHECK OPTION ;

```

รูปที่ 4.13 แสดงผลลัพธ์ที่ได้จากหน่วยสร้างภาษาสอบถามเชิงโครงสร้าง

## บทที่ 5

### การพัฒนาระบบ ER/5

โดยปกติในการพัฒนาโปรแกรมขึ้นใช้งานนั้น จำเป็นจะต้องมีเครื่องมือต่างๆที่ใช้ในการพัฒนาเพื่อให้สามารถสร้างโปรแกรมการทำงานได้ตามที่ต้องการ สำหรับการพัฒนาระบบ ER/5 นี้ใช้เครื่องมือหลายชนิดทั้งฮาร์ดแวร์และซอฟต์แวร์ ดังนั้นในบทนี้จะกล่าวถึงอุปกรณ์ที่ใช้ในการพัฒนาระบบ ER/5 และวิธีการพัฒนาโปรแกรมรวมถึงหน่วยโปรแกรมสำคัญต่างๆ

#### 5.1 อุปกรณ์ที่ใช้ในการพัฒนาระบบ ER/5

##### - เครื่องไมโครคอมพิวเตอร์ IBM PC

เครื่องไมโครคอมพิวเตอร์ IBM PC คอมแพคทีเบิลใช้เป็นฮาร์ดแวร์หลักในการพัฒนาระบบ ER/5 ประกอบด้วยหน่วยประมวลผลกลาง (Central Processing Unit) ขนาด 32 บิตของบริษัทอินเทล เบอร์ 80486 DX-33 หน่วยความจำขนาด 16 เมกกะไบต์ ฮาร์ดดิสก์ขนาด 345 เมกกะไบต์ ฟลอปปีดิสก์ขนาด 1.44 และ 1.2 เมกกะไบต์ อย่างละ 1 ตัว จอสี Super VGA (1024 X 768) 256 Color ของ Philips เป็นพิมพ์ 101 คีย์ ไมโครซอฟท์เมาท์ 1 ตัว ต่อเข้ากับระบบอินพุตเอาต์พุตแบบอนุกรมตามมาตรฐาน RS-232 และเครื่องพิมพ์เลเซอร์ HEWLETT PACKARD Laserjet 4L อีก 1 เครื่อง

##### - ระบบปฏิบัติการ MS-DOS

ระบบปฏิบัติการ MS-DOS ใช้เป็นซอฟต์แวร์ระบบปฏิบัติการบนเครื่อง IBM PC ผลิตโดยบริษัทไมโครซอฟท์โดยรุ่นที่ใช้ในการพัฒนาระบบ ER/5 นี้เป็นระบบปฏิบัติการ MS-DOS รุ่น 5.0

##### - โปรแกรมไมโครซอฟท์วินโดวส์

โปรแกรมไมโครซอฟท์วินโดวส์ [14] ใช้เป็นซอฟต์แวร์ระบบจัดการกราฟิกบนเครื่อง IBM PC ผลิตโดยบริษัทไมโครซอฟท์ซึ่งอยู่บนระบบปฏิบัติการ MS-DOS โดยรุ่นที่ใช้ในการพัฒนาระบบ ER/5 นี้เป็นโปรแกรมไมโครซอฟท์วินโดวส์รุ่น 3.1 ไทย

##### - โปรแกรมไมโครซอฟท์ Visual C++ สำหรับวินโดวส์

โปรแกรมไมโครซอฟท์ Visual C++ สำหรับวินโดวส์ ใช้เป็นซอฟต์แวร์แปลภาษา C ที่เขียนขึ้นเพื่อให้สามารถวิ่งบนไมโครซอฟท์วินโดวส์ได้ ผลิตโดยบริษัทไมโครซอฟท์โดยรุ่นที่ใช้ในการพัฒนาระบบ ER/5 นี้เป็นโปรแกรมไมโครซอฟท์ Visual C++ สำหรับวินโดวส์รุ่น 1.0

##### - ระบบจัดการฐานข้อมูลออราเคิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจัดการฐานข้อมูลออรากเคิลใช้เป็นซอฟต์แวร์ระบบจัดการฐานข้อมูลสำหรับเครื่อง IBM PC ผลิตโดยบริษัทออรากเคิลโดยรุ่นที่ใช้ในการพัฒนาระบบ ER/5 นี้เป็นระบบจัดการฐานข้อมูลออรากเคิลรุ่น 6.0 (รุ่นก่อนหน้านี้อาจไม่สามารถวิ่งบนไมโครซอฟท์วินโดวส์ได้)

- ตัวแปลภาษาขั้นต้น Pro\*C สำหรับวินโดวส์

ตัวแปลภาษาขั้นต้น (Precompiler) Pro\*C สำหรับวินโดวส์ ใช้เป็นซอฟต์แวร์ในการแปลคำสั่งภาษาสอบถามเชิงโครงสร้าง (SQL) ที่ฝังตัวอยู่กับภาษา C ในการติดต่อกับระบบจัดการฐานข้อมูลออรากเคิลให้เป็นภาษา C ทั้งหมดพร้อมที่จะนำไปผ่านการแปลภาษาด้วยโปรแกรมไมโครซอฟท์ Visual C++ สำหรับวินโดวส์ให้เป็นภาษาเครื่องที่สามารถทำงานได้ในที่สุด ซึ่งโปรแกรมไมโครซอฟท์ Visual C++ สำหรับวินโดวส์จะรู้จักแต่คำสั่งของภาษา C เท่านั้น สำหรับตัวแปลภาษาขั้นต้น Pro\*C สำหรับวินโดวส์ ผลิตโดยบริษัทออรากเคิลและที่ใช้ในการพัฒนาระบบ ER/5 นี้เป็นตัวแปลภาษาขั้นต้น Pro\*C สำหรับวินโดวส์ รุ่น 6.0

## 5.2 การเขียนโปรแกรมระบบ ER/5 บนไมโครซอฟท์วินโดวส์

### 5.2.1 ความแตกต่างของการเขียนโปรแกรมบนดอสกับไมโครซอฟท์วินโดวส์

จากการเขียนโปรแกรมบนดอสตั้งแต่รุ่นแรกๆจนถึงปัจจุบันนี้จุดหนึ่งที่ดอสไม่ได้ถูกพัฒนาเพิ่มเติมขึ้นมาเลยนั่นคือ การสนับสนุนระบบกราฟฟิกและระบบติดต่อกับผู้ใช้งานแบบกราฟฟิก ซึ่งในปัจจุบันนี้มีการใช้งานโปรแกรมทางด้านกราฟฟิกมากขึ้น ดังนั้นถึงเวลาแล้วที่จะต้องเลิกเขียนโปรแกรมกราฟฟิกบนดอสแล้วหันมาเขียนโปรแกรมบนไมโครซอฟท์วินโดวส์ซึ่งมีความสามารถมากกว่าและเป็นมาตรฐานเดียวกัน ความสามารถเด่นๆของไมโครซอฟท์วินโดวส์พอสรุปได้ดังนี้คือ

- ระบบติดต่อผู้ใช้

จุดประสงค์หลักอันหนึ่งของการออกแบบโปรแกรมไมโครซอฟท์วินโดวส์ก็เพื่อที่จะให้ผู้ใช้งานสามารถใช้งานแอปพลิเคชันหลายๆแอปพลิเคชันได้พร้อมกันรวมทั้งให้ผู้ใช้งานสามารถมองเห็นการทำงานของทุกๆแอปพลิเคชันได้อีกด้วย ดังนั้นแต่ละแอปพลิเคชันจึงต้องแบ่งปันจอภาพเพื่อแสดงผลต่อผู้ใช้งานโดยทั่วหน้า ใหญ่บ้าง เล็กบ้าง ทับกันบ้างตามแต่ผู้ใช้งานต้องการและสำหรับแอปพลิเคชันที่ยังไม่มีความจำเป็นที่ต้องการใช้จอภาพก็สามารถที่จะซ่อนตัวเองชั่วคราว (Background Process) เพื่อรอแสดงออกทางจอภาพในทันทีที่ต้องการแต่ละแอปพลิเคชันจะใช้จอภาพแสดงผลร่วมกับแอปพลิเคชันอื่นๆ โดยใช้งานเฉพาะส่วนที่อยู่ในกรอบสี่เหลี่ยมของตนที่เรียกกันว่าวินโดวส์ หากมองดูกันอย่างผิวเผินแล้ววินโดวส์จะเป็นเพียงรูปสี่เหลี่ยมที่มีกรอบธรรมดาเท่านั้นแต่ที่จริงแล้ววินโดวส์ไม่เพียงแต่จะเป็นอุปกรณ์แสดงผลชั้นพื้นฐานของแอปพลิเคชันเท่านั้นแต่ยังเป็นสิ่งที่ผู้ใช้งานสามารถติดต่อกับแอปพลิเคชันได้เป็นอย่างดีโดยผ่านส่วนประกอบต่างๆของวินโดวส์เช่น เมนู ปุ่มบังคับ สโคลบาร์ เป็นต้น

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากโปรแกรมไมโครซอฟท์วินโดวส์มีวินโดวส์ให้แอปพลิเคชันใช้หลายแบบ โปรแกรมไมโครซอฟท์วินโดวส์จึงไม่สร้างวินโดวส์ให้แก่แอปพลิเคชันใดๆ แอปพลิเคชันเหล่านั้นจะต้องสร้างวินโดวส์แบบที่ต้องการเอง (ซึ่งต่างจากคอสมอสที่เตรียมจอภาพพื้นฐานให้ใช้ได้ทั้งหมดทันที) โดยโปรแกรมไมโครซอฟท์วินโดวส์จะอำนวยความสะดวกต่างๆ ให้เช่น การเคลื่อนย้าย การเปลี่ยนขนาด การลดขนาดให้เป็นไอคอน ดังนั้นสามารถแน่ใจได้เลยว่าแอปพลิเคชันต่างๆจะไม่แย่งกันใช้งานจอภาพจนเกิดความสับสนอย่างแน่นอน

#### - ระบบการรับข้อมูลเข้า

ในระบบคอสมอสรวมเอาแอปพลิเคชันสามารถรับข้อมูลจากผู้ใช้งานได้นั้นจะต้องรอให้รับข้อมูลนั้นได้เสียก่อนจึงสามารถที่จะทำงานต่อไปได้ยกตัวอย่างเช่น เมื่อแอปพลิเคชันต้องการรับข้อมูลผ่านทางคีย์บอร์ดผู้ใช้งานจะต้องใส่ข้อมูลจริงผ่านทางคีย์บอร์ดโดยการกดปุ่มของคีย์บอร์ดเสียก่อนหากในระหว่างนั้นผู้ใช้งานจะใส่ข้อมูลทางอื่นเช่นเมาส์ก็ไม่สามารถรับข้อมูลเข้ามาได้แต่ในโปรแกรมไมโครซอฟท์วินโดวส์การรับข้อมูลจากผู้ใช้งานจะแตกต่างออกไปโดยโปรแกรมไมโครซอฟท์วินโดวส์จะรวบรวมข้อมูลเข้าจากทุกๆอุปกรณ์แล้วเก็บไว้ในเมสเสจคิว (Message Queue) เมื่อแอปพลิเคชันต้องการอ่านข้อมูลเข้าก็เพียงแต่อ่านจากเมสเสจคิวเท่านั้น แอปพลิเคชันจะได้รับข้อมูลจากผู้ใช้งานโดยไม่มีการสูญหายของข้อมูล ข้อมูลที่อยู่ในเมสเสจคิวจะอยู่ในรูปของเมสเสจที่มีขนาดในการเก็บเท่ากันและมีรูปแบบการใช้งานคล้ายกัน ในการอ่านข้อมูลแต่ละครั้งการพิจารณาข้อมูลก็จะใช้โปรแกรมลักษณะที่คล้ายกันทำให้การเขียนแอปพลิเคชันง่ายและสะดวกขึ้น การอ่านข้อมูลจากอุปกรณ์ต่างๆก็เช่นกัน ในระบบโปรแกรมไมโครซอฟท์วินโดวส์การอ่านค่าจากเมสเสจคิวจะใช้ฟังก์ชันเพียงฟังก์ชันเดียวเท่านั้นคือ GetMessage () แต่ในคอสมอสการรับข้อมูลจะต้องเจาะจงใช้ฟังก์ชันที่ติดต่อกับอุปกรณ์ชนิดนั้นโดยเฉพาะทำให้มีความยุ่งยากมากกว่า นอกจากนี้ข้อมูลที่เกิดจากผู้ใช้งานเลือกเมนูหรือปุ่มบังคับต่างๆก็ยังคงบรรจุไว้ในเมสเสจคิวนี้ด้วย ดังนั้นการรับรู้การกระทำทั้งหมดของผู้ใช้งานทุกอย่างจึงใช้เพียงฟังก์ชัน GetMessage () เพียงฟังก์ชันเดียวเท่านั้น

#### - ระบบกราฟฟิก

ระบบโปรแกรมไมโครซอฟท์วินโดวส์เต็มไปด้วยคำสั่งและวิธีการอันหลากหลายซับซ้อนและมีประสิทธิภาพสูงสำหรับจัดการกับระบบกราฟฟิกไม่ว่าจะเป็นการวาดเส้นตรง วงกลม รูปหลายเหลี่ยม รูปทึบ เส้นโค้ง การระบายสี หรือรูปภาพ ไม่เพียงเท่านั้นคำสั่งทุกคำสั่งยังสามารถใช้ได้กับอุปกรณ์แสดงผลทุกชนิด (Device independence) ที่สามารถแสดงผลในรูปแบบกราฟฟิก แอปพลิเคชันสามารถใช้คำสั่งเดียวกันวาดรูปต่างๆได้ทั้งบนจอภาพ เครื่องพิมพ์ (กราฟฟิก) หรือพลอตเตอร์ เนื่องจากระบบโปรแกรมไมโครซอฟท์วินโดวส์อาศัยดีไวส์ไดรเวอร์ในการแสดงผลให้เหมาะสมกับอุปกรณ์ที่ใช้แต่ละแบบ ดีไวส์ไดรเวอร์อาจเรียกได้ว่าเป็นเอกสารนี้เป็นเอกสารที่ส่งงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับว่าเห็นไปเชิงประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พื้นผิวอุปกรณ์ (device context) โดยการแสดงผลทุกครั้งจะต้องกระทำผ่านพื้นผิวอุปกรณ์นี้เสมอเพื่อความถูกต้องตามอุปกรณ์

#### - ระบบจัดการทรัพยากร

การทำงานบนคอนั้นไม่จำเป็นต้องมีการปันส่วนทรัพยากรเนื่องจากเป็นระบบแอฟพลีเคชันเดียว (Singletask) สามารถมีแอฟพลีเคชันได้เพียงแอฟพลีเคชันเดียว ดังนั้นการทำงานของแอฟพลีเคชันก็จะถือเสมือนว่าเป็นเจ้าของทรัพยากรทั้งหมดเช่น คีย์บอร์ด จอภาพ หน่วยความจำหรือแม้กระทั่งซีพียู แต่สำหรับการทำงานบนโปรแกรมไมโครซอฟท์วินโดวส์เป็นระบบแอฟพลีเคชันร่วม (Multitasking) ดังนั้นโปรแกรมไมโครซอฟท์วินโดวส์จึงต้องบังคับให้แอฟพลีเคชันมีการปันส่วนทรัพยากรต่างๆร่วมกันเสมอ พื้นที่หน่วยความจำซึ่งเป็นทรัพยากรที่มีจำนวนจำกัดจะต้องปันส่วนกันใช้หากทุกแอฟพลีเคชันต่างใช้อย่างสิ้นเปลืองหน่วยความจำก็จะถูกใช้จนหมดอย่างรวดเร็วทำให้แอฟพลีเคชันทำงานพร้อมกันได้ไม่มากจะนั้นในการทำงานจริงของโปรแกรมไมโครซอฟท์วินโดวส์จึงมีการเคลื่อนย้ายสลับเปลี่ยนหรือทิ้งส่วนต่างๆของหน่วยความจำตลอดเวลาเพื่อให้การทำงานทั้งระบบมีประสิทธิภาพสูงสุด อีกตัวอย่างหนึ่งในการจัดจอก็จะต้องแบ่งกันใช้ในหลายๆแอฟพลีเคชันดังนั้นในการแสดงผลจะต้องให้โปรแกรมไมโครซอฟท์วินโดวส์เป็นตัวจัดการแสดงผลให้ทุกครั้ง

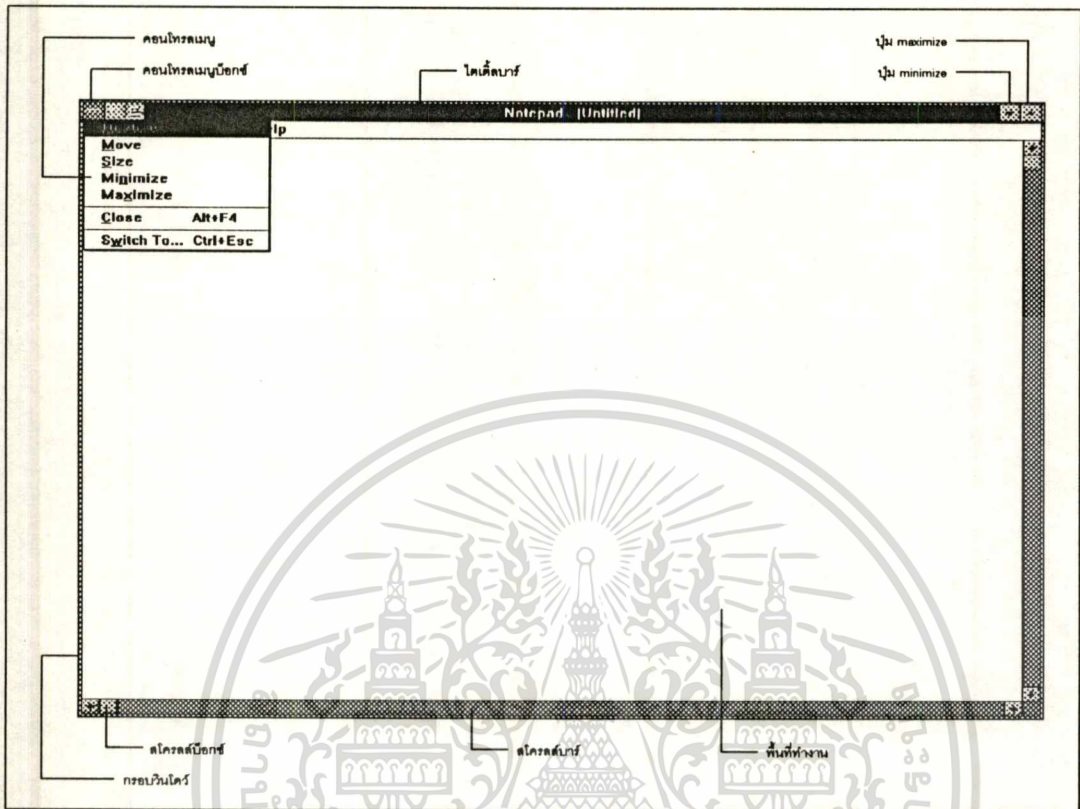
#### 5.2.2 รูปแบบของแอฟพลีเคชันบนไมโครซอฟท์วินโดวส์

จุดประสงค์หลักของโปรแกรมไมโครซอฟท์วินโดวส์คือ สร้างระบบติดต่อผู้ใช้งานแบบใหม่ที่ง่ายกว่าเดิม แอฟพลีเคชันจะติดต่อกับผู้ใช้งานโดยผ่านเครื่องมือแบบใหม่อันประกอบไปด้วยส่วนต่างๆดังต่อไปนี้

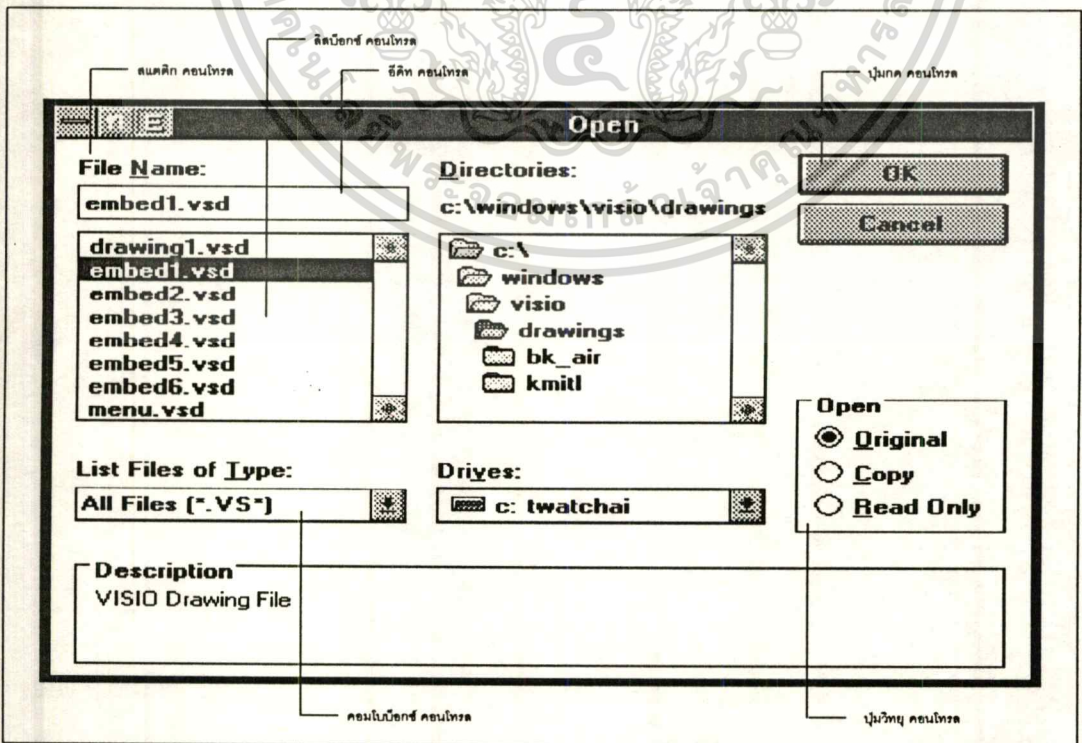
#### - วินโดวส์

เป็นสิ่งติดต่อกับผู้ใช้งานขั้นพื้นฐานที่ทุกแอฟพลีเคชันต้องมีใช้เสมอ วินโดวส์จะประกอบไปด้วยไคเต็ลบาร์ เมนูบาร์ สโครลบาร์ กรอบข้อความ เป็นต้น ดังแสดงในรูปที่ 5.1 เมื่อจะสร้างวินโดวส์ก็เพียงแต่เลือกสรรชิ้นส่วนต่างๆเพื่อมาประกอบเข้าด้วยกันเป็นวินโดวส์แบบที่ต้องการแล้วเรียกคำสั่ง CreateWindow () จากนั้นโปรแกรมไมโครซอฟท์วินโดวส์ก็จะทำหน้าที่วาดวินโดวส์ตามที่แอฟพลีเคชันต้องการ การสร้างวินโดวส์นั้นแอฟพลีเคชันจะเป็นผู้สร้างแต่การบริหารวินโดวส์จะเป็นหน้าที่ของโปรแกรมไมโครซอฟท์วินโดวส์ไม่ว่าจะเป็นการวาดวินโดวส์บนจอภาพด้วยรูปแบบต่างๆ ตำแหน่งของวินโดวส์ การเปลี่ยนขนาดวินโดวส์ การซ่อนวินโดวส์ เป็นต้น จะมีก็แต่เพียงส่วนหนึ่งของวินโดวส์ที่เรียกว่าพื้นที่ทำงาน (Client Area) ซึ่งส่วนนี้เองที่แอฟพลีเคชันที่เขียนขึ้นสามารถควบคุมรูปแบบการใช้งานต่างๆตามความต้องการได้อย่างเต็มที่และเนื่องจากแอฟพลีเคชันอันหนึ่งสามารถมีวินโดวส์ได้หลายวินโดวส์พื้นที่ทำงานของแต่ละวินโดวส์ย่อยในแอฟพลีเคชันนั้นก็แตกต่างกันออกไป จะมีการกำหนดให้เขียนส่วนตอบสนองของเอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ละวินโดว์ (หรือพื้นที่ทำงาน) แต่ละอันโดยเฉพาะในรูปแบบของฟังก์ชันประจำวินโดว์ซึ่งจะได้อธิบายโดยละเอียดจากการเขียนโปรแกรมระบบ ER/5 ต่อไป



รูปที่ 5.1 แสดงส่วนประกอบต่างๆของวินโดว์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 5.2 แสดงตัวอย่างของกรอบข้อความ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## - เมนู

เป็นสิ่งที่ใช้รับข้อมูลหลักของวินโดว์เมนูประกอบด้วยรายการคำสั่งเรียงกันไปซึ่งจะให้  
ผู้ใช้งานเลือกดูและใช้งานได้ง่าย โปรแกรมไมโครซอฟท์วินโดว์จะเป็นตัวจัดการแสดงเมนู  
และการเลือกรายการตามเมนู หลังจากนั้นก็จะส่งรายการที่เลือกผ่านมาทางเมสเสจคิวเพื่อ  
ให้แอปพลิเคชันที่เป็นเจ้าของเมื่อนั้นทำการตอบสนองต่อคำสั่งนั้นๆ ดังแสดงในรูปที่ 5.1

## - กรอบข้อความ (Dialog Boxes)

กรอบข้อความเป็นวินโดว์ชนิดหนึ่งที่ใช้งานชั่วคราวเพื่อให้แอปพลิเคชันติดต่อรับข้อ  
มูลจากผู้ใช้งาน ภายในกรอบข้อความจะประกอบด้วยคอนโทรล (Controls) ก็จัดเป็นวินโดว์  
ชนิดหนึ่ง) ต่างๆแล้วแต่จะกำหนดเช่น อีดิทคอนโทรล (Edit Control) ทำหน้าที่รับข้อมูลผ่าน  
ทางคีย์บอร์ด ปุ่มกดคอนโทรล (Push Button Control) ทำหน้าที่รับการกดจากเมาส์  
ลิสต์บ็อกซ์คอนโทรล (Listbox Control) ทำหน้าที่แสดงรายการต่างๆเพื่อให้ผู้ใช้งานเลือก จาก  
รูปที่ 5.2 เป็นตัวอย่างกรอบข้อความสำหรับเปิดเพิ่มข้อมูลซึ่งประกอบด้วยคอนโทรลต่างๆ  
ได้แก่ คอนโทรลสำหรับใส่ชื่อไฟล์เป็นอีดิทคอนโทรล คอนโทรลสำหรับเลือกไดเรกทอรีเป็น  
ลิสต์บ็อกซ์คอนโทรล คอนโทรลสำหรับบอกให้เริ่มการทำงานเป็นปุ่มกดคอนโทรล (OK) และ  
คอนโทรลสำหรับให้ยกเลิกการทำงานเป็นปุ่มกดคอนโทรล (Cancel) เป็นต้น

## - เมสเสจ

แอปพลิเคชันจะรับข้อมูลเข้าโดยผ่านทางเมสเสจคิว ส่วนประกอบหลักของแอปพลิเคชัน  
จึงต้องมีส่วนหนึ่งที่เรียกว่า เมสเสจลูป (Message Loop) เพื่อที่จะรับเมสเสจและแจก  
จ่ายไปยังวินโดว์ที่เหมาะสม จากรูปที่ 5.3 เมื่อเรากดคีย์บอร์ดหนึ่งคีย์โปรแกรมไมโครซอฟท์  
วินโดว์จะรับทราบและสร้างเมสเสจสำหรับคีย์บอร์ดที่ถูกกดนั้นวางลงบนคิวของระบบ  
(System que) แล้วส่งต่อไปยังคิวประจำแอปพลิเคชัน (Application que) ที่เหมาะสมจากนั้น  
ฟังก์ชันเมสเสจลูปของแอปพลิเคชันจะทำหน้าที่อ่านเมสเสจแล้วตีความซึ่งก็จะได้เมสเสจ  
WM\_CHAR โดยมีตัวอักษรในรูปของรหัส ASCII ฝากมากับเมสเสจนี้ด้วยแล้วส่งไปยังฟังก์ชัน  
ประจำวินโดว์ที่เป็นเจ้าของเมสเสจ หากฟังก์ชันประจำวินโดว์นั้นตอบสนองเมสเสจด้วยการ  
แสดงตัวหนังสือออกทางจอภาพก็จะใช้ฟังก์ชัน TextOut ในการแสดงตัวอักษรบนพื้นที่ทำงาน  
ของวินโดว์นั้น

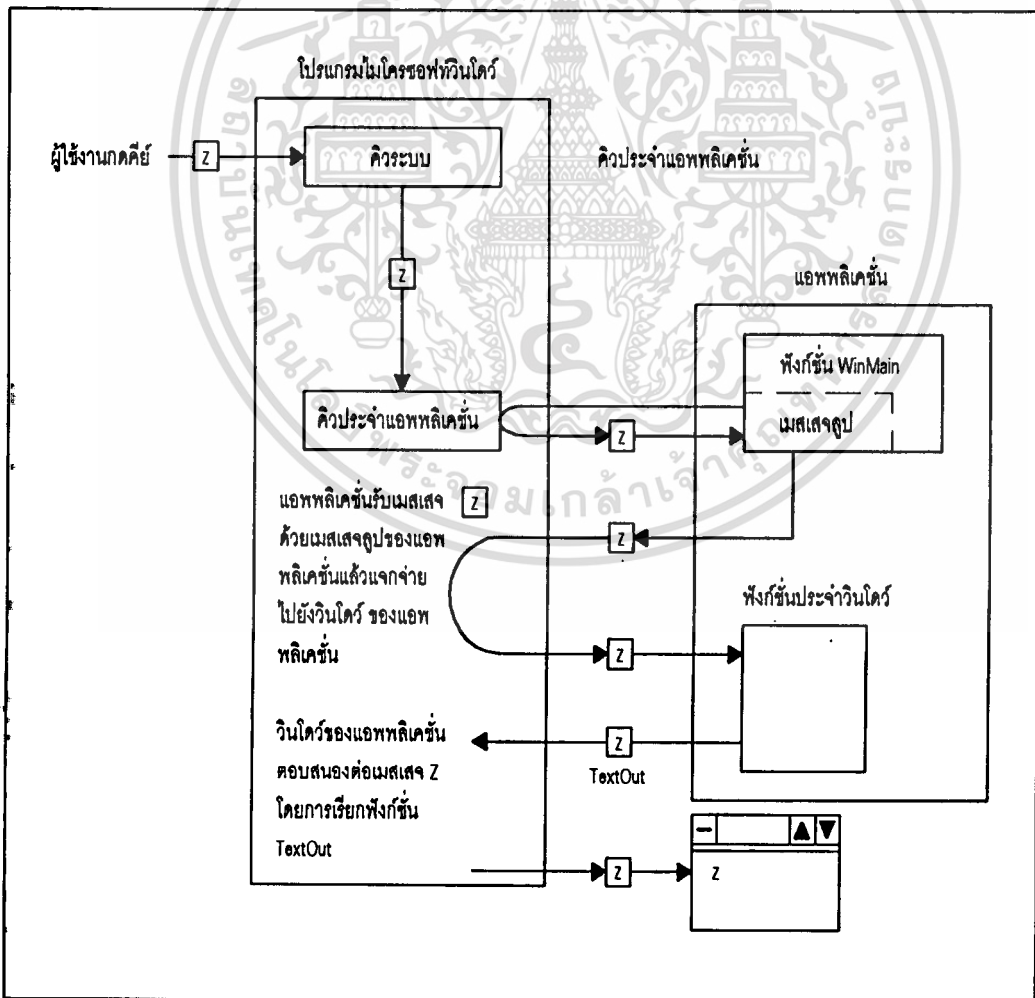
โปรแกรมไมโครซอฟท์วินโดว์ไม่เพียงแต่สามารถจัดการเมสเสจให้แก่แอปพลิเคชัน  
เดียวเท่านั้นแต่ยังสามารถที่จะรวบรวมเมสเสจให้แก่แอปพลิเคชันหลายๆแอปพลิเคชันพร้อม  
กันได้อีกด้วย จากรูปที่ 5.4 จะเห็นว่าโปรแกรมไมโครซอฟท์วินโดว์จะรวบรวมข้อมูลเข้าทั้ง  
หมดเก็บลงบนคิวของระบบแล้วจะทำการส่งเมสเสจไปยังคิวประจำแอปพลิเคชันแต่ละตัวโดย  
เมสเสจที่อยู่บนคิวประจำแอปพลิเคชันที่ได้รับนั้นจะเป็นเมสเสจที่ผู้ใช้งานได้ติดต่อกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้หาไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

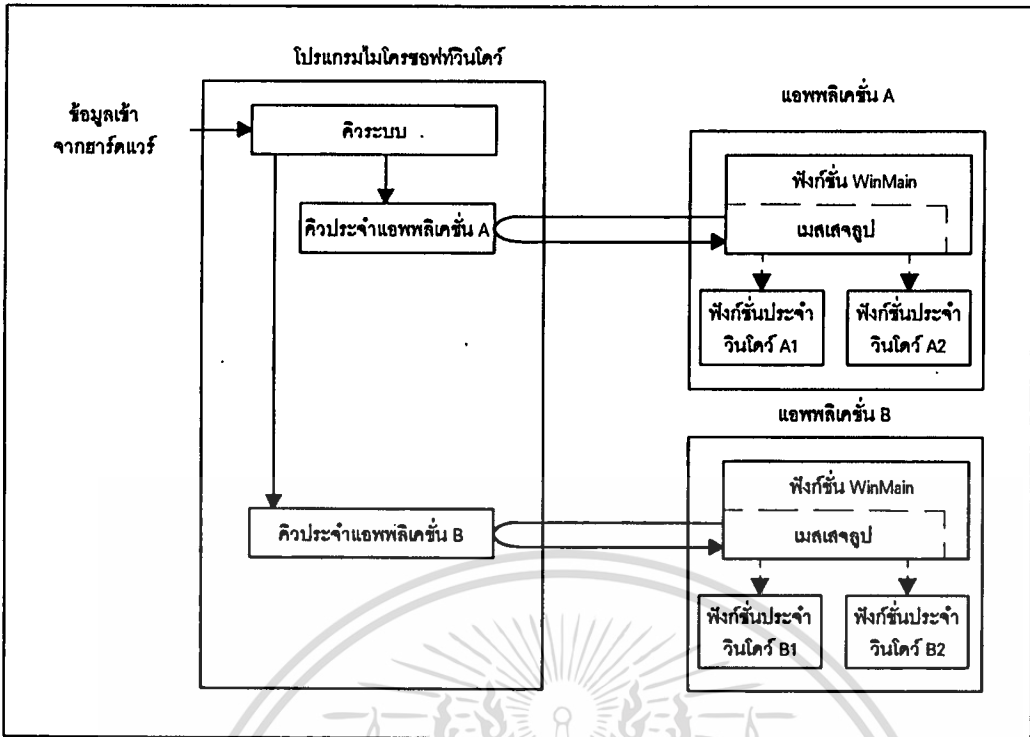
แอปพลิเคชันนั้นจริง จากนั้นแต่ละแอปพลิเคชันก็จะมีเมสเสจลูปคอยอ่านเมสเสจจากคิวของตนเองแล้วแจกจ่ายไปให้ฟังก์ชันประจำวินโดว์ที่เหมาะสมต่อไป

การส่งเมสเสจนอกจากจะส่งผ่านทางเมสเสจคิวประจำแอปพลิเคชันแล้ว โปรแกรมไม่ใครขอพท์วินโดว์ยังสามารถส่งเมสเสจไปยังฟังก์ชันประจำวินโดว์ของแต่ละแอปพลิเคชันโดยตรงได้อีกด้วยเช่นเมื่อผู้ใช้งานเลือกคำสั่ง Close ในเมนูระบบโปรแกรมไม่ใครขอพท์วินโดว์จะตอบรับการปิดระบบจากผู้ใช้งานโดยส่งเมสเสจชื่อ WM\_DESTROY ไปยังฟังก์ชันประจำวินโดว์โดยตรงไม่ผ่านเมสเสจคิวประจำแอปพลิเคชันเมื่อฟังก์ชันประจำวินโดว์ได้รับเมสเสจ WM\_DESTROY นี้ก็จะตอบสนองโดยการส่งสัญญาณให้แก่ WinMain ว่าแอปพลิเคชันวินโดว์ถูกสั่งปิดแล้วจะต้องเลิกทำงานด้วยการใช้คำสั่ง PostQuitMessage () เพื่อวางเมสเสจชื่อ WM\_QUIT ลงบนคิวประจำแอปพลิเคชันและเมื่อฟังก์ชัน GetMessage () ในเมสเสจลูปได้รับเมสเสจ WM\_QUIT ก็จะต้องเลิกทำงานพร้อมทั้งทำการปิดแอปพลิเคชันวินโดว์ทันที ดังแสดงในรูปที่ 5.5

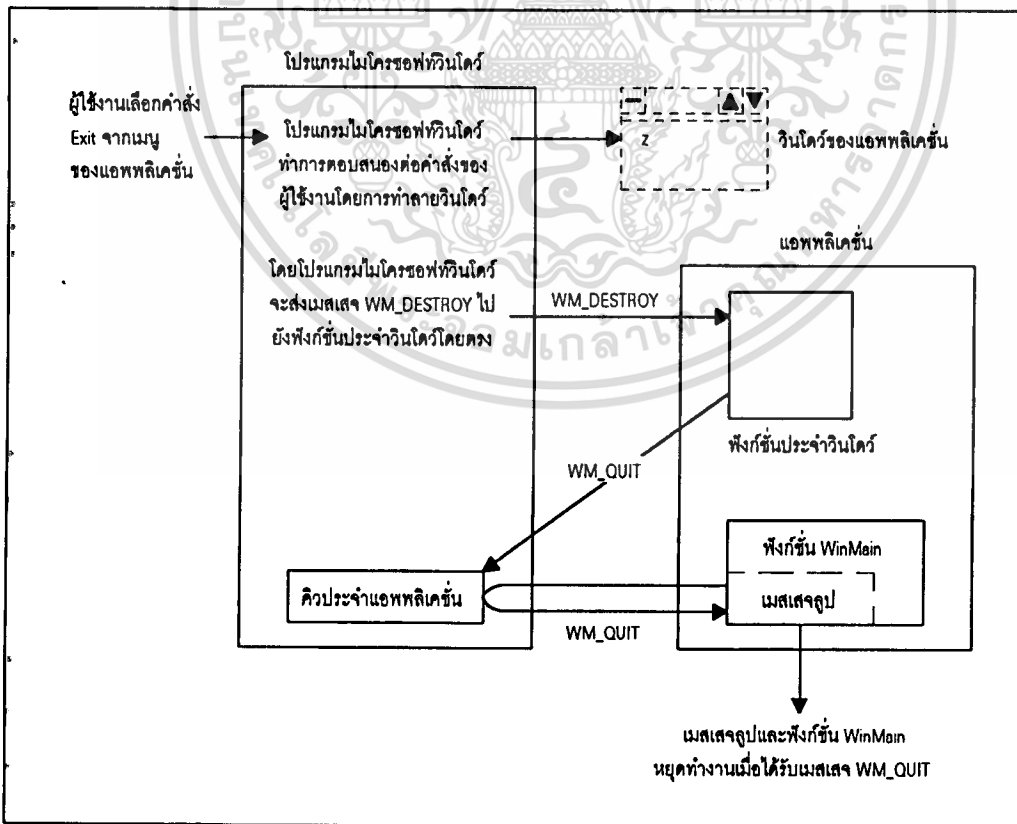


รูปที่ 5.3 แสดงการจัดการเข้าทางคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 แสดงการจัดการข้อมูลสำหรับแอปพลิเคชันสองตัว



รูปที่ 5.5 แสดงการจัดการเมสเสจของวินโดว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.2.3 แอปพลิเคชัน ER/5 บนไมโครซอฟท์วินโดว

จากที่ได้กล่าวมาแล้วจะเห็นว่าโปรแกรมใช้งานทางด้านกราฟฟิกสามารถเขียนบนโปรแกรมไมโครซอฟท์วินโดวได้โดยสะดวกและมีประสิทธิภาพ ดังนั้นแอปพลิเคชันโปรแกรมระบบ ER/5 ซึ่งจัดเป็นโปรแกรมใช้งานทางด้านกราฟฟิกก็ได้ถูกเขียนขึ้นบนโปรแกรมไมโครซอฟท์วินโดวโดยมีลักษณะการจัดการเมสเสจดังแสดงในรูปที่ 5.6 แอปพลิเคชัน ER/5 ประกอบด้วยวินโดวจำนวน 3 วินโดวดังแสดงในรูปที่ 5.6 (การสังเกตว่าเมสเสจนั้นเป็นของวินโดวใดให้สังเกตว่าเมาท์วางอยู่บนวินโดวใดเป็นหลัก) ซึ่งมีหน้าที่การทำงานแตกต่างกันคือ

#### - วินโดวหลัก (Main Window)

วินโดวหลัก ทำหน้าที่ในการรับข้อมูลจากเมนูของแอปพลิเคชัน ER/5 แล้วทำการประมวลผลตามรายการในเมนูที่ถูกเลือกซึ่งเมนูของแอปพลิเคชัน ER/5 ประกอบด้วย 2 เมนูหลักดังต่อไปนี้คือ

#### - เมนูหลักไฟล์ (File) ประกอบด้วยรายการย่อยจำนวน 6 รายการคือ

- เปิดแฟ้มข้อมูลของแบบจำลองอ็อบเจกต์ (Open...) เป็นคำสั่งให้แอปพลิเคชัน ER/5 ทำการแสดงกรอบข้อความ Open File สำหรับให้ผู้ใช้งานเลือกแฟ้มข้อมูลของแบบจำลองอ็อบเจกต์ที่มีอยู่เดิม เมื่อเลือกแฟ้มข้อมูลแล้วกรอบข้อความก็จะถูกปิดลงจากนั้นแอปพลิเคชัน ER/5 ก็จะนำข้อมูลของแบบจำลองอ็อบเจกต์ที่อยู่ในแฟ้มข้อมูลนั้นมาจัดเตรียมเพื่อรอการประมวลผลตามขั้นตอนที่ผู้ใช้งานต้องการต่อไป

- เก็บแฟ้มข้อมูลของแบบจำลองอ็อบเจกต์ (Save) เป็นคำสั่งให้แอปพลิเคชัน ER/5 ทำการเก็บข้อมูลทั้งหมดของแบบจำลองอ็อบเจกต์ที่ถูกวาดขึ้นเก็บไว้ในแฟ้มข้อมูลโดยไม่เปลี่ยนชื่อแฟ้มข้อมูล โดยยังสามารถที่จะใช้งานแอปพลิเคชัน ER/5 ได้ต่อไป

- เก็บแฟ้มข้อมูลของแบบจำลองอ็อบเจกต์ในชื่อใหม่ (Save As...) เป็นคำสั่งให้แอปพลิเคชัน ER/5 ทำการแสดงกรอบข้อความ Save As สำหรับให้ผู้ใช้งานตั้งชื่อแฟ้มข้อมูลที่ต้องการเก็บเมื่อผู้ใช้งานตั้งชื่อแฟ้มข้อมูลเป็นที่เรียบร้อยแล้วกรอบข้อความก็จะถูกปิดลง จากนั้นแอปพลิเคชัน ER/5 ก็จะเก็บข้อมูลทั้งหมดของแบบจำลองอ็อบเจกต์ที่ถูกวาดขึ้นเก็บไว้ในแฟ้มข้อมูลโดยเปลี่ยนชื่อแฟ้มข้อมูลเป็นชื่อที่ผู้ใช้งานตั้งขึ้นในกรอบข้อความ โดยยังสามารถที่จะใช้งานแอปพลิเคชัน ER/5 ได้ต่อไป

- พิมพ์แบบจำลองอ็อบเจกต์ (Print) เป็นคำสั่งให้แอปพลิเคชัน ER/5 ทำการพิมพ์ภาพแบบจำลองอ็อบเจกต์ที่ถูกวาดขึ้นออกทางเครื่องพิมพ์

- ข้อมูลเกี่ยวกับระบบ ER/5 (About ERFive...) เป็นคำสั่งให้แอปพลิเคชัน ER/5 ทำการแสดงกรอบข้อความ About ERFive อธิบายถึงข้อมูลและรายละเอียดต่างๆของ แอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ออกจากระบบ ER/5 (Exit) เป็นคำสั่งให้แอปพลิเคชัน ER/5 ทำการจบแอปพลิเคชัน และปิดวินโดว์ทั้งหมด

- เมนูหลักโปรเซส (Process) ประกอบด้วยรายการย่อยจำนวน 3 รายการคือ

- สร้างตารางฐานข้อมูลในบรรทัดฐานขั้นแรก (First Normal Form) เป็นคำสั่งให้ แอปพลิเคชัน ER/5 ทำการตรวจสอบความถูกต้องและประมวลผลข้อมูลจากแผนภาพอีอาร์ที วาดขึ้นแล้วสร้างคำสั่งภาษาสอบถามเชิงโครงสร้าง ในส่วนของการสร้างตาราง อินเด็กซ์ และ วิว โดยตารางฐานข้อมูลที่ได้นั้นจัดเป็นตารางฐานข้อมูลในบรรทัดฐานขั้นแรกและสร้างความ สัมพันธ์ระหว่างแอททริบิวต์เบื้องต้นขึ้น (Functional Dependency) เพื่อสามารถนำไปแก้ไขปรับ ปรุงในขั้นตอนการสร้างตารางฐานข้อมูลในบรรทัดฐานขั้น Boyce/Codd ต่อไป

- สร้างตารางฐานข้อมูลในบรรทัดฐานขั้น Boyce/Codd (Boyce Codd Normal Form) เป็นคำสั่งให้แอปพลิเคชัน ER/5 ทำการแสดงกรอบข้อความ FD พร้อมนำข้อมูลความสัมพันธ์ ระหว่างแอททริบิวต์ที่ถูกสร้างขึ้นในขั้นตอนการสร้างตารางในบรรทัดฐานขั้นแรกมาแสดง ซึ่ง ข้อมูลนี้สามารถที่จะเปลี่ยนแปลง แก้ไข เพิ่มเติม ได้จากผู้ใช้งานผ่านทางกรอบข้อความ FD และเมื่อผู้ใช้งานกดปุ่ม RUN แอปพลิเคชัน ER/5 ก็จะได้ทำการตรวจสอบความถูกต้องของ ความสัมพันธ์ระหว่างแอททริบิวต์และประมวลผลข้อมูลความสัมพันธ์ระหว่างแอททริบิวต์ทั้ง หมดแล้วสร้างคำสั่งภาษาสอบถามเชิงโครงสร้าง (SQL) ในส่วนของการสร้างตาราง อินเด็กซ์ และวิว โดยตารางฐานข้อมูลที่ได้นั้นจัดเป็นตารางฐานข้อมูลในบรรทัดฐานขั้น Boyce/Codd และปิดกรอบข้อความ FD

- สร้างตารางฐานข้อมูลในบรรทัดฐานขั้นที่ห้า (Fifth Normal Form) เป็นคำสั่งให้ แอปพลิเคชัน ER/5 ทำการตรวจสอบว่าตารางฐานข้อมูลที่สร้างขึ้นจากขั้นตอนการสร้าง ตารางฐานข้อมูลในบรรทัดฐานขั้น Boyce/Codd ว่าตารางใดยังสามารถที่จะมีความสัมพันธ์ เจตติ (Join Dependency) แอบแฝงอยู่ถ้าตรวจพบก็จะแสดงกรอบข้อความ JD เพื่อผู้ใช้งานให้ สามารถที่จะกำหนดความสัมพันธ์เจตติผ่านทางกรอบข้อความ JD และเมื่อผู้ใช้งานกดปุ่ม RUN แอปพลิเคชัน ER/5 ก็จะได้ทำการตรวจสอบความถูกต้องของความสัมพันธ์เจตติและ ประมวลผลข้อมูลเจตติทั้งหมดเพื่อสร้างคำสั่งภาษาสอบถามเชิงโครงสร้าง (SQL) ในส่วนของ การสร้างตาราง อินเด็กซ์ และวิว โดยตารางฐานข้อมูลที่ได้นั้นจัดเป็นตารางฐานข้อมูลใน บรรทัดฐานขั้นที่ห้าและปิดกรอบข้อความ JD

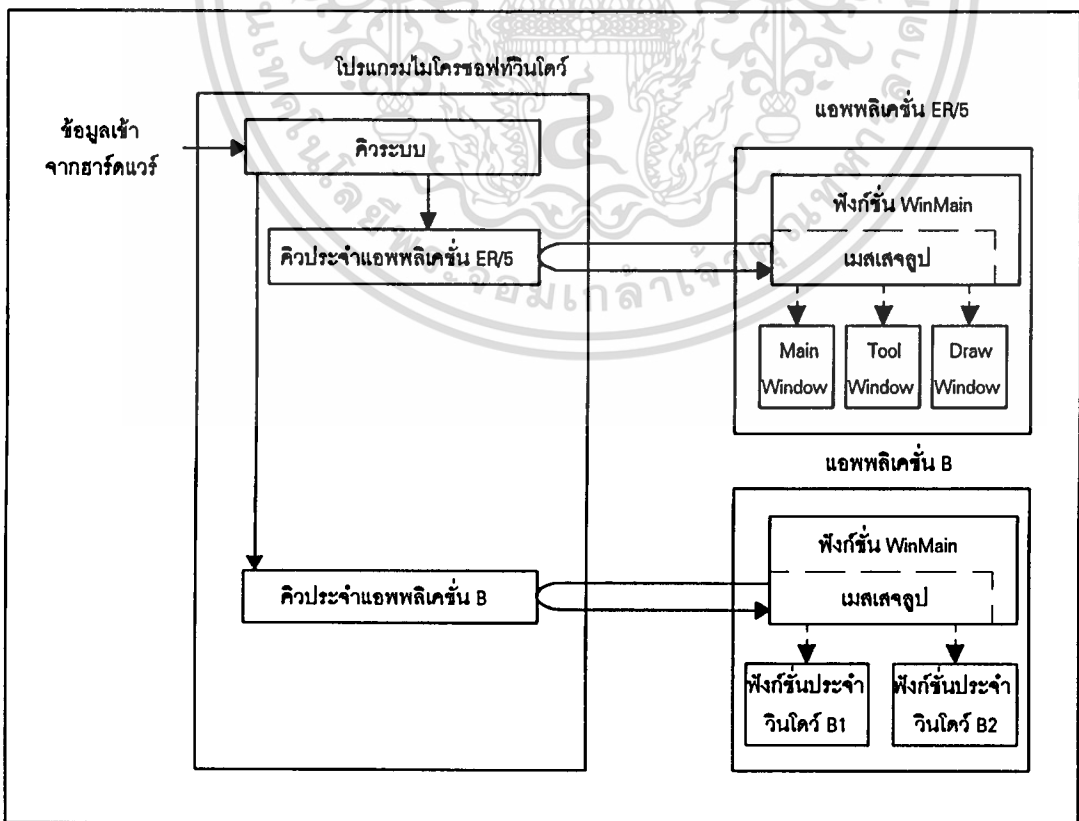
- วินโดว์เครื่องมือ (Tool Winow)

วินโดว์เครื่องมือ ทำหน้าที่แสดงสัญลักษณ์ทั้งหมดที่ใช้ในการวาดภาพแบบจำลอง อีอาร์ดีแสดงทางด้านซ้ายมือของรูปที่ 5.7 เพื่อให้ผู้ใช้งานเลือกว่าต้องการวาดสัญลักษณ์ใด มืออยู่ด้วยกัน 17 สัญลักษณ์เรียงลำดับกันจากบนลงล่างดังต่อไปนี้คือ การวาดเอ็นติตีชนิด เอกสารเป็นเอกสารที่ส่งไว้ในสำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนู๊ตเห็นไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปกติ การวาดเส้นติดกันชนิดอ่อน การวาดความเกี่ยวพันชนิดปกติ การวาดความเกี่ยวพันชนิดอ่อนการวาดแอทธิบิวต์ชนิดปกติ การวาดแอทธิบิวต์ชนิดกฏแฉ่แข่ง การวาดแอทธิบิวต์ชนิดกฏแฉ่หลัก การวาดแอทธิบิวต์ชนิดหลายค่า การวาดการเชื่อมต่อแอทธิบิวต์ การวาดการเชื่อมต่อความเกี่ยวพันแบบพาเซียลวัน (1P) การวาดการเชื่อมต่อความเกี่ยวพันแบบพาเซียลเอ็น (NP) การวาดการเชื่อมต่อความเกี่ยวพันแบบพาเซียลเอ็ม (MP) การวาดการเชื่อมต่อความเกี่ยวพันแบบโททอลวัน (1T) การวาดการเชื่อมต่อความเกี่ยวพันแบบโททอลเอ็น (NT) การวาดการเชื่อมต่อความเกี่ยวพันแบบโททอลเอ็ม (MT) การวาดข้อความ และการลบวัตถุของแบบจำลองอีอาร์ ในการเลือกสัญลักษณ์นั้นผู้ใช้งานจะเลือกสัญลักษณ์ได้ครั้งละหนึ่งสัญลักษณ์เท่านั้นด้วยเส้นเคอร์เซอร์เมาท์รูปมือไปวางบนสัญลักษณ์ที่ต้องการแล้วกดปุ่มซ้ายของเมาท์หนึ่งครั้งสัญลักษณ์ที่ถูกเลือกนั้นก็เปลี่ยนสีจากสีเขียวเป็นสีน้ำเงิน แอปพลิเคชัน ER/5 ก็จะมีเก็บค่าตัวเลขประจำสัญลักษณ์ที่เลือกนั้นไว้เพื่อให้วินโดว์วาดภาพสามารถวาดภาพได้ถูกต้องตามสัญลักษณ์ที่เลือก

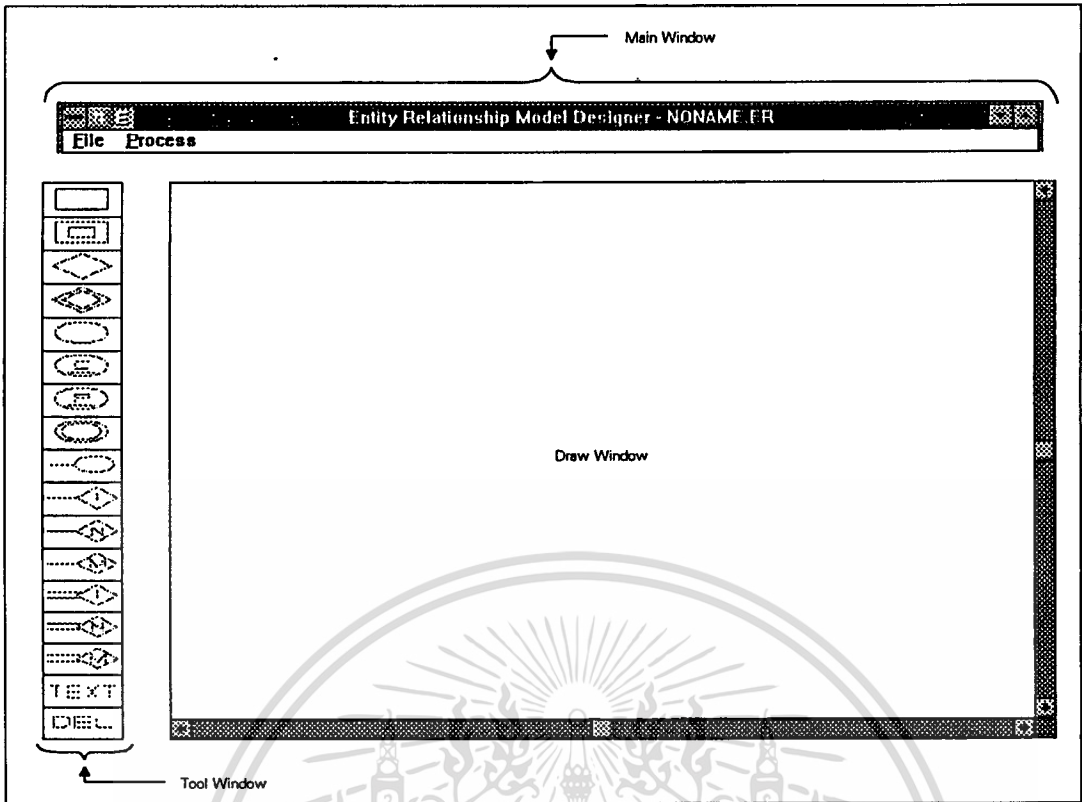
- วินโดว์วาดภาพ (Draw Window)

วินโดว์วาดภาพ ทำหน้าที่อ่านค่าตัวเลขประจำสัญลักษณ์ที่เลือกนั้นไว้จากวินโดว์เครื่องมือแล้วทำการวาดภาพตามสัญลักษณ์ที่เลือกนั้นซึ่งรายละเอียดของการวาดจะได้อธิบายโดยละเอียดในหัวข้อถัดไป



รูปที่ 5.6 แสดงการจัดการเมสเสจของแอปพลิเคชัน ER/5

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 แสดงวินโดว์ทั้งหมดของแอปพลิเคชัน ER/5

### 5.3 หน่วยโปรแกรมด้านกราฟฟิกของระบบ ER/5

หน่วยโปรแกรมด้านกราฟฟิกของระบบ ER/5 นั้นเป็นโปรแกรมที่ถูกเขียนขึ้นเพื่อให้ผู้ใช้งานสามารถใส่ข้อมูลของแบบจำลองอีอาร์ด้วยกราฟฟิกได้และเก็บข้อมูลรูปภาพลงบนฐานข้อมูลวินโดว์เมทต้าไฟล์ ข้อมูลแบบจำลองเก็บลงบนฐานข้อมูลอีอาร์เมทต้า และข้อมูลขอบเขตตำแหน่งของวัตถุที่วางอยู่บนจอภาพเก็บลงบนโครงสร้างข้อมูลวัตถุแบบเบิ้ลลิงคิสต์ ส่วนประกอบของหน่วยโปรแกรมด้านกราฟฟิกของระบบ ER/5 นี้สามารถแยกได้ตามสัญลักษณ์ของแบบจำลองอีอาร์ที่ปรากฏอยู่บนอีอาร์ทูลเพลทดังต่อไปนี้คือ

- ฟังก์ชัน R\_ENTITY\_PROC () ใช้วาดเอ็นติตี้ชนิดปกติ
- ฟังก์ชัน W\_ENTITY\_PROC () ใช้วาดเอ็นติตี้ชนิดอ่อน
- ฟังก์ชัน N\_RELATION\_PROC () ใช้วาดความเกี่ยวพันชนิดปกติ
- ฟังก์ชัน I\_RELATION\_PROC () ใช้วาดความเกี่ยวพันชนิดอ่อน
- ฟังก์ชัน N\_ATTRIBUTE\_PROC () ใช้วาดแอททริบิวต์ชนิดปกติ
- ฟังก์ชัน K\_ATTRIBUTE\_PROC () ใช้วาดแอททริบิวต์ชนิดกฤญแจคู่แข่ง
- ฟังก์ชัน P\_ATTRIBUTE\_PROC () ใช้วาดแอททริบิวต์ชนิดกฤญแจหลัก
- ฟังก์ชัน MV\_ATTRIBUTE\_PROC () ใช้วาดแอททริบิวต์ชนิดหลายค่า
- ฟังก์ชัน LINK\_PROC () ใช้วาดการเชื่อมต่อแอททริบิวต์

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน PARTIAL\_1\_PROC () ใช้วาดการเชื่อมต่อความเกี่ยวพันแบบพาเซี่ยลวัน  
ฟังก์ชัน PARTIAL\_N\_PROC () ใช้วาดการเชื่อมต่อความเกี่ยวพันแบบพาเซี่ยลเอ็น  
ฟังก์ชัน PARTIAL\_M\_PROC () ใช้วาดการเชื่อมต่อความเกี่ยวพันแบบพาเซี่ยลเอ็ม  
ฟังก์ชัน TOTAL\_1\_PROC () ใช้วาดการเชื่อมต่อความเกี่ยวพันแบบโททอลวัน  
ฟังก์ชัน TOTAL\_N\_PROC () ใช้วาดการเชื่อมต่อความเกี่ยวพันแบบโททอลเอ็น  
ฟังก์ชัน TOTAL\_M\_PROC () ใช้วาดการเชื่อมต่อความเกี่ยวพันแบบโททอลเอ็ม  
ฟังก์ชัน TEXT\_PROC () ใช้วาดข้อความ  
ฟังก์ชัน INDICATOR\_PROC () ใช้ลบลักษณะของแบบจำลองอ็อบเจกต์

ฟังก์ชันทั้งหมดนี้เป็นฟังก์ชันประจำวินโดว์วาดภาพจะถูกเรียกมาทำงานก็ต่อเมื่อมีเมสเสจส่งเข้ามายังวินโดว์วาดภาพนี้และฟังก์ชันใดจะถูกใช้งานก็จะขึ้นอยู่กับค่าตัวเลขประจำสัญลักษณ์ที่เลือกจากวินโดว์เครื่องมือ ในการวาดภาพตามสัญลักษณ์ต่างๆนั้นบางสัญลักษณ์ก็มีลักษณะการทำงานที่คล้ายคลึงกันดังนั้น ฟังก์ชันที่คล้ายคลึงกันจึงสามารถจัดรวมกันเป็นกลุ่มได้ 7 กลุ่มดังต่อไปนี้คือ

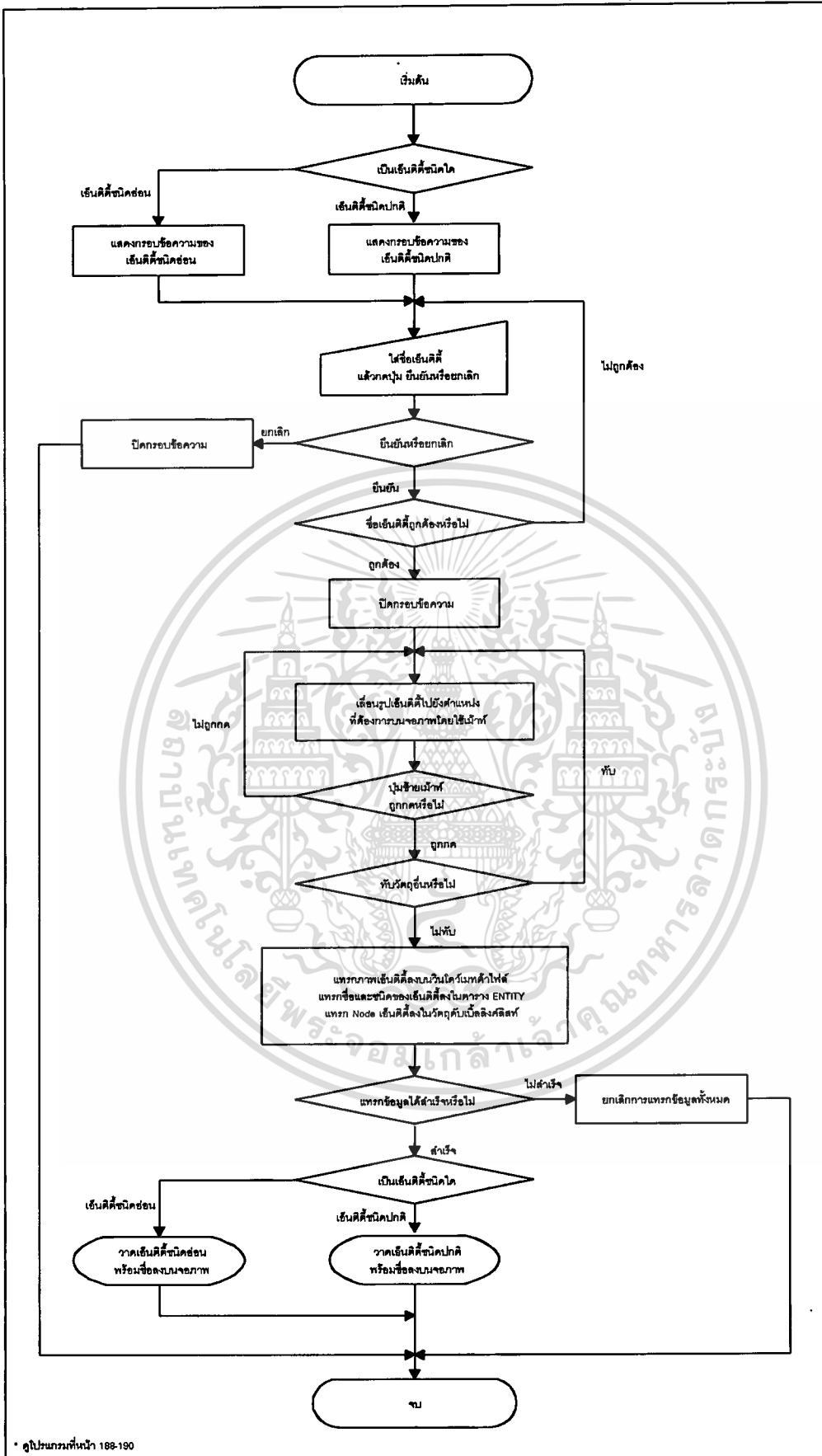
- การวาดเอ็นติตี้ ฟังก์ชันการวาดเอ็นติตี้ ENTITY\_PROC () เป็นฟังก์ชันที่เขียนขึ้นเพื่อสนับสนุนการทำงานของฟังก์ชันการวาดเอ็นติตี้ชนิดปกติ R\_ENTITY\_PROC () และเอ็นติตี้ชนิดอ่อน W\_ENTITY\_PROC () มีลักษณะการทำงานดังแสดงในรูปที่ 5.8

- การวาดความเกี่ยวพัน ฟังก์ชันการวาดความเกี่ยวพัน RELATION\_PROC () เป็นฟังก์ชันที่เขียนขึ้นเพื่อสนับสนุนการทำงานของฟังก์ชันการวาดความเกี่ยวพันชนิดปกติ N\_RELATION\_PROC () และความเกี่ยวพันชนิดอ่อน I\_RELATION\_PROC () มีลักษณะการทำงานดังแสดงในรูปที่ 5.9

- การวาดแอททริบิวต์ ฟังก์ชันการวาดแอททริบิวต์ ATTRIBUTE\_PROC () เป็นฟังก์ชันที่เขียนขึ้นเพื่อสนับสนุนการทำงานของฟังก์ชันการวาดแอททริบิวต์ชนิดปกติ N\_ATTRIBUTE\_PROC () แอททริบิวต์ชนิดถูกผูกมัด K\_ATTRIBUTE\_PROC () แอททริบิวต์ชนิดถูกผูกมัดหลัก P\_ATTRIBUTE\_PROC () และแอททริบิวต์ชนิดหลายค่า MV\_ATTRIBUTE\_PROC () มีลักษณะการทำงานดังแสดงในรูปที่ 5.10

- การวาดการเชื่อมต่อแอททริบิวต์ ฟังก์ชันการวาดการเชื่อมต่อแอททริบิวต์ LINK\_PROC () เป็นฟังก์ชันที่เขียนขึ้นเพื่อให้ผู้ใช้งานสามารถที่จะเชื่อมแอททริบิวต์ที่ได้วาดขึ้นกับวัตถุอื่น มีลักษณะการทำงานดังแสดงในรูปที่ 5.11

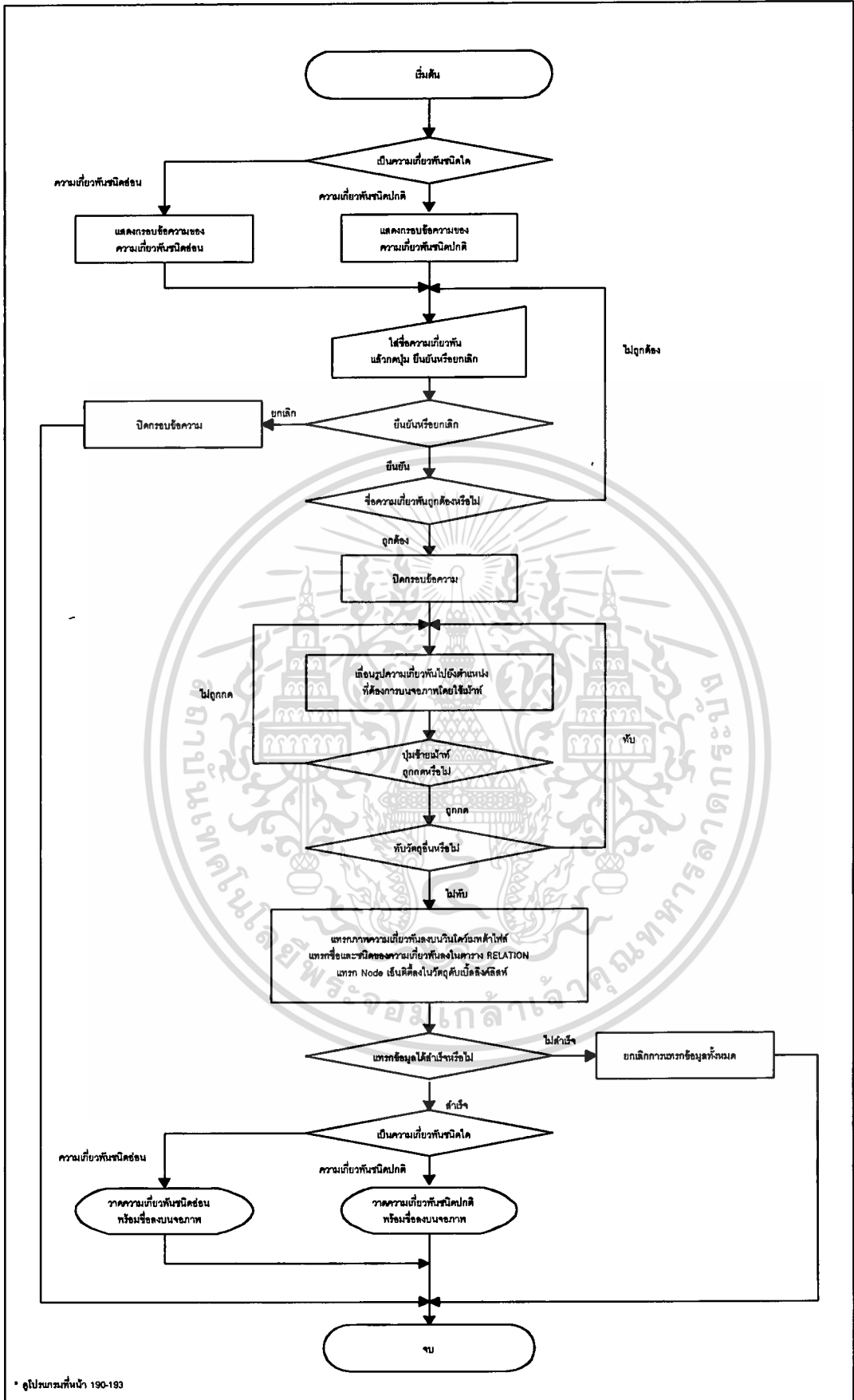
- การวาดการเชื่อมต่อความเกี่ยวพัน ฟังก์ชันการวาดการเชื่อมต่อความเกี่ยวพัน ROLE\_PROC () เป็นฟังก์ชันที่เขียนขึ้นเพื่อสนับสนุนการทำงานของฟังก์ชันการเชื่อมต่อความเกี่ยว



\* คู่มือกรมท่งหน้า 188-190

รูปที่ 5.8 แสดงผังการทำงานในการวาดเอนทิตี

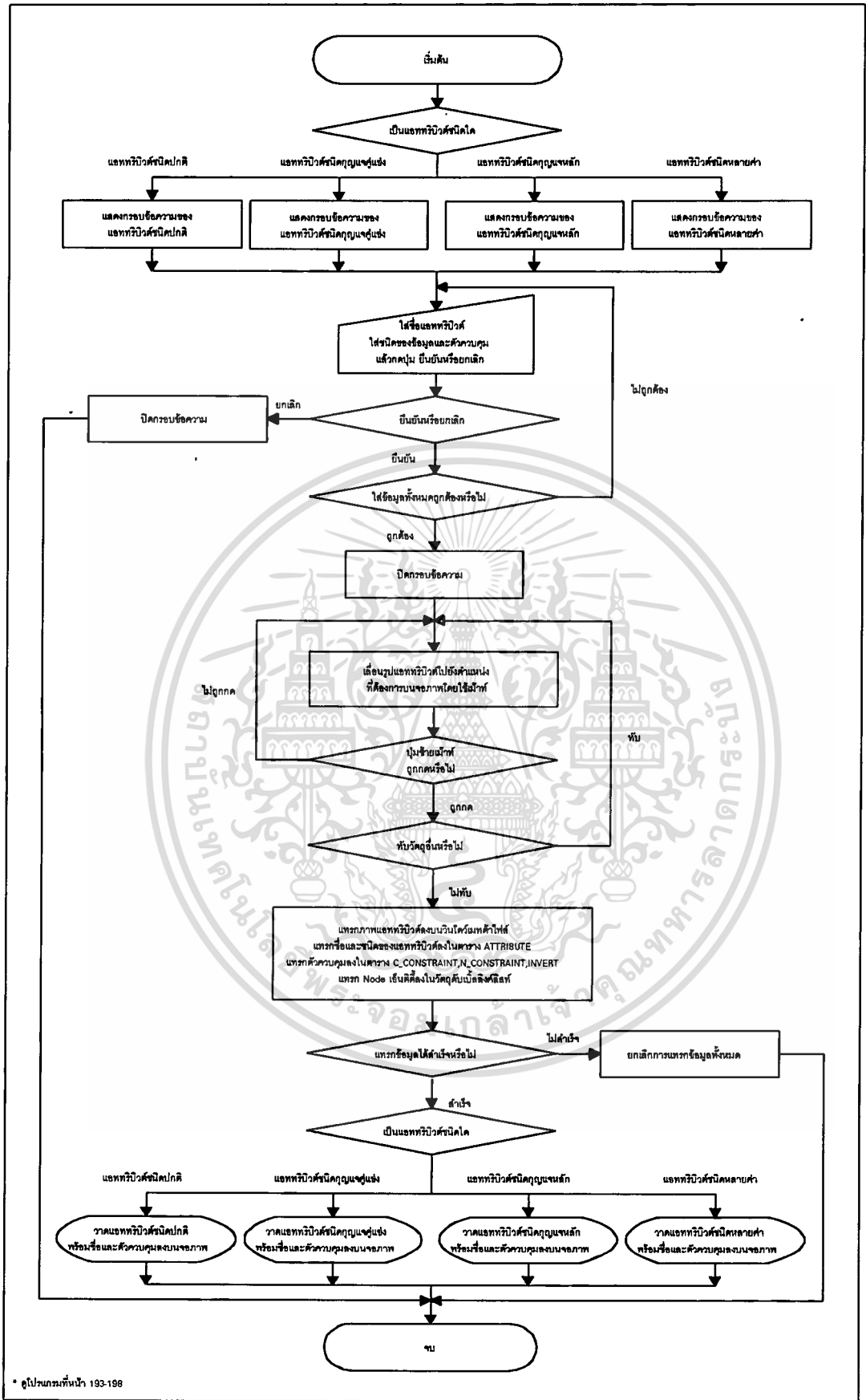
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



• คู่มือระบบที่หน้า 190-193

รูปที่ 5.9 แสดงผังการทำงานในการวาดความสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



\* ดูโปรแกรมที่หน้า 193-198

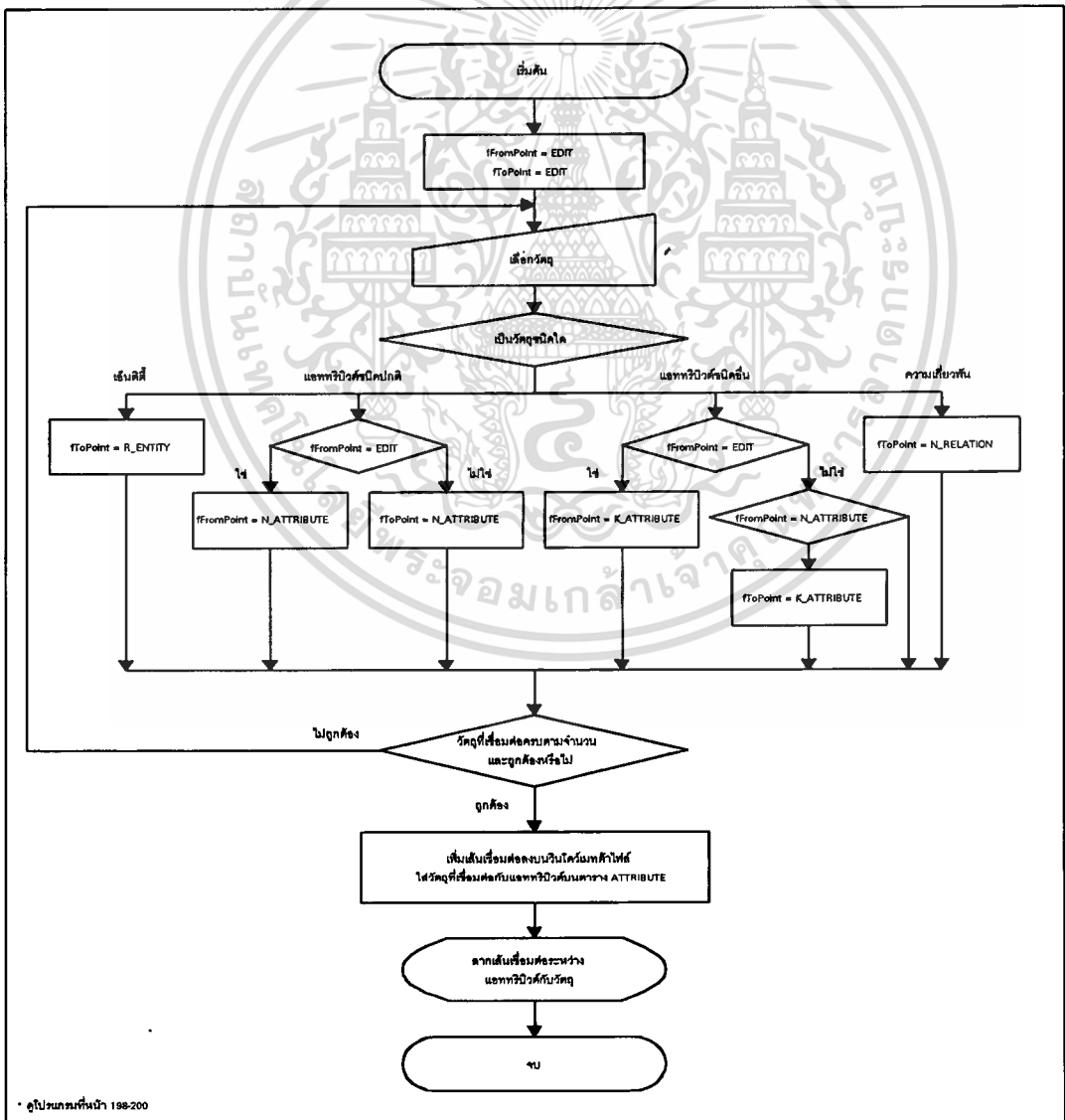
รูปที่ 5.10 แสดงผังการทำงานในการวาดแอททริบิวต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พันธแบบพาเซี่ยลวัน PARTIAL\_1\_PROC () (1P) การเชื่อมต่อความเกี่ยวพันแบบพาเซี่ยลเอ็น PARTIAL\_N\_PROC () (NP) การเชื่อมต่อความเกี่ยวพันแบบพาเซี่ยลเอ็ม PARTIAL\_M\_PROC () (MP) เชื่อมต่อความเกี่ยวพันแบบโททอลวัน TOTAL\_1\_PROC () (1T) การเชื่อมต่อความเกี่ยวพันแบบโททอลเอ็น TOTAL\_N\_PROC () (NT) และการเชื่อมต่อความเกี่ยวพันแบบโททอลเอ็ม TOTAL\_M\_PROC () (MT) มีลักษณะการทำงานดังแสดงในรูปที่ 5.12

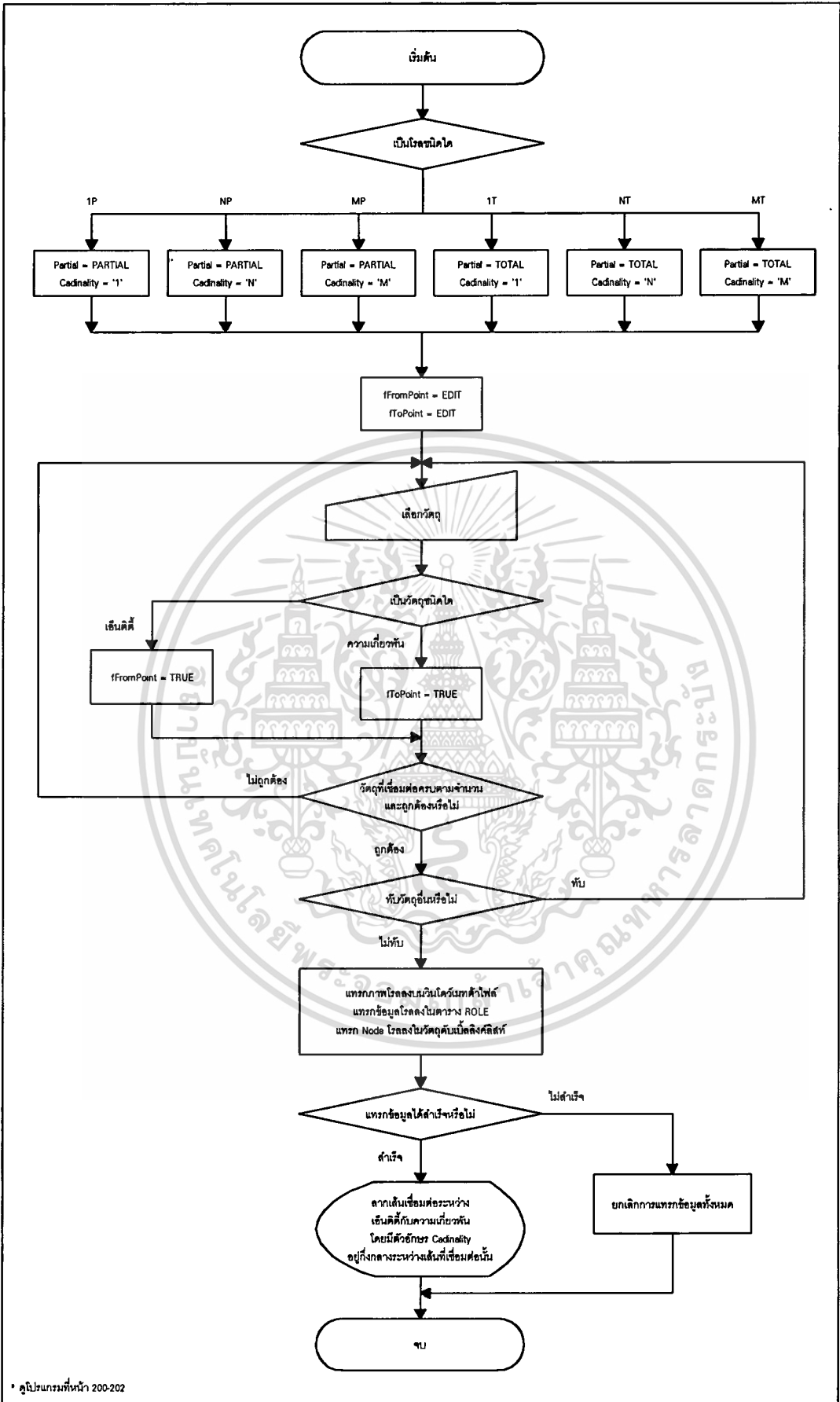
- การวาดข้อความ ฟังก์ชันการวาดข้อความ TEXT\_PROC () เป็นฟังก์ชันที่เขียนขึ้นเพื่อให้ผู้ใช้งานสามารถที่จะเขียนข้อความลงบนแบบจำลองอ็อบเจกต์ได้ มีลักษณะการทำงานดังแสดงในรูปที่ 5.13

- การลบข้อมูลวัตถุออกจากแบบจำลองอ็อบเจกต์ ฟังก์ชันการลบวัตถุบนแบบจำลองอ็อบเจกต์ INDICATOR\_PROC () เป็นฟังก์ชันที่เขียนขึ้นเพื่อให้ผู้ใช้งานสามารถที่จะลบวัตถุได้ทุกชนิดบนแบบจำลองอ็อบเจกต์ได้ มีลักษณะการทำงานดังแสดงในรูปที่ 5.14 และ รูปที่ 5.15



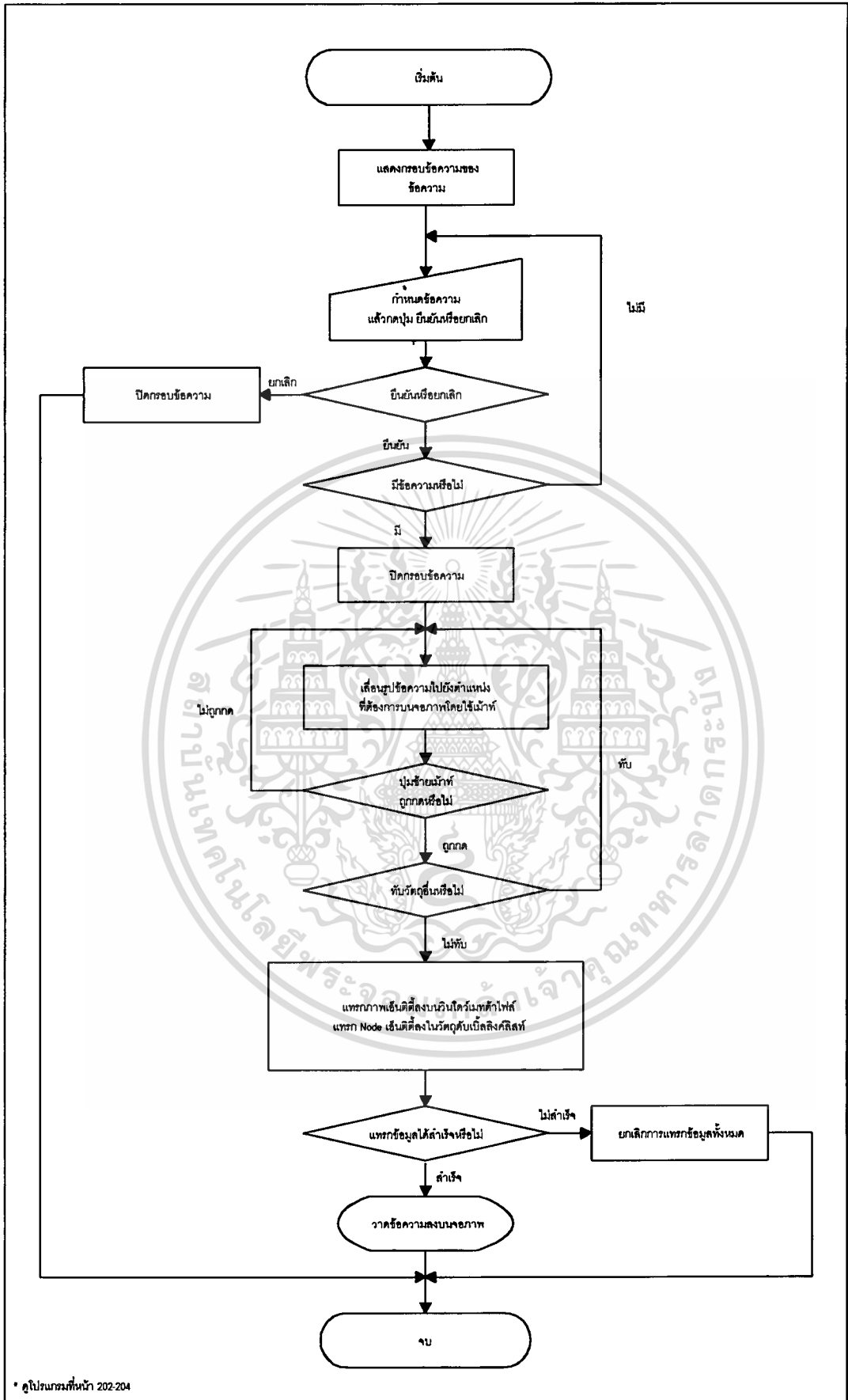
รูปที่ 5.11 แสดงผังการทำงานในการวาดการเชื่อมต่อแอททริบิวต์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานเท่านั้น เมื่อผู้ดูแลเห็นนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



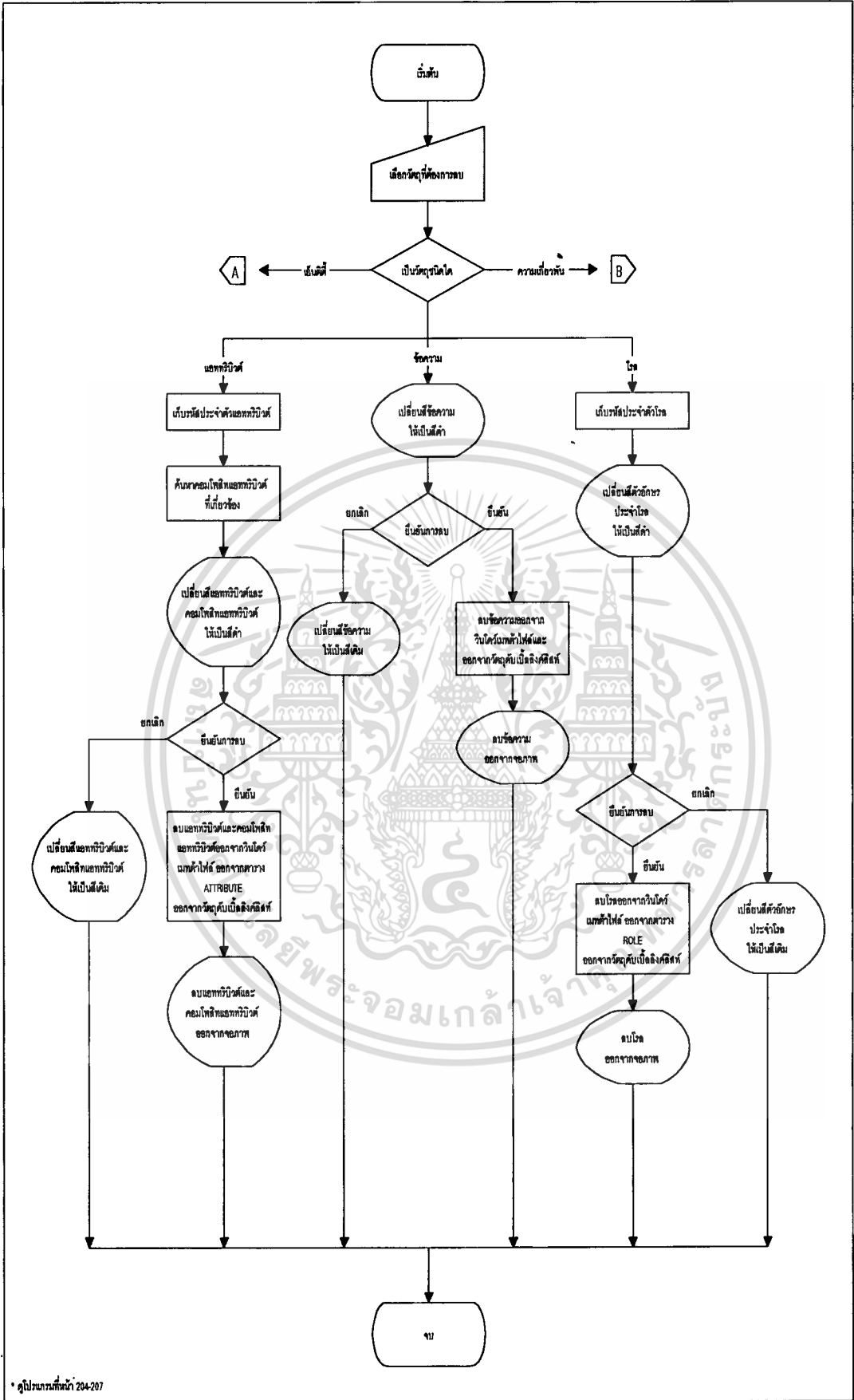
• คู่มือโปรแกรมที่หน้า 200-202

รูปที่ 5.12 แสดงผังการทำงานในการวาดการเชื่อมต่อความสัมพันธ์ เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ใช้ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



\* คู่มือระบบต้นฉบับ 202-204

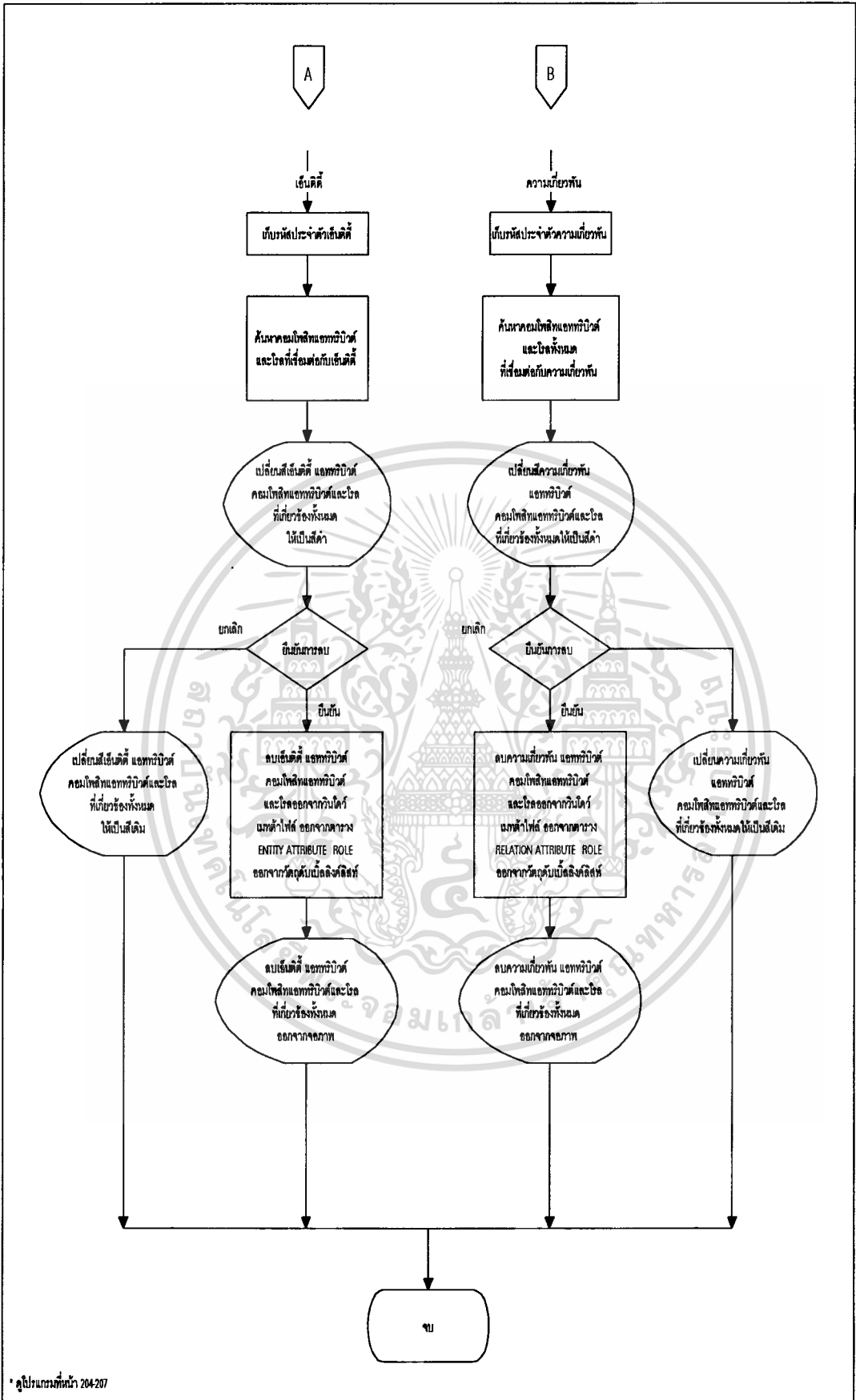
รูปที่ 5.13 แสดงผังการทำงานในการวาดข้อความ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



\* คู่มือกรมที่หน้า 204-207

รูปที่ 5.14 แสดงผังการทำงานในการลบวัตถุออกจากแบบจำลองอ็อบเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



\* ดูโปรแกรมหน้า 204-207

รูปที่ 5.15 แสดงผังการทำงานในการลบวัตถุจากแบบจำลองอ็อบเจกต์

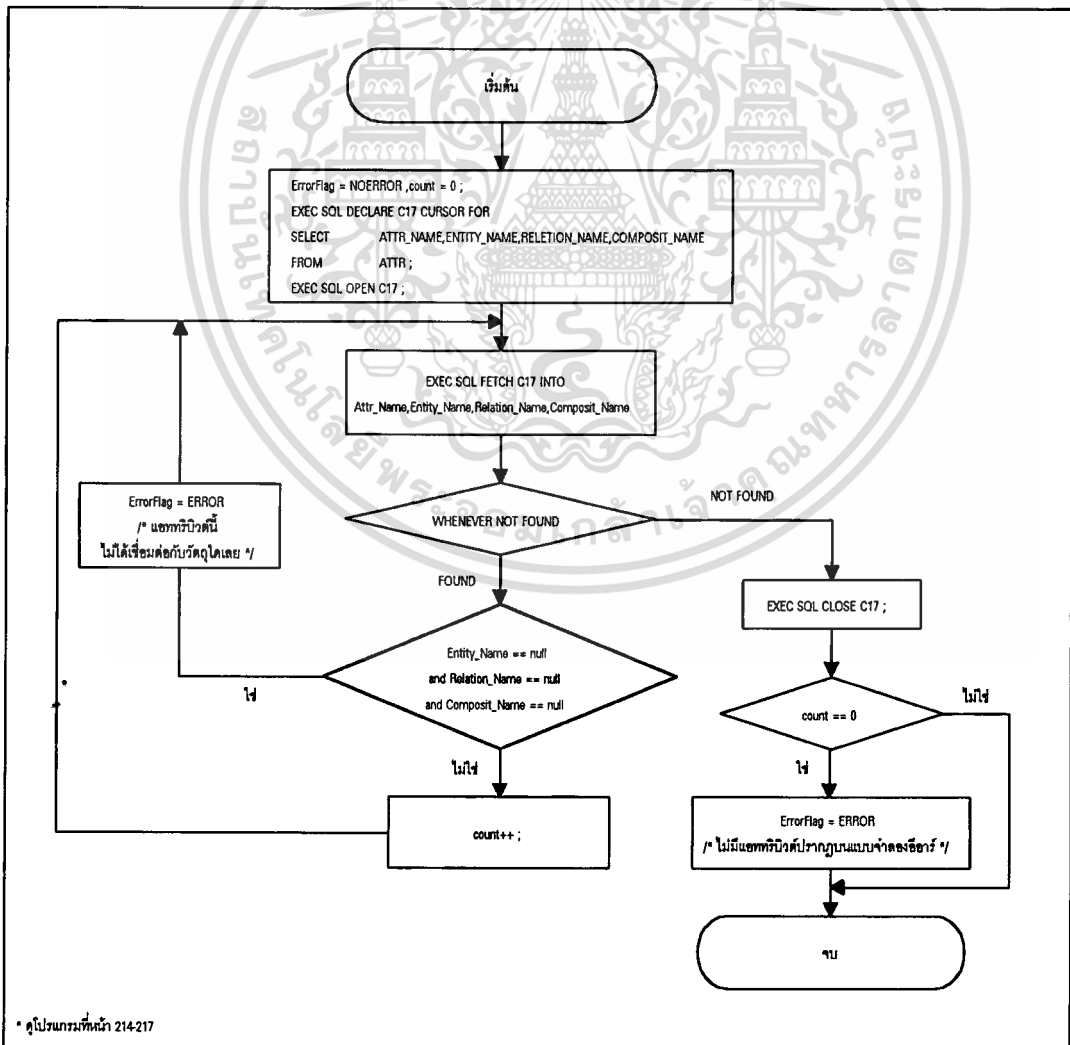
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.4 หน่วยโปรแกรมประมวลผลข้อมูลสร้างตารางในบรรทัดฐานขั้นแรก

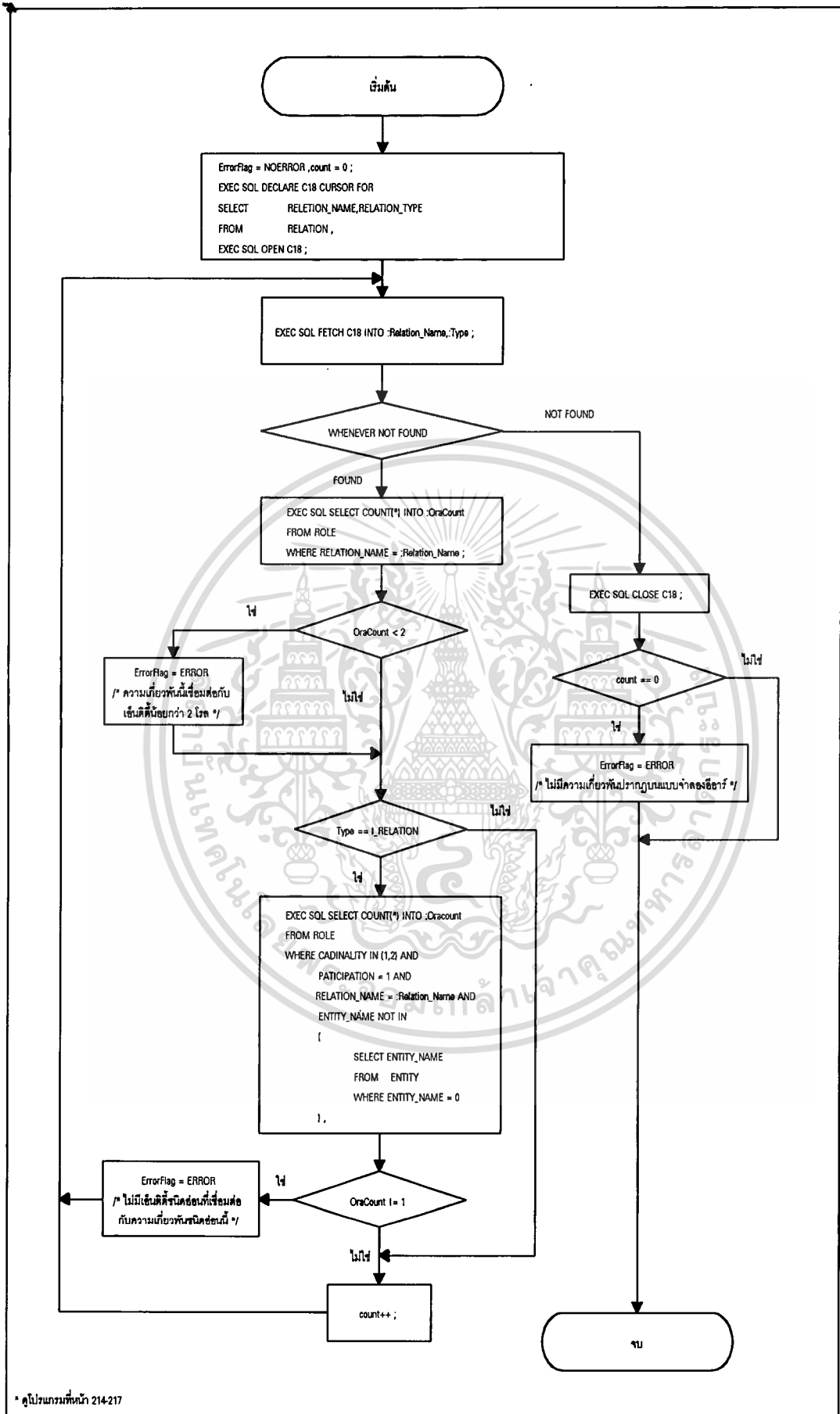
### 5.4.1 ส่วนตรวจสอบความถูกต้องของแบบจำลองอ็อร์

ส่วนตรวจสอบความถูกต้องของแบบจำลองอ็อร์ จะถูกเรียกมาทำงานเมื่อผู้ใช้งานได้เลือกเมนู First Normal Form จากเมนูหลัก Process ทำหน้าที่ตรวจสอบว่าแบบจำลองอ็อร์ที่ถูกวาดขึ้นนั้นถูกต้องหรือไม่โดยใช้วิธีการตรวจสอบข้อมูลของแบบจำลองอ็อร์ที่วาดขึ้นบนฐานข้อมูลอ็อร์เมทาดำตามกฎต่างๆ 6 ข้อดังต่อไปนี้คือ

- กฎข้อที่ 1 ต้องมีแอททริบิวต์บนแบบจำลองอ็อร์
- กฎข้อที่ 2 ทุกแอททริบิวต์ต้องมีการเชื่อมต่อกับวัตถุ
- กฎข้อที่ 3 ต้องมีความเกี่ยวพันบนแบบจำลองอ็อร์
- กฎข้อที่ 4 ทุกความเกี่ยวพันจะต้องเชื่อมต่อกับโวลอย่างน้อย 2 โวล
- กฎข้อที่ 5 ต้องมีเอ็นติตีบนแบบจำลองอ็อร์
- กฎข้อที่ 6 ทุกเอ็นติตีจะต้องเชื่อมต่อแอททริบิวต์ชนิดกุญแจ



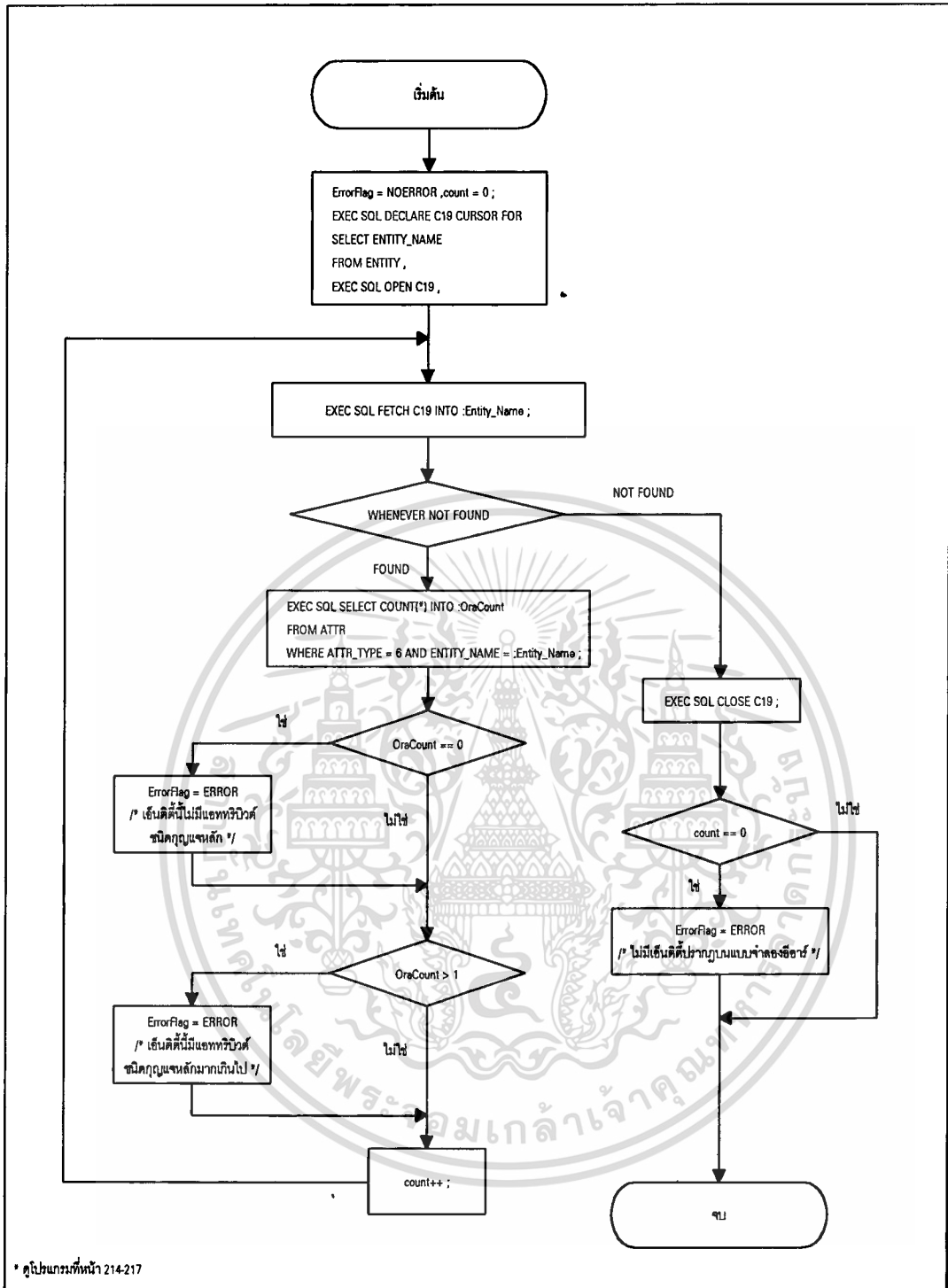
รูปที่ 5.16 แสดงผังการทำงานในการตรวจสอบความถูกต้องของแอททริบิวต์ตามกฎข้อที่ 1 และ 2



\* คู่มือโปรแกรมที่หน้า 214-217

รูปที่ 5.17 แสดงผังการทำงานในการตรวจสอบความถูกต้องของความสัมพันธ์ตามกฎข้อที่ 3 และ 4

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



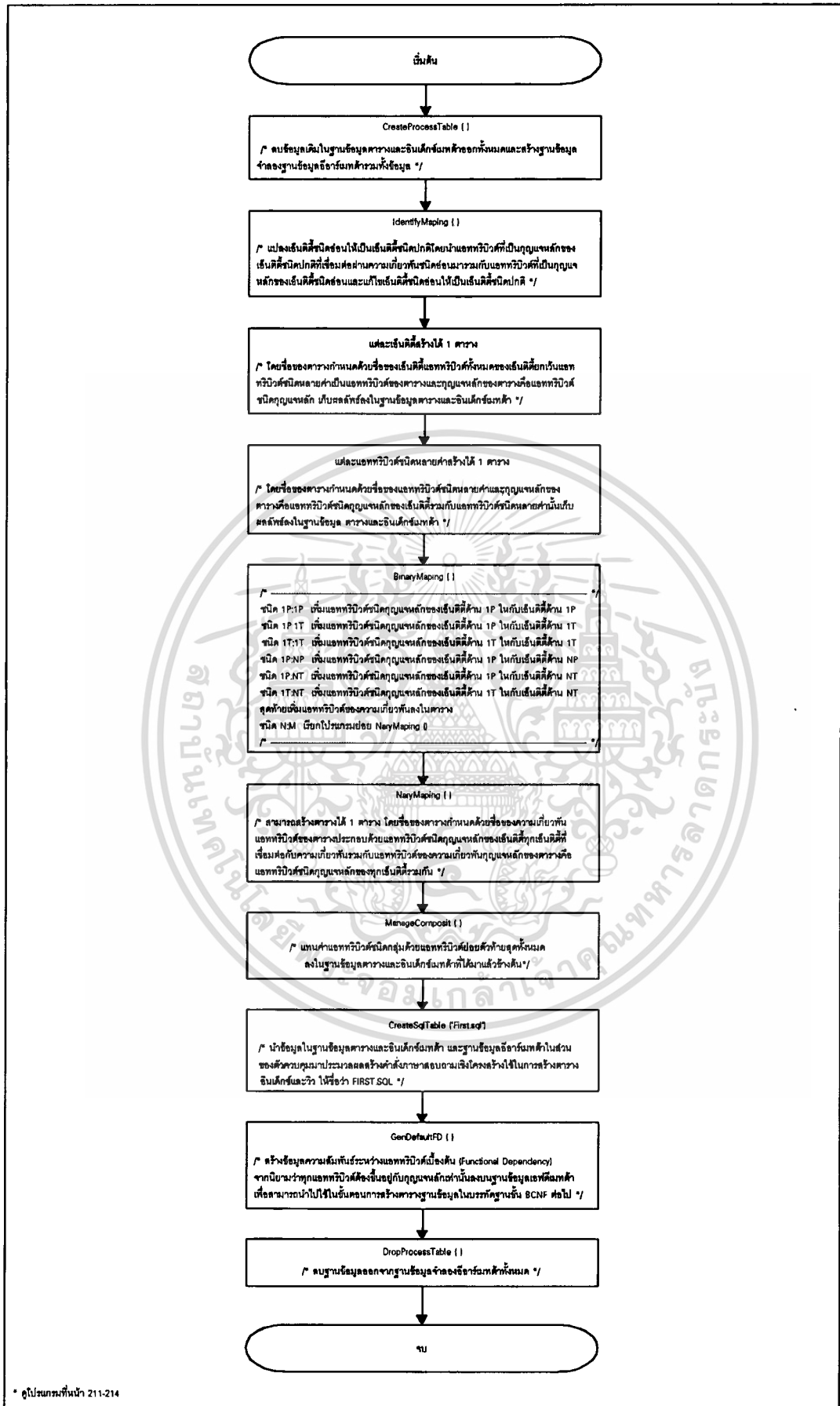
\* คู่มือโปรแกรมที่หน้า 214-217

รูปที่ 5.18 แสดงผังการทำงานในการตรวจสอบความถูกต้องของเอ็นติตี้ตามกฎข้อที่ 5 และ 6

### 5.4.2 ส่วนสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นแรก

ส่วนสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นแรก เป็นส่วนที่จะทำงานหลังจากได้ตรวจสอบจากข้อมูลแผนภาพอีอาร์ว่าถูกต้องโดยจะทำการประมวลผลข้อมูลในฐานข้อมูลอีอาร์เมทาดาและสร้างข้อมูลผลลัพธ์ที่ได้เก็บไว้ในฐานข้อมูลตารางและอินเด็กซ์เมทาดาตามกฎต่างๆที่ได้กล่าวมาแล้วในบทที่ 3 มีลักษณะการทำงานดังแสดงในรูปที่ 5.19

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ทางเชิงพาณิชย์ที่จัดทำขึ้นโดยผู้จัดทำเพื่อไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



\* คู่มือบทที่หน้า 211-214

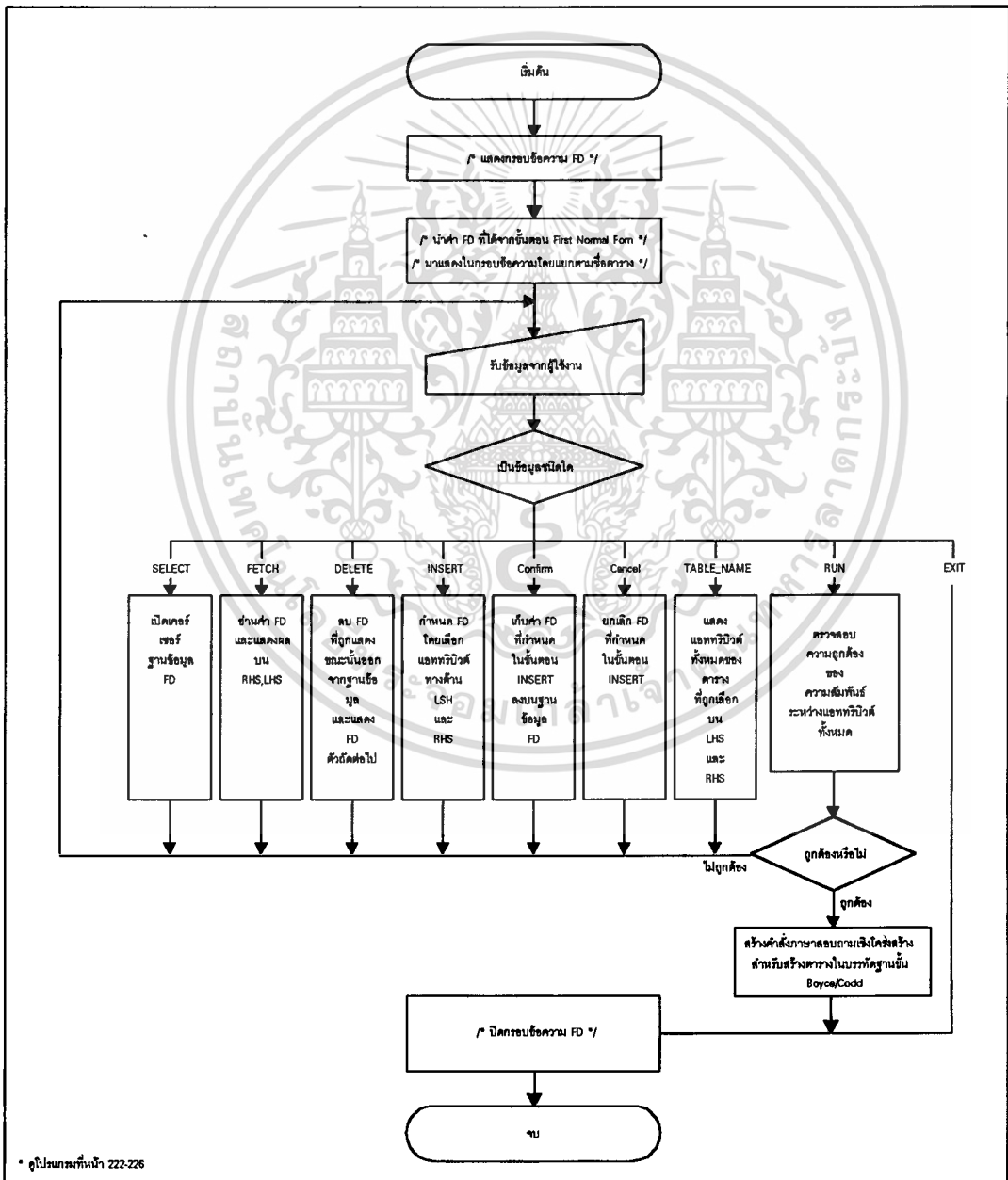
เอกสารรูปที่ 5.19 แสดงผังการทำงานในการสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นแรกจากแบบจำลองอีอาร์เอ็มขั้นตอนการคำนวณว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.5 หน่วยโปรแกรมประมวลผลข้อมูลสร้างตารางในบรรทัดฐานชั้น

### Boyce/Codd

#### 5.5.1 ส่วนติดต่อเอพีดี (Functional Dependency)

ส่วนติดต่อเอพีดี เป็นส่วนที่จะทำงานเมื่อผู้ใช้งานเลือกเมนู Boyce Codd Normal Form จากเมนูหลัก Process ทำหน้าที่แสดงความสัมพันธ์ระหว่างแอททริบิวต์ที่ถูกสร้างขึ้นในขั้นตอนการสร้างตารางฐานข้อมูลในบรรทัดฐานชั้นแรก สำหรับให้ผู้ใช้งานเพิ่ม ลบ แก้ไข เปลี่ยนแปลงความสัมพันธ์ระหว่างแอททริบิวต์ได้ตามต้องการ มีลักษณะการทำงานดังแสดงในรูปที่ 5.20



รูปที่ 5.20 แสดงผังการทำงานในส่วนติดต่อเอพีดี  
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.5.2 ส่วนตรวจสอบความถูกต้องของความสัมพันธ์ระหว่างแอททริบิวต์

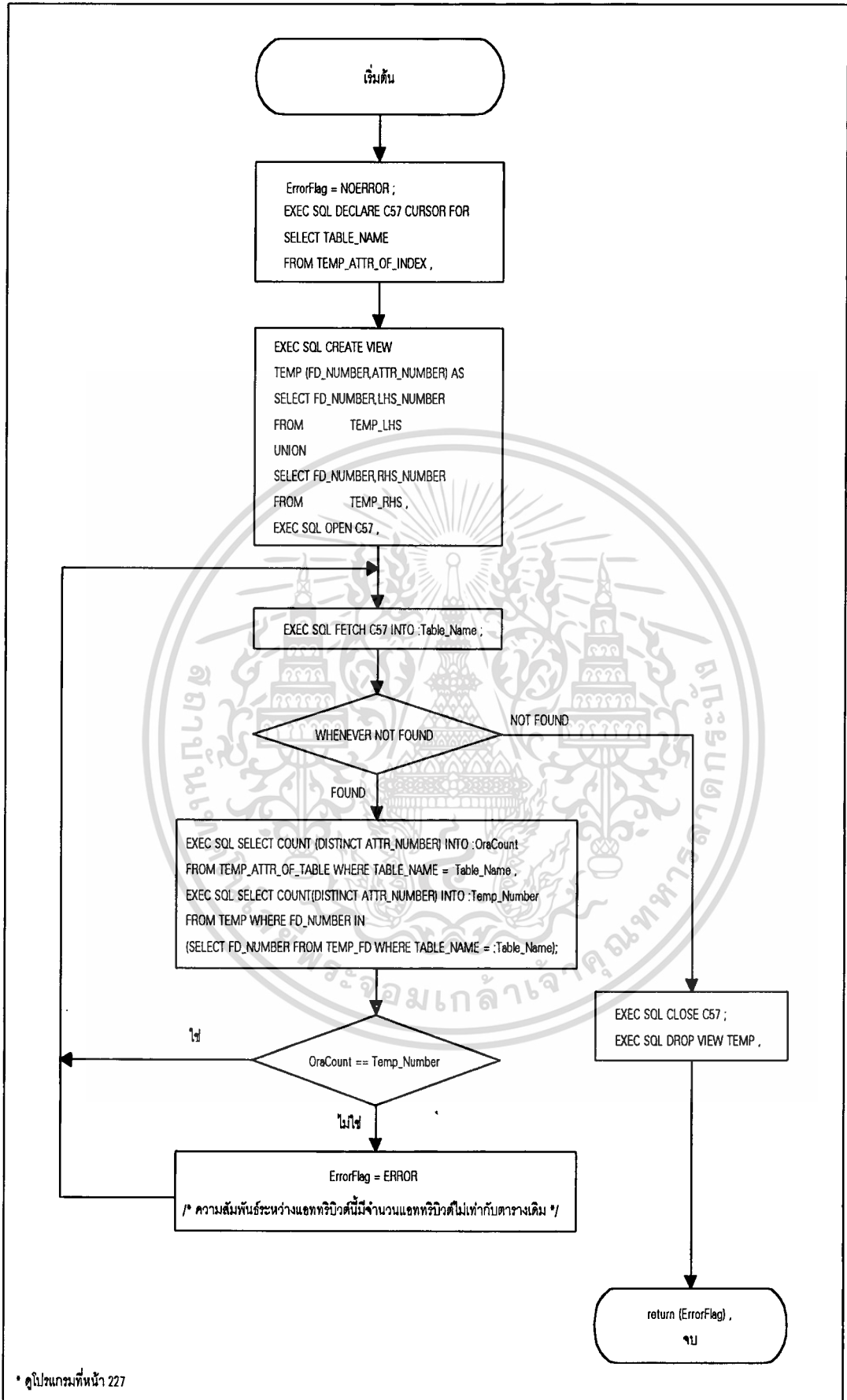
ส่วนตรวจสอบความถูกต้องของความสัมพันธ์ระหว่างแอททริบิวต์ จะถูกเรียกมาทำงานเมื่อผู้ใช้งานกดปุ่ม Confirm หรือ RUN ของกรอบข้อความ FD ทำหน้าที่ตรวจสอบว่าความสัมพันธ์ระหว่างแอททริบิวต์ที่ถูกแก้ไข เพิ่มเติม หรือเปลี่ยนแปลง ในส่วนติดต่อเอฟดีนั้นถูกต้องหรือไม่ โดยใช้วิธีการตรวจสอบจากข้อมูลภายในกรอบข้อความ FD เองและจากข้อมูลบนฐานข้อมูลเอฟดีเมทต้าตามกฎต่างๆ 2 ข้อดังต่อไปนี้คือ

**กฎข้อที่ 1** ความสัมพันธ์ระหว่างแอททริบิวต์จะต้องประกอบด้วยแอททริบิวต์ด้านซ้าย (Determinant) และแอททริบิวต์ด้านขวา (Dependent)

การตรวจสอบความถูกต้องของความสัมพันธ์ระหว่างแอททริบิวต์ตามกฎข้อที่ 1 นี้จะทำการตรวจสอบโดยมีขั้นตอนดังนี้คือ เมื่อผู้ใช้งานกดปุ่ม INSERT บนกรอบข้อความ FD แอททริบิวต์ที่อยู่ภายในลิสต์บ็อกซ์ Left Hand Side และ Right Hand Side จะยังไม่มีแถบสีดำพาดอยู่บนแอททริบิวต์ใดเลยแสดงว่ายังไม่มีทางเลือกแอททริบิวต์เกิดขึ้น ผู้ใช้งานจะต้องเลือกแอททริบิวต์ตามความสัมพันธ์ระหว่างแอททริบิวต์ที่ต้องการ ซึ่งแอททริบิวต์ที่อยู่ด้านซ้ายแสดงถึง Determinant และแอททริบิวต์ที่อยู่ด้านขวาแสดงถึง Dependent โดยการเลื่อนเมาส์ไปวางบนแอททริบิวต์ที่ต้องการแล้วกดปุ่มซ้ายของเมาส์แอททริบิวต์ที่ถูกเลือกก็จะปรากฏแถบสีดำพาดแอททริบิวต์นั้น (การเลือกสามารถเลือกแอททริบิวต์ได้ทั้งสองด้านโดยแต่ละด้านสามารถเลือกได้มากกว่าหนึ่งแอททริบิวต์ขึ้นอยู่กับความสัมพันธ์ระหว่างแอททริบิวต์ที่ต้องการ) เมื่อกำหนดความสัมพันธ์ระหว่างแอททริบิวต์ที่ต้องการเสร็จเรียบร้อยแล้วให้ผู้ใช้งานกดปุ่ม Confirm ข้อมูลความสัมพันธ์ระหว่างแอททริบิวต์นี้ก็จะถูกเก็บลงบนฐานข้อมูลเอฟดีเมทต้า แต่ก่อนที่จะมีการเก็บข้อมูลลงบนฐานข้อมูลเอฟดีเมทต้าจะต้องมีการตรวจสอบความถูกต้อง ตามกฎข้อที่ 1 ก่อนว่าถูกต้องหรือไม่ โดยทันทีที่ผู้ใช้งานกดปุ่ม Confirm ของกรอบข้อความ FD โปรแกรมก็จะส่งเมลเสกสอบถาม (LclItems = SendDlgMessage (hDlg, IDC\_LHS, LB\_GETSELCOUNT, 0, 0); และ RclItems = SendDlgMessage (hDlg, IDC\_RHS, LB\_GETSELCOUNT, 0, 0);) ไปยังลิสต์บ็อกซ์คอนโทรล Left Hand Side และ Right Hand Side ว่าจำนวนของแอททริบิวต์ที่ถูกเลือกภายในลิสต์บ็อกซ์ทั้งสองมีจำนวนเท่าใด ถ้าแอททริบิวต์ทั้งสองด้านไม่ถูกเลือก (แสดงว่า LclItems == 0 และ RclItems == 0) หรือแอททริบิวต์ถูกเลือกเพียงด้านใดด้านหนึ่ง (แสดงว่า LclItems == 0 หรือ RclItems == 0) แสดงว่าข้อมูลความสัมพันธ์ระหว่างแอททริบิวต์นี้ไม่ถูกต้องก็จะไม่ยอมให้เก็บข้อมูลลงบนฐานข้อมูลเอฟดีเมทต้า

**กฎข้อที่ 2** ความสัมพันธ์ระหว่างแอททริบิวต์ทั้งหมดในแต่ละตารางเมื่อนำแอททริบิวต์มารวมกันจะต้องประกอบด้วยแอททริบิวต์ของตารางเดิมครบทุกแอททริบิวต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

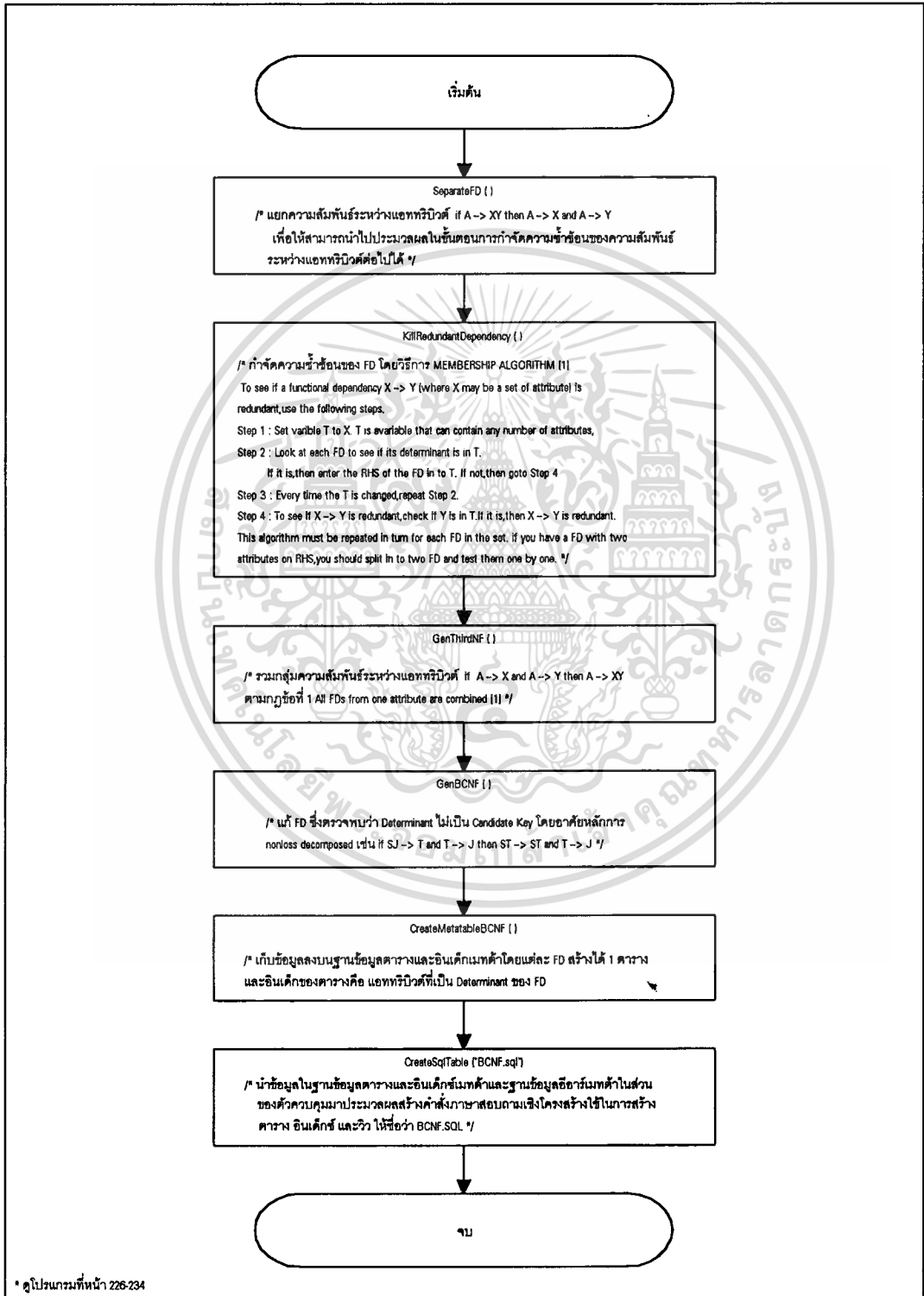


\* คู่มือกรมที่หน้า 227

รูปที่ 5.21 แสดงผังการทำงานในการตรวจสอบความถูกต้องของความสัมพันธ์ระหว่างแอตทริบิวต์ตามกฎข้อที่ 2 ด้านการคำนวณ  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.5.3 ส่วนสร้างตารางฐานข้อมูลในบรรทัดฐานชั้น Boyce/Codd

ส่วนสร้างตารางฐานข้อมูลในบรรทัดฐานชั้น Boyce/Codd เป็นส่วนที่จะทำงานเมื่อผู้ใช้งานกดปุ่ม RUN บนกรอบข้อความ FD ทำหน้าที่ประมวลผลข้อมูล FD ทั้งหมดแล้วสร้างตารางฐานข้อมูลในบรรทัดฐานชั้น Boyce/Codd มีลักษณะการทำงานดังแสดงในรูปที่ 5.22

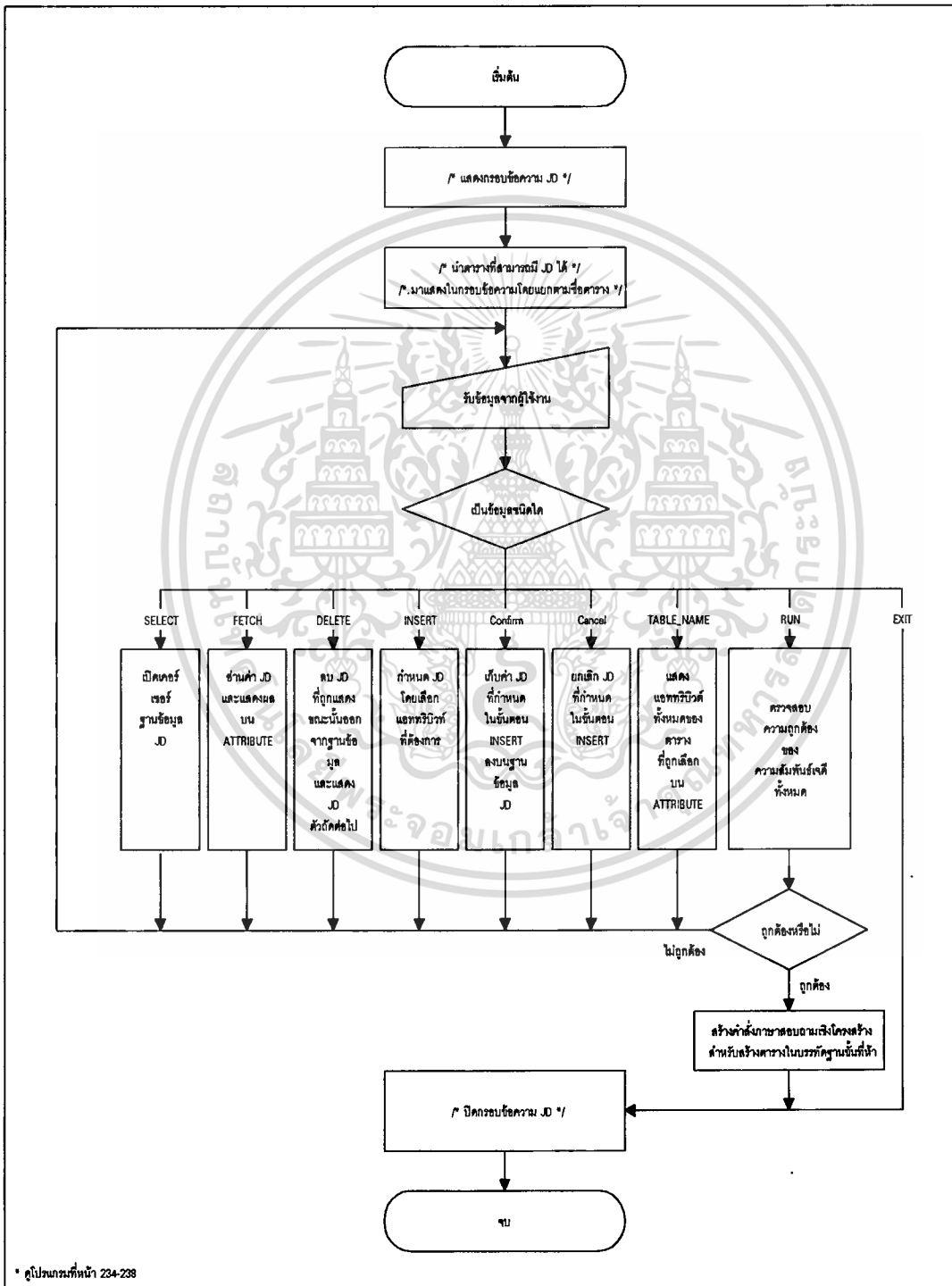


รูปที่ 5.22 แสดงผังการทำงานในส่วนสร้างตารางฐานข้อมูลในชั้น Boyce/Codd

## 5.6 หน่วยโปรแกรมประมวลผลข้อมูลสร้างตารางในบรรทัดฐานขั้นที่ห้า

### 5.6.1 ส่วนติดต่อเจติ (Join Dependency)

ส่วนติดต่อเจติ เป็นส่วนที่จะทำงานเมื่อผู้ใช้งานเลือกเมนู Fifth Normal Form จากเมนูหลัก Process ทำหน้าที่แสดงความสัมพันธ์เจติ สำหรับให้ผู้ใช้งานเพิ่ม ลบ แก้ไขเปลี่ยนแปลงความสัมพันธ์เจติได้ตามต้องการ มีลักษณะการทำงานดังแสดงในรูปที่ 5.23



รูปที่ 5.23 แสดงผังการทำงานในส่วนติดต่อเจติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.6.2 ส่วนตรวจสอบความถูกต้องของความสัมพันธ์เจดี

ส่วนตรวจสอบความถูกต้องของความสัมพันธ์เจดี จะถูกเรียกมาทำงานเมื่อผู้ใช้งาน กดปุ่ม Confirm หรือ RUN ของกรอบข้อความ JD ทำหน้าที่ตรวจสอบว่าความสัมพันธ์เจดีที่ถูกแก้ไข เพิ่มเติม หรือเปลี่ยนแปลง ในส่วนติดต่อเจดีนั้นถูกต้องหรือไม่ โดยใช้วิธีการตรวจสอบจากข้อมูลภายในกรอบข้อความ JD เองและจากข้อมูลบนฐานข้อมูลเจดีเมทาดาตามกฎต่างๆ 3 ข้อดังต่อไปนี้คือ

**กฎข้อที่ 1** การกำหนดแอททริบิวต์ในความสัมพันธ์เจดีในแต่ละครั้งจะต้องประกอบด้วยแอททริบิวต์จำนวนไม่ต่ำกว่า 2 แอททริบิวต์

การกำหนดความสัมพันธ์เจดีนั้นผู้ใช้งานจะเป็นผู้กำหนดว่ามีตารางใดบ้างที่สามารถนำมา Join กันแล้วเป็น Lossless Join ซึ่งกรอบข้อความ JD ทำหน้าที่เพียงช่วยให้ผู้ใช้งานกำหนดว่าแต่ละตารางนั้นประกอบด้วยแอททริบิวต์อะไรบ้าง ยกตัวอย่างเช่น ความสัมพันธ์เจดี  $JD*((S\#,P\#),(P\#,J\#),(J\#,S\#))$  ในรูปที่ 2.13 ซึ่งแอบแฝงอยู่ในตาราง SPJ แสดงให้เห็นว่าความสัมพันธ์เจดีนี้ประกอบด้วยตารางจำนวน 3 ตารางคือ

1. ตาราง SP ประกอบด้วยแอททริบิวต์จำนวน 2 แอททริบิวต์ได้แก่ S# กับ P#
2. ตาราง PJ ประกอบด้วยแอททริบิวต์จำนวน 2 แอททริบิวต์ได้แก่ P# กับ J#
3. ตาราง JS ประกอบด้วยแอททริบิวต์จำนวน 2 แอททริบิวต์ได้แก่ J# กับ S#

ถ้าผู้ใช้งานต้องการกำหนดความสัมพันธ์เจดีนั้นบนกรอบข้อความ JD จะต้องกำหนดได้ด้วยการเลือกแอททริบิวต์ S# กับ P# เป็นลำดับแรก P# กับ J# เป็นลำดับที่สอง และ J# กับ S# เป็นลำดับสุดท้าย จะเห็นว่าถ้ามีการเลือกแอททริบิวต์เพียงแอททริบิวต์เดียวนั้นจะทำให้ได้ตารางที่มีแอททริบิวต์เดียวทำให้เมื่อนำไป Join กับตารางใดก็ตามจะทำให้ได้ตารางนั้นซึ่งไม่มีประโยชน์ ดังนั้นจึงต้องมีการตรวจสอบความถูกต้องของความสัมพันธ์เจดี ตามกฎข้อที่ 1 โดยมีขั้นตอนดังนี้คือ เมื่อผู้ใช้งานกดปุ่ม INSERT บนกรอบข้อความ JD แอททริบิวต์ที่อยู่ในรายชื่อแอททริบิวต์ ATTRIBUTE จะยังไม่มีแถบสีดำพาดอยู่บนแอททริบิวต์ใดเลยแสดงว่ายังไม่มีการเลือกแอททริบิวต์เกิดขึ้น ผู้ใช้งานจะต้องเลือกแอททริบิวต์โดยการเลื่อนเมาส์ที่ไปวางบนแอททริบิวต์ที่ต้องการแล้วกดปุ่มซ้ายของเมาส์แอททริบิวต์ที่ถูกเลือกก็จะปรากฏแถบสีดำพาดแอททริบิวต์นั้น เมื่อเลือกแอททริบิวต์เสร็จเรียบร้อยแล้วให้ผู้ใช้งานกดปุ่ม Confirm ข้อมูลความสัมพันธ์เจดีนี้ก็จะถูกเก็บลงบนฐานข้อมูลเจดีเมทาดา แต่ก่อนที่จะมีการเก็บข้อมูลลงบนฐานข้อมูลเจดีเมทาดาจะต้องมีการตรวจสอบความถูกต้อง ตามกฎข้อที่ 1 ก่อนว่าถูกต้องหรือไม่ โดยทันทีที่ผู้ใช้งานกดปุ่ม Confirm ของกรอบข้อความ JD โปรแกรมก็จะส่งเมสเสจสอบถาม (cltems = SendDlgMessage (hDlg, IDC\_ATTR, LB\_GETSELCOUNT, 0, 0);) ไปยังรายชื่อแอททริบิวต์คอนโทรล ATTRIBUTE ว่าจำนวนของแอททริบิวต์ที่ถูกเลือกภายในรายชื่อแอททริบิวต์ทั้งหมดมี

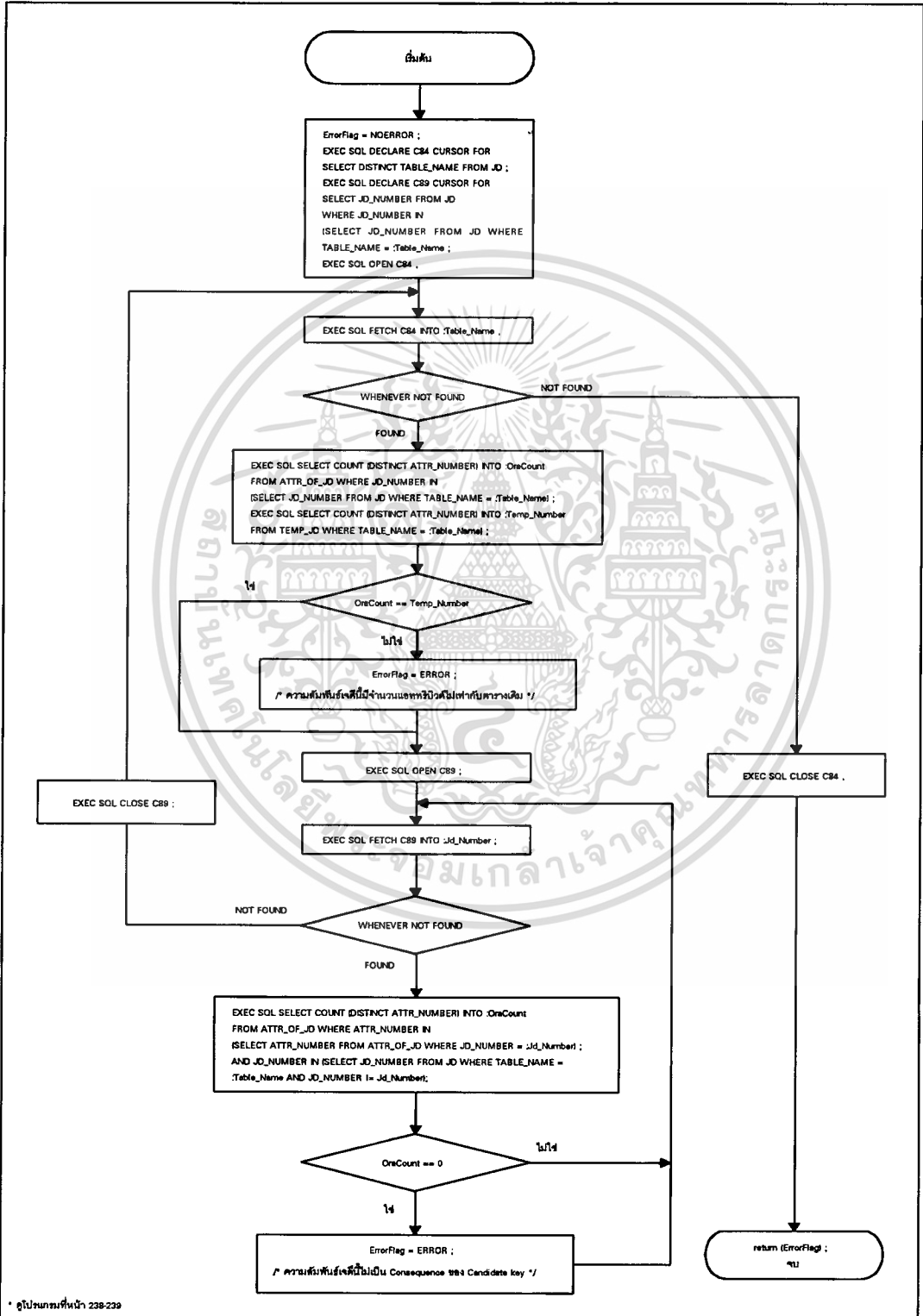
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนเท่าใดถ้าแอททริบิวต์ที่ถูกเลือกมีจำนวนน้อยกว่าสองแอททริบิวต์ (แสดงว่า citems < 2)

แสดงว่าข้อมูลไม่ถูกต้องก็จะไม่ยอมให้เก็บข้อมูลลงบนฐานข้อมูลเจดีเมทต้า

กฎข้อที่ 2 ความสัมพันธ์เจดีจะต้องครอบคลุมทุกแอททริบิวต์

กฎข้อที่ 3 ความสัมพันธ์เจดีจะต้องเป็น Consequence ของ Candidate Key

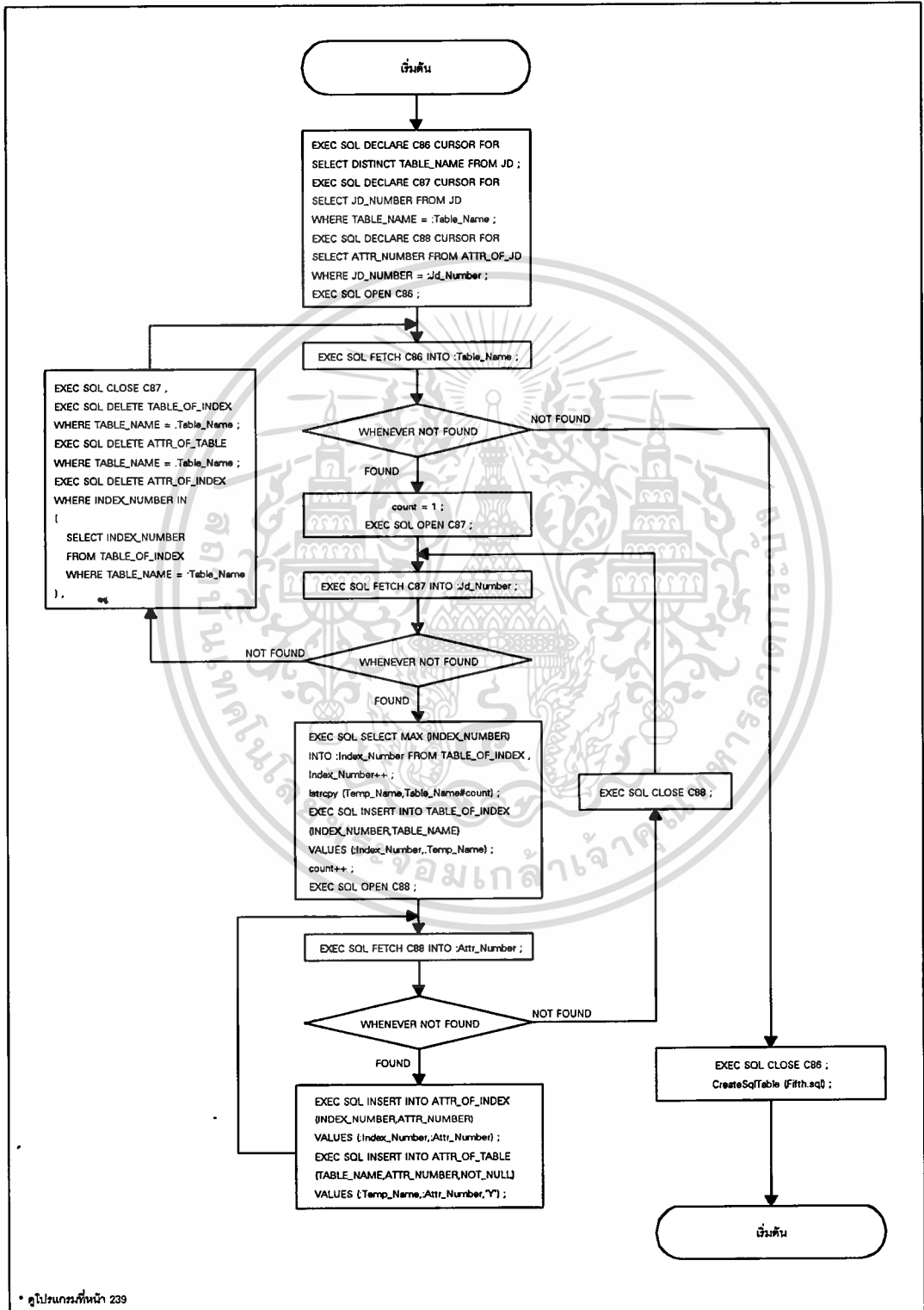


\* ดูโปรแกรมที่หน้า 239-239

รูปที่ 5.24 แสดงผังการทำงานในการตรวจสอบความถูกต้องของความสัมพันธ์เจดีตามกฎข้อที่ 2 และ 3 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.6.3 ส่วนสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นที่ห้า

ส่วนสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นที่ห้า เป็นส่วนที่จะทำงานเมื่อผู้ใช้งานกดปุ่ม RUN บนกรอบข้อความ JD ทำหน้าที่ประมวลผลข้อมูล JD ทั้งหมดแล้วสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นที่ห้า มีลักษณะการทำงานดังแสดงในรูปที่ 5.25



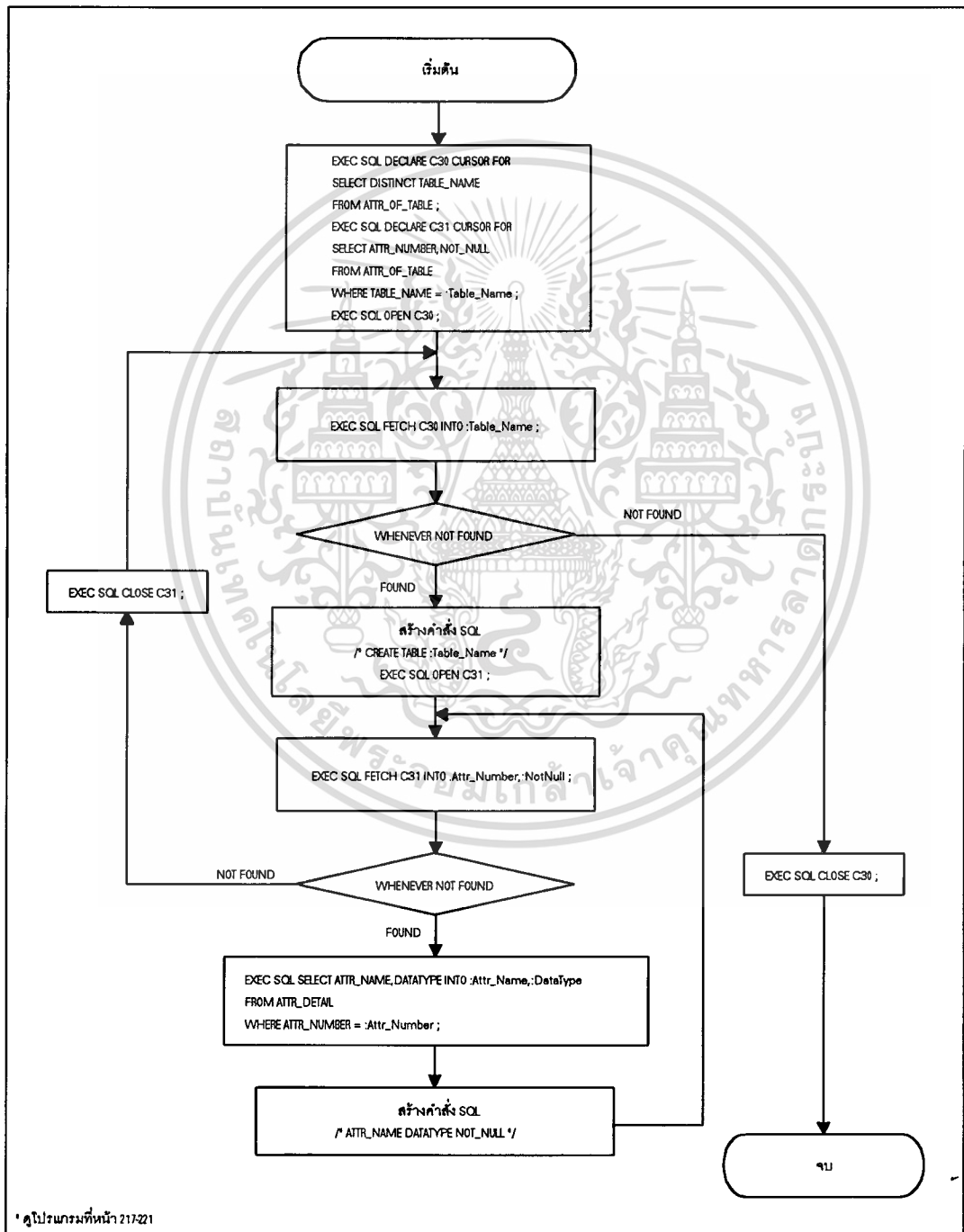
รูปที่ 5.25 แสดงผังการทำงานในส่วนสร้างตารางฐานข้อมูลในขั้นที่ห้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.7 หน่วยสร้างภาษาสอบถามเชิงโครงสร้างของระบบ ER/5

### 5.7.1 ส่วนสร้างตาราง

ส่วนสร้างตาราง ทำหน้าที่สร้างคำสั่งภาษาสอบถามเชิงโครงสร้าง เพื่อใช้ในการสร้างตาราง โดยจะทำหน้าที่แปลงข้อมูลที่อยู่บนฐานข้อมูลตารางและอินเด็กซ์เมทาดั (ข้อมูลของฐานข้อมูลตารางและอินเด็กซ์เมทาดัถูกกำหนดจากขั้นตอนการสร้างตารางในบรรทัดฐานขั้นแรก ชั้น Boyce/Codd หรือ ชั้นที่ห้า) มีลักษณะการทำงานดังแสดงในรูปที่ 5.26

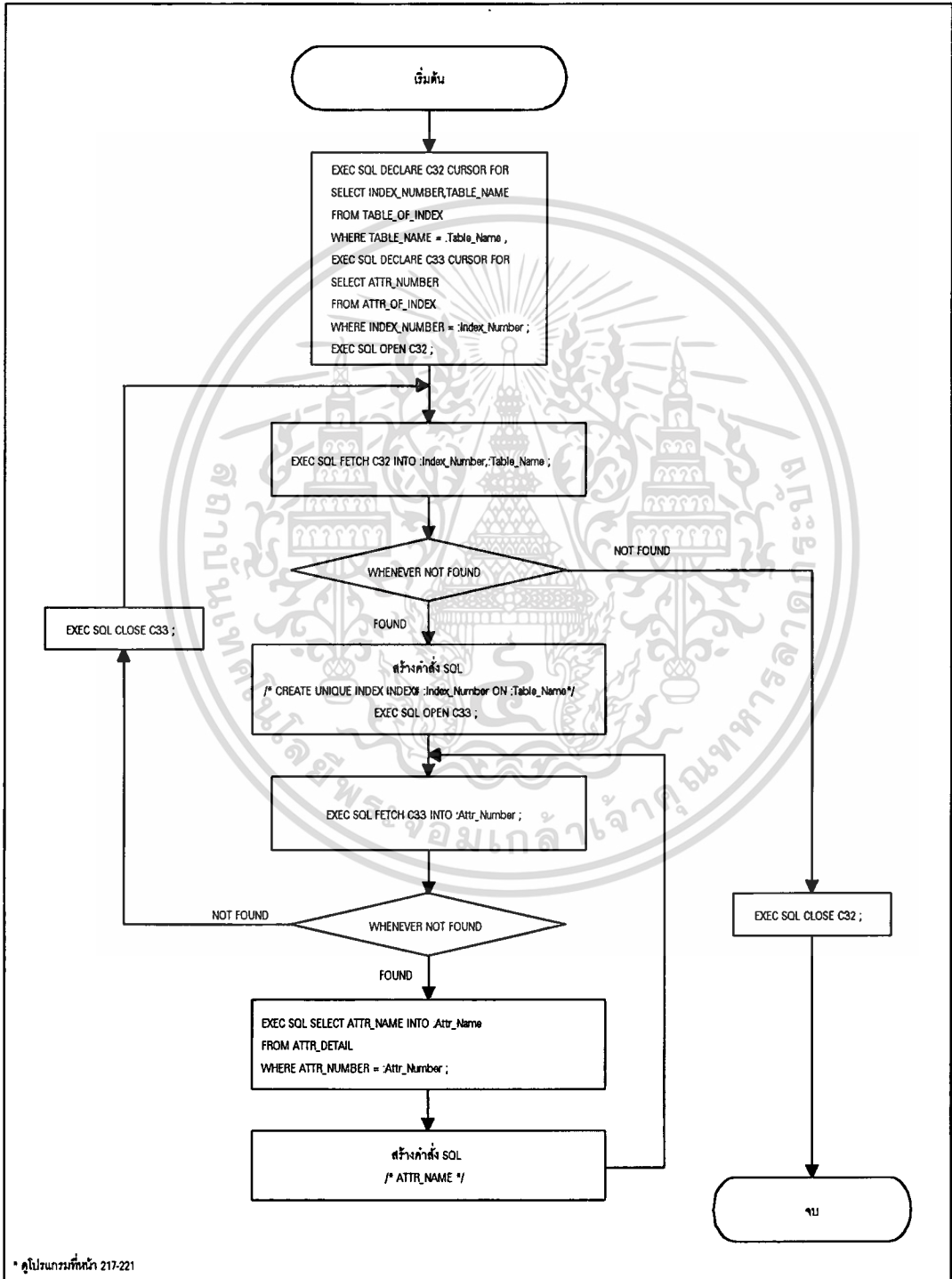


รูปที่ 5.26 แสดงผังการทำงานในส่วนสร้างตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.7.2 ส่วนสร้างอินเด็กซ์

ส่วนสร้างอินเด็กซ์ ทำหน้าที่สร้างคำสั่งภาษาสอบถามเชิงโครงสร้าง เพื่อใช้ในการสร้างอินเด็กซ์ โดยจะทำหน้าที่แปลงข้อมูลที่อยู่บนฐานข้อมูลตารางและอินเด็กซ์เมทาดา (ข้อมูลของฐานข้อมูลตารางและอินเด็กซ์เมทาดาถูกกำหนดจากขั้นตอนการสร้างตารางในบรรทัดฐานขั้นแรก ชั้น Boyce/Codd หรือ ชั้นที่ห้า) มีลักษณะการทำงานดังแสดงในรูปที่ 5.27

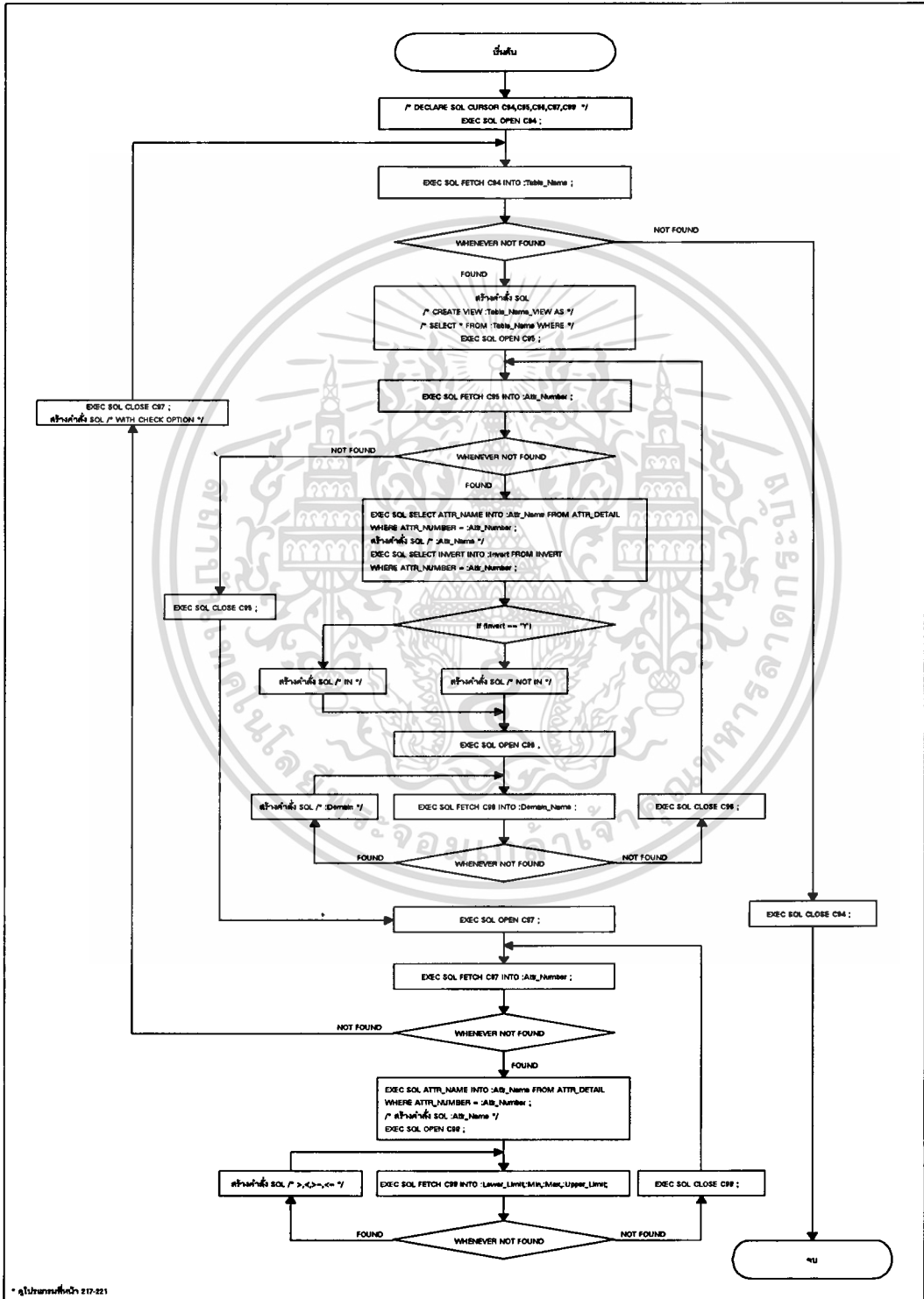


รูปที่ 5.27 แสดงผังการทำงานในส่วนสร้างอินเด็กซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.7.3 ส่วนสร้างวิว

ส่วนสร้างวิว ทำหน้าที่สร้างคำสั่งภาษาสอบถามเชิงโครงสร้างเพื่อใช้ในการสร้างวิวเพื่อควบคุมความถูกต้องของข้อมูลภายในแอททริบิวต์ที่มีตัวควบคุมกำกับอยู่ โดยจะทำหน้าที่แปลงข้อมูลที่อยู่บนฐานข้อมูลตารางและอินเด็กซ์เมทด้ากับข้อมูลที่อยู่บนฐานข้อมูลอ็อบเจกต์เมทด้าในส่วนของตัวควบคุม มีลักษณะการทำงานดังแสดงในรูปที่ 5.28



รูปที่ 5.28 แสดงผังการทำงานในส่วนสร้างวิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### การทดสอบและประเมินผลระบบ ER/5

#### 6.1 คุณสมบัติโดยทั่วไปของระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูล

ปัจจุบันนี้การออกแบบฐานข้อมูลส่วนใหญ่จะใช้แบบจำลองฐานข้อมูลช่วยในการออกแบบทำให้มีการคิดค้นแบบจำลองฐานข้อมูลขึ้นมากมาย แบบจำลองฐานข้อมูลที่ถูกคิดค้นขึ้นโดยมากจะมีลักษณะเป็นภาพที่แสดงถึงส่วนประกอบและความสัมพันธ์ของข้อมูลทั้งหมดจากนั้นก็จะต้องนำภาพดังกล่าวไปสร้างเป็นตารางฐานข้อมูลตามหลักและวิธีการของแต่ละแบบจำลองเพื่อให้ได้คำสั่งในการสร้างตารางฐานข้อมูลที่สามารถนำไปใช้งานกับระบบจัดการฐานข้อมูลต่างๆได้

การสร้างระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลจะต้องกำหนดว่าใช้กับแบบจำลองฐานข้อมูลชนิดใด ระบบดังกล่าวก็จะสามารถเข้าใจสัญลักษณ์ต่างๆที่ปรากฏอยู่บนรูปภาพ สามารถนำภาพไปสร้างตารางฐานข้อมูลได้อย่างถูกต้อง ดังนั้นระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลจึงจะต้องมีความสามารถขั้นพื้นฐานดังต่อไปนี้คือ

##### - ความสามารถทางกราฟฟิก

ระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลจะต้องสามารถรับอินพุตในลักษณะของกราฟฟิกจากผู้ใช้งาน สามารถที่จะแสดงภาพกราฟฟิคนั้นออกทางจอภาพ สามารถที่จะพิมพ์ภาพกราฟฟิกออกทางเครื่องพิมพ์ และสามารถจัดเก็บลงบนแฟ้มข้อมูลเพื่อนำมาใช้ปรับปรุงแก้ไขได้ในภายหลัง

##### - ความสามารถในการสร้างตารางฐานข้อมูลจากแบบจำลอง

เมื่อผู้ใช้งานได้ออกแบบจำลองฐานข้อมูลเสร็จเป็นที่เรียบร้อยแล้ว ระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลจะต้องสามารถตรวจสอบได้ว่าแบบจำลองที่ถูกออกแบบขึ้นนั้นมีความถูกต้องหรือไม่ เมื่อไม่ถูกต้องก็สามารถที่จะให้ผู้ใช้งานนำกลับไปแก้ไขใหม่ ถ้าถูกต้องแล้วก็สามารถแปลงรูปภาพแบบจำลองฐานข้อมูลนั้นให้เป็นตารางฐานข้อมูลเพื่อนำไปใช้งานกับตัวจัดการฐานข้อมูลได้จริง โดยที่ผู้ใช้งานไม่จำเป็นต้องทราบขั้นตอนและวิธีการแปลงรูปภาพแบบจำลองฐานข้อมูลให้เป็นตารางฐานข้อมูลมาก่อน

##### - ความสามารถในการควบคุมความถูกต้องของข้อมูล

ในขั้นตอนการออกแบบฐานข้อมูลด้วยแบบจำลองฐานข้อมูล เมื่อผู้ใช้งานได้กำหนดตัวควบคุม (Constraint) ลงบนแบบจำลองฐานข้อมูล ระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลจะต้องสามารถทำให้ตัวควบคุมที่ผู้ใช้งานกำหนดขึ้นนั้นมีผลบังคับใช้ได้จริงในขณะนำไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

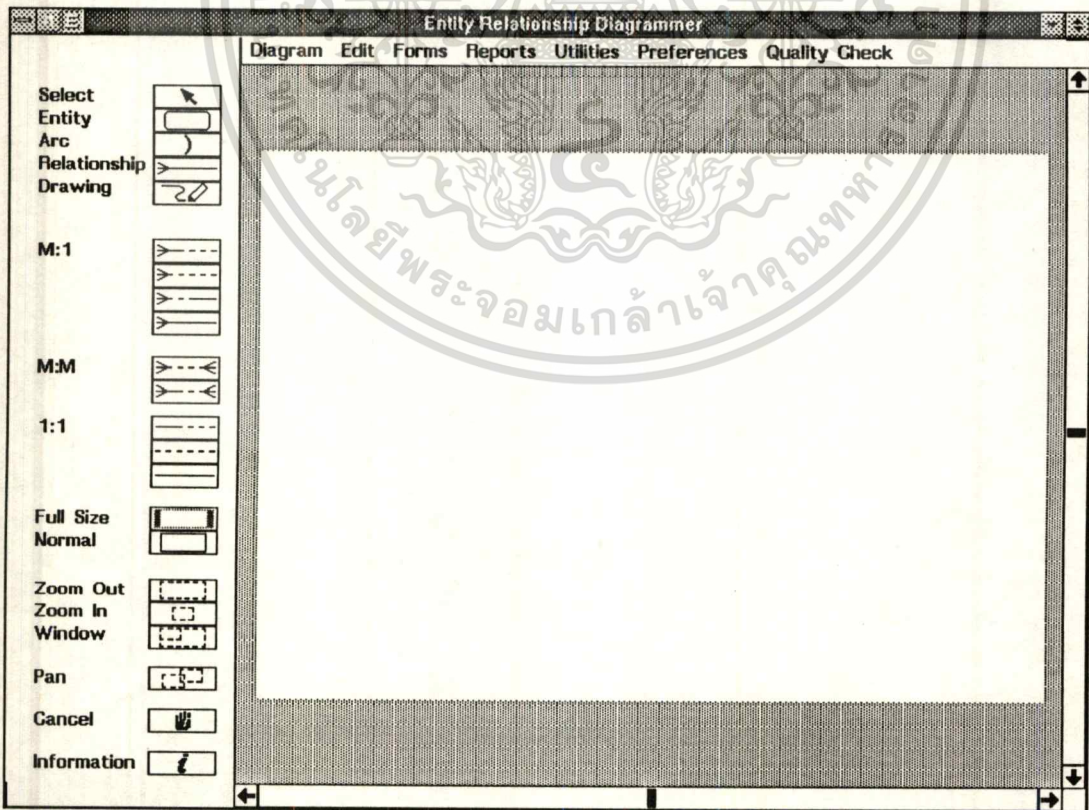
## 6.2 ระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลของตัวจัดการฐานข้อมูล

### ORACLE

ระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลของตัวจัดการฐานข้อมูล ORACLE เป็นระบบที่สร้างขึ้นโดย บริษัท ออราเคิล จำกัด มีวัตถุประสงค์ก็เพื่อที่จะสนับสนุนตัวจัดการฐานข้อมูลออราเคิลซึ่งเป็นสินค้าหลัก ใช้แบบจำลองอีอาร์ที่ดัดแปลงขึ้นเองเป็นอินพุต ดังนั้นผู้ใช้งานที่ต้องการใช้ระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลของตัวจัดการฐานข้อมูล ORACLE [5] นี้จำเป็นที่จะต้องศึกษาแบบจำลองอีอาร์ที่ดัดแปลงขึ้นนี้เพิ่มเติมจากคู่มือการใช้งานเพื่อที่สามารถออกแบบได้อย่างถูกต้อง

จากรูปที่ 6.1 แสดงหน้าจอของระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลของตัวจัดการฐานข้อมูลออราเคิล ซึ่งประกอบด้วย

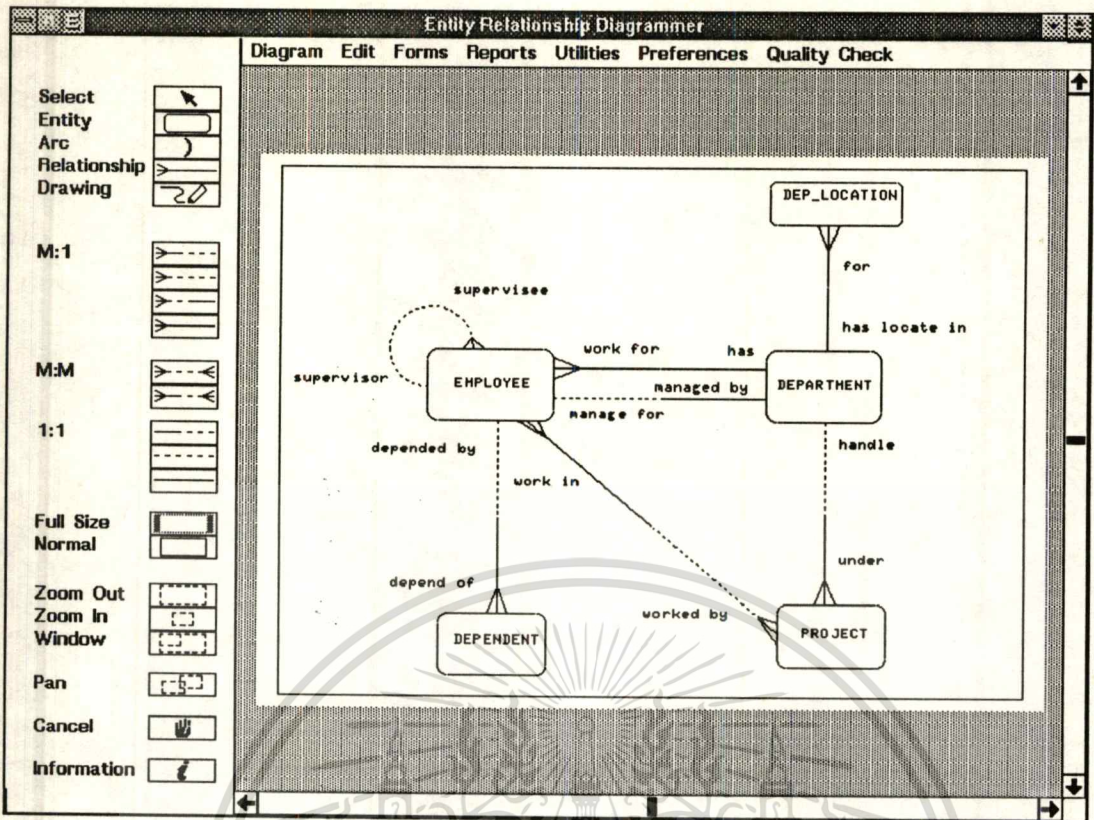
- 1.เมนูการประมวลผลต่างๆ อยู่ทางด้านบนของจอภาพ
2. สัญลักษณ์ต่างๆของแบบจำลองอีอาร์อยู่ทางด้านซ้ายมือใช้เป็นเครื่องมือช่วยในการวาดแบบจำลองอีอาร์
- 3.พื้นที่ทำงานเป็นพื้นที่สำหรับผู้ใช้งานวาดแบบจำลองอีอาร์ลงบนพื้นที่ว่าง ดังแสดงในรูปที่ 6.2



รูปที่ 6.1 แสดงหน้าจอของระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลของตัวจัดการฐานข้อมูลออราเคิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

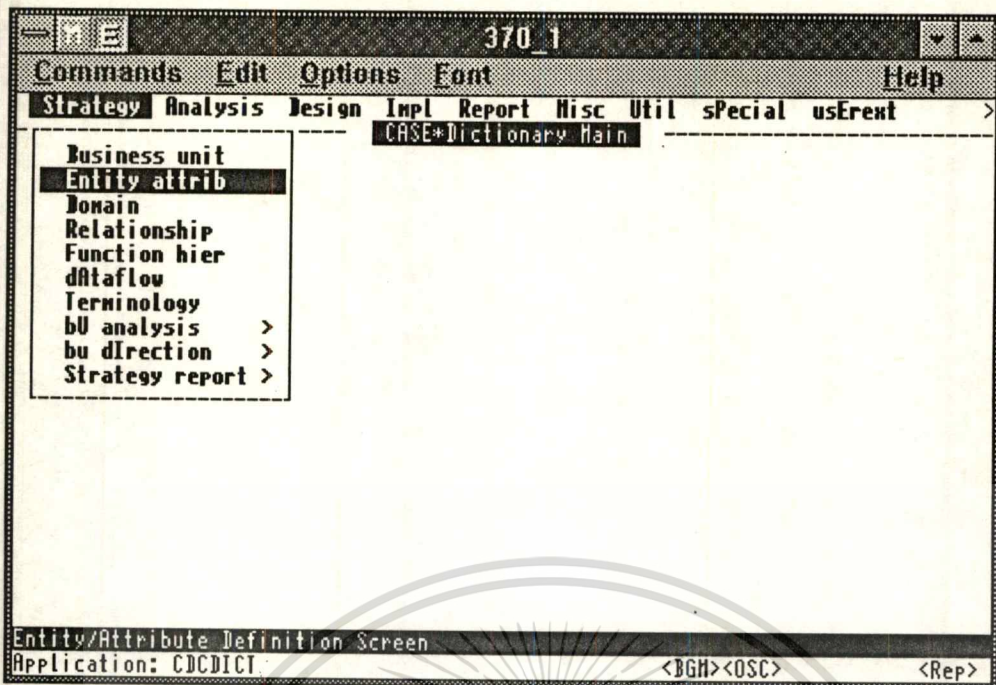
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



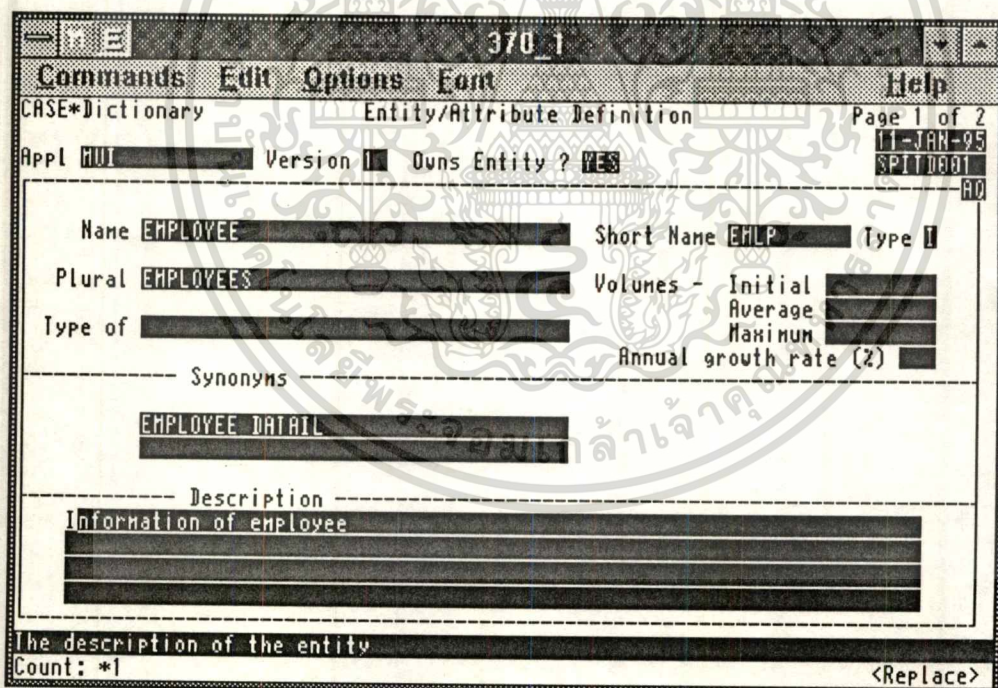
รูปที่ 6.2 แสดงแบบจำลองอ็อบเจกต์ที่ถูกต้อง

เนื่องจากแบบจำลองอ็อบเจกต์ของตัวจัดการฐานข้อมูลออบเจกต์นี้ไม่สามารถที่จะแสดงแอททริบิวต์ต่างๆของเอนิตีบนแบบจำลองอ็อบเจกต์ได้ดังนั้นหลังจากที่ผู้ใช้งานได้วาดแบบจำลองอ็อบเจกต์เสร็จเป็นที่เรียบร้อยแล้วผู้ใช้งานจะต้องกำหนดว่าในแต่ละเอนิตีประกอบด้วยแอททริบิวต์อะไรบ้างโดยการเลือกเมนู Entity attrib ดังแสดงในรูปที่ 6.3 เมื่อเข้าไปยังเมนูนี้แล้วก็จะปรากฏชื่อของเอนิตีทั้งหมดให้เห็น จากนั้นก็ให้เลือกเอนิตีที่ต้องการกำหนดแอททริบิวต์ จากรูปที่ 6.4 แสดงการเลือกกำหนดแอททริบิวต์ให้กับเอนิตี EMPLOYEE เมื่อเลือกเอนิตีแล้วก็สามารถที่จะกำหนดแอททริบิวต์ได้ จากรูปที่ 6.5 แสดงการกำหนดแอททริบิวต์ SSN SALARY และ SEX (สามารถกำหนดแอททริบิวต์ได้จำนวนมากกว่านี้แต่จะแสดงให้เห็นได้บนจอภาพเพียงครั้งละ 3 แอททริบิวต์เท่านั้น) ลงบนเอนิตี EMPLOYEE ในการกำหนดแอททริบิวต์สามารถที่จะกำหนดตัวควบคุมลงบนแอททริบิวต์ที่ต้องการควบคุมขอบเขตของข้อมูล (Domain Constraint) ได้อีกด้วยโดยอ้างถึงด้วยชื่อของตัวควบคุม เช่น แอททริบิวต์ SEX มีตัวควบคุมชื่อ SEX\_DOMAIN กำกับอยู่ แอททริบิวต์ SALARY มีตัวควบคุมชื่อ SALARY\_DOMAIN กำกับอยู่ เป็นต้น เมื่อกำหนดแอททริบิวต์ให้กับเอนิตี EMPLOYEE เป็นที่เรียบร้อยแล้ว ก็จะต้องกำหนดแอททริบิวต์ให้กับเอนิตีอื่นๆที่เหลืออยู่ให้ครบทุกเอนิตีเพื่อความถูกต้องของแบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3 แสดงเมนูสำหรับกำหนดแอททริบิวต์ให้กับเอนิตี



รูปที่ 6.4 แสดงการเลือกกำหนดแอททริบิวต์ให้กับเอนิตี EMPLOYEE

หลังจากที่กำหนดว่าแต่ละเอนิตีประกอบด้วยแอททริบิวต์อะไรบ้างเสร็จเรียบร้อยแล้วต่อไปก็ต้องตามไปกำหนดขอบเขตการควบคุมของตัวควบคุมทุกตัวเพื่อความถูกต้องของแอททริบิวต์ที่มีตัวควบคุมกำกับอยู่โดยเลือกเมนู Domain ดังแสดงในรูปที่ 6.6 หลังจากนั้นก็ให้เลือกชื่อของตัวควบคุมที่ต้องการกำหนดขอบเขตการควบคุม จากรูปที่ 6.7 แสดงการดำเนินการคำนวณราคา

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกกำหนดการควบคุมให้กับตัวควบคุมชื่อ SEX\_DOMAIN ว่าแอททริบิวต์ SEX จะต้องเป็นข้อมูล 'M' หรือ 'F' เท่านั้น และจากรูปที่ 6.8 แสดงการเลือกกำหนดการควบคุมให้กับตัวควบคุมชื่อ SALARY\_DOMAIN ว่าแอททริบิวต์ SALARY จะต้องมียู่ในช่วง 100 ถึง 500 กับ 1000 ถึง 5000 บาท เท่านั้น เป็นต้น

370\_1

Commands Edit Options Font Help

CASE\*Dictionary Entity/Attribute Definition Page 2 of 2

11-JAN-95  
SPITDAI  
AD

Appl HUI Version 1 Ovns Entity ? YES

Entity Name EMPLOYEE

Attribute Name	Opt?	Format	Length	Create In UID?	In UIDs
SALARY	Y	NUMBER			0
SEX	Y	CHAR	1		0
SSN	Y	NUMBER			1

The name of a domain this attribute is part of  
Count: \*8 <List><Replace>

รูปที่ 6.5 แสดงแอททริบิวต์บางส่วนของเอนิตี EMPLOYEE ที่กำหนดขึ้น

370\_1

Commands Edit Options Font Help

Strategy Analysis Design Impl Report Misc Util sPecial usEnt

CASE\*Dictionary Main

- Business unit
- Entity attrib
- Domain
- Relationship
- Function hier
- dataflow
- Terminology
- bu analysis >
- bu direction >
- Strategy report >

Domain Definition Screen  
Application: CDCDICT <BGH><OSCC> <Rep>

รูปที่ 6.6 แสดงเมนูสำหรับกำหนดตัวควบคุมให้กับแอททริบิวต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

370\_1

Commands Edit Options Font Help

CASE\*Dictionary Domain Definition Page 2 of 2

11-JAN-95  
SPIT D001  
AD

Appl AUT Version 1

Domain SEX\_DOMAIN

----- Valid Domain Values -----

Value	High value	Abbreviation
M		MALE
	Meaning	GENDER
F		FEMALE
	Meaning	GENDER

----- Validation Rules -----

The meaning for the value or range  
Count: \*0 <Replace>

รูปที่ 6.7 แสดงเมนูสำหรับกำหนดตัวควบคุม SEX ให้กับแอททริบิวต์ SEX (ชนิดตัวอักษร)

370\_1

Commands Edit Options Font Help

CASE\*Dictionary Domain Definition Page 2 of 2

11-JAN-95  
SPIT D001  
AD

Appl AUT Version 1

Domain SALARY\_DOMAIN

----- Valid Domain Values -----

Value	High value	Abbreviation
100	500	
	Meaning	BAHT
1000	5000	
	Meaning	BAHT

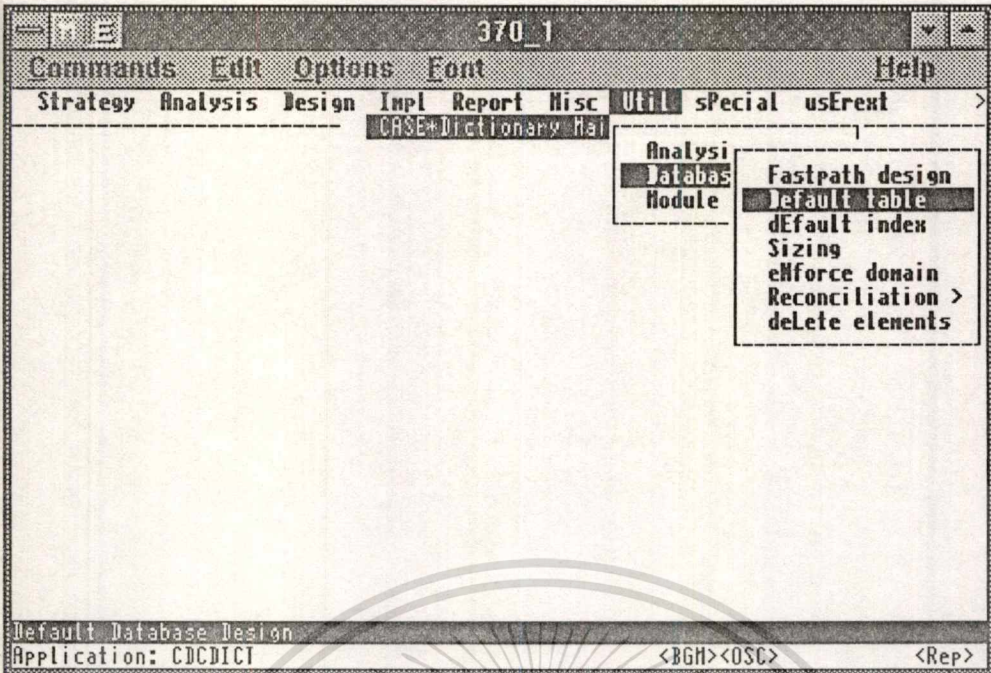
----- Validation Rules -----

The meaning for the value or range  
Count: \*2 <Replace>

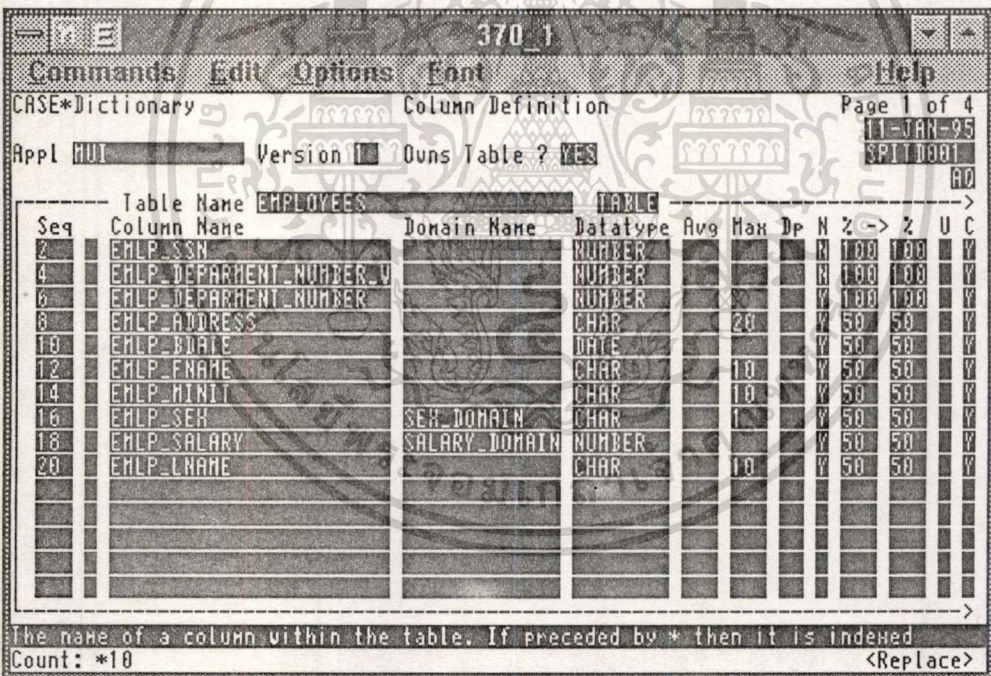
รูปที่ 6.8 แสดงเมนูสำหรับกำหนดตัวควบคุม SALARY ให้กับแอททริบิวต์ SALARY (ชนิดตัวเลข)

จากนั้นก็ถึงขั้นตอนของการแปลงแบบจำลองอ็อบเจกต์ให้เป็นตารางฐานข้อมูลโดยการเลือกเมนู Default table มาใช้งาน ดังแสดงในรูปที่ 6.9 ผลลัพธ์ที่ได้จะออกมาเป็น 2 แบบคือ Table Definition และ Data Definition Language ดังแสดงในรูปที่ 6.10 และ 6.11 ตามลำดับสามารถนำผลลัพธ์ไปใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.9 แสดงเมนูสำหรับสร้างตารางฐานข้อมูล



รูปที่ 6.10 แสดงผลลัพธ์ในรูปของ Table Definition

```

REM This ORACLE V6 RDBMS command file was generated by CASE*Dictionary
REM on 07-JAN-95
REM For application system MUI version 1
SET SCAN OFF
REM Objects being generated in this file are:-

REM TABLE
REM   DEPARTMENTS
REM   DEPENDENTS
REM   EMPLOYEES
REM   LOCATIONS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษายเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

REM    PROJECTS
REM Created from Entity DEPARTMENT by SPITD001 on 07-JAN-95
PROMPT
PROMPT Creating Table DEPARTMENTS
CREATE TABLE departments
(
    depa_number    NUMBER    NOT NULL,
    depa_name      CHAR(20)  NOT NULL
);
COMMENT ON TABLE departments
    IS 'Created from Entity DEPARTMENT by SPITD001 on 07-JAN-95';
REM Created from Entity DEPENDENT by SPITD001 on 07-JAN-95
PROMPT
PROMPT Creating Table DEPENDENTS
CREATE TABLE dependents
(
    depe_name      CHAR(20)  NOT NULL,
    depe_birth_date DATE    NOT NULL,
    depe_sex       CHAR(1)   NOT NULL,
    depe_relationship CHAR(5) NULL
);
COMMENT ON TABLE dependents
    IS 'Created from Entity DEPENDENT by SPITD001 on 07-JAN-95';
REM Created from Entity EMPLOYEE by SPITD001 on 07-JAN-95
PROMPT
PROMPT Creating Table EMPLOYEES
CREATE TABLE employees
(
    emp_ssn        NUMBER    NOT NULL,
    emp_department_number_work_for NUMBER NOT NULL,
    emp_department_number NUMBER  NULL,
    emp_address    CHAR(20)  NULL,
    emp_bdate     DATE      NULL,
    emp_fname     CHAR(10)  NULL,
    emp_minit     CHAR(10)  NULL,
    emp_sex       CHAR(1)   NULL,
    emp_salary    NUMBER    NULL,
    emp_lname     CHAR(10)  NULL
);
COMMENT ON TABLE employees
    IS 'Created from Entity EMPLOYEE by SPITD001 on 07-JAN-95';
REM Created from Entity DEP_LOCATION by SPITD001 on 07-JAN-95
PROMPT
PROMPT Creating Table LOCATIONS
CREATE TABLE locations
(
    loca_location CHAR(10)  NOT NULL,
    loca_department_number NUMBER NOT NULL
);
COMMENT ON TABLE locations
    IS 'Created from Entity DEP_LOCATION by SPITD001 on 07-JAN-95';
REM Created from Entity PROJECT by SPITD001 on 07-JAN-95
PROMPT
PROMPT Creating Table PROJECTS
CREATE TABLE projects
(
    proj_number    NUMBER    NOT NULL,
    proj_name      CHAR(10)  NOT NULL,
    proj_department_number NUMBER NOT NULL,
    proj_locate    CHAR(10)  NULL
);
COMMENT ON TABLE projects
    IS 'Created from Entity PROJECT by SPITD001 on 07-JAN-95';
REM End of command file
EXIT

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ที่ 6.11 แสดงผลลัพธ์ในรูปแบบของ Data Definition Language ที่ให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่สามารถนำเอาทั้งสั้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.3 การเปรียบเทียบระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูล ของตัวจัดการฐานข้อมูล ORACLE กับระบบ ER/5

จากที่ได้กล่าวมาแล้วในหัวข้อที่ 6.1 ถึงคุณสมบัติทั่วไปของระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลว่าควรมีความสามารถอย่างไรบ้างซึ่งทั้งระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลของตัวจัดการฐานข้อมูล ORACLE และของระบบ ER/5 ก็มีความสามารถดังที่ได้กล่าวมาแล้วทั้งสิ้นแต่ยังมีความแตกต่างกันในเรื่องของประสิทธิภาพข้อดีและข้อเสียในรายละเอียดดังจะได้กล่าวถึงดังต่อไปนี้

#### - ความสามารถทางกราฟฟิก

##### ORACLE

###### ข้อดี

1. เมื่อวาดแบบจำลองอีอาร์เสร็จเป็นที่เรียบร้อยแล้วสามารถที่จะเคลื่อนย้ายรูปภาพไปยังที่ต่างๆได้โดยง่ายไม่ต้องมีการลบทิ้งแล้ววาดใหม่

###### ข้อเสีย

1. การกำหนดแอททริบิวต์และตัวควบคุมจะต้องออกจากระบบกราฟฟิกแล้วมาใส่เพิ่มเติมทีหลังทำให้มีความยุ่งยากและอาจเกิดการผิดพลาดได้ในกรณีที่กำหนดไม่ครบตามจำนวนเป็นต้น

2. ไม่สามารถแสดงแอททริบิวต์และตัวควบคุม (Domain Constraint) ให้เห็นบนแบบจำลองอีอาร์ทำให้เกิดการสับสนในการออกแบบได้ง่าย

##### ER/5

###### ข้อดี

1. การกำหนดแอททริบิวต์และตัวควบคุมไม่จำเป็นต้องออกจากระบบกราฟฟิกสามารถกำหนดลงบนแผนภาพได้ทันทีทำให้ง่ายและไม่เกิดการผิดพลาด

2. สามารถแสดงแอททริบิวต์และตัวควบคุม (Domain Constraint) ให้เห็นบนแบบจำลองอีอาร์ทำให้ข้อมูลของแบบจำลองอีอาร์ที่อยู่บนฐานข้อมูลสัมพันธ์กับรูปภาพโดยตลอดไม่เกิดการผิดพลาด

###### ข้อเสีย

1. เมื่อวาดแบบจำลองอีอาร์เสร็จเป็นที่เรียบร้อยแล้วไม่สามารถที่จะเคลื่อนย้ายรูปภาพไปยังที่ต่างๆได้แต่สามารถเพียงเหตุการณ์ดังกล่าวได้ด้วยการลบวัตถุบางอย่างทิ้งแล้ววาดลงที่ตำแหน่งใหม่แทน

#### - ความสามารถในการสร้างตารางฐานข้อมูลจากแบบจำลอง

##### ORACLE

###### ข้อดี

เอกสารนี้เป็นเอกสารที่เสิร์ฟไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. มีระบบสนับสนุนในการพัฒนาโปรแกรมทำให้สามารถนำผลลัพธ์ที่ได้ไปสร้างเป็นโปรแกรมได้โดยง่าย

### ข้อเสีย

1. ตารางฐานข้อมูลที่ได้จากแบบจำลองเป็นตารางฐานข้อมูลที่มีความซ้ำซ้อนซึ่งจัดเป็นฐานข้อมูลในบรรทัดฐานขั้นแรกและไม่สามารถที่จะปรับปรุงให้เป็นฐานข้อมูลในบรรทัดฐานขั้นสูงจนถึงขั้นที่ห้าซึ่งเป็นฐานข้อมูลที่ปราศจากความซ้ำซ้อนได้

### ER/5

### ข้อดี

1. ตารางฐานข้อมูลที่ได้จากแบบจำลองเป็นตารางฐานข้อมูลที่มีความซ้ำซ้อนซึ่งจัดเป็นฐานข้อมูลในบรรทัดฐานขั้นแรกแต่สามารถที่จะปรับปรุงให้เป็นตารางฐานข้อมูลในบรรทัดฐานขั้นสูงจนถึงขั้นที่ห้าได้โดยการกำหนดความสัมพันธ์ระหว่างแอททริบิวต์ (Functional Dependency) และความสัมพันธ์เจดี (Join Dependency) เพิ่มเติมแล้วนำไปประมวลผลทำให้ผลลัพธ์สุดท้ายเป็นตารางฐานข้อมูลในบรรทัดฐานขั้นที่ห้าซึ่งเป็นตารางฐานข้อมูลที่ปราศจากความซ้ำซ้อนโดยสิ้นเชิง

### ข้อเสีย

1. ไม่มีระบบสนับสนุนในการพัฒนาโปรแกรมทำให้ผลลัพธ์ที่ได้ไปเพียงตารางฐานข้อมูลเบื้องต้นเท่านั้นเมื่อต้องการสร้างโปรแกรมก็ต้องเขียนโปรแกรมจัดการเองซึ่งค่อนข้างยุ่งยากและไม่มีมาตรฐานที่แน่นอน

- ความสามารถในการควบคุมความถูกต้องของข้อมูล

### ORACLE

### ข้อดี

1. แอททริบิวต์ต่างๆที่ต้องการข้อมูลในขอบเขตเดียวกันสามารถอ้างถึงชื่อของตัวควบคุมชื่อเดียวกันได้ทำให้การกำหนดตัวควบคุมไม่ซ้ำซ้อน

### ข้อเสีย

1. การนำตัวควบคุมไปใช้งานจะสามารถควบคุมความถูกต้องได้ในระดับแอปพลิเคชันเท่านั้นยกตัวอย่างเช่นแอททริบิวต์ SEX สามารถมีข้อมูลได้คือ 'M' กับ 'F' เมื่อผู้ใช้งานแอปพลิเคชันกำหนดค่าข้อมูลให้กับแอททริบิวต์ SEX แอปพลิเคชันจะนำข้อมูลที่กำหนดขึ้นนี้ไปค้นหาภายใต้ตัวควบคุมชื่อ SEX\_DOMAIN ว่าข้อมูลที่กำหนดนั้นเป็น 'M' กับ 'F' หรือไม่ถ้าใช่ก็จะยอมให้เก็บข้อมูลลงบนฐานข้อมูลได้ แต่ถ้าไม่ใช่ก็จะไม่ยอมให้เก็บข้อมูลลงบนฐานข้อมูล\* แต่เมื่อจบแอปพลิเคชันแล้วเกิดมีผู้ใช้งานอื่นเข้าไปแก้ไขข้อมูลของแอททริบิวต์ SEX ผ่านทางโปรแกรม SQLPLUS ตัวควบคุมดังกล่าวก็ไม่สามารถควบคุมได้เนื่องจากไม่มีแอปพลิเคชันมาตรวจสอบความถูกต้องของข้อมูลเพื่อการศึกษานี้ ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ไม่สามารถกำหนดตัวควบคุมชนิดนอกเซตได้ ยกตัวอย่างเช่น ข้อมูลของแอททริบิวต์ Proj\_Location ต้องเป็นชื่อของประเทศที่ไม่ใช่ประเทศสหรัฐอเมริกาและประเทศแคนาดา ('USA','CANADA') แต่ถ้าต้องการดัดแปลงการควบคุมแอททริบิวต์ Proj\_Location ให้สามารถควบคุมความถูกต้องดังกล่าวได้ ก็สามารถทำได้โดยการกำหนดชื่อของประเทศทุกประเทศในโลกลงไปเป็นเซตยกเว้นประเทศสหรัฐอเมริกาและประเทศแคนาดาซึ่งมีมากมายและไม่สะดวกและในการเปรียบเทียบค้นหาที่จะเสียเวลามาก

#### ER/5

##### **ข้อดี**

1. การนำตัวควบคุมไปใช้งานจะสามารถควบคุมความถูกต้องได้ในระดับ SYSTEM เนื่องจากมีการสร้างวิวของตารางที่ต้องการควบคุมดังนั้น ผู้ใช้งานอื่นไม่สามารถเข้าไปแก้ไขข้อมูลของแอททริบิวต์ SEX ผ่านทางโปรแกรม SQLPLUS ได้เว้นเสียแต่จะได้แก้ไขผ่านตารางจริงโดยตรง

2. สามารถกำหนดตัวควบคุมชนิดนอกเซตได้ ยกตัวอย่างเช่น ข้อมูลของแอททริบิวต์ Proj\_Location ต้องเป็นชื่อของประเทศที่ไม่ใช่ประเทศสหรัฐอเมริกาและประเทศแคนาดา ก็สามารถทำได้โดยการกำหนดตัวควบคุมดังนี้คือ !('USA','CANADA')

##### **ข้อเสีย**

1. แอททริบิวต์ต่างๆที่ต้องการข้อมูลในขอบเขตเดียวกันไม่สามารถอ้างถึงชื่อของตัวควบคุมชื่อเดียวกันได้ทำให้การกำหนดตัวควบคุมเกิดซ้ำซ้อน

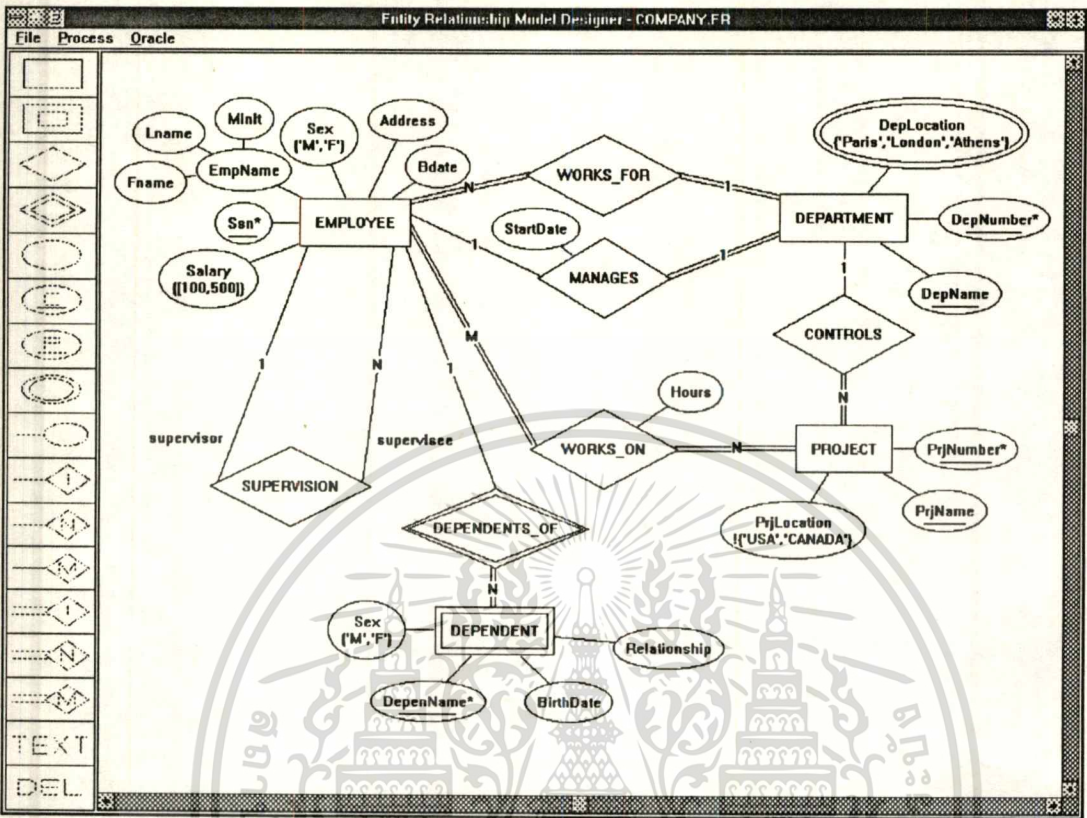
#### **6.4 ผลการทดลองใช้ระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูล ER/5**

สำหรับการทดลองใช้ระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูล ER/5 สามารถแบ่งผลการทดลองออกเป็น 3 ระดับดังต่อไปนี้คือ

##### **6.4.1 ผลการทดลองระดับฐานข้อมูลในบรรทัดฐานขั้นแรก**

การทดลองระดับฐานข้อมูลในบรรทัดฐานขั้นแรกได้ทำการทดลองด้วยการออกแบบฐานข้อมูลด้วยแบบจำลองอีอาร์จำนวน 3 แบบจำลองได้แก่ แบบจำลองอีอาร์ที่ใช้สำหรับเก็บข้อมูลของบริษัท ดังแสดงในรูปที่ 6.12 แบบจำลองอีอาร์ที่ใช้สำหรับเก็บข้อมูลของสนามบิน ดังแสดงในรูปที่ 6.14 และแบบจำลองอีอาร์ที่ใช้สำหรับเก็บข้อมูลของมหาวิทยาลัย ดังแสดงในรูปที่ 6.16 ซึ่งผลของการทดลองปรากฏว่าสามารถสร้างผลลัพธ์เป็นภาษาสอบถามเชิงโครงสร้าง (Structure Query Language) ในส่วนของการกำหนดโครงสร้างตาราง (Data Definition Language) ซึ่งสามารถสร้างตารางฐานข้อมูลที่จัดอยู่ในบรรทัดฐานขั้นแรกได้อย่างถูกต้องและสามารถสร้างวิวสำหรับควบคุมความถูกต้องของข้อมูลได้ดียิ่งอีกด้วย โดยผลลัพธ์ของแบบจำลองอีอาร์รูปที่ 6.12 ได้แก่คำสั่งภาษาสอบถามเชิงโครงสร้างในรูปที่ 6.13 ผลลัพธ์ของแบบจำ

ลองอีอาร์รูปที่ 6.14 ได้แก่คำสั่งภาษาสอบถามเชิงโครงสร้างในรูปที่ 6.15 และผลลัพธ์ของแบบจำลองอีอาร์รูปที่ 6.16 ได้แก่คำสั่งภาษาสอบถามเชิงโครงสร้างในรูปที่ 6.17 ตามลำดับ



รูปที่ 6.12 แสดงแบบจำลองอีอาร์ที่ใช้เก็บข้อมูลของบริษัทแห่งหนึ่ง

```

CREATE TABLE DEPARTMENT
(
    Ssn          NUMBER          NOT NULL,
    DepNumber    NUMBER          NOT NULL,
    DepName      CHAR(30)       NOT NULL,
    StartDate    DATE
);
CREATE TABLE DEPENDENT
(
    Sex          CHAR(1),
    DepenName    CHAR(30)       NOT NULL,
    BirthDate    DATE,
    Relationship  CHAR(30),
    Ssn          NUMBER          NOT NULL
);
CREATE TABLE DepLocation
(
    DepNumber    NUMBER          NOT NULL,
    DepLocation  CHAR(30)       NOT NULL
);
CREATE TABLE EMPLOYEE
(
    Fname        CHAR(1),
    Lname        CHAR(10),
    Minit        CHAR(1),
    Ssn          NUMBER          NOT NULL,
    Salary       NUMBER,
    Sex          CHAR(1),
    Address      CHAR(30),
    Bdate        DATE,
    DepNumber    NUMBER          NOT NULL,
    Ssn23       NUMBER
    );
    
```

);สารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**CREATE TABLE PROJECT**

```
(
    DepNumber    NUMBER,
    PrjNumber    NUMBER    NOT NULL,
    PrjName      CHAR(30)  NOT NULL,
    PrjLocation  CHAR(30)
);
```

**CREATE TABLE WORKS\_ON**

```
(
    Ssn          NUMBER    NOT NULL,
    PrjNumber    NUMBER    NOT NULL,
    Hours        DATE
);
```

```
CREATE UNIQUE INDEX INDEX_#1 ON EMPLOYEE (Ssn) ;
CREATE UNIQUE INDEX INDEX_#2 ON DEPARTMENT (DepNumber) ;
CREATE UNIQUE INDEX INDEX_#3 ON PROJECT (PrjNumber) ;
CREATE UNIQUE INDEX INDEX_#4 ON DEPENDENT (DepenName,Ssn) ;
CREATE UNIQUE INDEX INDEX_#5 ON DepLocation (DepNumber,DepLocation) ;
CREATE UNIQUE INDEX INDEX_#6 ON WORKS_ON (Ssn,PrjNumber) ;
```

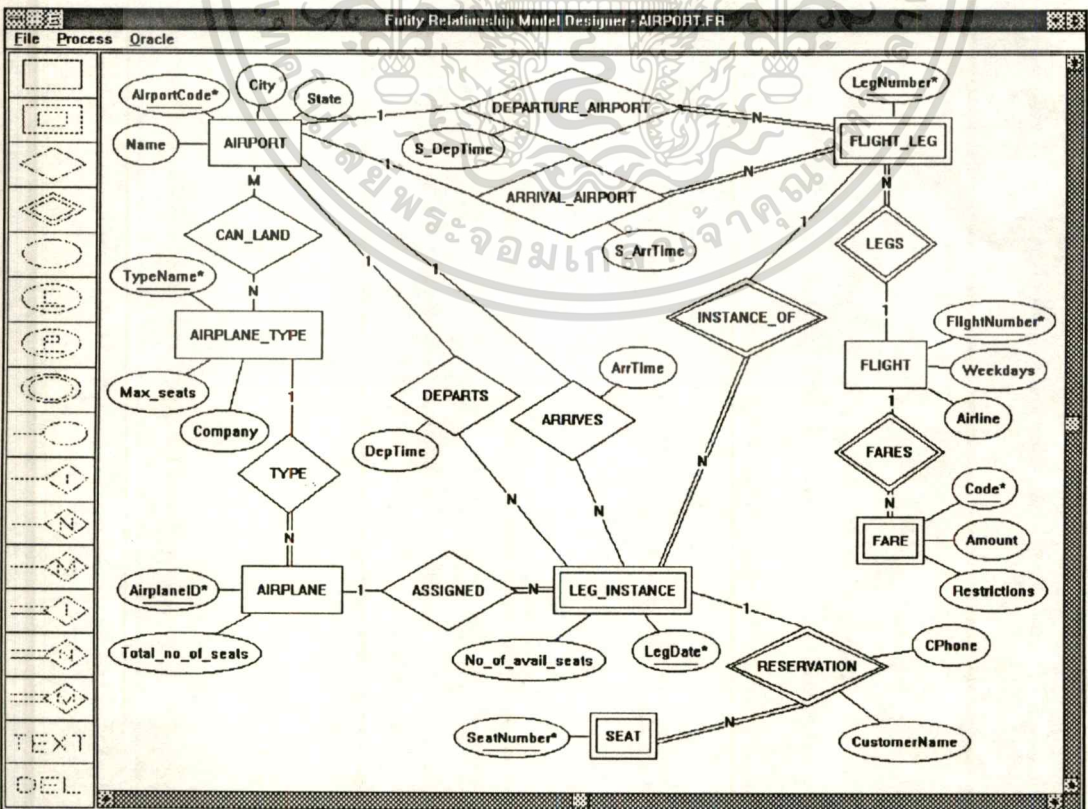
```
CREATE VIEW DEPENDENT_VIEW AS SELECT *
FROM DEPENDENT
WHERE Sex IN ('F','M')
WITH CHECK OPTION ;
```

```
CREATE VIEW DepLocation_VIEW AS SELECT *
FROM DepLocation
WHERE DepLocation IN ('Athens','London','Paris')
WITH CHECK OPTION ;
```

```
CREATE VIEW EMPLOYEE_VIEW AS SELECT *
FROM EMPLOYEE
WHERE Sex IN ('F','M') AND ((Salary >= 100 AND Salary <= 500))
WITH CHECK OPTION ;
```

```
CREATE VIEW PROJECT_VIEW AS SELECT *
FROM PROJECT
WHERE PrjLocation NOT IN ('CANADA','USA')
WITH CHECK OPTION ;
```

รูปที่ 6.13 แสดงผลลัพธ์ที่ได้จากแบบจำลองอ็อบเจกต์ในรูปที่ 6.12



เอกสารนี้เป็นเอกสารรูปที่ 6.14 แสดงแบบจำลองอ็อบเจกต์ที่ใช้เก็บข้อมูลของสนามบินแห่งหนึ่ง นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

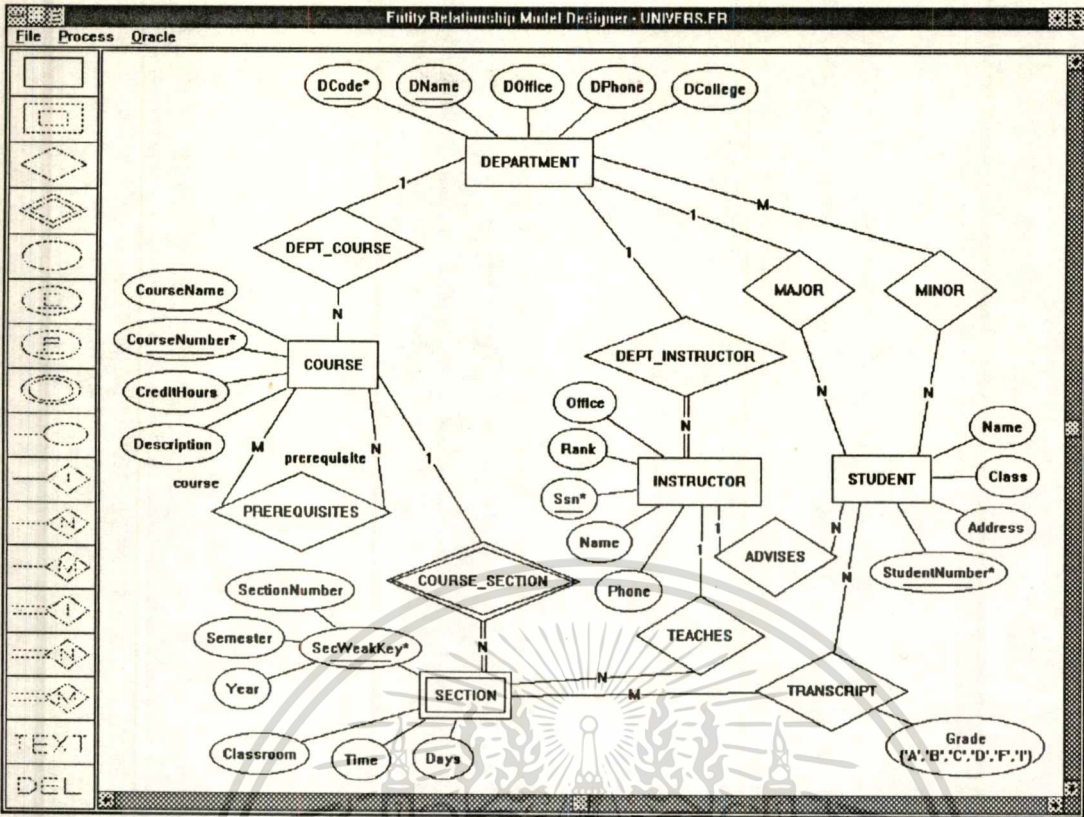
CREATE TABLE AIRPLANE
(
    TypeName          CHAR(30),
    AirplaneID        NUMBER          NOT NULL,
    Total_no_of_seats NUMBER
);
CREATE TABLE AIRPLANE_TYPE
(
    TypeName          CHAR(30)          NOT NULL,
    Max_seats         NUMBER,
    Company           CHAR(30)
);
CREATE TABLE AIRPORT
(
    AirportCode       NUMBER          NOT NULL,
    Name              CHAR(30),
    City              CHAR(40),
    State             CHAR(20)
);
CREATE TABLE CAN_LAND
(
    AirportCode       NUMBER          NOT NULL,
    TypeName          CHAR(30)          NOT NULL
);
CREATE TABLE FARE
(
    Code              NUMBER          NOT NULL,
    Amount            NUMBER,
    Restrictions      NUMBER,
    FlightNumber      NUMBER          NOT NULL
);
CREATE TABLE FLIGHT
(
    FlightNumber      NUMBER          NOT NULL,
    Weekdays         DATE,
    Airline           CHAR(30)
);
CREATE TABLE FLIGHT_LEG
(
    AirportCode       NUMBER          NOT NULL,
    LegNumber         NUMBER          NOT NULL,
    S_DepTime         DATE,
    S_ArrTime         DATE,
    FlightNumber      NUMBER          NOT NULL,
    AirportCode33     NUMBER          NOT NULL
);
CREATE TABLE LEG_INSTANCE
(
    AirportCode       NUMBER,
    AirplaneID        NUMBER          NOT NULL,
    LegDate           DATE          NOT NULL,
    No_of_avail_seats NUMBER,
    DepTime           DATE,
    ArrTime           DATE,
    LegNumber         NUMBER          NOT NULL,
    FlightNumber      NUMBER          NOT NULL,
    AirportCode34     NUMBER
);
CREATE TABLE SEAT
(
    SeatNumber        NUMBER          NOT NULL,
    CustomerName      CHAR(30),
    CPhone            NUMBER,
    LegDate           DATE          NOT NULL,
    LegNumber         NUMBER          NOT NULL,
    FlightNumber      NUMBER          NOT NULL
);
CREATE UNIQUE INDEX INDEX_#1 ON FLIGHT_LEG (LegNumber,FlightNumber);
CREATE UNIQUE INDEX INDEX_#2 ON FARE (Code,FlightNumber);
CREATE UNIQUE INDEX INDEX_#3 ON AIRPLANE_TYPE (TypeName);
CREATE UNIQUE INDEX INDEX_#4 ON AIRPORT (AirportCode);
CREATE UNIQUE INDEX INDEX_#5 ON FLIGHT (FlightNumber);
CREATE UNIQUE INDEX INDEX_#6 ON AIRPLANE (AirplaneID);
CREATE UNIQUE INDEX INDEX_#7 ON LEG_INSTANCE (LegDate,LegNumber,FlightNumber);
CREATE UNIQUE INDEX INDEX_#8 ON SEAT (SeatNumber,LegDate,LegNumber,FlightNumber);
CREATE UNIQUE INDEX INDEX_#9 ON CAN_LAND (AirportCode,TypeName);

```

รูปที่ 6.15 แสดงผลลัพธ์ที่ได้จากแบบจำลองอ็อบเจกต์ในรูปที่ 6.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.16 แสดงแบบจำลองอ็อร์ที่ใช้เก็บข้อมูลของมหาวิทยาลัยแห่งหนึ่ง

```

CREATE TABLE COURSE
(
    DCode          NUMBER,
    CourseName     CHAR(30),
    CourseNumber   NUMBER          NOT NULL,
    CreditHours    NUMBER,
    Description     CHAR(40)
);
CREATE TABLE DEPARTMENT
(
    DCode          NUMBER          NOT NULL,
    DName          CHAR(30)        NOT NULL,
    DOffice        CHAR(20),
    DPhone         NUMBER,
    DCollege       CHAR(30)
);
CREATE TABLE INSTRUCTOR
(
    DCode          NUMBER          NOT NULL,
    Name           CHAR(30),
    Ssn            NUMBER          NOT NULL,
    Rank           CHAR(30),
    Office         CHAR(30),
    Phone         NUMBER
);
CREATE TABLE MINOR
(
    DCode          NUMBER          NOT NULL,
    StudentNumber  NUMBER          NOT NULL
);
CREATE TABLE PREREQUISITES
(
    CourseNumber   NUMBER          NOT NULL,
    CourseNumber28 NUMBER          NOT NULL
);
CREATE TABLE SECTION
(
    Classroom      NUMBER,

```

เอกสารนี้สงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Time          NUMBER,
Days          DATE,
Year          DATE          NOT NULL,
Semester      NUMBER       NOT NULL,
SectionNumber NUMBER       NOT NULL,
Ssn           NUMBER,
CourseNumber  NUMBER       NOT NULL
);
CREATE TABLE STUDENT
(
  DCode        NUMBER,
  Ssn          NUMBER,
  StudentNumber NUMBER     NOT NULL,
  Address      CHAR(30),
  Class       NUMBER,
  Name        CHAR(30)
);
CREATE TABLE TRANSCRIPT
(
  Year          DATE          NOT NULL,
  Semester      NUMBER       NOT NULL,
  SectionNumber NUMBER       NOT NULL,
  Grade         CHAR(1),
  StudentNumber NUMBER     NOT NULL,
  CourseNumber  NUMBER       NOT NULL
);
CREATE UNIQUE INDEX INDEX_#1 ON COURSE (CourseNumber) ;
CREATE UNIQUE INDEX INDEX_#2 ON SECTION (Year,Semester,SectionNumber,CourseNumber) ;
CREATE UNIQUE INDEX INDEX_#3 ON INSTRUCTOR (Ssn) ;
CREATE UNIQUE INDEX INDEX_#4 ON DEPARTMENT (DCode) ;
CREATE UNIQUE INDEX INDEX_#5 ON STUDENT (StudentNumber) ;
CREATE UNIQUE INDEX INDEX_#6 ON PREREQUISITES (CourseNumber,CourseNumber28) ;
CREATE UNIQUE INDEX INDEX_#7 ON MINOR (DCode,StudentNumber) ;
CREATE UNIQUE INDEX INDEX_#8 ON TRANSCRIPT
(Year,Semester,SectionNumber,StudentNumber,CourseNumber) ;
CREATE VIEW TRANSCRIPT_VIEW AS SELECT *
FROM TRANSCRIPT
WHERE
Grade IN ('A','B','C','D','F','I')
WITH CHECK OPTION ;

```

รูปที่ 6.17 แสดงผลลัพธ์ที่ได้จากแบบจำลองอ็อบเจกต์ในรูปที่ 6.16

#### 6.4.2 ผลการทดลองระดับฐานข้อมูลในบรรทัดฐานชั้น Boyce/Codd

จากตารางฐานข้อมูลในรูปที่ 2.8 จะเห็นว่ามีความสัมพันธ์ระหว่างแอททริบิวต์อยู่จำนวน 4 ความสัมพันธ์ได้แก่

- 1.TEACHER,SEMERTER,SECTION -> ATTENDANCES
- 2.SUBJECT,SEMESTER,SECTION -> ATTENDANCES
- 3.TEACHER,SEMESTER -> SUBJECT
- 4.SUBJECT,SEMESTER -> TEACHER

จากความสัมพันธ์ระหว่างแอททริบิวต์ดังกล่าวสามารถบอกได้ว่าตารางฐานข้อมูลในรูปที่ 2.8 เป็นฐานข้อมูลในบรรทัดฐานชั้นที่สามเนื่องจากมีแอททริบิวต์ที่ไม่เป็นกุญแจหลักของตารางอยู่เพียงแอททริบิวต์เดียวคือ ATTENDANCES แต่ไม่เป็นฐานข้อมูลในบรรทัดฐานชั้น Boyce/Codd เนื่องจาก Determinant ของความสัมพันธ์ระหว่างแอททริบิวต์ที่ 3 และ 4 ไม่เป็น Candidate Key ดังนั้นถ้าต้องการให้ตารางฐานข้อมูลในบรรทัดฐานชั้น Boyce/Codd จึงเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้เชิงพาณิชย์ใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องแยกตารางฐานข้อมูลในรูปที่ 2.8 โดยใช้อาศัยหลักการของ nonloss decomposition ออกเป็นสองตารางดังแสดงในรูปที่ 2.9

จากที่ได้กล่าวมาแล้วเป็นตัวอย่างของการแปลงตารางฐานข้อมูลในบรรทัดฐานขั้นแรกให้เป็นฐานข้อมูลในบรรทัดฐานขั้น Boyce/Codd ดังนั้นเพื่อให้ง่ายในการทดลองใช้ระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูล ER/5 ในขั้น Boyce/Codd นี้จะขอสมมติให้แอททริบิวต์ของตาราง DEPENDENT บนกรอบข้อความ FD ซึ่งมีจำนวนแอททริบิวต์ 5 แอททริบิวต์ เท่ากับจำนวนแอททริบิวต์ของตารางในรูปที่ 2.8 โดยกำหนดให้

แอททริบิวต์ TEACHER	แทนด้วยแอททริบิวต์	BirthDate
แอททริบิวต์ SEMESTER	แทนด้วยแอททริบิวต์	DepenName
แอททริบิวต์ SUBJECT	แทนด้วยแอททริบิวต์	Relationship
แอททริบิวต์ SECTION	แทนด้วยแอททริบิวต์	Sex
แอททริบิวต์ ATTENDANCES	แทนด้วยแอททริบิวต์	Ssn

หลังจากนั้นก็ทำการลบความสัมพันธ์ระหว่างแอททริบิวต์ที่ถูกสร้างขึ้นในขั้นตอนการสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นแรกของตาราง DEPENDENT ออกให้หมดแล้วใส่ความสัมพันธ์ระหว่างแอททริบิวต์ใหม่ดังนี้

1. BirthDate, DepenName, Sex --> Ssn
2. Relationship, DepenName, Sex --> Ssn
3. BirthDate, DepenName --> Relationship
4. Relationship, DepenName --> BirthDate

ความสัมพันธ์ระหว่างแอททริบิวต์เปรียบเสมือนกับความสัมพันธ์ระหว่างแอททริบิวต์ของตารางที่ 2.8 ทุกประการ และลบความสัมพันธ์ระหว่างแอททริบิวต์ที่ถูกสร้างขึ้นในขั้นตอนการสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นแรกของตาราง WORKS\_ON ออกให้หมดแล้วใส่ความสัมพันธ์ระหว่างแอททริบิวต์ Hours, PrijNumber, Ssn --> Hours, PrijNumber, Ssn แทนสัมพันธ์ระหว่างแอททริบิวต์ของตาราง WORKS\_ON ที่กำหนดขึ้นใหม่นี้เพื่อที่จะทำให้สามารถนำไปทดสอบในระดับฐานข้อมูลในบรรทัดฐานขั้นที่ห้าซึ่งจะได้กล่าวถึงในหัวข้อต่อไป จากนั้นก็กดปุ่ม RUN จะได้ผลลัพธ์ของตารางดังแสดงในรูปที่ 6.20 โดยจะสังเกตเห็นว่าตาราง DEPENDENT เดิมถูกแยกออกเป็น 2 ตารางคือ

DEPENDENT#1 ประกอบด้วยแอททริบิวต์ DepenName, BirthDate และ Relationship

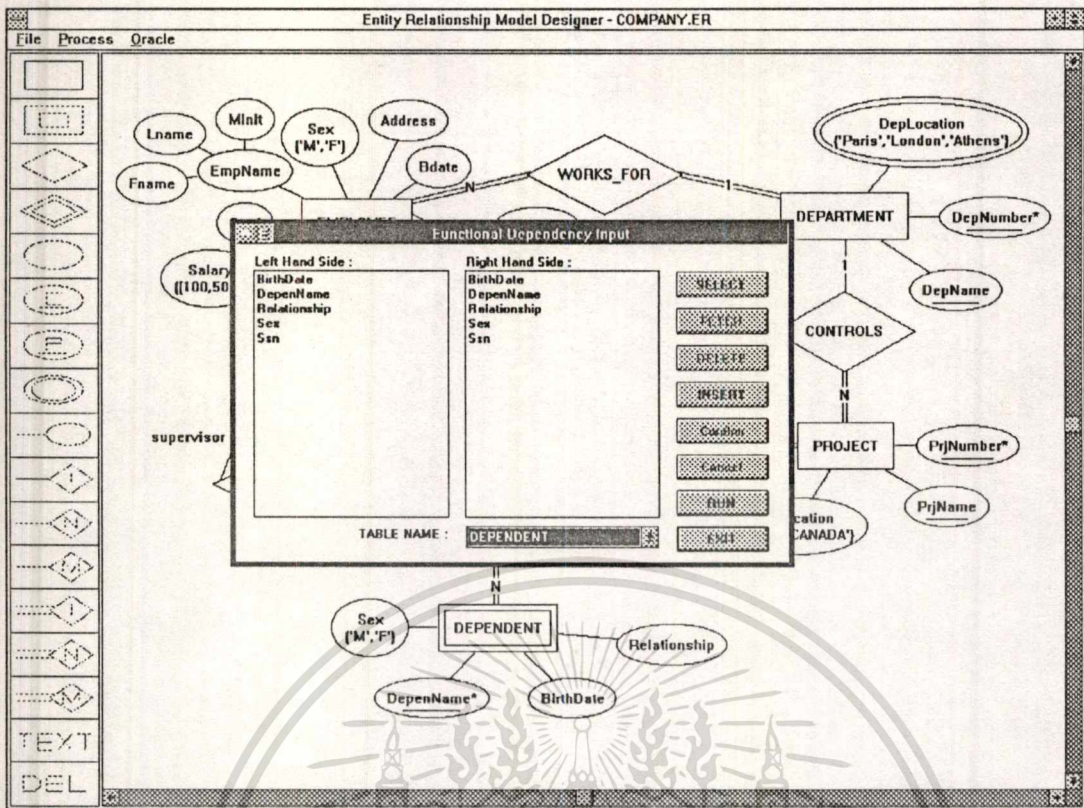
DEPENDENT#2 ประกอบด้วยแอททริบิวต์ Sex, DepenName, Relationship และ Ssn

ซึ่งผลลัพธ์ของตารางที่ได้ก็เปรียบเสมือนตารางในรูปที่ 2.9 ซึ่งจัดเป็นตารางฐานข้อมูลในขั้น

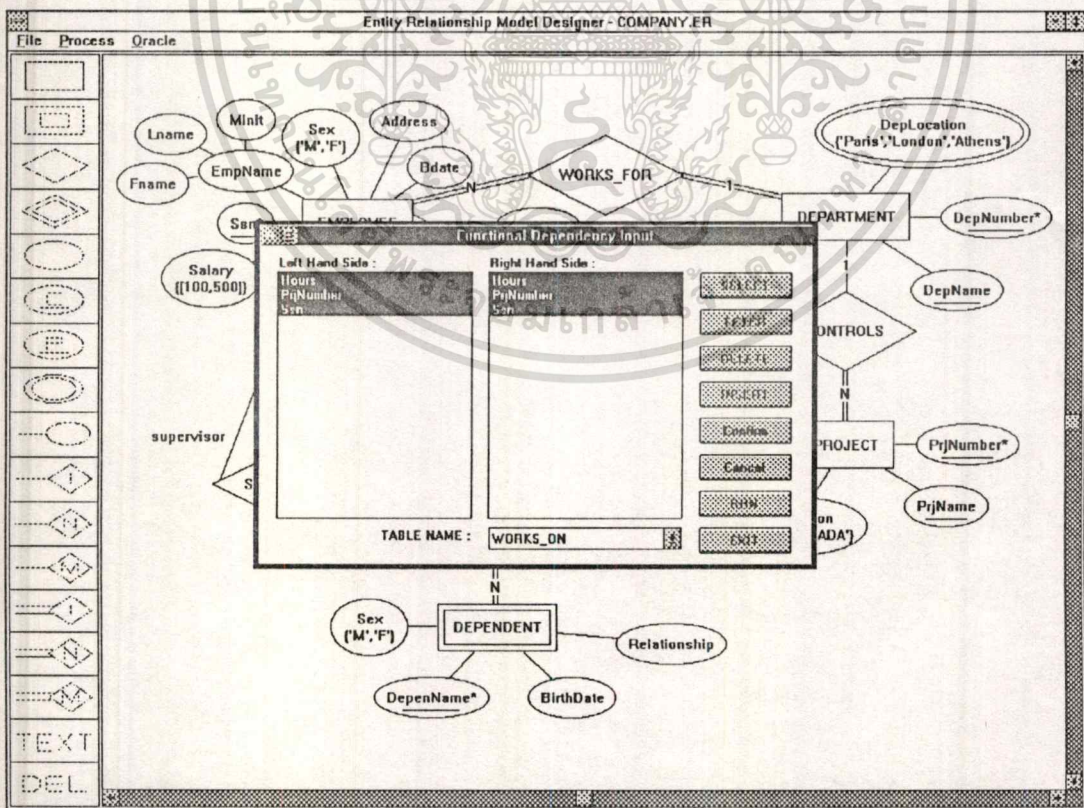
Boyce/Codd อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.18 แสดงการกำหนด FD ให้กับตาราง DEPENDENT



รูปที่ 6.19 แสดงการกำหนด FD ให้กับตาราง WORKS\_ON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CREATE TABLE DEPARTMENT
(
    Ssn          NUMBER,
    DepNumber    NUMBER          NOT NULL,
    DepName      CHAR(30),
    StartDate    DATE
);
CREATE TABLE DEPENDENT#1
(
    DepenName    CHAR(30)        NOT NULL,
    BirthDate    DATE            NOT NULL,
    Relationship  CHAR(30)
);
CREATE TABLE DEPENDENT#2
(
    Sex          CHAR(1)         NOT NULL,
    DepenName    CHAR(30)        NOT NULL,
    Relationship  CHAR(30)        NOT NULL,
    Ssn          NUMBER
);
CREATE TABLE DepLocation
(
    DepNumber    NUMBER          NOT NULL,
    DepLocation  CHAR(30)        NOT NULL
);
CREATE TABLE EMPLOYEE
(
    Fname        CHAR(1),
    Lname        CHAR(10),
    Minit        CHAR(1),
    Ssn          NUMBER          NOT NULL,
    Salary       NUMBER,
    Sex          CHAR(1),
    Address      CHAR(30),
    Bdate        DATE,
    DepNumber    NUMBER,
    Ssn23        NUMBER
);
CREATE TABLE PROJECT
(
    DepNumber    NUMBER,
    PrjNumber    NUMBER          NOT NULL,
    PrjName      CHAR(30),
    PrjLocation  CHAR(30)
);
CREATE TABLE WORKS_ON
(
    Ssn          NUMBER          NOT NULL,
    PrjNumber    NUMBER          NOT NULL,
    Hours        DATE            NOT NULL
);
CREATE UNIQUE INDEX INDEX_#1 ON DEPARTMENT (DepNumber);
CREATE UNIQUE INDEX INDEX_#2 ON DEPENDENT#1 (DepenName,BirthDate);
CREATE UNIQUE INDEX INDEX_#3 ON DEPENDENT#2 (Sex,DepenName,Relationship);
CREATE UNIQUE INDEX INDEX_#4 ON DepLocation (DepNumber,DepLocation);
CREATE UNIQUE INDEX INDEX_#5 ON EMPLOYEE (Ssn);
CREATE UNIQUE INDEX INDEX_#6 ON PROJECT (PrjNumber);
CREATE UNIQUE INDEX INDEX_#7 ON WORKS_ON (Ssn,PrjNumber,Hours);

CREATE VIEW DEPENDENT#2_VIEW AS SELECT *
FROM DEPENDENT#2
WHERE Sex IN ('F','M')
WITH CHECK OPTION;

CREATE VIEW DepLocation_VIEW AS SELECT *
FROM DepLocation
WHERE DepLocation IN ('Athens','London','Paris')
WITH CHECK OPTION;

CREATE VIEW EMPLOYEE_VIEW AS SELECT *
FROM EMPLOYEE
WHERE Sex IN ('F','M') AND ((Salary >= 100 AND Salary <= 500))
WITH CHECK OPTION;

CREATE VIEW PROJECT_VIEW AS SELECT *
FROM PROJECT
WHERE PrjLocation NOT IN ('CANADA','USA')
WITH CHECK OPTION;

```

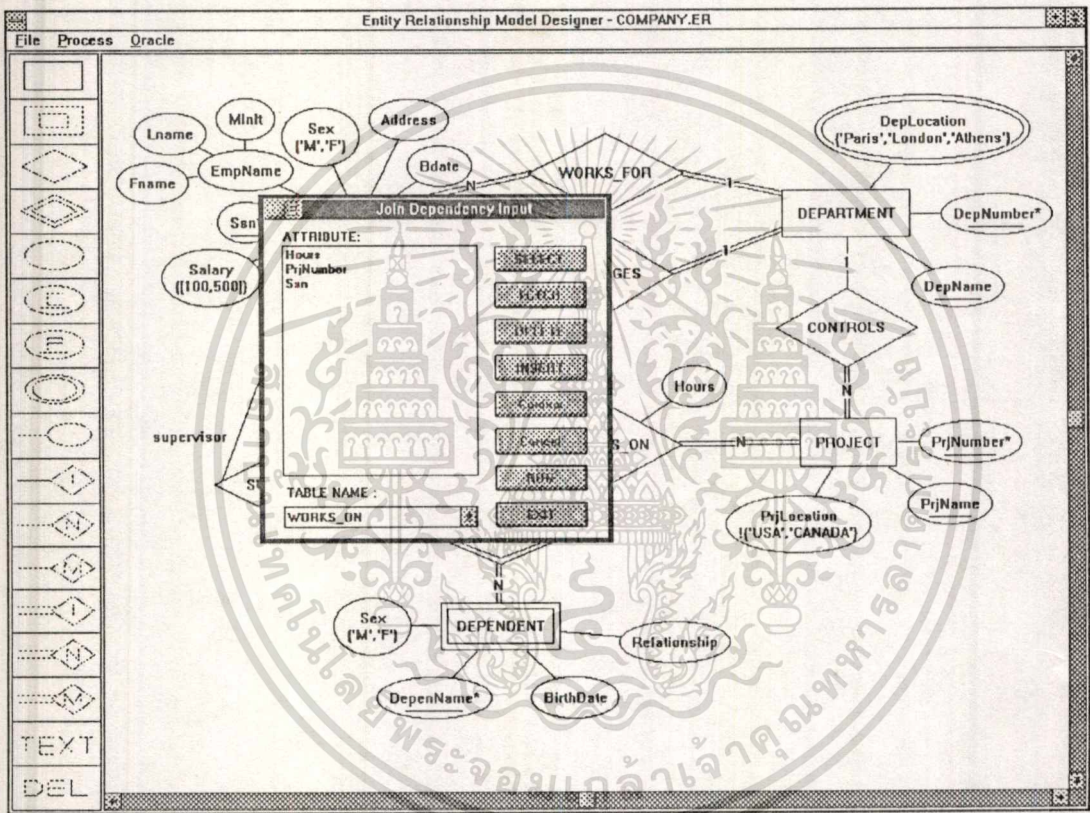
รูปที่ 6.20 แสดงผลลัพธ์ของตารางที่ได้ในบรรทัดฐานชั้น Boyce/Codd

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.4.3 ผลการทดลองระดับฐานข้อมูลในบรรทัดฐานขั้นที่ห้า

จากตารางฐานข้อมูลในรูปที่ 2.12 ประกอบด้วยแอททริบิวต์ S,P และ J ที่มีความสัมพันธ์ระหว่างแอททริบิวต์คือ S,P,J → S,P,J แสดงให้เห็นว่าเป็นตารางฐานข้อมูลในบรรทัดฐานขั้น Boyce/Codd เนื่องจากแอททริบิวต์ทั้งหมดเป็นกุญแจหลักของตาราง แต่ยังไม่จัดว่าเป็นตารางฐานข้อมูลในบรรทัดฐานขั้นที่ห้าเนื่องจากว่ายังมีความสัมพันธ์เจต JD\*((S,P),(P,J),(J,S)) แอบแฝงอยู่ดังนั้นถ้าต้องการตารางฐานข้อมูลในบรรทัดฐานขั้นที่ห้าจะต้องแยกตารางฐานข้อมูลในรูปที่ 2.12 ออกเป็น 3 ตารางตามหลักของ Lossless Join Property ดังแสดงในรูปที่ 2.13



รูปที่ 6.21 แสดงการกำหนด JD ให้กับตาราง WORKS\_ON

จากที่ได้กล่าวมาแล้วเป็นตัวอย่างของการแปลงตารางฐานข้อมูลในบรรทัดฐานขั้น Boyce/Codd ให้เป็นตารางฐานข้อมูลในบรรทัดฐานขั้นที่ห้า ดังนั้นเพื่อให้ง่ายในการทดลองใช้ระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูล ER/5 ในบรรทัดฐานขั้นที่ห้านี้ จะขอสมมติให้แอททริบิวต์ของตาราง WORKS\_ON บนกรอบข้อความ JD (เป็นตารางเดียวที่เป็นไปได้ที่จะมีความสัมพันธ์เจตแอบแฝงอยู่เนื่องจากทุกแอททริบิวต์ของตารางเป็นกุญแจหลักของตาราง) ซึ่งมีจำนวนแอททริบิวต์ 3 แอททริบิวต์เท่ากับจำนวนแอททริบิวต์ของตารางในรูปที่ 2.12 โดยกำหนดให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอดทริบิวต์	S	แทนด้วยแอดทริบิวต์	Hours
แอดทริบิวต์	P	แทนด้วยแอดทริบิวต์	PrjNumber
แอดทริบิวต์	J	แทนด้วยแอดทริบิวต์	Ssn

หลังจากนั้นก็ใส่ความสัมพันธ์เจตีดังนี้ JD\*((Hours,PrjNumber),(PrjNumber,Ssn),(Ssn,Hours)) ความสัมพันธ์เจตีดีเปรียบเสมือนกับความสัมพันธ์เจตีดีของตารางที่ 2.12 ทุกประการ จากนั้นก็ กดปุ่ม RUN จะได้ผลลัพธ์ของตารางดังแสดงในรูปที่ 6.22 โดยจะสังเกตเห็นว่าตาราง WORKS\_ON เดิมถูกแยกออกเป็น 3 ตารางคือ

WORKS\_ON#1 ประกอบด้วยแอดทริบิวต์ Hours กับแอดทริบิวต์ PrjNumber

WORKS\_ON#2 ประกอบด้วยแอดทริบิวต์ PrjNumber กับแอดทริบิวต์ Ssn

WORKS\_ON#3 ประกอบด้วยแอดทริบิวต์ Ssn กับแอดทริบิวต์ Hours

ซึ่งผลลัพธ์ของตารางที่ได้ก็เปรียบเสมือนตารางในรูปที่ 2.13 ซึ่งจัดเป็นตารางฐานข้อมูลในชั้นที่ห้าอย่างถูกต้อง

```

CREATE TABLE DEPARTMENT
(
    Ssn          NUMBER,
    DepNumber    NUMBER          NOT NULL,
    DepName     CHAR(30),
    StartDate   DATE
);
CREATE TABLE DEPENDENT#1
(
    DepenName   CHAR(30)          NOT NULL,
    BirthDate   DATE              NOT NULL,
    Relationship CHAR(30)
);
CREATE TABLE DEPENDENT#2
(
    Sex          CHAR(1)          NOT NULL,
    DepenName   CHAR(30)          NOT NULL,
    Relationship CHAR(30)          NOT NULL,
    Ssn         NUMBER
);
CREATE TABLE DepLocation
(
    DepNumber    NUMBER          NOT NULL,
    DepLocation  CHAR(30)        NOT NULL
);
CREATE TABLE EMPLOYEE
(
    Fname       CHAR(1),
    Lname       CHAR(10),
    Minit       CHAR(1),
    Ssn         NUMBER          NOT NULL,
    Salary      NUMBER,
    Sex         CHAR(1),
    Address     CHAR(30),
    Bdate      DATE,
    DepNumber   NUMBER,
    Ssn23      NUMBER
);
CREATE TABLE PROJECT
(
    DepNumber    NUMBER,
    PrjNumber    NUMBER          NOT NULL,
    PrjName     CHAR(30),

```

เอกสารนี้เป็นลิขสิทธิ์สงวนไว้ใช้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์อื่นใดเป็นการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    PrjLocation    CHAR(30)
);
CREATE TABLE WORKS_ON#1
(
    PrjNumber      NUMBER          NOT NULL,
    Hours          DATE            NOT NULL
);
CREATE TABLE WORKS_ON#2
(
    Ssn            NUMBER          NOT NULL,
    PrjNumber      NUMBER          NOT NULL
);
CREATE TABLE WORKS_ON#3
(
    Ssn            NUMBER          NOT NULL,
    Hours          DATE            NOT NULL
);
CREATE UNIQUE INDEX INDEX_#1 ON DEPARTMENT (DepNumber) ;
CREATE UNIQUE INDEX INDEX_#2 ON DEPENDENT#1 (DepenName,BirthDate) ;
CREATE UNIQUE INDEX INDEX_#3 ON DEPENDENT#2 (Sex,DepenName,Relationship) ;
CREATE UNIQUE INDEX INDEX_#4 ON DepLocation (DepNumber,DepLocation) ;
CREATE UNIQUE INDEX INDEX_#5 ON EMPLOYEE (Ssn) ;
CREATE UNIQUE INDEX INDEX_#6 ON PROJECT (PrjNumber) ;
CREATE UNIQUE INDEX INDEX_#8 ON WORKS_ON#1 (PrjNumber,Hours) ;
CREATE UNIQUE INDEX INDEX_#9 ON WORKS_ON#2 (Ssn,PrjNumber) ;
CREATE UNIQUE INDEX INDEX_#10 ON WORKS_ON#3 (Ssn,Hours) ;

CREATE VIEW DEPENDENT#1_VIEW AS SELECT *
FROM DEPENDENT#1
WHERE Sex IN ('F','M')
WITH CHECK OPTION ;

CREATE VIEW DepLocation_VIEW AS SELECT *
FROM DepLocation
WHERE DepLocation IN ('Athens','London','Paris')
WITH CHECK OPTION ;

CREATE VIEW EMPLOYEE_VIEW AS SELECT *
FROM EMPLOYEE
WHERE Sex IN ('F','M') AND ((Salary >= 100 AND Salary <= 500))
WITH CHECK OPTION ;

CREATE VIEW PROJECT_VIEW AS SELECT *
FROM PROJECT
WHERE PrjLocation NOT IN ('CANADA','USA')
WITH CHECK OPTION ;

```

รูปที่ 6.22 แสดงผลลัพธ์ของตารางที่ได้ในบรรทัดฐานขั้นที่ห้า

## บทที่ 7

### สรุปและแนวทางการวิจัย

#### 7.1 สรุป

วัตถุประสงค์ในการทำวิทยานิพนธ์ฉบับนี้ก็เพื่อที่จะสร้างระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลแบบรีเลชันแนลชื่อ ER/5 ที่มีประสิทธิภาพสูงโดยการนำแบบจำลองอีอาร์ของ Ramez Elmasri/Shamkant B.Navathe [2] ซึ่งเป็นที่นิยมใช้กันอย่างแพร่หลายมาพัฒนาเพิ่มประสิทธิภาพจนสามารถที่จะออกแบบฐานข้อมูลได้ตารางฐานข้อมูลในรูปแบบบรรทัดฐานขั้นที่ห้า ซึ่งในทางทฤษฎีแล้วถือว่าเป็นตารางฐานข้อมูลที่ปราศจากความซ้ำซ้อนโดยสิ้นเชิง ซึ่งฐานข้อมูลใดก็ตามที่ยังมีความซ้ำซ้อนอยู่อาจเป็นสาเหตุทำให้เกิดการประมวลผลข้อมูลที่ไม่ตรงกับความเป็นจริง (Database inconsistency and modification anomalies) เช่น

PROJECTS

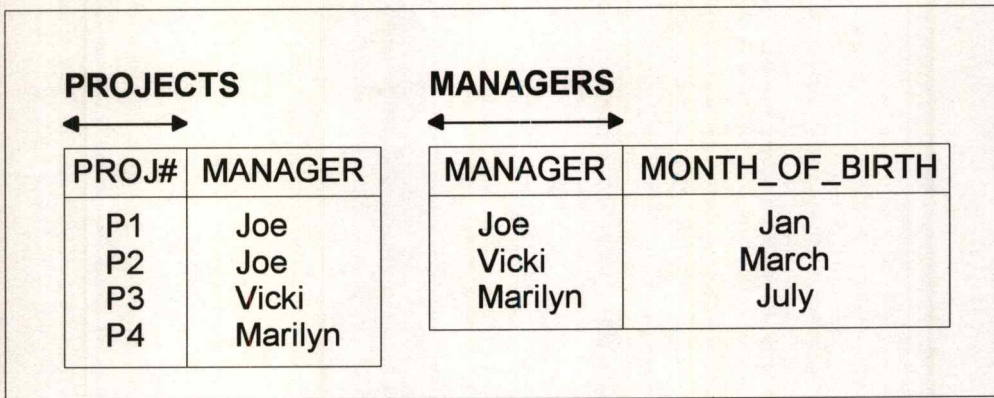
PROJ#	MANAGER	MONTH_OF_BIRTH
P1	Joe	Jan
P3	Vicki	March
P2	Joe	Jan
P4	Marilyn	July

รูปที่ 7.1 แสดงตารางฐานข้อมูลยังมีความซ้ำซ้อน

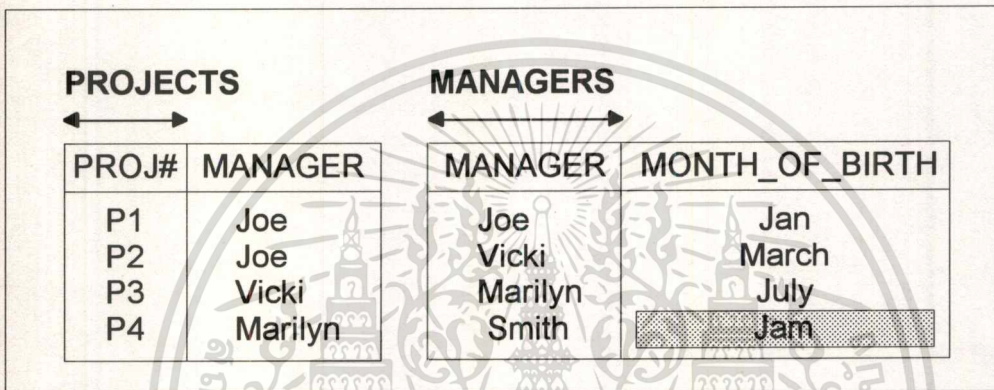
จากตารางฐานข้อมูลในรูปที่ 7.1 เป็นตารางฐานข้อมูลที่ยังมีความซ้ำซ้อนเนื่องจากแอททริบิวต์ที่ไม่ได้เป็นกุญแจหลักของตารางคือ MONTH\_OF\_BIRTH สามารถขึ้นอยู่กับแอททริบิวต์ MANAGER ซึ่งเป็นแอททริบิวต์ที่ไม่เป็นกุญแจหลักของตารางเช่นเดียวกัน ทำให้เกิดความซ้ำซ้อนของตารางฐานข้อมูลขึ้นได้แก่ เมื่อไหร่ก็ตามที่เรากำหนด MANAGER จะได้ผลลัพธ์ของ MONTH\_OF\_BIRTH ซึ่งเป็นค่าเดียวกันจำนวนหลายแถว ข้อเสียคือถ้าต้องการแก้ไขข้อมูลของ MANAGER จะต้องแก้ไขข้อมูลของ MONTH\_OF\_BIRTH ด้วยค่าที่สอดคล้องกัน จะแก้ไขเพียงแอททริบิวต์ใดแอททริบิวต์หนึ่งไม่ได้ จากเหตุผลดังกล่าวทำให้ข้อมูลที่ถูกเก็บลงในตารางฐานข้อมูลนี้อาจจะไม่ตรงกับความจริงได้ถ้าเมื่อไหร่ก็ตามที่มีการแก้ไข MANAGER หรือ MONTH\_OF\_BIRTH เพียงแอททริบิวต์เดียว วิธีการกำจัดความซ้ำซ้อนของตารางฐานข้อมูลนี้เราต้องแยกตารางฐานข้อมูลนี้ออกเป็นสองตารางดังแสดงในรูปที่ 7.2 ซึ่งตารางฐานข้อมูลทั้งสองที่เกิดขึ้นใหม่นี้จัดอยู่ในตารางฐานข้อมูลในรูปแบบบรรทัดฐานขั้นที่ห้าซึ่งปราศจาก

ความซ้ำซ้อนโดยสิ้นเชิง

ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.2 แสดงตารางฐานข้อมูลที่ไม่มีความซ้ำซ้อน

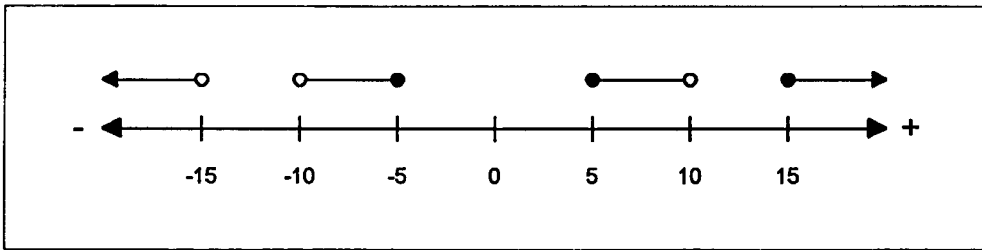


รูปที่ 7.3 แสดงตารางฐานข้อมูลที่ใส่ข้อมูลผิดพลาดจากผู้ใช้งาน

จากรูปที่ 7.3 ถึงแม้ว่าจะเป็นตารางฐานข้อมูลที่ไม่ซ้ำซ้อนแต่ข้อมูลอยู่บนตารางฐานข้อมูลยังไม่ถูกต้องซึ่งเป็นความผิดพลาดดังกล่าวเกิดจากผู้ใช้ข้อมูลนั้นต้องการใส่ข้อมูลว่า Smith เป็น MANAGER ที่ยังไม่มี PROJ และเกิดในเดือนมกราคม (Jan) แต่ได้ใส่ข้อมูลเดือนที่เกิดของผู้จัดการ Smith ผิดกลายเป็น (Jam) ซึ่งไม่ได้หมายถึงชื่อของเดือนต่างๆในรอบปี ทำให้ข้อมูลที่ผิดพลาดดังกล่าวถูกเก็บลงบนตารางฐานข้อมูลและเมื่อต้องการเรียกดูฐานข้อมูลว่าผู้จัดการคนใดที่เกิดในเดือนมกราคม (Jan) ผู้จัดการ Smith ก็จะไม่ถูกแสดงออกให้เห็น ทั้งที่ในความต้องการจริงๆแล้วจะต้องมีผู้จัดการ Smith ปรากฏให้เห็น จากการผิดพลาดของผู้ใช้งานดังกล่าวถ้าเกิดขึ้นกับแอททริบิวต์อื่นๆและเป็นจำนวนมากแล้วจะเกิดผลเสียหายต่อระบบจัดเก็บฐานข้อมูลจนไม่สามารถที่จะยอมรับได้

ดังนั้นในระบบ ER/5 จึงสร้างระบบป้องกันความผิดพลาดจากผู้ใช้งานดังกล่าว ด้วยการให้ผู้ออกแบบสามารถที่จะกำหนดตัวควบคุม (Domain constraint) ลงในแอททริบิวต์ต่างๆบนแบบจำลองอีอาร์ได้ซึ่งตัวควบคุมดังกล่าวสามารถใช้ได้กับแอททริบิวต์ที่มีคุณลักษณะ (Datatype) ดังต่อไปนี้คือ

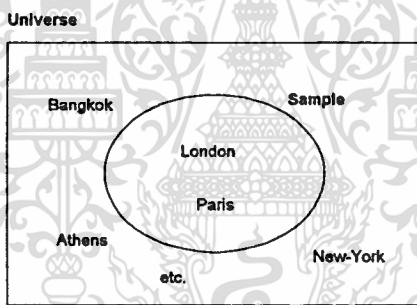
- การกำหนดตัวควบคุมเป็นตัวเลข



รูปที่ 7.4 แสดงข้อมูลที่อยู่ในเส้นจำนวน

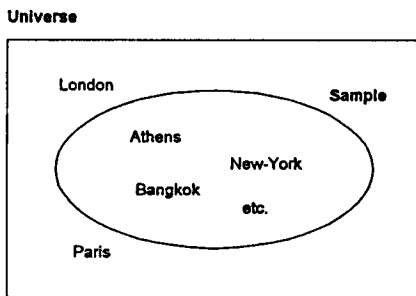
จะใช้กับแอททริบิวต์ที่มีคุณลักษณะเป็นตัวเลข โดยจะกำหนดเป็นลักษณะของเส้นจำนวน เช่นถ้าต้องการให้แอททริบิวต์ Sample มีข้อมูลที่อยู่บนเส้นจำนวนดังรูปที่ 7.4 จะต้องกำหนดตัวควบคุมที่มีรูปแบบดังนี้  $\{ (<,-15),(-10,-5],[5,10],[15,>) \}$  เป็นต้น

- การกำหนดตัวควบคุมเป็นตัวอักษร



รูปที่ 7.5 แสดงข้อมูลที่อยู่ในเซต Sample

จะใช้กับแอททริบิวต์ที่มีคุณลักษณะเป็นตัวอักษรโดยจะกำหนดเป็นลักษณะของเซต เช่นถ้าต้องการให้แอททริบิวต์ Sample มีข้อมูลที่อยู่ในเซตดังรูปที่ 7.5 จะต้องกำหนดตัวควบคุมที่มีรูปแบบดังนี้  $\{ 'London','Paris' \}$



รูปที่ 7.6 แสดงข้อมูลไม่อยู่ในเซต Sample

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการแข่งขันเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าไม่ต้องการให้แอททริบิวต์ Sample มีข้อมูลอยู่ในเซตดังรูปที่ 7.6 ต้องกำหนดตัวควบคุมที่มีรูปแบบดังนี้ ! { 'London','Paris' }

ดังนั้นถ้าผู้ออกแบบฐานข้อมูลไม่ต้องการให้เกิดความผิดพลาดขึ้นดังแสดงในรูปที่ 7.3 จะต้องกำหนดตัวควบคุมให้กับแอททริบิวต์ MONTH\_OF\_BIRTH ลงบนแบบจำลองอีอาร์ ดังนี้คือ { 'Jan','Feb','March','April','May','June','July','Aug','Sep','Oct','Nov','Dec' } ระบบ ER/5 จะทำการสร้างวิวขึ้นมาตรวจสอบข้อมูลของแอททริบิวต์ MONTH\_OF\_BIRTH ว่าจะต้องเป็นสมาชิกของเซตชื่อเดือนต่างๆในรอบปีเท่านั้นจึงจะยอมให้ข้อมูลถูกเก็บลงบนตารางฐานข้อมูลได้

สุดท้ายนี้กระผม นายธวัชชัย สุทธิธรรม หวังเป็นอย่างยิ่งว่าระบบ ER/5 ที่สร้างขึ้นจากวิทยานิพนธ์ฉบับนี้จะถูกนำไปใช้งานกันอย่างแพร่หลาย เพราะการออกแบบฐานข้อมูลด้วยระบบ ER/5 นี้สามารถใช้งานได้กับเครื่องไมโครคอมพิวเตอร์ IBM PC ซึ่งมีอยู่เป็นจำนวนมากในประเทศไทยและตัวแบบจำลองฐานข้อมูลอีอาร์ที่ใช้ในการออกแบบฐานข้อมูลก็เป็นที่ยอมรับกันดีในกลุ่มผู้ออกแบบฐานข้อมูลว่าออกแบบได้ง่ายไม่ซับซ้อนอีกทั้งระบบ ER/5 ยังติดต่อกับผู้ออกแบบฐานข้อมูลในแบบกราฟฟิกทำให้ผู้ออกแบบฐานข้อมูลสามารถวาดรูปแบบจำลองอีอาร์ได้สะดวกรวดเร็วยิ่งขึ้น และได้ผลงานที่สวยงาม ผลลัพธ์ของการออกแบบฐานข้อมูลด้วยระบบ ER/5 ยังเป็นตารางฐานข้อมูลที่ไม่ซ้ำซ้อนและสามารถควบคุมความถูกต้องของข้อมูลในระดับแอททริบิวต์ได้ดีอีกด้วย

## 7.2 แนวทางการวิจัย

สำหรับการวิจัยในครั้งนี้ได้ตั้งวัตถุประสงค์ว่าจะต้องสร้างระบบอัตโนมัติช่วยในการออกแบบฐานข้อมูลที่ยอมรับกันอย่างแพร่หลายใช้งานได้สะดวกบนเครื่องไมโครคอมพิวเตอร์ และได้ผลลัพธ์ในการออกแบบเป็นตารางฐานข้อมูลในบรรทัดฐานขั้นที่ห้าซึ่งปราศจากความซ้ำซ้อนสามารถควบคุมความถูกต้องของข้อมูลในระดับแอททริบิวต์ได้ ดังนั้นระบบ ER/5 นี้จึงยังขาดความสามารถอีกหลายอย่างที่จะทำให้เป็น CASE ในระบบฐานข้อมูลได้อย่างสมบูรณ์ เช่น หน่วยสร้างโปรแกรมประยุกต์ (Application Generator Unit) หน่วยสร้างรายงาน (Report Generator Unit) หน่วยสร้างเมนู (Menu Generator Unit) หน่วยสร้างฟอร์ม (Form Generator Unit) และหน่วยปรับแต่งตารางฐานข้อมูล (Database Tuning Unit) เป็นต้น

แนวทางการวิจัยต่อจากวิทยานิพนธ์ฉบับนี้เป็นได้ทั้งหมดจากที่กล่าวมาเพื่อให้ระบบ ER/5 เป็น CASE ในระบบฐานข้อมูลอย่างสมบูรณ์และมีประสิทธิภาพสูง

## เอกสารอ้างอิง

- [1] I.T. Hawryskiewicz, *"RELATIONAL DATABASE DESIGN AN INTRODUCTION"*, PRENTICE HALL, 1990.
- [2] Ramez Elmasri/Shamkant B.Navathe, *"FUNDAMENTALS OF DATABASE SYSTEMS"*, The Benjamin/Cummings Publishing Company,Inc., 1989.
- [3] Alan F.Dutka and Howard H.Hanson,*"Fundamentals of Data Normalization"*, ADDISON-WESLEY PUBLISHING COMPANY, 1989.
- [4] C.J. Date, *"AN INTRODUCTION TO Database Systems VOLUME I"*, ADDISON-WESLEY PUBLISHING COMPANY, 1986.
- [5] RICHARD BARKER,*"CASE\*METHOD™ Entity Relationship Modelling"*, ADDISON-WESLEY PUBLISHING COMPANY, 1989.
- [6] G.M Nijssen and T.A. Halpin, *"CONCEPTUAL SCHEMA and RELATIONAL DATA BASE DESIGN"*, PRENTICE HALL, 1989.
- [7] C.J. Date,*"A Guide to THE SQL STANDARD"*, ADDISON-WESLEY PUBLISHING COMPANY, MASSACHUSATTS, 1987.
- [8] C.J. Date with COLIN J.WHITE, *"A Guide to SQL/SD"*, ADDISON-WESLEY PUBLISHING COMPANY, 1989.
- [9] David C. Kay and John R. Levine, *"Graphics File Formats"*, WINDCREST/McGRAW-HILL,1992
- [10] Brent Rector, *"Developing Windows™ 3 Applications with Microsoft SDK"*, SAMS, 1992
- [11] Jonathan Sachs, *"SQL Plus REFERENCE GUIDE"*, ORACLE, 1987
- [12] Donna Neville, *"PRO\*C USER'S GUIDE"*, ORACLE, 1987
- [13] Gerald E.Sobelman and David E.Krekelberg, *"Advance C: Techniques and Applications"*, Que Corporation Indianapolis,Indiana, 1985
- [14] Microsoft Corporation, *"Microsoft Windows Software Development Kit Guide to Programming Version 3.0"*, 1990

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก การติดตั้ง การเรียกโปรแกรม และการใช้งานระบบ ER/5

### อุปกรณ์ที่ต้องใช้ขั้นต่ำสำหรับระบบ ER/5

1. เครื่องไมโครคอมพิวเตอร์ CUP เบอร์ 80386 หรือสูงกว่า
2. หน่วยความจำขนาด 16 เมกกะไบต์
3. ฮาร์ดดิสก์
3. ระบบปฏิบัติการ MS-DOS
4. โปรแกรมไมโครซอฟท์วินโดวเวอร์ชัน 3.1
5. ระบบจัดการฐานข้อมูลออรากเคิล เวอร์ชัน 6.0

### การติดตั้งระบบ ER/5

หลังจากที่ติดตั้งระบบปฏิบัติการ MS-DOS และโปรแกรมไมโครซอฟท์วินโดว 3.1 เป็นที่เรียบร้อยแล้วให้ปฏิบัติตามขั้นตอนดังต่อไปนี้

1. ติดตั้งระบบจัดการฐานข้อมูลออรากเคิลในแบบโปรเท็กท์โหมดเฉพาะส่วน RDBMS เท่านั้น (หาอ่านได้จากคู่มือการติดตั้งของ บริษัท ออรากเคิล ประเทศไทย จำกัด) หลังจากติดตั้งเสร็จเป็นที่เรียบร้อยแล้วให้เพิ่มประโยค `DYNAMIC_MEMORY=3072` ลงบนไฟล์ `CONFIG.ORA` เป็นการกำหนดให้ระบบจัดการฐานข้อมูลออรากเคิลใช้หน่วยความจำขนาด 3 เมกกะไบต์ จะได้มีหน่วยความจำเหลือพอที่จะสามารถโหลดโปรแกรมไมโครซอฟท์วินโดวมาใช้งานได้

2. ใส่แผ่นดิสก์ระบบ ER/5 ที่ไดรฟ์ a:
3. พิมพ์คำว่า `c:\> a:install` แล้วกด Enter
4. เรียกโปรแกรมไมโครซอฟท์วินโดวให้ใช้เมนู File ของ Program manager ทำการสร้าง ER/5 ไอคอนของโปรแกรมระบบ ER/5 พร้อมกำหนดว่าให้ใช้โปรแกรม `ERFIVEPC.EXE` โดยสามารถใส่ไว้ที่กลุ่มใดก็ได้ตามชอบใจ

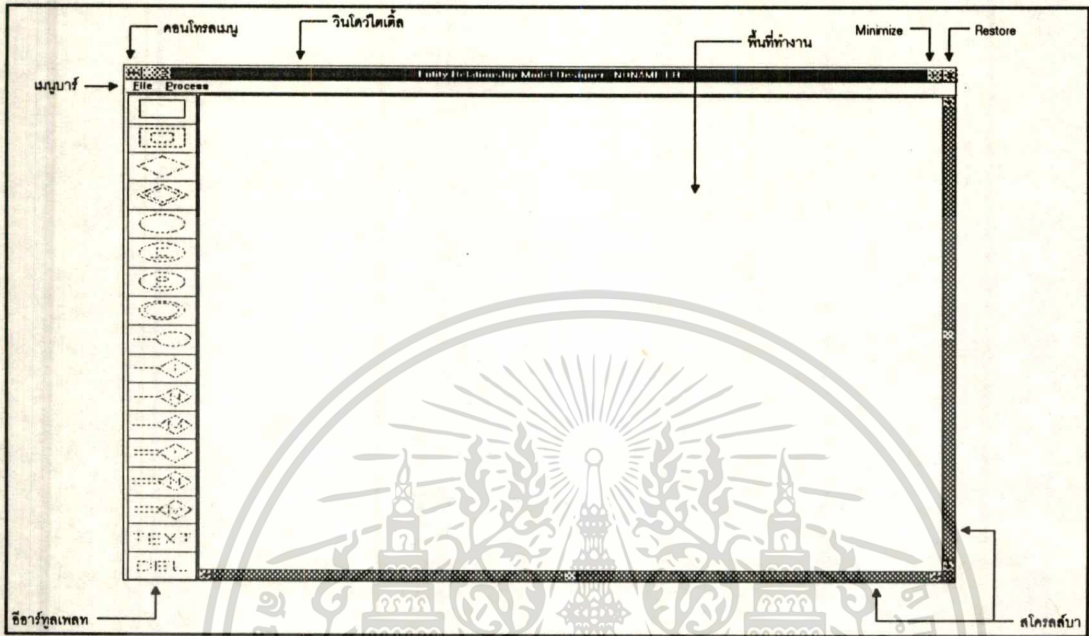
### การเรียกโปรแกรม ER/5

การเรียกโปรแกรม ER/5 มาใช้งานนั้นผู้ใช้งานจะต้องแน่ใจว่ามีตัวโปรแกรมจัดการฐานข้อมูลออรากเคิลอยู่บนหน่วยความจำเรียบร้อยแล้วถ้ายังไม่มีให้โหลดตัวโปรแกรมจัดการฐานข้อมูลออรากเคิลมาไว้บนหน่วยความจำโดยพิมพ์คำว่า `C:\>ORACLE` แล้วกดปุ่ม Enter หลังจากนั้นให้เรียกโปรแกรมไมโครซอฟท์วินโดวมาทำงานใน Standard โหมดโดยพิมพ์คำว่า `C:\>win /s` แล้วกดปุ่ม Enter ก็จะเข้าสู่ระบบจัดการโปรแกรมของไมโครซอฟท์วินโดวให้เลื่อนเมาส์ไปยัง ER/5 ไอคอนแล้วกดดับเบิลคลิกที่ปุ่มซ้ายของเมาส์โปรแกรมระบบ ER/5 ก็จะถูกเรียกมาใช้งานทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การใช้งานระบบ ER/5

การใช้งานระบบ ER/5 นี้จะอธิบายถึงส่วนประกอบวินโดว์ของระบบ ER/5 วิธีการวาดแบบจำลองอีอาร์และการสร้างตารางฐานข้อมูล โดยจะจำลองการออกแบบจริงเป็นขั้นตอนเพื่อให้เข้าใจง่าย



รูปที่ A1 แสดงหน้าจอของระบบ ER/5

### - ส่วนประกอบวินโดว์ของระบบ ER/5

จากรูปที่ A1 แสดงหน้าจอของระบบ ER/5 ซึ่งจะประกอบด้วย

1. คอนโทรลเมนู เป็นเมนูของระบบไมโครซอฟท์วินโดว์ใช้ในการเปลี่ยนแปลงขนาดของวินโดว์ ปิดวินโดว์ หรือเปลี่ยนไปทำงานที่วินโดว์อื่น
2. วินโดว์ไตเติ้ล มีไว้แสดงชื่อไฟล์ที่กำลังทำงานอยู่
3. ปุ่ม Minimize ใช้สั่งให้วินโดว์รันเป็นแบลคกราวนโปรเซสพร้อมทั้งซ่อนวินโดว์ของตัวเองให้อยู่ในรูปไอคอน
4. ปุ่ม Restore ใช้สั่งให้ขนาดของวินโดว์กลับไปมีขนาดเหมือนเดิมก่อนหน้า
5. เมนูบาร์ แสดงเมนูของระบบ ER/5 ซึ่งประกอบด้วยเมนูหลัก 2 เมนูคือ
  - 5.1 FILE เมนู ใช้ในการสร้าง แก้ไข เก็บ พิมพ์ และปิดเพิ่มข้อมูล
  - 5.2 PROCESS เมนู ใช้ในการสร้างตารางฐานข้อมูลในรูปบรรทัดฐานขั้นแรก ชั้น Boyce/Codd และชั้นที่ห้า
6. อีอาร์ทูลเพลท ใช้เป็นเพลทสำหรับช่วยในการวาดสัญลักษณ์ต่างๆของแบบจำลองอีอาร์ ซึ่งจะอธิบายการใช้งานโดยละเอียดต่อไป
7. พื้นที่ทำงาน เป็นพื้นที่สำหรับวาดแบบจำลองอีอาร์

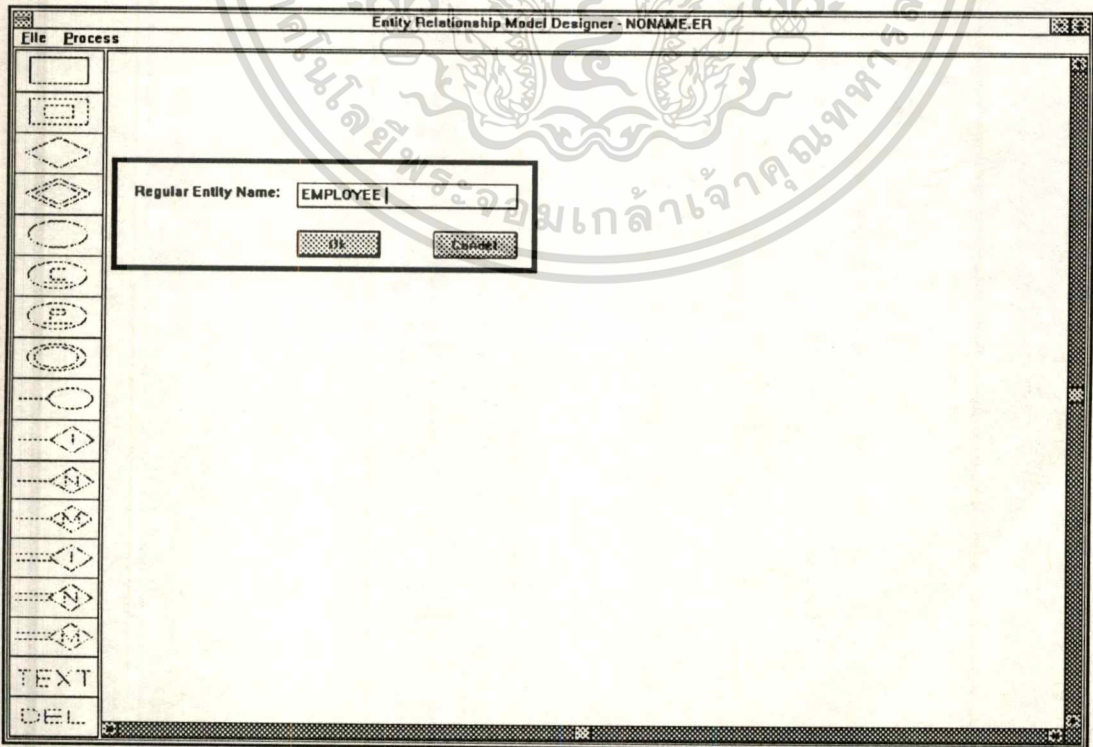
เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของสำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิสารสนเทศ (องค์การมหาชน) กระทรวงวิทยาศาสตร์และเทคโนโลยี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. โครลล์บาร์ ใช้เลื่อนพื้นที่ทำงานในกรณีที่ไม่สามารถแสดงแบบจำลองอ็อบจ็องที่สร้างขึ้นได้ในหน้าจอเดียว

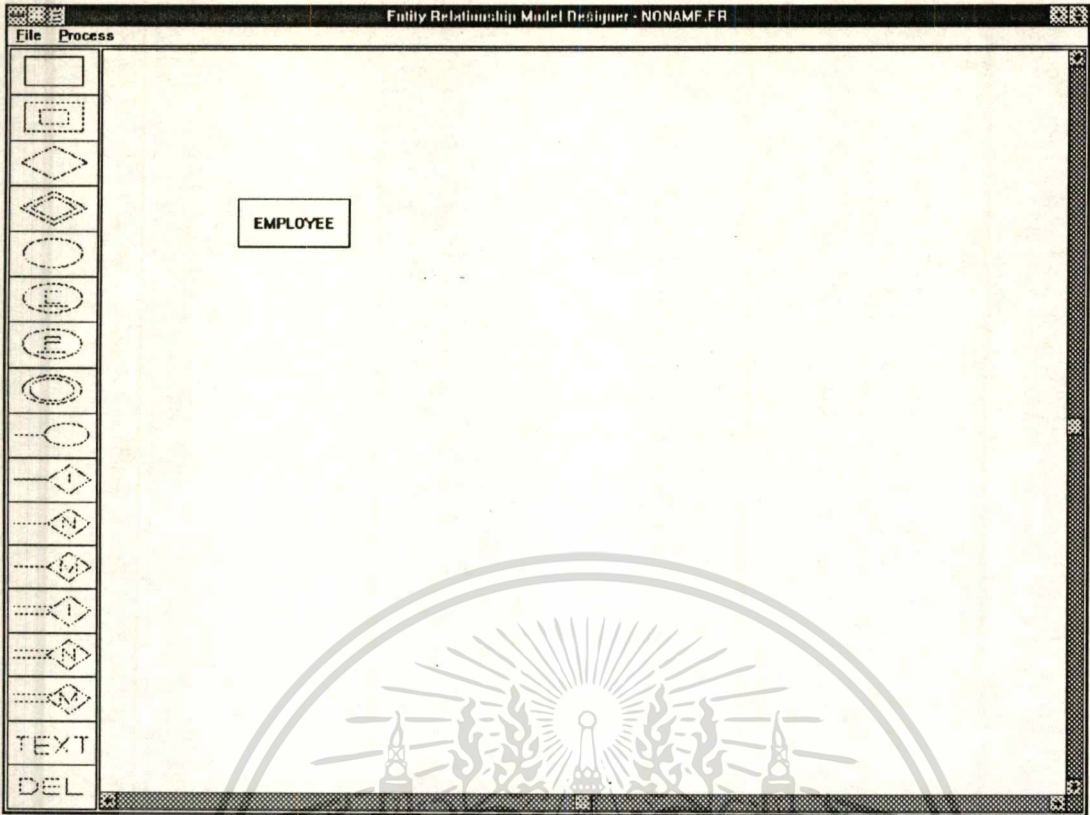
- การวาดแบบจำลองอ็อบจ็อง

- การวาดเอนติตี้ชนิดปกติ

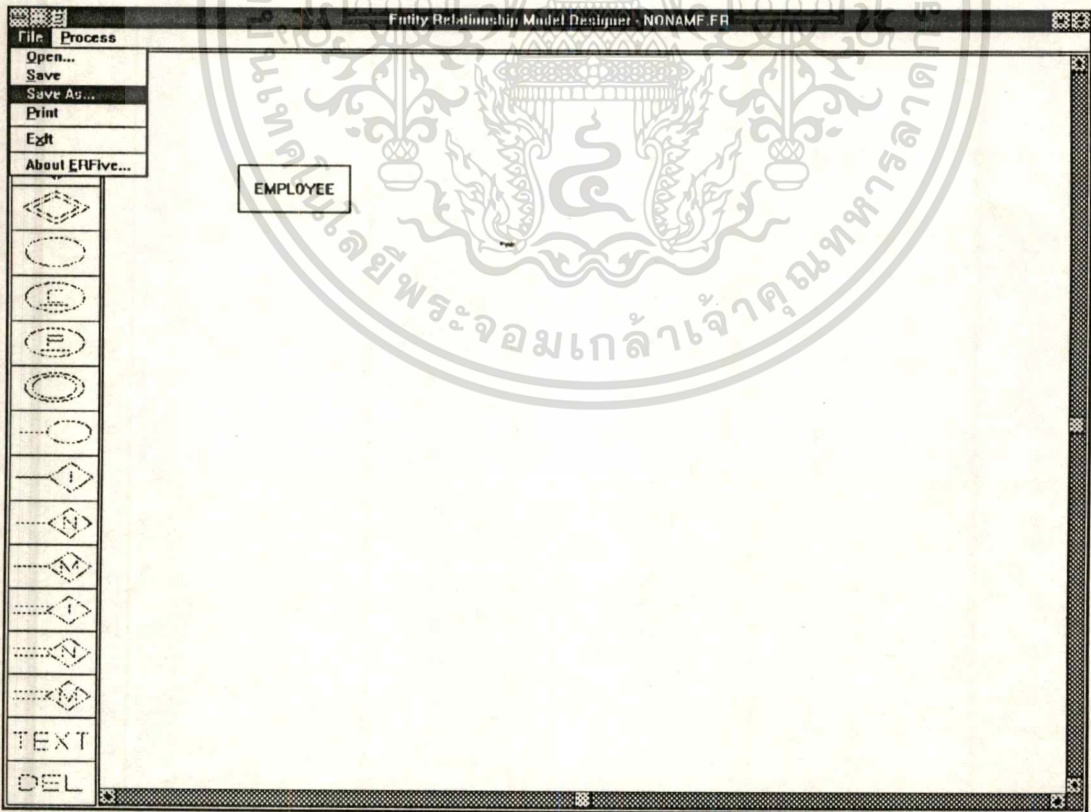
ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของเอนติตี้ชนิดปกติซึ่งเป็นรูปสี่เหลี่ยมผืนผ้าบนแถวแรกของอ็อบจ็องทูลเพลทเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปมือที่ใช้ชี้เลือกเพลทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว่าสัญลักษณ์เอนติตี้ชนิดปกติจะเปลี่ยนจากสี่เหลี่ยมเป็นสี่น้ำเงิน (ถ้าเป็นสี่น้ำเงินอยู่แต่เดิมแสดงว่าถูกเลือกไว้ก่อนแล้ว) หลังจากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นตัวอักษร A แล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งจะมีกรอบข้อความเกิดขึ้นบนพื้นที่ทำงานดังแสดงในรูปที่ A2 ให้ใส่ชื่อของเอนติตี้ชนิดปกตินั้นตามต้องการโดยที่ชื่อของเอนติตี้ชนิดปกติที่ตั้งขึ้นนั้นจะต้องสามารถใช้เป็นชื่อของตารางบนฐานข้อมูลได้เสร็จแล้วให้กดปุ่ม OK เพื่อแสดงการยืนยันกรอบข้อความจะหายไปแต่จะมีกรอบสี่เหลี่ยมผืนผ้าว่างๆ แสดงขึ้นบนพื้นที่ทำงานแทนให้ใช้เมาท์เลื่อนรูปสี่เหลี่ยมผืนผ้าที่ว่างๆ ไปยังตำแหน่งที่ต้องการบนพื้นที่ทำงานโดยจะต้องไม่ไปทับกับวัตถุเดิมที่มีปรากฏอยู่ก่อนแล้วบนพื้นที่ทำงานแล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งรูปสี่เหลี่ยมผืนผ้าที่ว่างๆ จะถูกวางลงบนพื้นที่ทำงานพร้อมทั้งมีชื่อของเอนติตี้ชนิดปกติที่ใส่ลงในกรอบข้อความมาปรากฏอยู่ตรงกลางภายในกรอบสี่เหลี่ยมผืนผ้าที่ว่างๆ ดังแสดงในรูปที่ A3



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสำนักงานคณะกรรมการการอุดมศึกษาและวิทยาศาสตร์ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

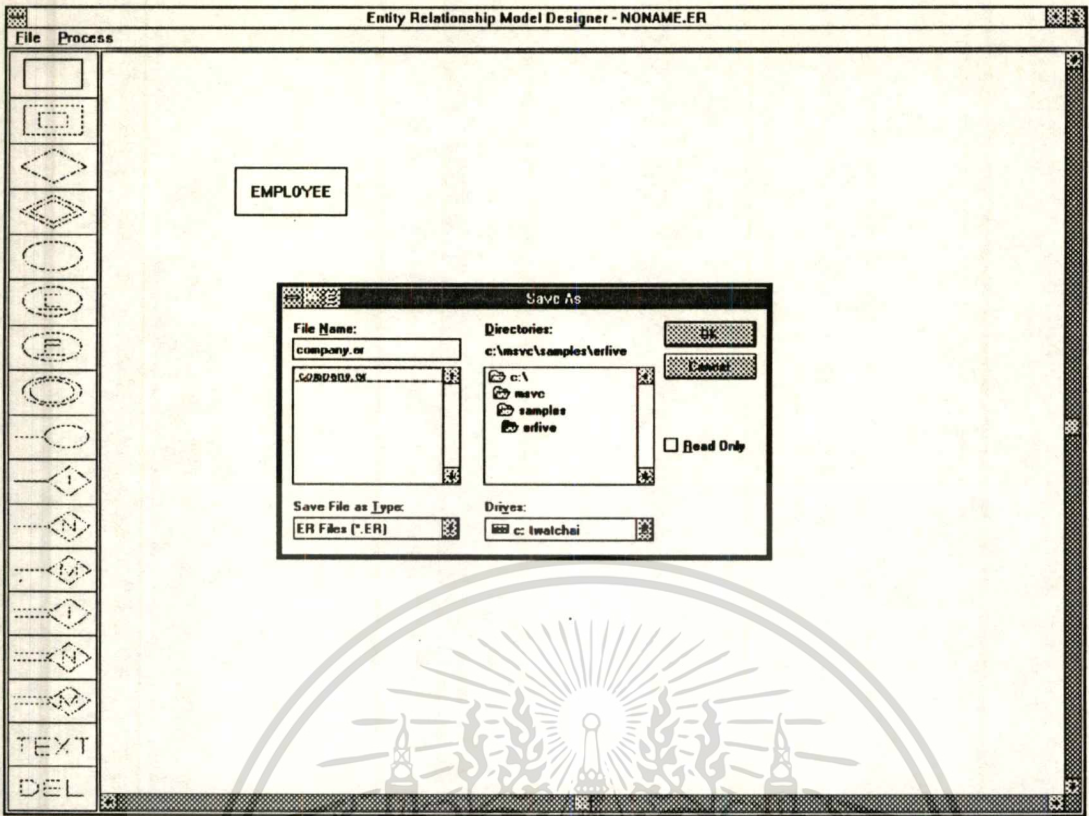


รูปที่ A3 แสดงภาพเอนิตีชนิดปกติ EMPLOYEE ที่ถูกวาดขึ้น

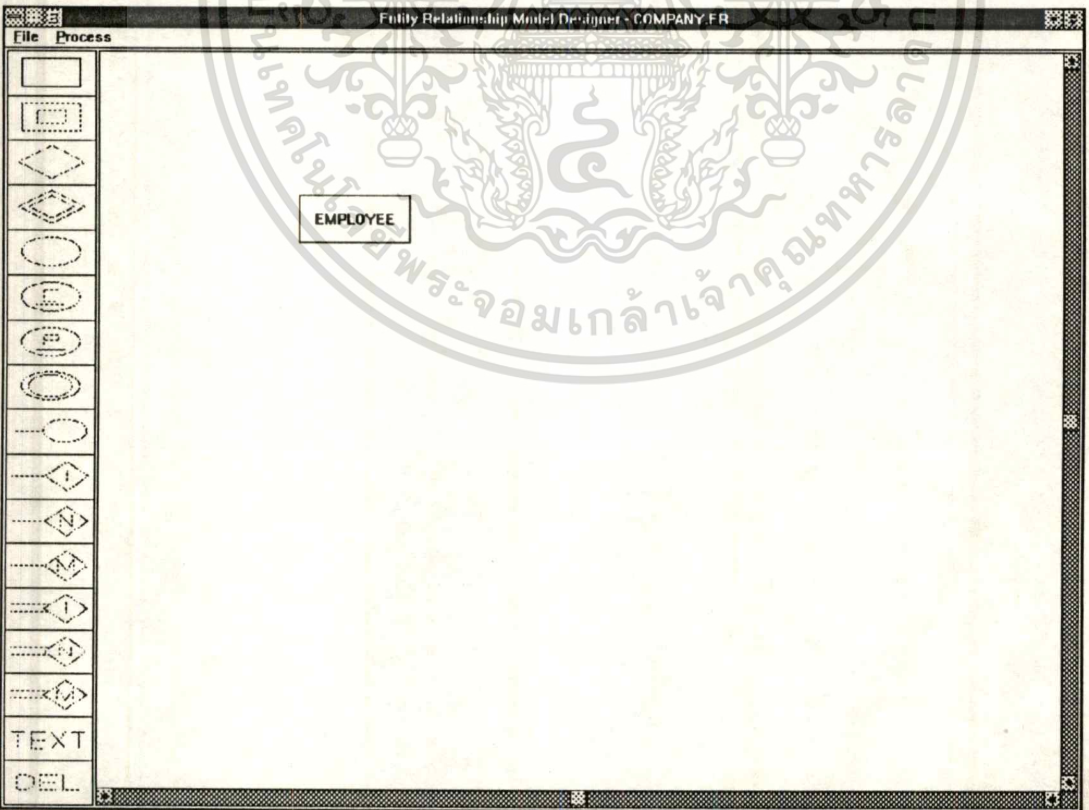


รูปที่ A4 แสดงการเลือกเมนู Save As... บนเมนูหลัก File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ A5 แสดงการเก็บข้อมูลโดยใช้ชื่อไฟล์ COMPANY.ER



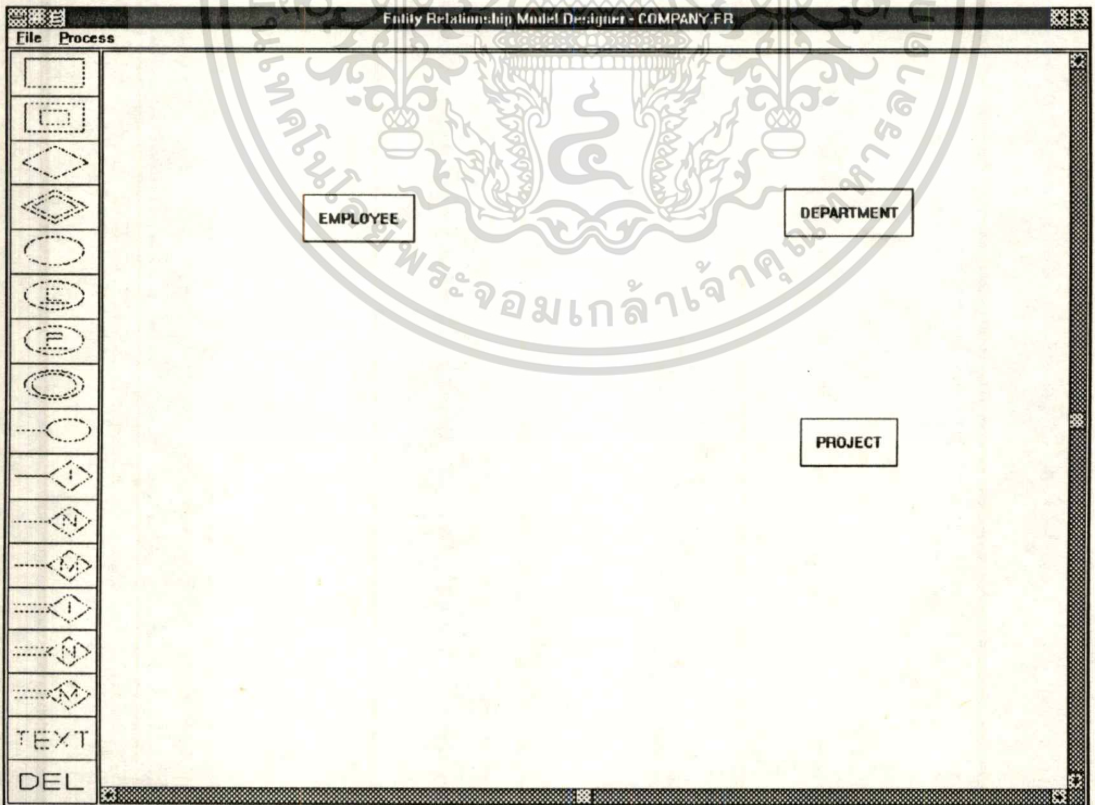
รูปที่ A6 แสดงไฟล์ไต่เตลที่เปลี่ยนจาก NONAME.ER เป็น COMPANY.ER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ A4 A5 และ A6 แสดงขั้นตอนการจับเก็บข้อมูลของแบบจำลองอ็อบเจกต์ที่วาดขึ้นเพื่อให้สามารถนำมาปรับปรุงแก้ไขได้ในภายหลังและต้องการเปลี่ยนชื่อไฟล์ที่ใช้จับเก็บ ผู้ใช้งานจะต้องเลือกเมนู Save As... บนเมนูหลัก File ดังแสดงในรูปที่ A4 เมื่อผู้ใช้งานได้เลือกเมนูนี้แล้วจะมีกรอบข้อความเกิดขึ้น ให้ผู้ใช้งานใส่ชื่อไฟล์ตามต้องการ ซึ่งในรูปที่ A5 ชื่อไฟล์ที่ใช้เก็บแบบจำลองอ็อบเจกต์นี้ใช้ชื่อไฟล์ว่า company.er หลังจากนั้นให้กดปุ่ม OK เพื่อแสดงการยืนยันข้อมูลของแบบจำลองอ็อบเจกต์ที่วาดขึ้นทั้งหมดก็จะถูกเก็บลงในไฟล์ชื่อ company.er ตามต้องการและจะสังเกตเห็นว่าวินโดว์ได้เปลี่ยนชื่อจาก NONAME.ER เป็น COMPANY.ER ดังแสดงในรูปที่ A6

ในกรณีที่ผู้ใช้งานต้องการจับเก็บข้อมูลของแบบจำลองอ็อบเจกต์โดยใช้ชื่อไฟล์เดิม ผู้ใช้งานจะต้องเลือกเมนู Save บนเมนูหลัก File ข้อมูลของแบบจำลองอ็อบเจกต์ที่วาดขึ้นทั้งหมดก็就会被เก็บลงในชื่อไฟล์เดิม

การนำแบบจำลองอ็อบเจกต์ที่ถูกจับเก็บกลับมาแก้ไขใหม่ ผู้ใช้งานจะต้องเลือกเมนู Open บนเมนูหลัก File จะมีกรอบข้อความเกิดขึ้น ให้ผู้ใช้งานใส่ชื่อไฟล์ข้อมูลของแบบจำลองอ็อบเจกต์ที่ต้องการ แบบจำลองอ็อบเจกต์ที่ใช้งานอยู่ในขณะนั้นจะถูกยกเลิกเปลี่ยนเป็นไฟล์ที่ผู้ใช้งานได้เลือกใหม่แทน

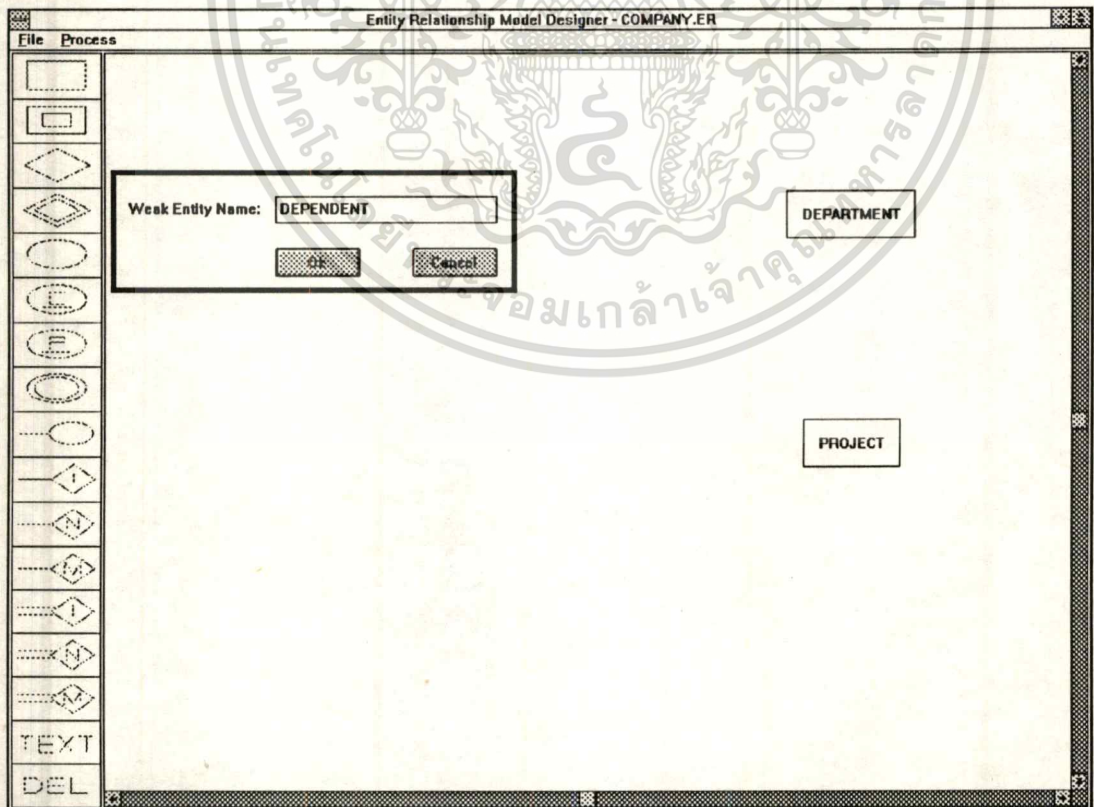


รูปที่ A7 แสดงการวาดเอนิตีเซตชนิดปกติทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

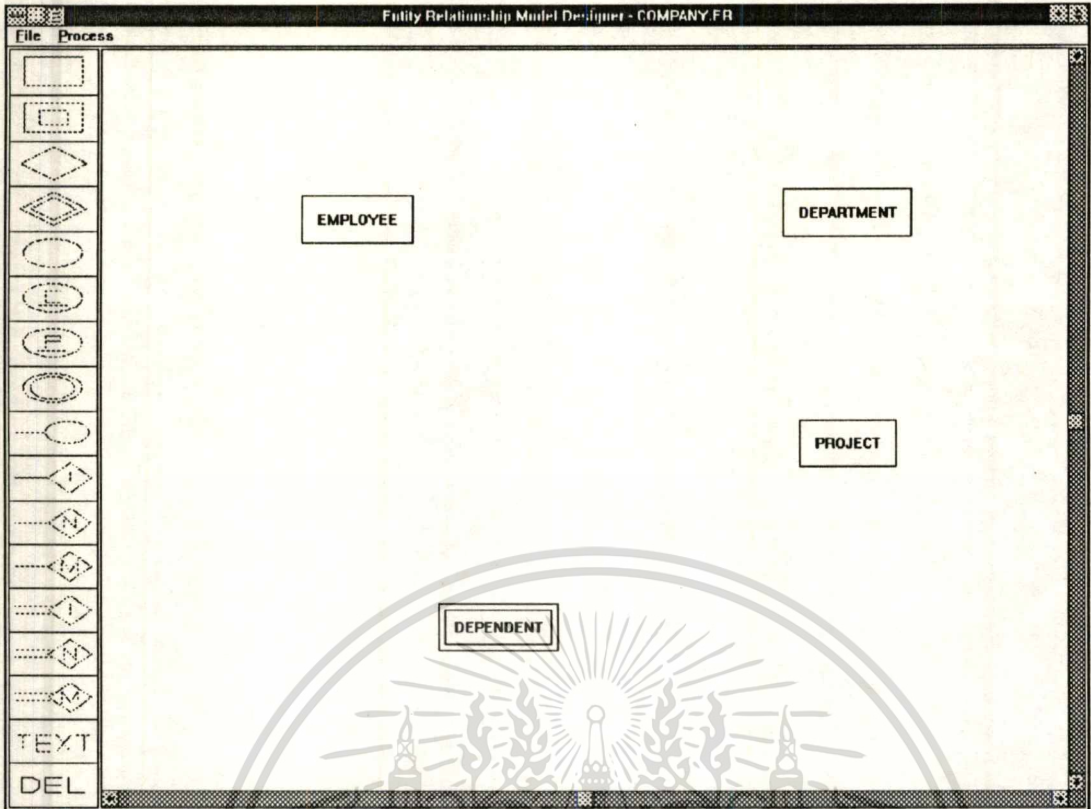
- การวาดเอนติตี้ชนิดอ่อน

ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของเอนติตี้ชนิดอ่อนซึ่งเป็นรูปสี่เหลี่ยมผืนผ้าซ้อนกันสองรูปบนแถวที่สองของอาร์ทูลเพลทเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปมือที่ใช้ชี้เลือกเพลทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว่าสัญลักษณ์เอนติตี้ชนิดอ่อนจะเปลี่ยนจากสี่เหลี่ยมเป็นสี่น้ำเงิน (ถ้าเป็นสี่น้ำเงินอยู่แต่เดิมแสดงว่าถูกเลือกไว้ก่อนแล้ว) หลังจากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นตัวอักษร A แล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งจะมีกรอบข้อความเกิดขึ้นบนพื้นที่ทำงานดังแสดงในรูปที่ A8 ให้ใส่ชื่อของเอนติตี้ชนิดอ่อนนั้นตามต้องการโดยที่ชื่อของเอนติตี้ชนิดอ่อนที่ตั้งขึ้นนั้นจะต้องสามารถใช้เป็นชื่อของตารางบนฐานข้อมูลได้เสร็จแล้วให้กดปุ่ม OK เพื่อแสดงการยืนยันกรอบข้อความจะหายไปแต่จะมีกรอบสี่เหลี่ยมผืนผ้าว่างๆ แสดงขึ้นบนพื้นที่ทำงานแทนให้ใช้เมาท์เลื่อนรูปสี่เหลี่ยมผืนผ้า นั้นไปยังตำแหน่งที่ต้องการบนพื้นที่ทำงานโดยจะต้องไม่ไปทับกับวัตถุเดิมที่มีปรากฏอยู่ก่อนแล้วบนพื้นที่ทำงานแล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งรูปสี่เหลี่ยมผืนผ้า นั้นจะถูกวางลงบนพื้นที่ทำงานพร้อมทั้งมีชื่อของเอนติตี้ชนิดอ่อนที่ใส่ลงในกรอบข้อความมาปรากฏอยู่ตรงกลางภายในกรอบสี่เหลี่ยมผืนผ้าที่ซ้อนกันสองรูปนั้นดังแสดงในรูปที่ A9



รูปที่ A8 แสดงกรอบข้อความของเอนติตี้ชนิดอ่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

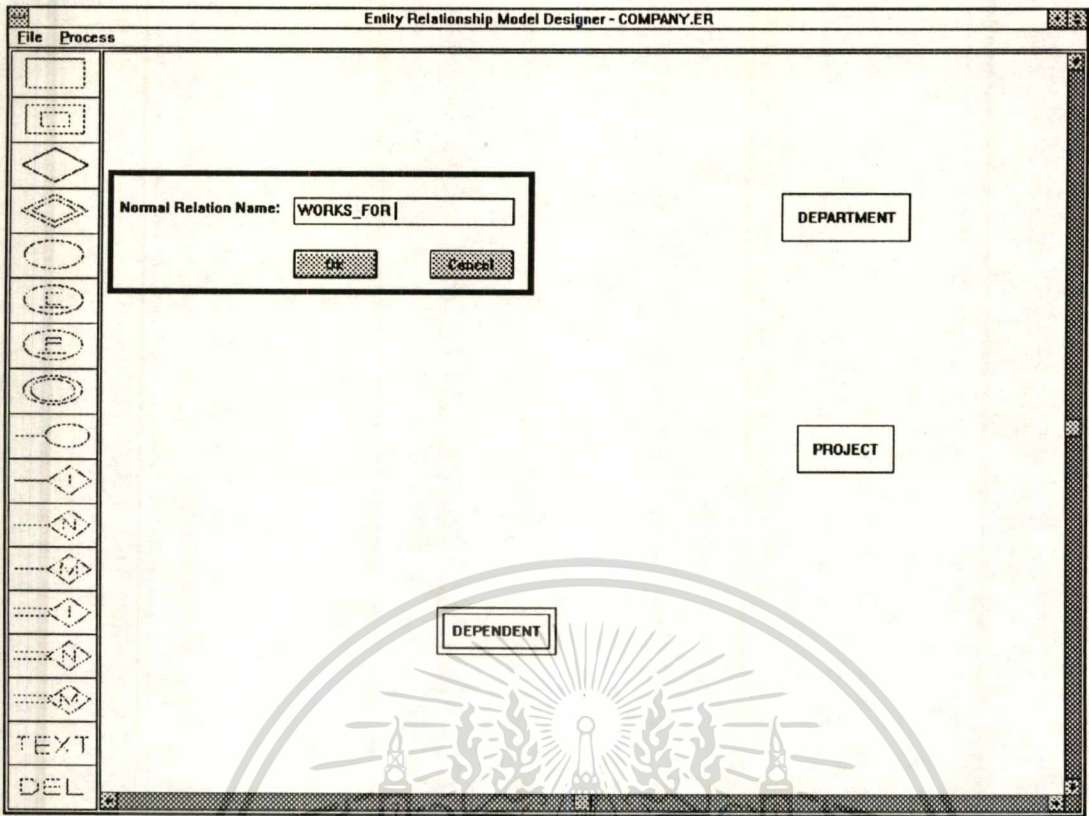


รูปที่ A9 แสดงภาพเอนิติตซ์ชนิดอ้อน DEPENDENTS ที่ถูกวาดขึ้น

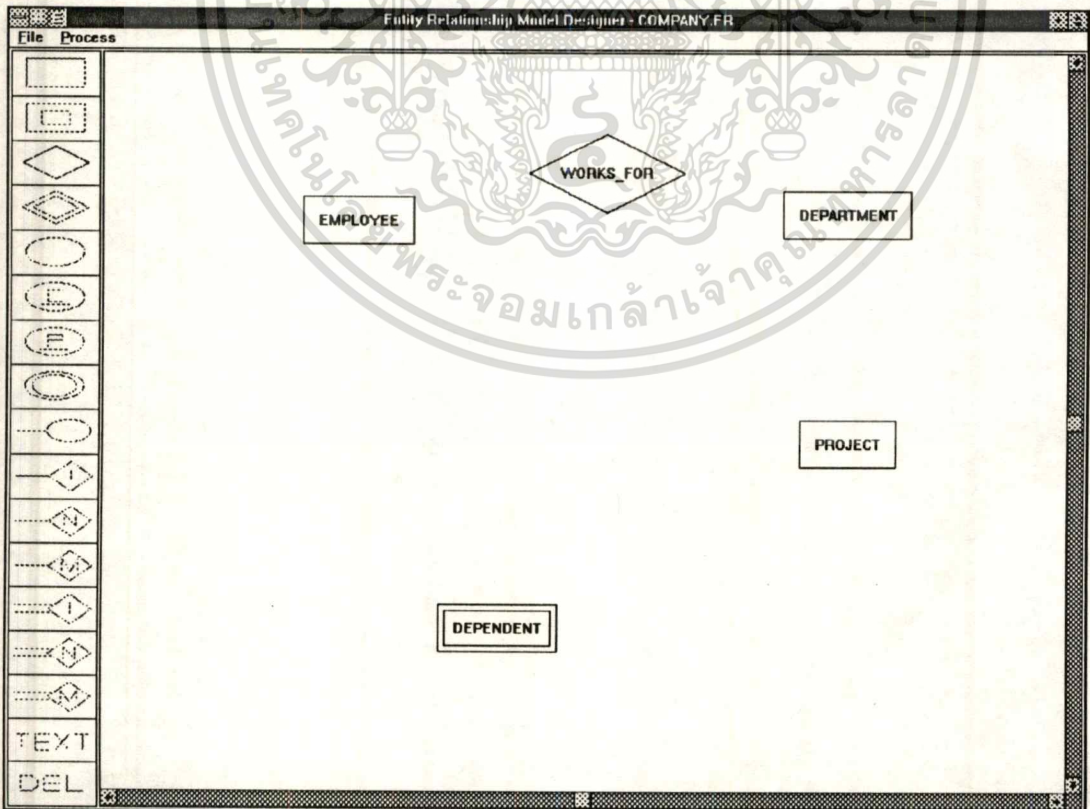
- การวาดความเกี่ยวพันชนิดปกติ

ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของความเกี่ยวพันชนิดปกติซึ่งเป็นรูปสี่เหลี่ยมข้าวหลามตัดบนแถวที่สามของอีอาร์ทูลเพลทเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปมือที่ใช้ชี้เลือกเพลทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว่าสัญลักษณ์ความเกี่ยวพันชนิดปกติจะเปลี่ยนจากสี่เหลี่ยมเป็นสีน้ำเงิน (ถ้าเป็นสีน้ำเงินอยู่แต่เดิมแสดงว่าถูกเลือกไว้ก่อนแล้ว) หลังจากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นตัวอักษร A แล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งจะมีกรอบข้อความเกิดขึ้นบนพื้นที่ทำงานดังแสดงในรูปที่ A10 ให้ใส่ชื่อของความเกี่ยวพันชนิดปกตินั้นตามต้องการโดยที่ชื่อของความเกี่ยวพันชนิดปกติที่ตั้งขึ้นนั้นจะต้องสามารถใช้เป็นชื่อของตารางบนฐานข้อมูลได้เสร็จแล้วให้กดปุ่ม OK เพื่อแสดงการยืนยันกรอบข้อความจะหายไปแต่จะมีกรอบสี่เหลี่ยมข้าวหลามตัดว่างๆ แสดงขึ้นบนพื้นที่ทำงานแทนให้ใช้เมาท์เลื่อนรูปสี่เหลี่ยมข้าวหลามตัดนั้นไปยังตำแหน่งที่ต้องการบนพื้นที่ทำงานโดยจะต้องไม่ไปทับกับวัตถุเดิมที่มีปรากฏอยู่ก่อนแล้วบนพื้นที่ทำงานแล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งรูปสี่เหลี่ยมข้าวหลามตัดนั้นจะถูกวางลงบนพื้นที่ทำงานพร้อมทั้งมีชื่อของความเกี่ยวพันชนิดปกติที่ใส่ลงในกรอบข้อความมาปรากฏอยู่ตรงกลางภายในกรอบสี่เหลี่ยมข้าวหลามตัดนั้นดังแสดงในรูปที่ A11

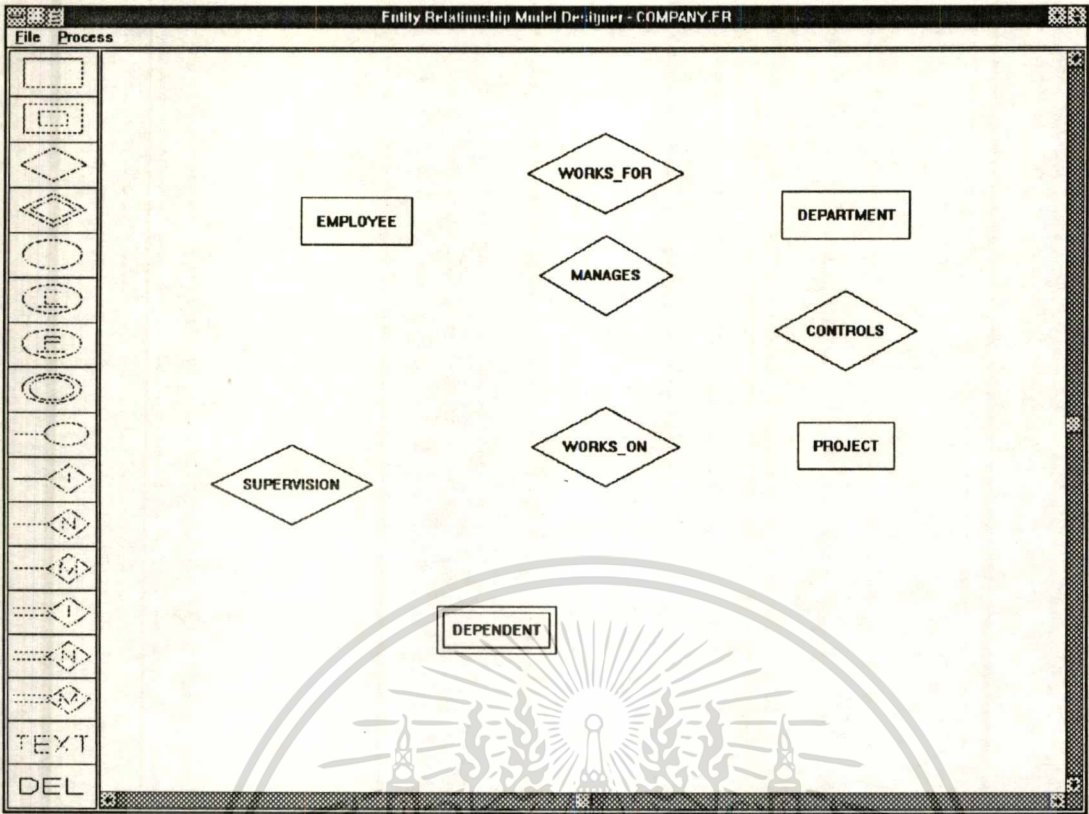
เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ A10 แสดงกรอบข้อความของความเกี่ยวพันชนิดปกติ



เอกสารนี้เป็นเอกสารรูปที่ A11 แสดงภาพความเกี่ยวพันชนิดปกติ WORKS\_FOR ที่ถูกวาดขึ้นนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

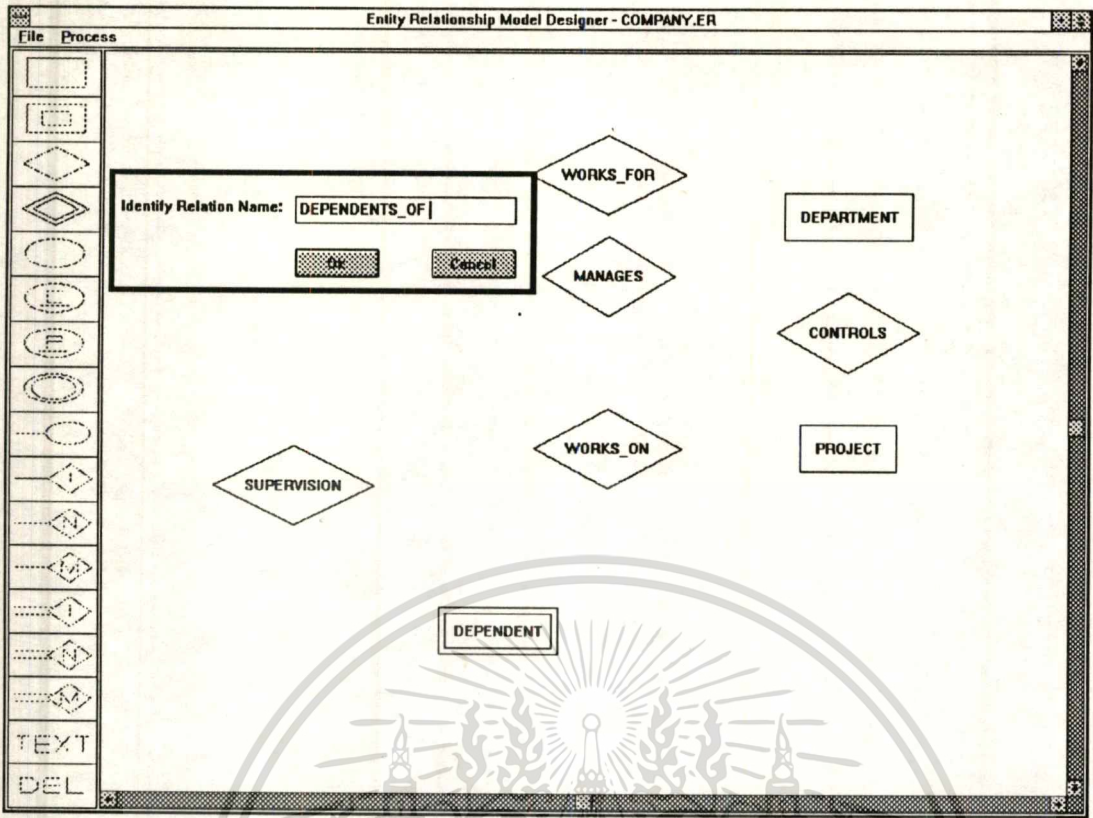


รูปที่ A12 แสดงการวาดความสัมพันธ์ชนิดปกติทั้งหมด

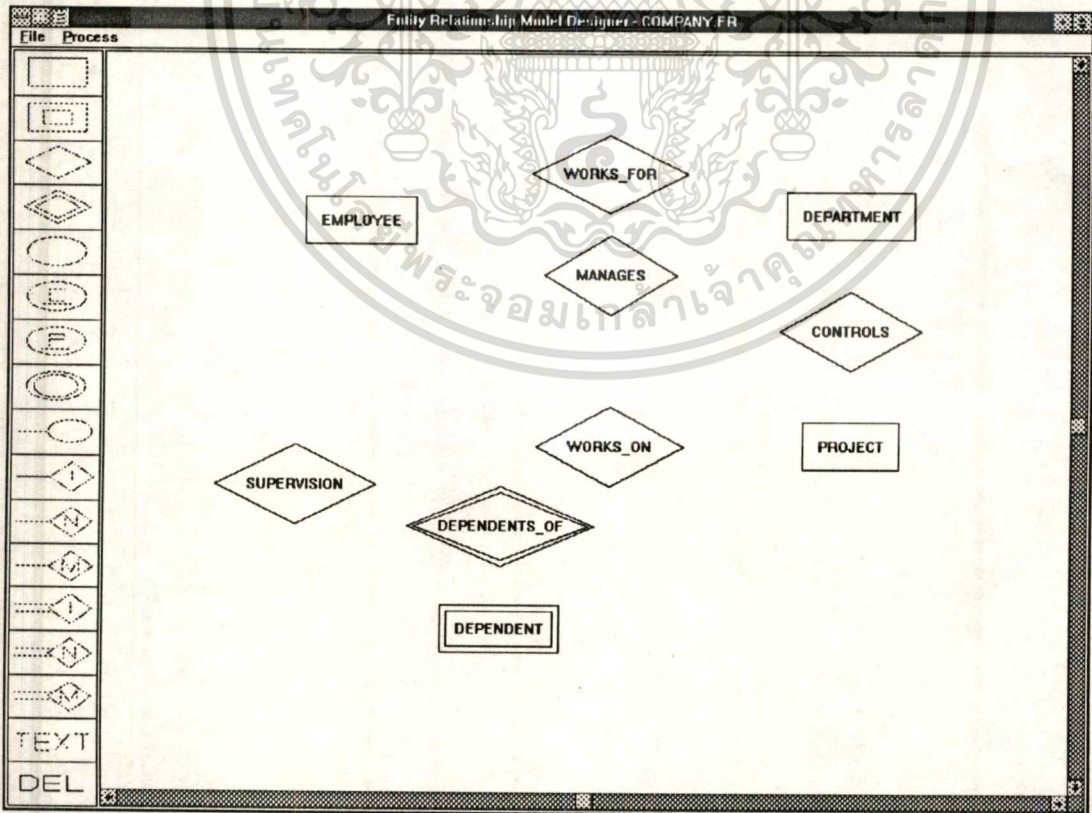
- การวาดความสัมพันธ์ชนิดอ่อน

ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของความสัมพันธ์ชนิดอ่อนซึ่งเป็นรูปสี่เหลี่ยมข้าวหลามตัดซ้อนกันสองรูปบนแถวที่สี่ของอาร์ทูลเพลทเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปมือที่ใช้ชี้เลือกเพลทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว่าสัญลักษณ์ความสัมพันธ์ชนิดอ่อนจะเปลี่ยนจากสี่เหลี่ยมเป็นสี่น้ำเงิน หลังจากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นตัวอักษร A แล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งจะมีกรอบข้อความเกิดขึ้นบนพื้นที่ทำงานดังแสดงในรูปที่ A13 ให้ใส่ชื่อของความสัมพันธ์ชนิดอ่อนนั้นตามต้องการโดยที่ชื่อของความสัมพันธ์ชนิดอ่อนที่ตั้งขึ้นนั้นจะต้องสามารถใช้เป็นชื่อของตารางบนฐานข้อมูลได้เสร็จแล้วให้กดปุ่ม OK เพื่อแสดงการยืนยันกรอบข้อความจะหายไปแต่จะมีกรอบสี่เหลี่ยมข้าวหลามตัดวางๆ แสดงขึ้นบนพื้นที่ทำงานแทนให้ใช้เมาท์เลื่อนรูปสี่เหลี่ยมข้าวหลามตัดนั้นไปยังตำแหน่งที่ต้องการบนพื้นที่ทำงานโดยจะต้องไม่ไปทับกับวัตถุเดิมที่มีปรากฏอยู่ก่อนแล้วบนพื้นที่ทำงานแล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งรูปสี่เหลี่ยมข้าวหลามตัดนั้นจะถูกวางลงบนพื้นที่ทำงานพร้อมทั้งมีชื่อของความสัมพันธ์ชนิดอ่อนที่ใส่ลงในกรอบข้อความมาปรากฏอยู่ตรงกลางภายในกรอบสี่เหลี่ยมข้าวหลามตัดที่ซ้อนกันสองรูปนั้นดังแสดงในรูปที่ A14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



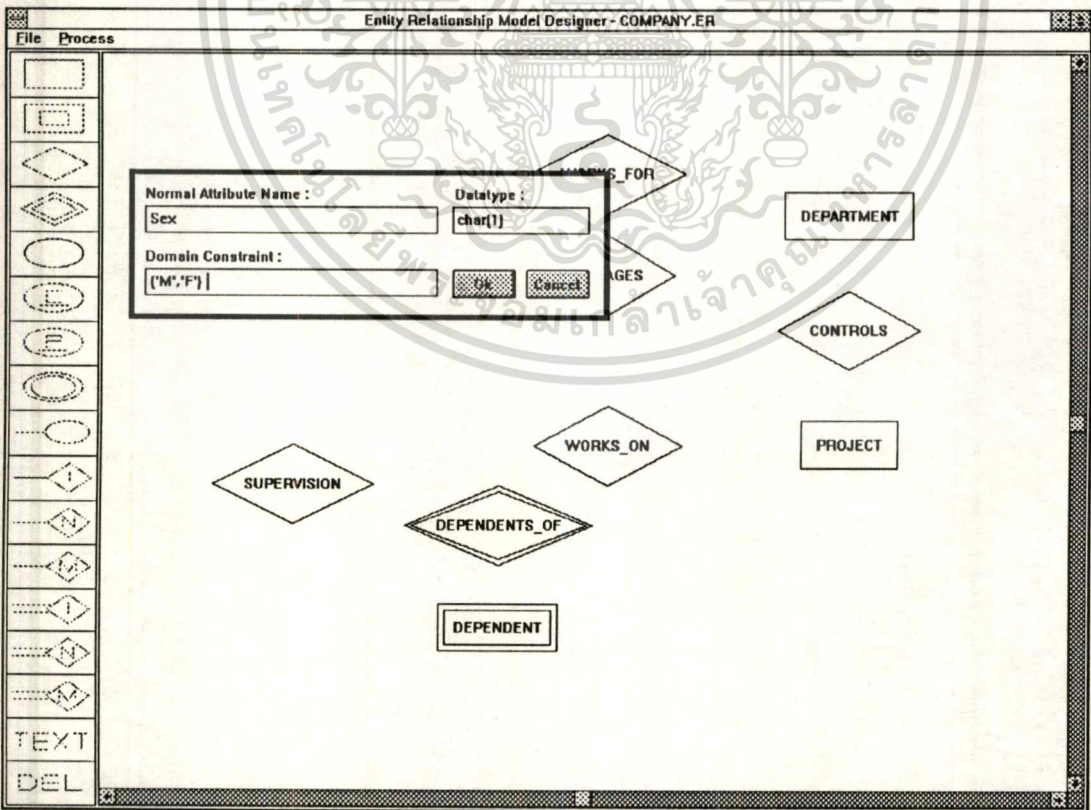
รูปที่ A13 แสดงกรอบข้อความของความเกี่ยวข้องชนิดอื่น



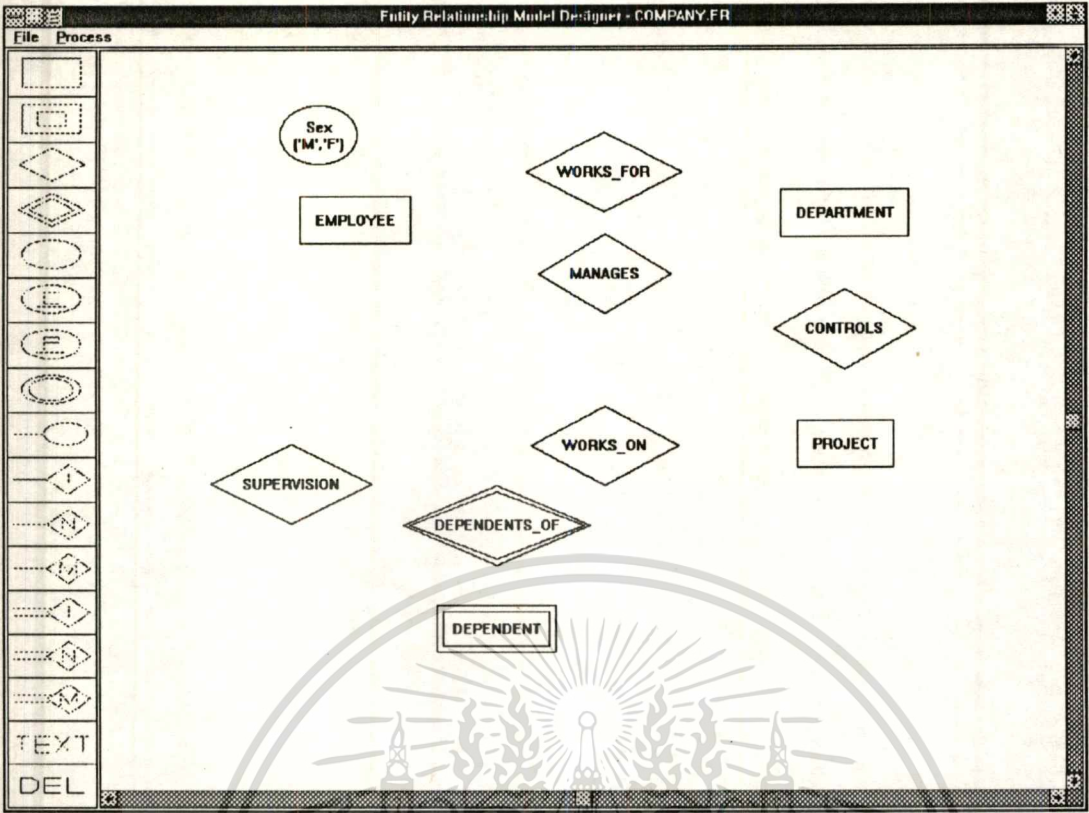
เอกสารนี้เป็นเอกสารที่แสดงภาพของความเกี่ยวข้องชนิดอื่น DEPENDENTS\_OF ที่ถูกวาดขึ้นโดยใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การวาดแอททริบิวต์ชนิดปกติ

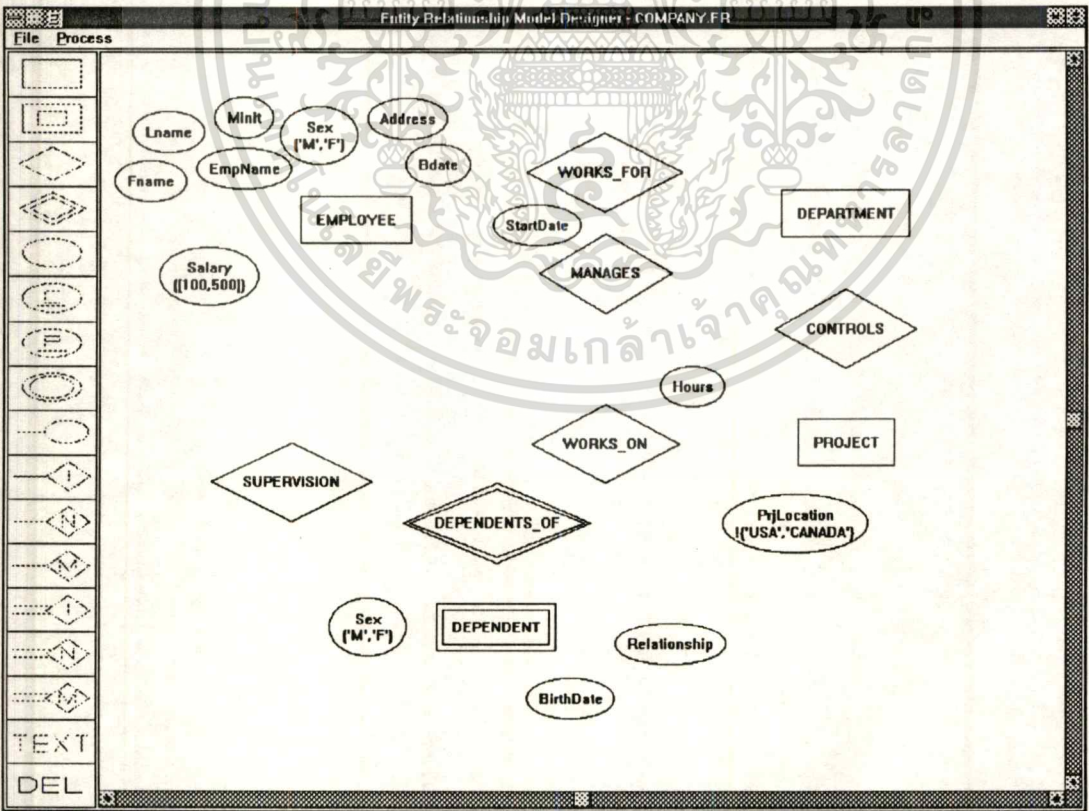
ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของแอททริบิวต์ชนิดปกติซึ่งเป็นรูปวงรีบนแถวที่ห้าของอ็อบเจกต์เพลทเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปมือที่ใช้ชี้เลือกเพลทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว่าสัญลักษณ์แอททริบิวต์ชนิดปกติจะเปลี่ยนจากสีเขียวเป็นสีน้ำเงิน หลังจากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นตัวอักษร A แล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งจะมีกรอบข้อความเกิดขึ้นบนพื้นที่ทำงานดังแสดงในรูปที่ A15 ให้ใส่ชื่อของแอททริบิวต์ชนิดปกติ ลักษณะข้อมูล (Datatype) และตัวควบคุม โดยที่ชื่อของแอททริบิวต์ชนิดปกติที่ตั้งขึ้นนั้นจะต้องสามารถใช้เป็นชื่อของแอททริบิวต์บนตารางฐานข้อมูลได้และตัวควบคุมไม่จำเป็นต้องใส่ก็ได้แต่ถ้าใส่ก็จะต้องใส่ให้ถูกต้องตามโครงสร้างของตัวควบคุมที่ได้กล่าวมาแล้วในบทที่ 3 เสร็จแล้วให้กดปุ่ม OK เพื่อแสดงการยืนยันกรอบข้อความจะหายไปแต่จะมีกรอบวงรีว่างๆ แสดงขึ้นบนพื้นที่ทำงานแทนให้ใช้เมาท์เลื่อนรูปวงรีนั้นไปยังตำแหน่งที่ต้องการบนพื้นที่ทำงานโดยจะต้องไม่ไปทับกับวัตถุเดิมที่มีปรากฏอยู่ก่อนแล้วบนพื้นที่ทำงานแล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งรูปวงรีนั้นจะถูกวางลงบนพื้นที่ทำงานพร้อมทั้งมีชื่อของแอททริบิวต์ชนิดปกติและตัวควบคุมที่ใส่ลงในกรอบข้อความมาปรากฏอยู่ตรงกลางภายในกรอบวงรีนั้นดังแสดงในรูปที่ A16



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของโรงเรียนเพื่อการศึกษาเท่านั้น เมื่อผู้ญาติให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ A16 แสดงภาพแอททริบิวต์ชนิดปกติ Sex ตัวควบคุม ('M','F') ที่ถูกวาดขึ้น

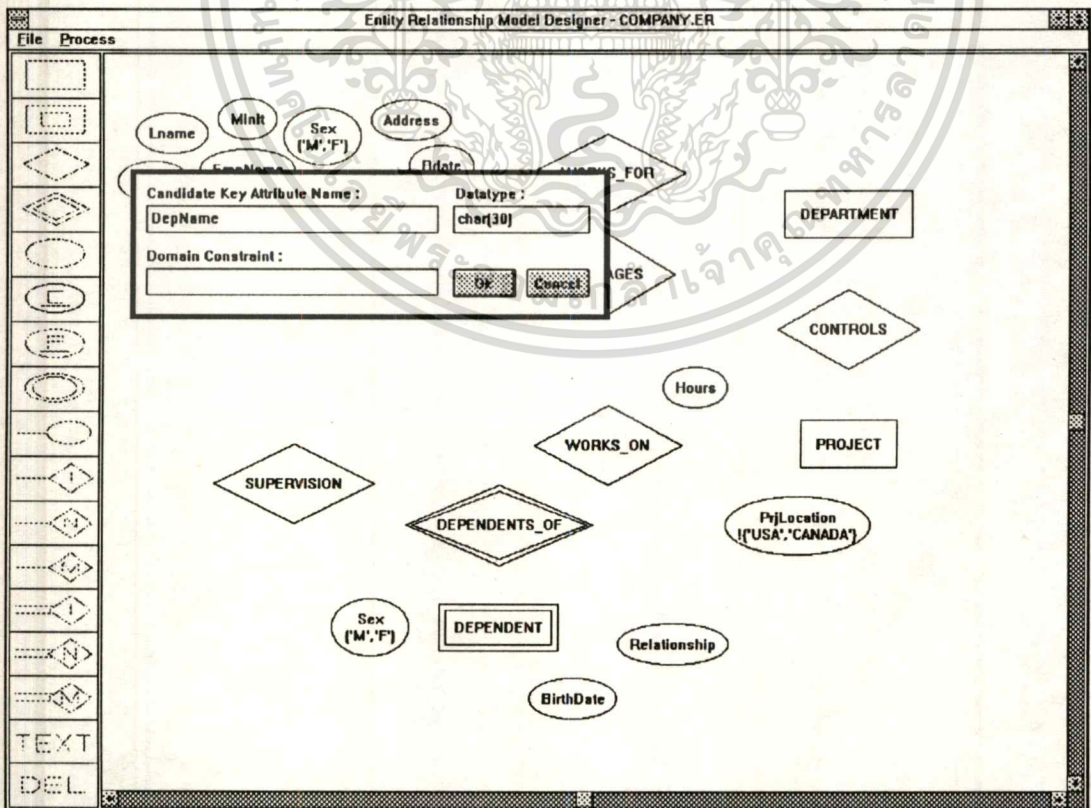


รูปที่ A17 แสดงการวาดแอททริบิวต์ชนิดปกติทั้งหมด

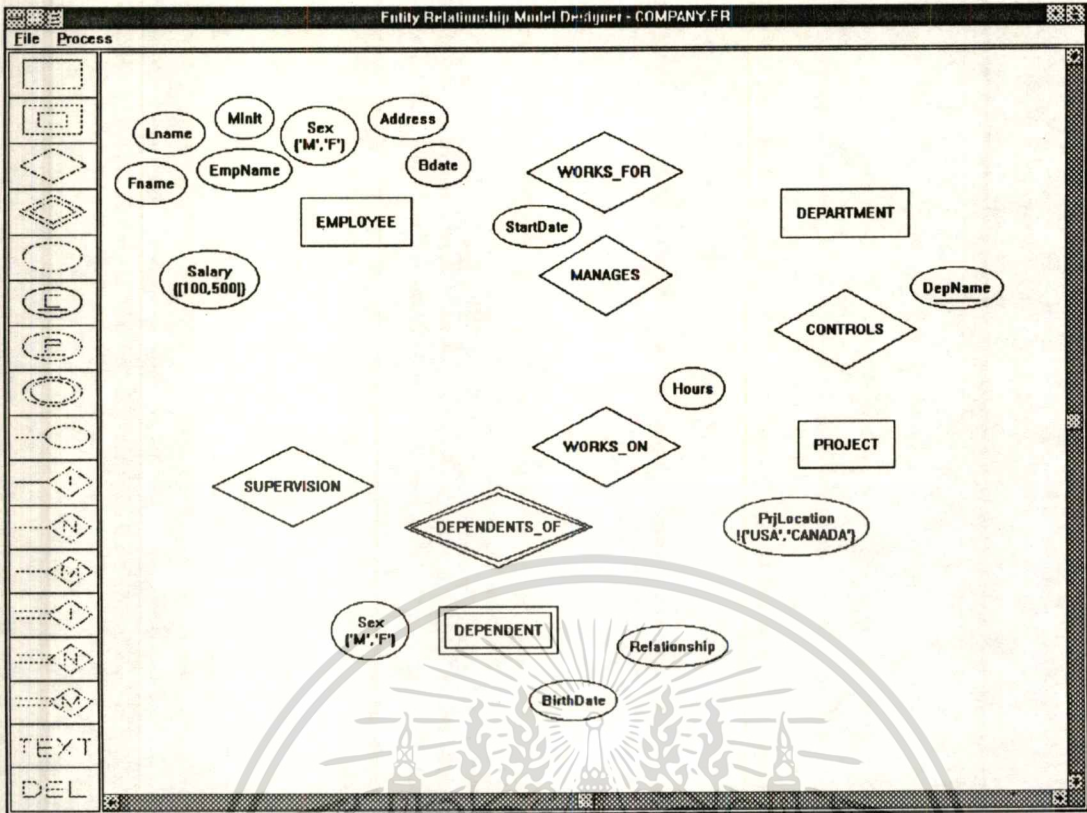
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีก้นำไปใช้

- การวาดแอททริบิวต์ชนิดกุญแจคู่แข่ง

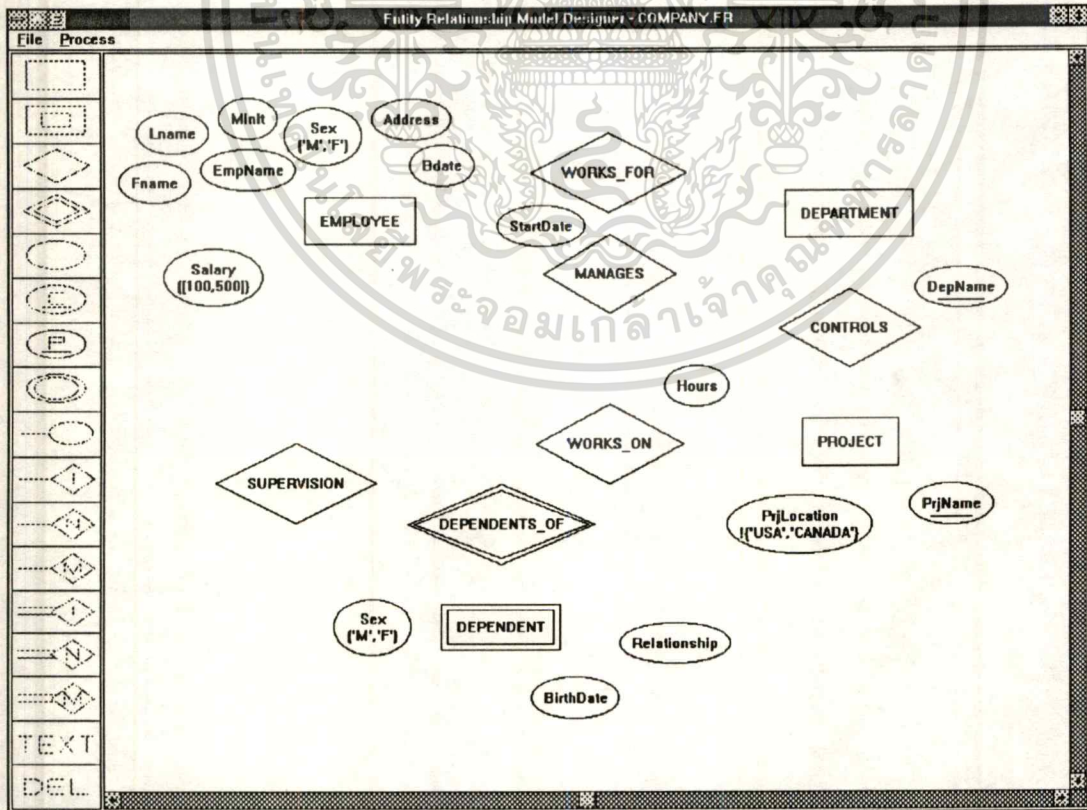
ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของแอททริบิวต์ชนิดกุญแจคู่แข่ง ซึ่งเป็นรูปวงรีมีตัวอักษร C อยู่ภายในบนแถวที่หกของอีอาร์ทูลเพลทเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปมือที่ใช้ชี้เลือกเพลทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว่าสัญลักษณ์แอททริบิวต์ชนิดกุญแจคู่แข่งจะเปลี่ยนจากสีเขียวเป็นสีน้ำเงิน หลังจากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นตัวอักษร A แล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งจะมีกรอบข้อความเกิดขึ้นบนพื้นที่ทำงานดังแสดงในรูปที่ A18 ให้ใส่ชื่อของแอททริบิวต์ชนิดกุญแจคู่แข่ง ลักษณะข้อมูล (Datatype) และตัวควบคุม โดยที่ชื่อของแอททริบิวต์ชนิดกุญแจคู่แข่งที่ตั้งขึ้นนั้นจะต้องสามารถใช้เป็นชื่อของแอททริบิวต์บนตารางฐานข้อมูลได้และตัวควบคุมไม่จำเป็นต้องใส่ก็ได้แต่ถ้าใส่ก็ต้องใส่ให้ถูกต้องตามโครงสร้างของตัวควบคุมที่ได้กล่าวมาแล้วในบทที่ 3 เสร็จแล้วให้กดปุ่ม OK เพื่อแสดงการยืนยันกรอบข้อความจะหายไปแต่จะมีกรอบวงรีว่างๆ แสดงขึ้นบนพื้นที่ทำงานแทนให้ใช้เมาท์เลื่อนรูปวงรีนั้นไปยังตำแหน่งที่ต้องการบนพื้นที่ทำงานโดยจะต้องไม่ไปทับกับวัตถุเดิมที่มีปรากฏอยู่ก่อนแล้วบนพื้นที่ทำงานแล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งรูปวงรีนั้นจะถูกวางลงบนพื้นที่ทำงานพร้อมทั้งมีชื่อของแอททริบิวต์ชนิดกุญแจคู่แข่งขีดเส้นใต้และตัวควบคุมที่ใส่ลงในกรอบข้อความมาปรากฏอยู่ตรงกลางภายในกรอบวงรีนั้นดังแสดงในรูปที่ A19



รูปที่ A18 แสดงกรอบข้อความของแอททริบิวต์ชนิดกุญแจคู่แข่ง  
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการวิจัยเท่านั้น เมื่อผู้ผู้จัดทำนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ A19 แสดงภาพแอททริบิวต์ชนิดทวิภาคของ DepName ที่ถูกวาดขึ้น

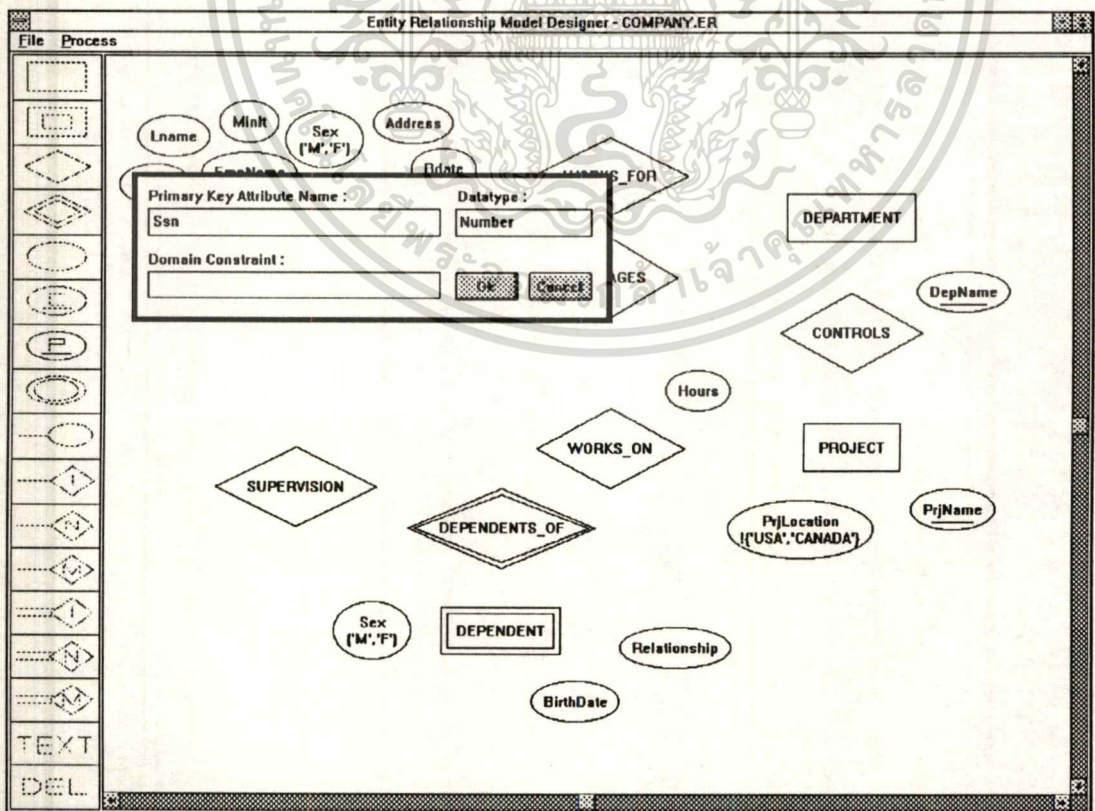


รูปที่ A20 แสดงการวาดแอททริบิวต์ชนิดทวิภาคของทั้งหมด

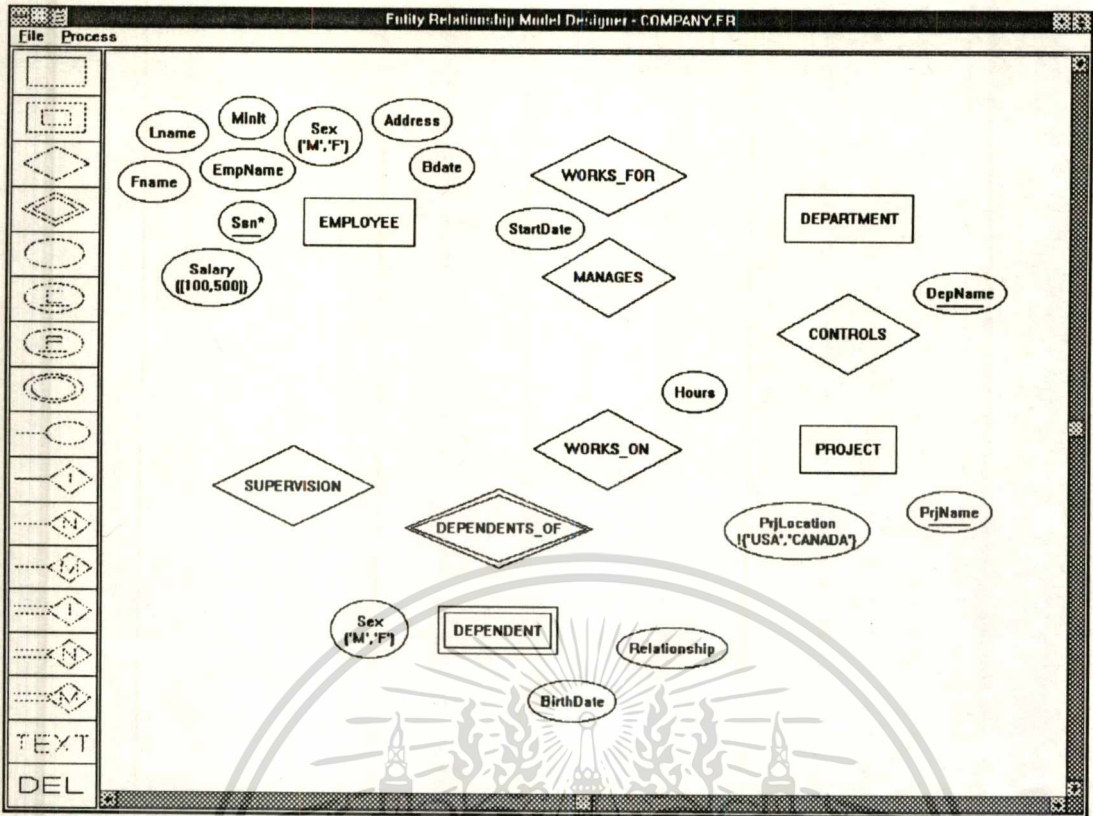
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การวาดแอททริบิวต์ชนิดกุญแจหลัก

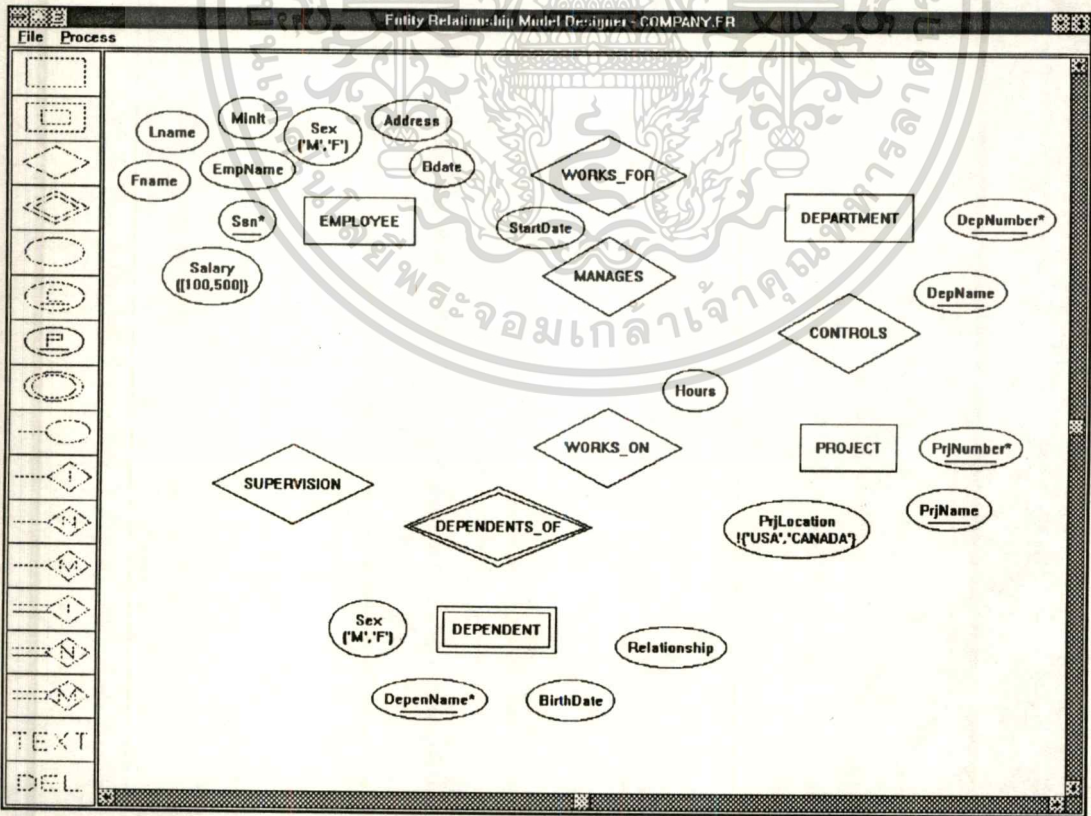
ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของแอททริบิวต์ชนิดกุญแจหลัก ซึ่งเป็นรูปวงรีมีตัวอักษร P อยู่ภายในบนแถวที่เจ็ดของอ็อบเจกต์เพลทเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปมือที่ใช้ชี้เลือกเพลทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว่าสัญลักษณ์แอททริบิวต์ชนิดกุญแจหลักจะเปลี่ยนจากสี่เหลี่ยมเป็นสี่น้ำเงิน หลังจากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นตัวอักษร A แล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งจะมีกรอบข้อความเกิดขึ้นบนพื้นที่ทำงานดังแสดงในรูปที่ A21 ให้ใส่ชื่อของแอททริบิวต์ชนิดกุญแจหลัก ลักษณะข้อมูล (Datatype) และตัวควบคุม โดยที่ชื่อของแอททริบิวต์ชนิดกุญแจหลักที่ตั้งขึ้นนั้นจะต้องสามารถใช้เป็นชื่อของแอททริบิวต์บนตารางฐานข้อมูลได้และตัวควบคุมไม่จำเป็นต้องใส่ก็ได้แต่ถ้าใส่ก็ต้องใส่ให้ถูกต้องตามโครงสร้างของตัวควบคุมที่ได้กล่าวมาแล้วในบทที่ 3 เสร็จแล้วให้กดปุ่ม OK เพื่อแสดงการยืนยันกรอบข้อความจะหายไปแต่จะมีกรอบว่างๆ แสดงขึ้นบนพื้นที่ทำงานแทนให้ใช้เมาท์เลื่อนรูปวงรีนั้นไปยังตำแหน่งที่ต้องการบนพื้นที่ทำงานโดยจะต้องไม่ไปทับกับวัตถุเดิมที่มีปรากฏอยู่ก่อนแล้วบนพื้นที่ทำงานแล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งรูปวงรีนั้นจะถูกวางลงบนพื้นที่ทำงานพร้อมทั้งมีชื่อของแอททริบิวต์ชนิดกุญแจหลักตามด้วยตัวอักษรดอกจันทร์ \* พร้อมขีดเส้นใต้และตัวควบคุมที่ใส่ลงในกรอบข้อความปรากฏอยู่ตรงกลางภายในกรอบวงรีดังแสดงในรูปที่ A22



เอกสารนี้เป็นเอกสารที่รูปที่ A21 แสดงกรอบข้อความของแอททริบิวต์ชนิดกุญแจหลักให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ A22 แสดงภาพแอททริบิวต์ชนิดทูลงแหล่ง Ssn ที่ถูกวาดขึ้น

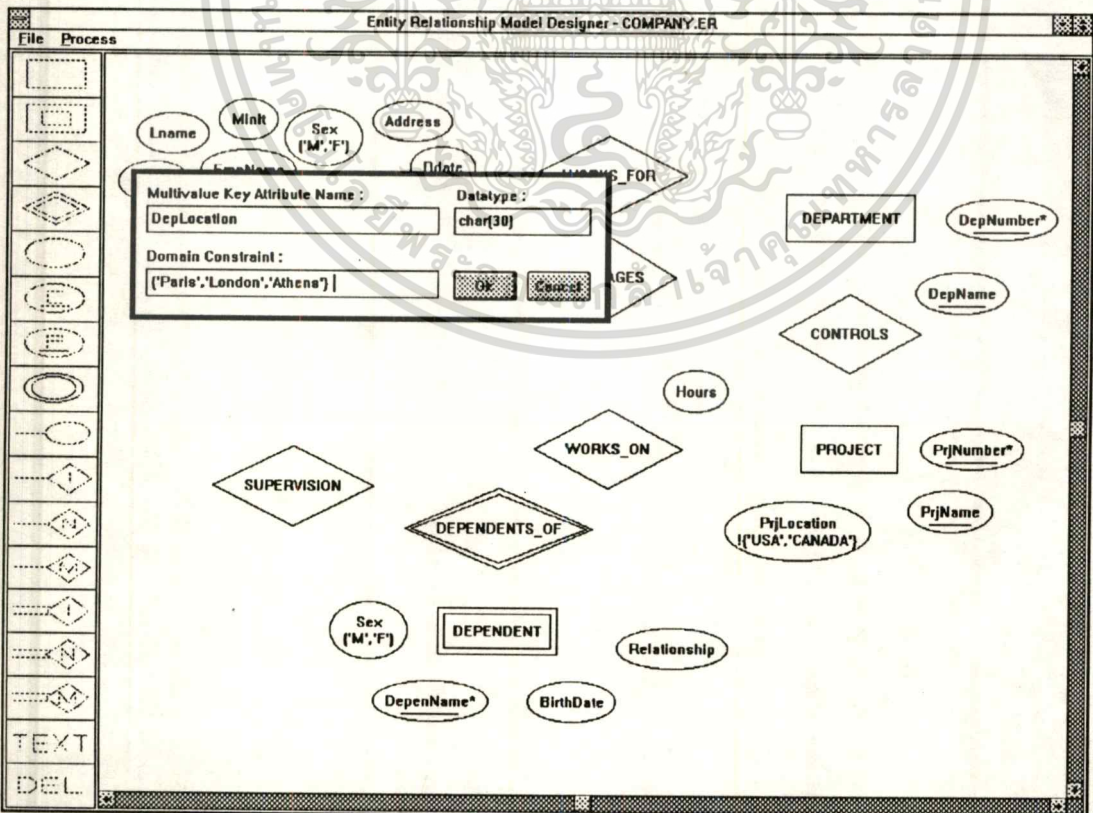


รูปที่ A23 แสดงการวาดแอททริบิวต์ชนิดทูลงแหล่งทั้งหมด

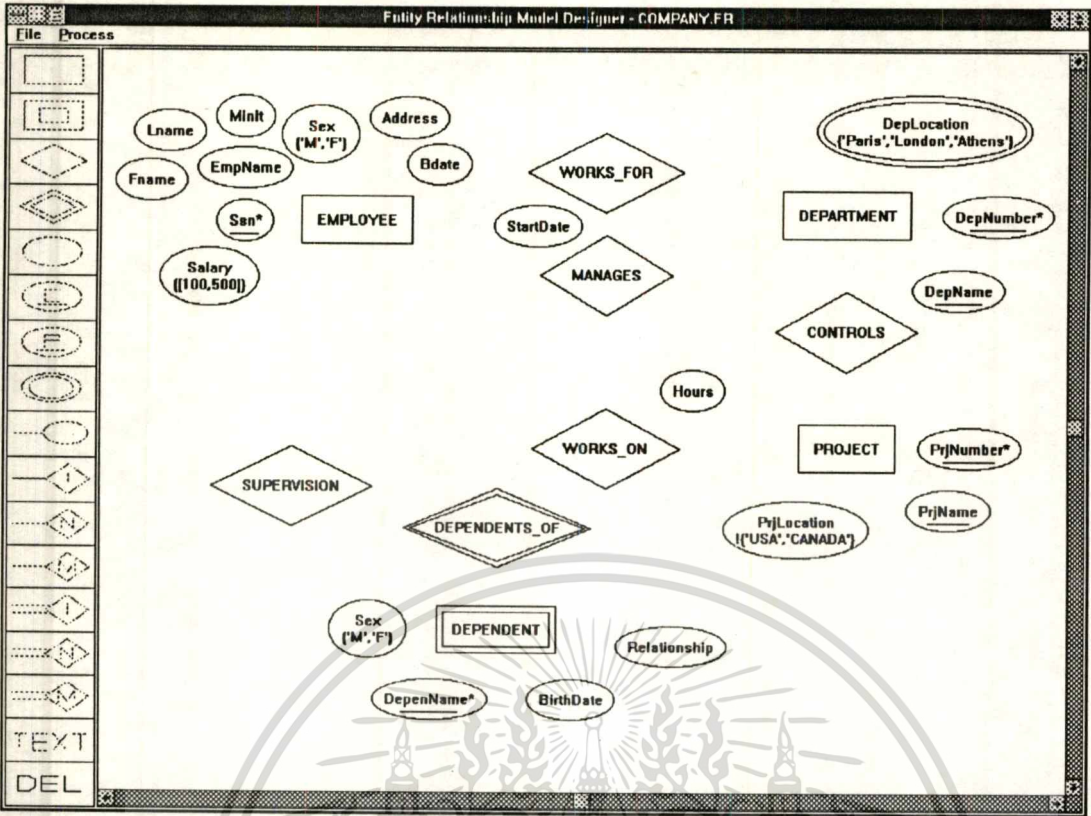
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การวาดแอททริบิวต์ชนิดหลายค่า

ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของแอททริบิวต์ชนิดหลายค่าซึ่งเป็นรูปวงรีซ้อนกันสองวงบนแถวที่แปดของอีอาร์ทูลเพลทเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปมือที่ใช้ชี้เลือกเพลทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว่าสัญลักษณ์แอททริบิวต์ชนิดหลายค่าจะเปลี่ยนจากสี่เหลี่ยมเป็นสี่น้ำเงิน หลังจากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นตัวอักษร A แล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งจะมีกรอบข้อความเกิดขึ้นบนพื้นที่ทำงานดังแสดงในรูปที่ A24 ให้ใส่ชื่อของแอททริบิวต์ชนิดหลายค่า ลักษณะข้อมูล (Datatype) และตัวควบคุม โดยที่ชื่อของแอททริบิวต์ชนิดหลายค่าที่ตั้งขึ้นนั้นจะต้องสามารถใช้เป็นชื่อของตารางและแอททริบิวต์บนตารางฐานข้อมูลได้และตัวควบคุมไม่จำเป็นต้องใส่ก็ได้แต่ถ้าใส่ก็ต้องใส่ให้ถูกต้องตามโครงสร้างของตัวควบคุมที่ได้กล่าวมาแล้วในบทที่ 3 เสร็จแล้วให้กดปุ่ม OK เพื่อแสดงการยืนยันกรอบข้อความจะหายไปแต่จะมีกรอบวงรีว่างๆ แสดงขึ้นบนพื้นที่ทำงานแทนให้ใช้เมาท์เลื่อนรูปวงรีนั้นไปยังตำแหน่งที่ต้องการบนพื้นที่ทำงานโดยจะต้องไม่ไปทับกับวัตถุเดิมที่มีปรากฏอยู่ก่อนแล้วบนพื้นที่ทำงานแล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งรูปวงรีนั้นจะถูกวางลงบนพื้นที่ทำงานพร้อมทั้งมีชื่อของแอททริบิวต์ชนิดหลายค่าและตัวควบคุมที่ใส่ลงในกรอบข้อความมาปรากฏอยู่ตรงกลางภายในกรอบวงรีที่ซ้อนกันสองวงนั้นดังแสดงในรูปที่ A25



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการขงนสิทธิ์เพื่อการค้าเท่านั้น เมื่อผู้จัดทำให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ A25 แสดงภาพแอททริบิวต์ชนิดหลายค่า Locations ตัวควบคุม ('Paris', 'London', 'Athens') ที่ถูกวาดขึ้น

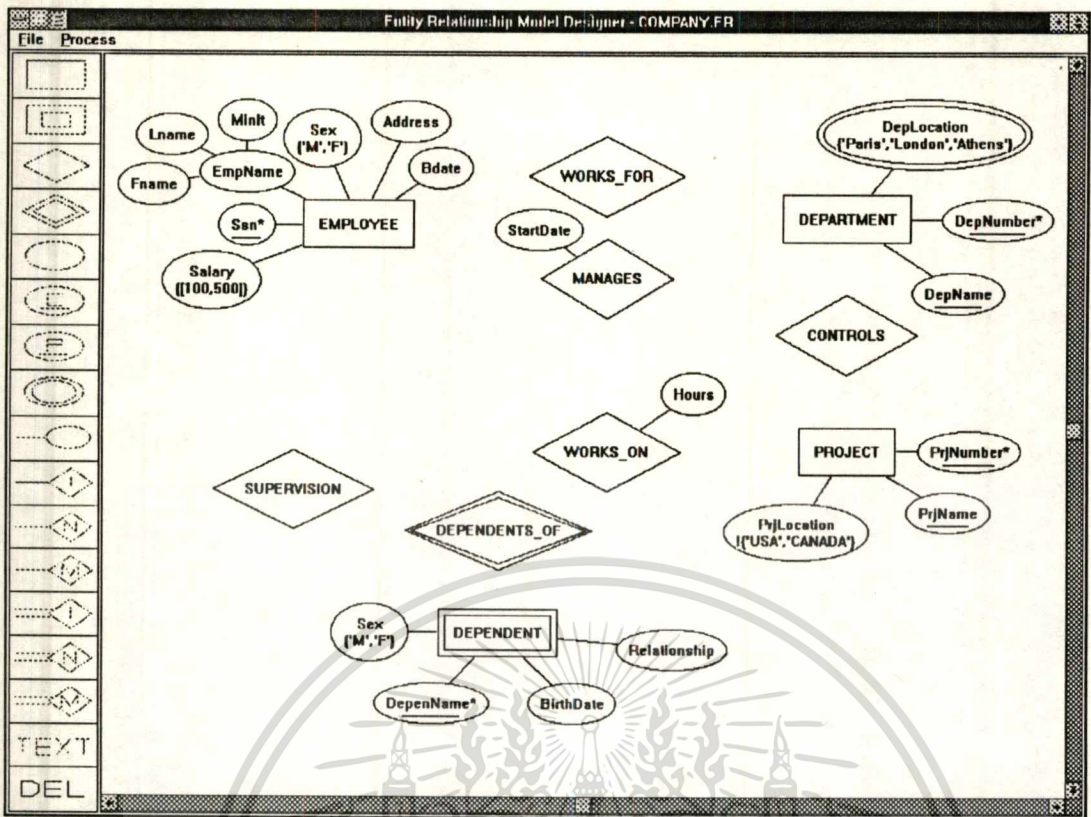
- การเชื่อมต่อแอททริบิวต์

การเชื่อมต่อของแอททริบิวต์มีอยู่ด้วยกัน 5 วิธีคือ

1. การเชื่อมต่อแอททริบิวต์กับแอททริบิวต์
2. การเชื่อมต่อแอททริบิวต์กับเอ็นติตี้ชนิดปกติ
3. การเชื่อมต่อแอททริบิวต์กับเอ็นติตี้ชนิดอ่อน
4. การเชื่อมต่อแอททริบิวต์กับความเกี่ยวพันชนิดปกติ
5. การเชื่อมต่อแอททริบิวต์กับความเกี่ยวพันชนิดอ่อน

ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของการเชื่อมต่อแอททริบิวต์ซึ่งเป็นรูปเส้นตรงแนวนอนเชื่อมกับวงรีบนแถวที่เก้าของอีอาร์ทูลเพลทเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปมือที่ใช้ชี้เลือกเพลทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว่าสัญลักษณ์การเชื่อมต่อแอททริบิวต์จะเปลี่ยนจากสีเขียวเป็นสีน้ำเงิน หลังจากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปดินสอให้กดปุ่มซ้ายของเมาท์บนแอททริบิวต์ที่ต้องการเชื่อมต่อหนึ่งครั้งแล้วเลื่อนเมาท์มายังวัตถุที่ต้องการเชื่อมกับแอททริบิวต์นี้กดปุ่มซ้ายของเมาท์บนวัตถุอีกครั้งหนึ่งจะเกิดเส้นตรงขึ้นหนึ่งเส้นระหว่างแอททริบิวต์กับวัตถุนั้น ดังแสดงในรูปที่ A26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



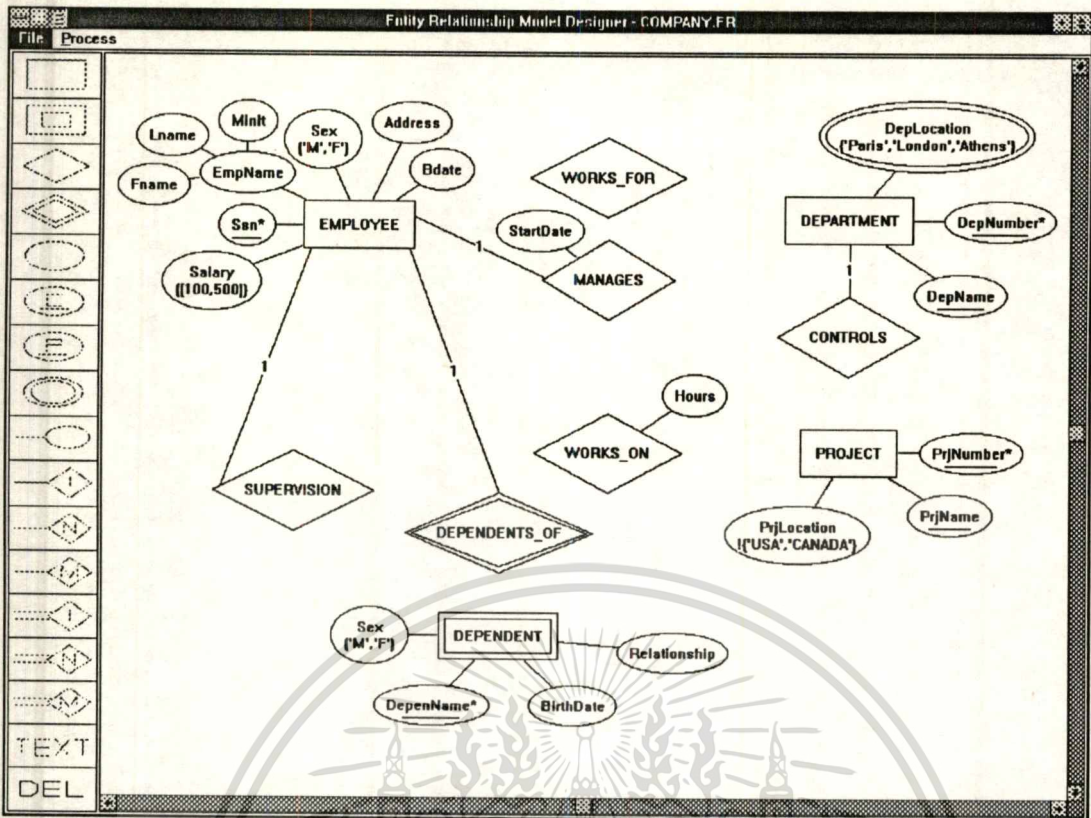
รูปที่ A26 แสดงการเชื่อมต่อของแอททริบิวต์ทั้งหมด

- การเชื่อมต่อความสัมพันธ์ชนิดพาเซี่ยลวัน (1P)

ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของการเชื่อมต่อความสัมพันธ์ชนิดพาเซี่ยลวันซึ่งเป็นรูปเส้นตรงแนวนอนเชื่อมกับรูปสี่เหลี่ยมข้าวหลามตัดที่มีตัวเลข 1 อยู่ภายในรูปสี่เหลี่ยมข้าวหลามตัดนั้นบนแถวที่สิบของอีอาร์ทูลเพลทเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปมือที่ใช้ชี้เลือกเพลทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว่าสัญลักษณ์การเชื่อมต่อความสัมพันธ์ชนิดพาเซี่ยลวันจะเปลี่ยนจากสีเขียวเป็นสีน้ำเงิน (ถ้าเป็นสีน้ำเงินอยู่แต่เดิมแสดงว่าถูกเลือกไว้ก่อนแล้ว) จากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปดินสอ ให้กดปุ่มซ้ายของเมาท์บนความสัมพันธ์ที่ต้องการเชื่อมต่อหนึ่งครั้งแล้วเลื่อนเมาท์มายังเอ็นติตี้ที่ต้องการเชื่อมกับความสัมพันธ์นี้ กดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งหรือจะกดปุ่มซ้ายของเมาท์บนเอ็นติตี้ที่ต้องการเชื่อมต่อก่อนหนึ่งครั้งแล้วเลื่อนเมาท์มายังความสัมพันธ์ที่ต้องการเชื่อมกับเอ็นติตี้กดปุ่มซ้ายของเมาท์ทีหลังก็ได้โดยใดอย่างหนึ่ง จะเกิดเส้นตรงขึ้นหนึ่งเส้นระหว่างความสัมพันธ์กับเอ็นติตี้ที่ต้องการเชื่อมต่อนั้นและมีตัวเลข 1 ปรากฏอยู่กึ่งกลางของเส้นตรงที่เชื่อมต่อนั้นดังแสดงในรูปที่

A27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

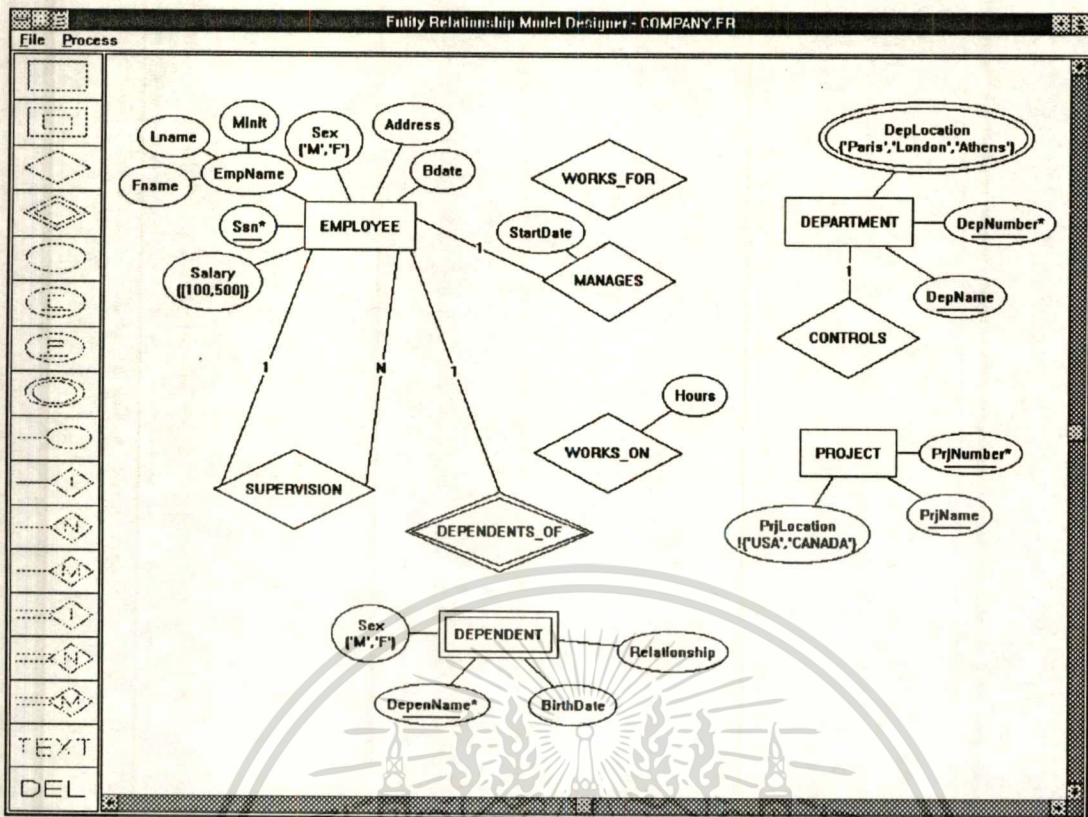


รูปที่ A27 แสดงการเชื่อมต่อความสัมพันธ์ชนิดพาเซิลวัน (1P)

- การเชื่อมต่อความสัมพันธ์ชนิดพาเซิลเอ็น (NP)

ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของการเชื่อมต่อความสัมพันธ์ชนิดพาเซิลเอ็นซึ่งเป็นรูปเส้นตรงแนวนอนเชื่อมกับรูปสี่เหลี่ยมข้าวหลามตัดที่มีตัวอักษร N อยู่ภายในรูปสี่เหลี่ยมข้าวหลามตัดนั้นบนแถวที่สิบเอ็ดของอาร์ทูลเพทเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปมือที่ใช้ชี้เลือกเพทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว่าสัญลักษณ์การเชื่อมต่อความสัมพันธ์ชนิดพาเซิลเอ็นจะเปลี่ยนจากสี่เหลี่ยมเป็นสีน้ำเงิน (ถ้าเป็นสีน้ำเงินอยู่แต่เดิมแสดงว่าถูกเลือกไว้ก่อนแล้ว) จากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปดินสอ ให้กดปุ่มซ้ายของเมาท์บนความสัมพันธ์ที่ต้องการเชื่อมต่อหนึ่งครั้งแล้วเลื่อนเมาท์มายังเอ็นติตี้ที่ต้องการเชื่อมกับความสัมพันธ์นี้ กดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งหรือจะกดปุ่มซ้ายของเมาท์บนเอ็นติตี้ที่ต้องการเชื่อมต่อก่อนหนึ่งครั้งแล้วเลื่อนเมาท์มายังความสัมพันธ์ที่ต้องการเชื่อมกับเอ็นติตี้กดปุ่มซ้ายของเมาท์ทีหลังก็ได้โดยวิธีใดวิธีหนึ่ง จะเกิดเส้นตรงขึ้นหนึ่งเส้นระหว่างความสัมพันธ์กับเอ็นติตี้ที่ต้องการเชื่อมต่อนั้นและมีตัวอักษร N ปรากฏอยู่กึ่งกลางของเส้นตรงที่เชื่อมต่อนั้น ดังแสดงในรูปที่ A28

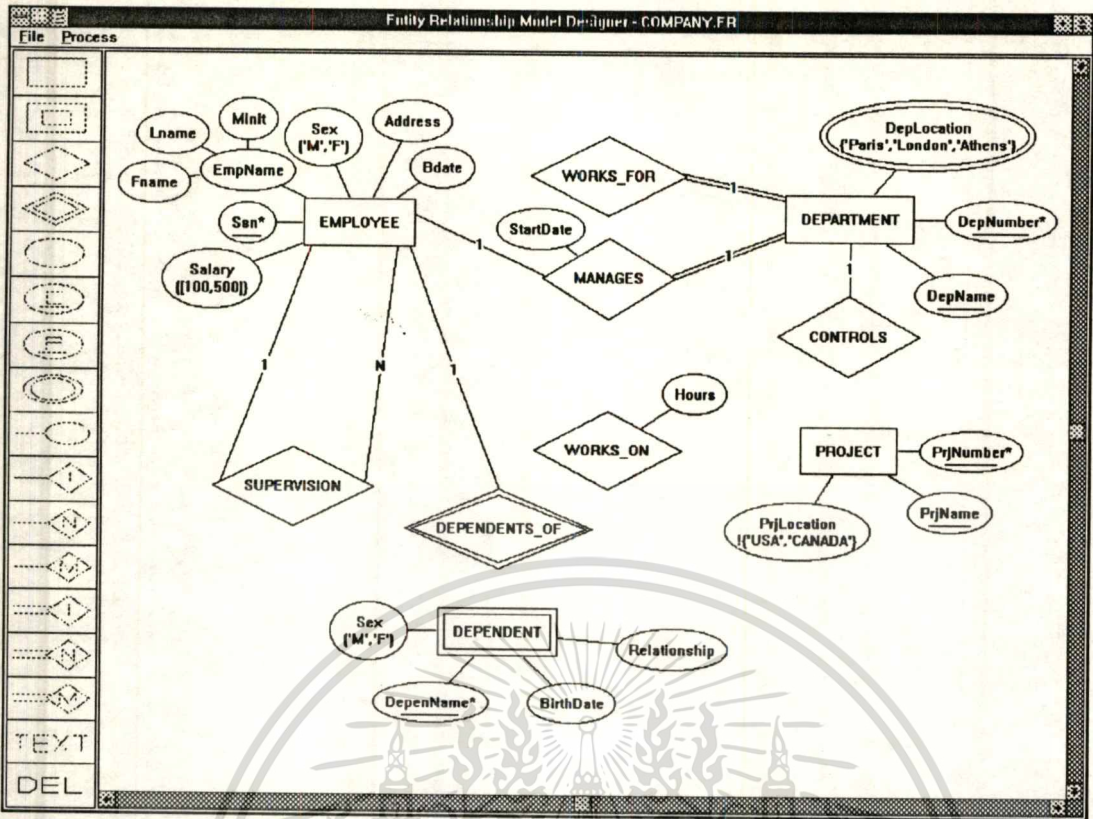
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะวิธีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ A28 แสดงการเชื่อมต่อความสัมพันธ์ชนิดพาเซิลเอ็น (NP)

- การเชื่อมต่อความสัมพันธ์ชนิดโททอลวัน (1T)

ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของการเชื่อมต่อความสัมพันธ์ชนิดโททอลวันซึ่งเป็นรูปเส้นตรงแนวนอนขนานกันสองเส้นเชื่อมกับรูปสี่เหลี่ยมข้าวหลามตัดที่มีตัวเลข 1 อยู่ภายในรูปสี่เหลี่ยมข้าวหลามตัดนั้นบนแถวที่สิบสามของอ็อบเจกต์เพลทเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปมือที่ใช้ชี้เลือกเพลทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว่าสัญลักษณ์การเชื่อมต่อความสัมพันธ์ชนิดโททอลวันจะเปลี่ยนจากสีเขียวเป็นสีน้ำเงิน (ถ้าเป็นสีน้ำเงินอยู่แต่เดิมแสดงว่าถูกเลือกไว้ก่อนแล้ว) จากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปดินสอ ให้กดปุ่มซ้ายของเมาท์บนความสัมพันธ์ที่ต้องการเชื่อมต่อหนึ่งครั้งแล้วเลื่อนเมาท์มายังเอ็นติตี้ที่ต้องการเชื่อมกับความสัมพันธ์นี้กดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งหรือจะกดปุ่มซ้ายของเมาท์บนเอ็นติตี้ที่ต้องการเชื่อมต่อก่อนหนึ่งครั้งแล้วเลื่อนเมาท์มายังความสัมพันธ์ที่ต้องการเชื่อมกับเอ็นติตี้กดปุ่มซ้ายของเมาท์ทีหลังก็ได้โดยบังเอิญจะเกิดเส้นตรงแนวนอนขนานกันสองเส้นระหว่างความสัมพันธ์กับเอ็นติตี้ที่ต้องการเชื่อมต่อนั้นและมีตัวเลข 1 ปรากฏอยู่กึ่งกลางของเส้นตรงที่เชื่อมต่อนั้น ดังแสดงในรูปที่ A29

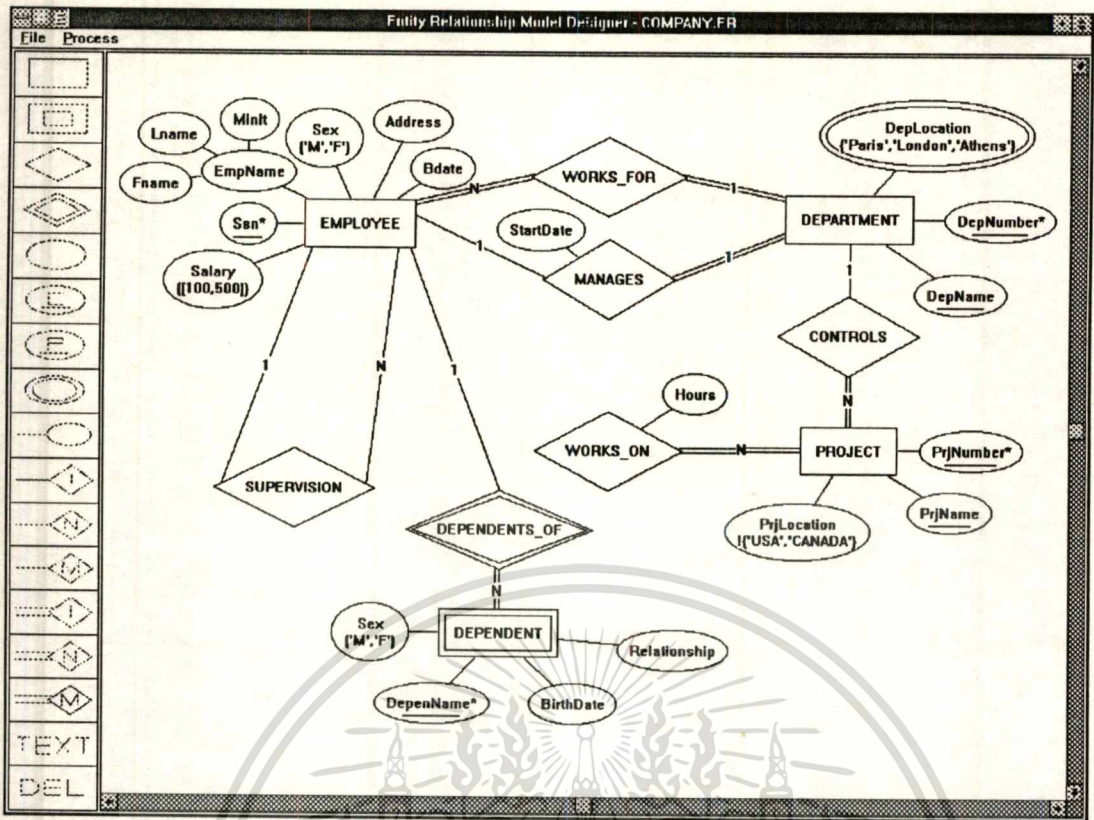


รูปที่ A29 แสดงการเชื่อมต่อความสัมพันธ์โททอลวัน (1T)

- การเชื่อมต่อความสัมพันธ์ชนิดโททอลเอ็น (NT)

ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของการเชื่อมต่อความสัมพันธ์ชนิดโททอลเอ็นซึ่งเป็นรูปเส้นตรงแนวนอนขนานกันสองเส้นเชื่อมกับรูปสี่เหลี่ยมข้าวหลามตัดที่มีตัวอักษร N อยู่ภายในรูปสี่เหลี่ยมข้าวหลามตัดนั้นบนแถวที่สิบสี่ของอ็อบเจกต์เพลทเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปมือที่ใช้ชี้เลือกเพลทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว่าสัญลักษณ์การเชื่อมต่อความสัมพันธ์ชนิดโททอลเอ็นจะเปลี่ยนจากสีเขียวเป็นสีน้ำเงิน (ถ้าเป็นสีน้ำเงินอยู่แต่เดิมแสดงว่าถูกเลือกไว้ก่อนแล้ว) จากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปดินสอ ให้กดปุ่มซ้ายของเมาท์บนความสัมพันธ์ที่ต้องการเชื่อมต่อหนึ่งครั้งแล้วเลื่อนเมาท์มายังเอ็นติตี้ที่ต้องการเชื่อมกับความสัมพันธ์นี้กดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งหรือจะกดปุ่มซ้ายของเมาท์บนเอ็นติตี้ที่ต้องการเชื่อมต่อก่อนหนึ่งครั้งแล้วเลื่อนเมาท์มายังความสัมพันธ์ที่ต้องการเชื่อมกับเอ็นติตี้กดปุ่มซ้ายของเมาท์ทีหลังก็ได้ อย่างไรก็ตามจะเกิดเส้นตรงแนวนอนขนานกันสองเส้นระหว่างความสัมพันธ์กับเอ็นติตี้ที่ต้องการเชื่อมต่อนั้นและมีตัวอักษร N ปรากฏอยู่กึ่งกลางของเส้นตรงที่เชื่อมต่อนั้น ดังแสดงในรูปที่ A30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

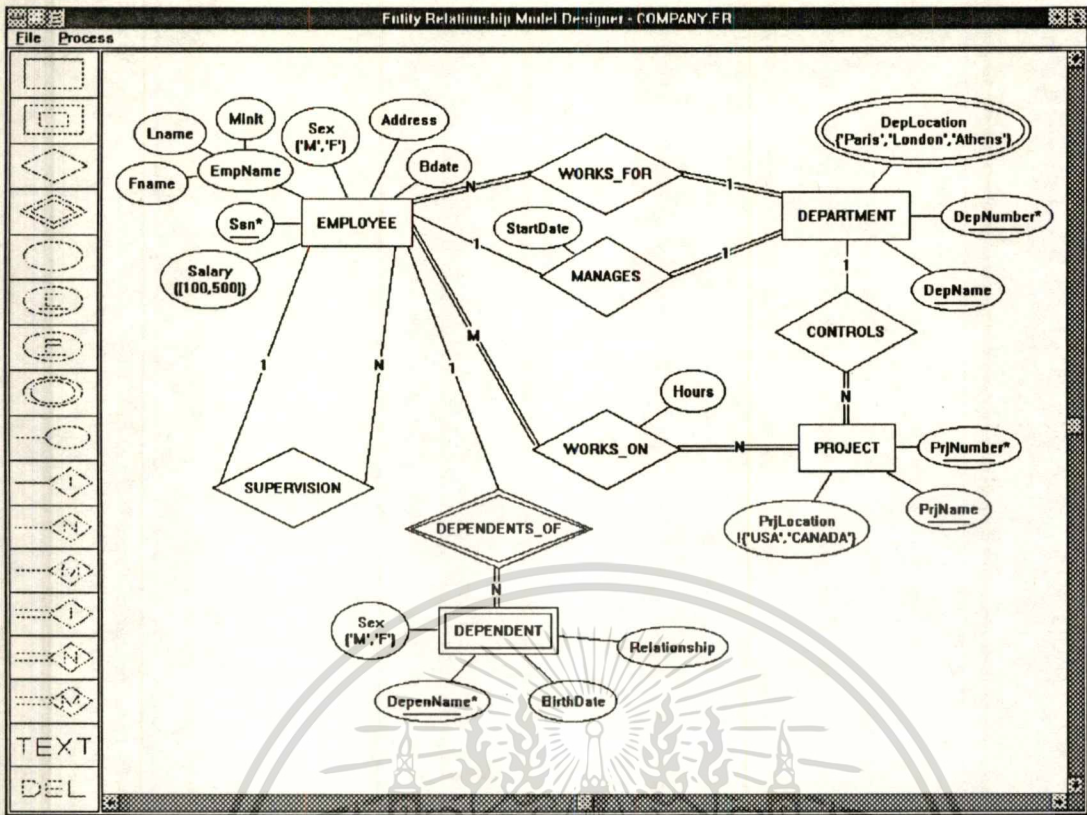


รูปที่ A30 แสดงการเชื่อมต่อความเกี่ยวพันโททอลเอ็น (NT)

- การเชื่อมต่อความเกี่ยวพันชนิดโททอลเอ็ม (MT)

ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของการเชื่อมต่อความเกี่ยวพันชนิดโททอลเอ็มซึ่งเป็นรูปเส้นตรงแนวนอนขนานกันสองเส้นเชื่อมกับรูปสี่เหลี่ยมข้าวหลามตัดที่มีตัวอักษร M อยู่ภายในรูปสี่เหลี่ยมข้าวหลามตัดนั้นบนแถวที่สิบห้าของอีอาร์ทูลเพทเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปมือที่ใช้ชี้เลือกเพทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว่าสัญลักษณ์การเชื่อมต่อความเกี่ยวพันชนิดโททอลเอ็มจะเปลี่ยนจากสีเขียวเป็นสีน้ำเงิน (ถ้าเป็นสีน้ำเงินอยู่แต่เดิมแสดงว่าถูกเลือกไว้ก่อนแล้ว) จากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปดินสอ ให้กดปุ่มซ้ายของเมาท์บนความเกี่ยวพันที่ต้องการเชื่อมต่อหนึ่งครั้งแล้วเลื่อนเมาท์มายังเอ็นติตี้ที่ต้องการเชื่อมกับความเกี่ยวพันนี้กดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งหรือจะกดปุ่มซ้ายของเมาท์บนเอ็นติตี้ที่ต้องการเชื่อมต่อก่อนหนึ่งครั้งแล้วเลื่อนเมาท์มายังความเกี่ยวพันที่ต้องการเชื่อมกับเอ็นติตี้กดปุ่มซ้ายของเมาท์ทีหลังก็ได้โดยวิธีใดวิธีหนึ่ง จะเกิดเส้นตรงแนวนอนขนานกันสองเส้นระหว่างความเกี่ยวพันกับเอ็นติตี้ที่ต้องการเชื่อมต่อนั้นและมีตัวอักษร M ปรากฏอยู่กึ่งกลางของเส้นตรงที่เชื่อมต่อนั้น ดังแสดงในรูปที่ A31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

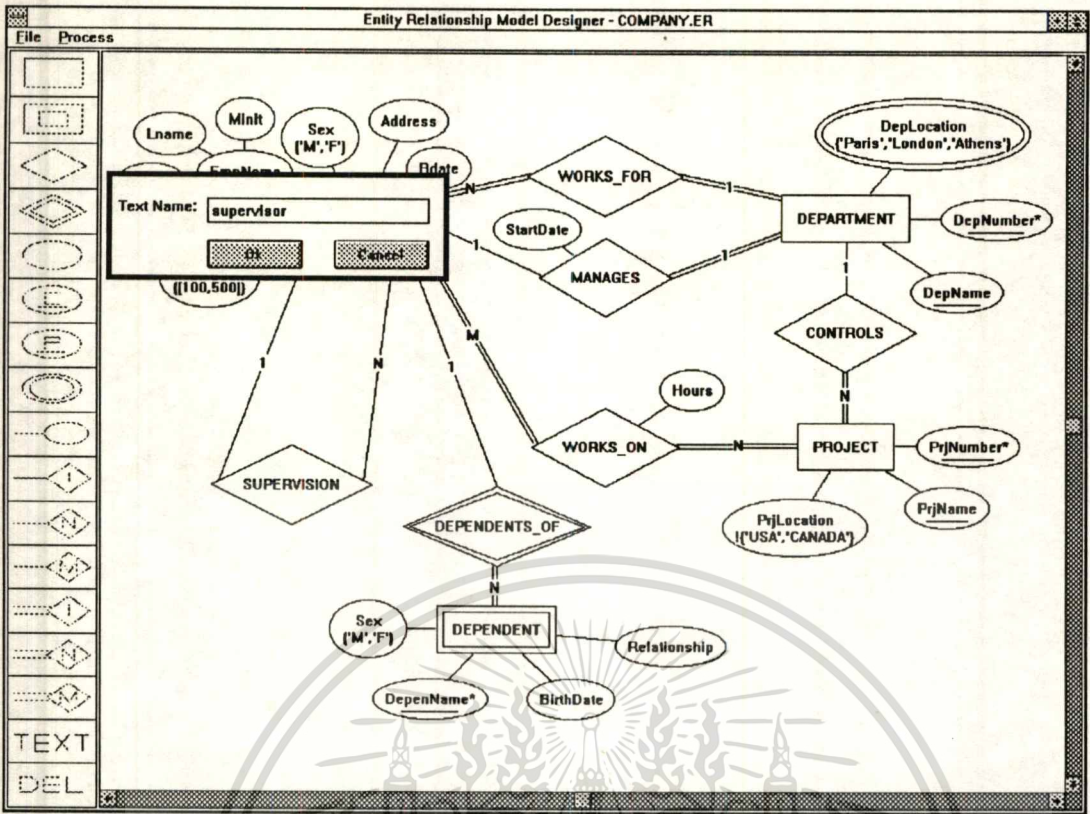


รูปที่ A31 แสดงการเชื่อมต่อความสัมพันธ์โททอลเอ็ม (MT)

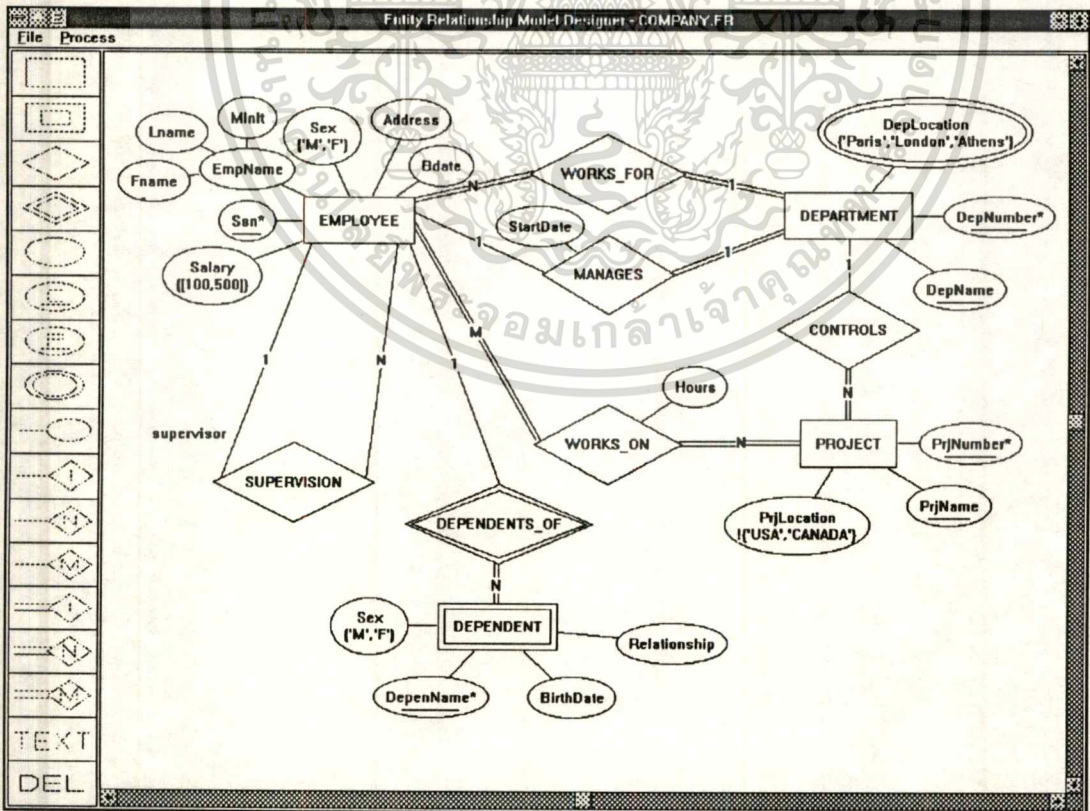
- การเขียนข้อความลงบนพื้นที่ทำงาน

ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของการเขียนข้อความซึ่งเป็น  
 ประโยคคำว่า TEXT บนแถวที่สิบหกของอีอาร์ทูลเพลทเคอร์เซอร์ของเมาท์จะเปลี่ยน  
 เป็นรูปมือที่ใช้ชี้เลือกเพลทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว  
 ประโยคคำว่า TEXT จะเปลี่ยนจากสีเขียวเป็นสีน้ำเงิน (ถ้าเป็นสีน้ำเงินอยู่แต่เดิมแสดงว่าถูก  
 เลือกไว้ก่อนแล้ว) หลังจากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยน  
 เป็นตัวอักษร A แล้วกดปุ่มซ้ายของเมาท์อีกครั้งหนึ่งจะมีกรอบข้อความเกิดขึ้นบนพื้นที่ทำงาน  
 ดังแสดงในรูปที่ A32 ให้ใส่ข้อความที่ต้องการลงในกรอบข้อความเสร็จแล้วให้กดปุ่ม OK  
 เพื่อแสดงการยืนยันกรอบข้อความจะหายไปแต่จะมีกรอบสี่เหลี่ยมผืนผ้าว่างๆ แสดงขึ้นบน  
 พื้นที่ทำงานแทนให้ใช้เมาท์เลื่อนรูปสี่เหลี่ยมผืนผ้า นั้นไปยังตำแหน่งที่ต้องการบนพื้นที่ทำงาน  
 โดยจะต้องไม่ไปทับกับวัตถุเดิมที่มีปรากฏอยู่ก่อนแล้วบนพื้นที่ทำงานแล้วกดปุ่มซ้ายของเมาท์  
 อีกครั้งหนึ่งรูปสี่เหลี่ยมผืนผ้า นั้นจะหายไปแต่จะมีข้อความที่ใส่ลงในกรอบข้อความมาปรากฏ  
 อยู่ดังแสดงในรูปที่ A33 และในรูปที่ A34 แสดงภาพแบบจำลองอีอาร์ที่ได้ถูกวาดขึ้นตาม  
 ตัวอย่างตั้งแต่ต้นจนเสร็จสมบูรณ์พร้อมที่จะนำไปผ่านกระบวนการแปลงให้เป็นตารางฐานข้อ  
 มูลในรูปบรรทัดฐานขั้นแรกต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

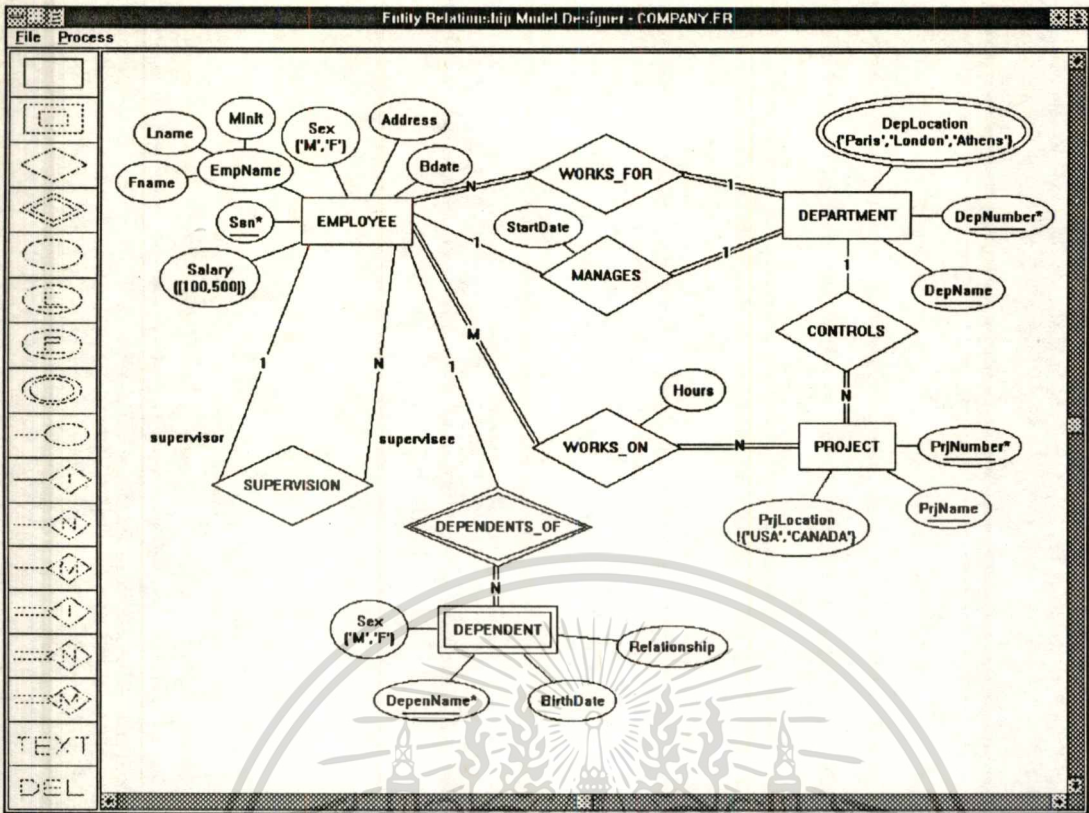


รูปที่ A32 แสดงกรอบข้อความของการเขียนข้อความลงบนพื้นที่ทำงาน



รูปที่ A33 แสดงภาพการเขียนข้อความว่า supervisor ลงบนพื้นที่ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ A34 แสดงแบบจำลองอ็อบเจกต์ที่ได้วาดขึ้นจนเสร็จสมบูรณ์

### - การลบบัตถุบนแบบจำลองอ็อบเจกต์

ผู้ใช้งานจะต้องเลื่อนเคอร์เซอร์เมาท์ไปที่สัญลักษณ์ของการลบบัตถุบนแบบจำลองอ็อบเจกต์ซึ่งเป็นรูปตัวหนังสือคำว่า DEL บนแถวที่สิบเจ็ดของอ็อบเจกต์ทูลเพลทเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปมือที่ใช้ชี้เลือกเพลทด้วยปลายนิ้วชี้ของเคอร์เซอร์แล้วกดปุ่มซ้ายของเมาท์จะเห็นว่าสัญลักษณ์รูปตัวหนังสือคำว่า DEL จะเปลี่ยนจากสีเขียวเป็นสีน้ำเงิน (ถ้าเป็นสีน้ำเงินอยู่แต่เดิมแสดงว่าถูกเลือกไว้ก่อนแล้ว) หลังจากนั้นให้เลื่อนเมาท์มายังพื้นที่ทำงานเคอร์เซอร์ของเมาท์จะเปลี่ยนเป็นรูปหัวลูกศรสีดำให้กดปุ่มซ้ายของเมาท์บนการลบบัตถุที่ต้องการลบซึ่งมีวิธีการลบบัตถุด้วยกันหกแบบคือ

#### 1. การลบแอททริบิวต์

ผู้ใช้งานต้องเลื่อนเคอร์เซอร์เมาท์รูปหัวลูกศรสีดำไปยังแอททริบิวต์ที่ต้องการลบให้กดปุ่มซ้ายของเมาท์หนึ่งครั้งจะเห็นว่าแอททริบิวต์ที่ต้องการลบทั้งนั้นเปลี่ยนสีเป็นดำและปรากฏกรอบข้อความเกิดขึ้นดังแสดงในรูปที่ A35 จากนั้นให้กดปุ่ม Yes ของกรอบข้อความเพื่อแสดงการยืนยันกรอบข้อความนั้นก็หายไปและแอททริบิวต์ที่ถูกเลือกก็ถูกลบออกดังแสดงในรูปที่ A36

#### 2. การลบแอททริบิวต์ชนิดกลุ่ม

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับใช้ในการเรียนการสอน การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้งานต้องเลือกเคอร์เซอร์เมา์ที่รูปหัวลูกศรสีดำไปยังแอททริบิวต์ชนิดกลุ่มที่ต้องการลบให้กดปุ่มซ้ายของเมา์หนึ่งครั้งจะเห็นว่าแอททริบิวต์ชนิดกลุ่มที่ต้องการลบที่นั้นเปลี่ยนสีเป็นดำทั้งกลุ่มและปรากฏกรอบข้อความเกิดขึ้นดังแสดงในรูปที่ A37 จากนั้นให้กดปุ่ม Yes ของกรอบข้อความเพื่อแสดงการยืนยันกรอบข้อความนั้นก็หายไปและแอททริบิวต์ชนิดกลุ่มที่ถูกเลือกทั้งกลุ่มก็จะถูกลบออก ดังแสดงในรูปที่ A38

### 3. การลบโวล

ผู้ใช้งานต้องเลือกเคอร์เซอร์เมา์ที่รูปหัวลูกศรสีดำไปยังตัวอักษรหรือตัวเลขที่อยู่กึ่งกลางของโวลที่ต้องการลบให้กดปุ่มซ้ายของเมา์หนึ่งครั้งจะเห็นว่าตัวอักษรโวลที่ต้องการลบที่นั้นเปลี่ยนสีเป็นดำและปรากฏกรอบข้อความเกิดขึ้นดังแสดงในรูปที่ A39 จากนั้นให้กดปุ่ม Yes ของกรอบข้อความเพื่อแสดงการยืนยันกรอบข้อความนั้นก็หายไปและโวลที่ถูกเลือกก็จะถูกลบออก ดังแสดงในรูปที่ A40

### 4. การลบความเกี่ยวพัน

ผู้ใช้งานต้องเลือกเคอร์เซอร์เมา์ที่รูปหัวลูกศรสีดำไปยังความเกี่ยวพันที่ต้องการลบให้กดปุ่มซ้ายของเมา์หนึ่งครั้งจะเห็นว่าความเกี่ยวพัน แอททริบิวต์ของความเกี่ยวพัน และโวลที่เชื่อมต่อกับความเกี่ยวพันที่ต้องการลบที่นั้นเปลี่ยนสีเป็นดำทั้งหมดและปรากฏกรอบข้อความเกิดขึ้นดังแสดงในรูปที่ A41 จากนั้นให้กดปุ่ม Yes ของกรอบข้อความเพื่อแสดงการยืนยันกรอบข้อความนั้นก็หายไป ความเกี่ยวพัน แอททริบิวต์ของความเกี่ยวพัน และโวลที่เชื่อมต่อกับความเกี่ยวพันที่ถูกเลือกก็จะถูกลบออกดังแสดงในรูปที่ A42

### 5. การลบเอ็นติตี้

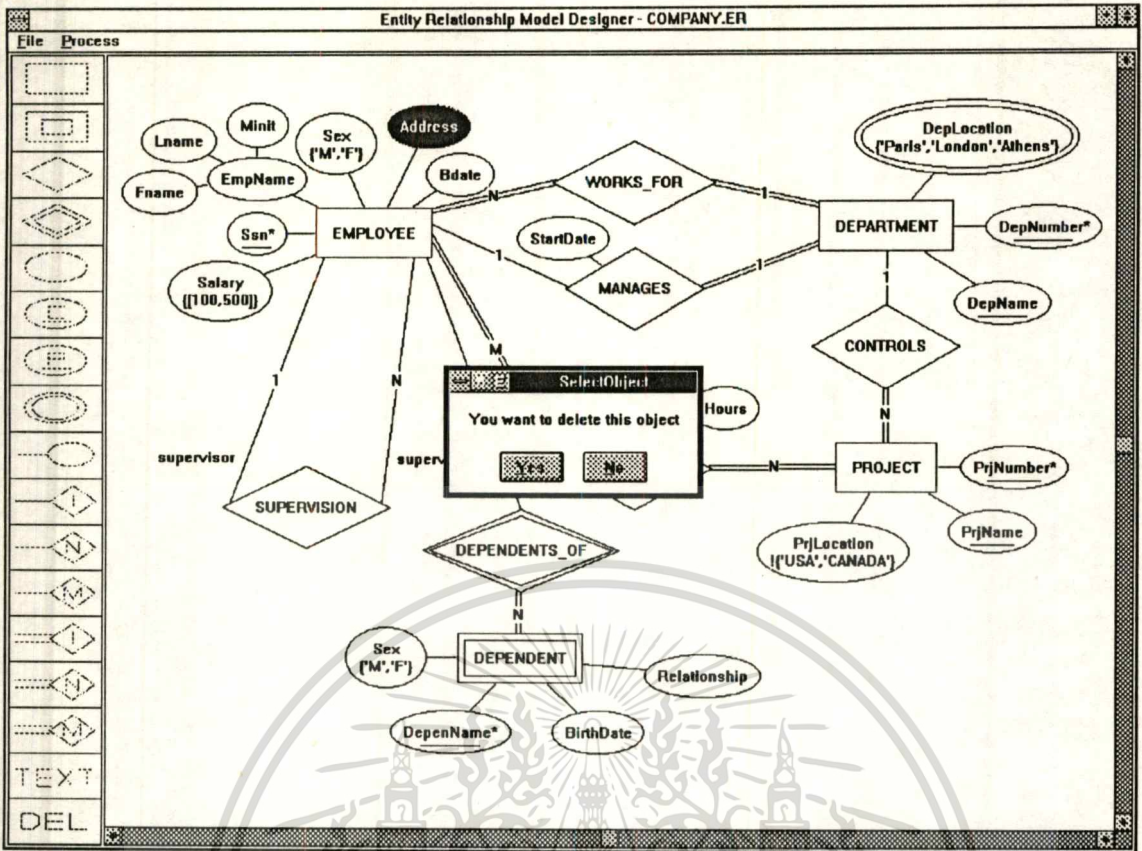
ผู้ใช้งานต้องเลือกเคอร์เซอร์เมา์ที่รูปหัวลูกศรสีดำไปยังเอ็นติตี้ที่ต้องการลบให้กดปุ่มซ้ายของเมา์หนึ่งครั้งจะเห็นว่าเอ็นติตี้ แอททริบิวต์ของเอ็นติตี้ และโวลที่เชื่อมต่อกับเอ็นติตี้ที่ต้องการลบที่นั้นเปลี่ยนสีเป็นดำทั้งหมดและปรากฏกรอบข้อความเกิดขึ้นดังแสดงในรูปที่ A43 จากนั้นให้กดปุ่ม Yes ของกรอบข้อความเพื่อแสดงการยืนยันกรอบข้อความนั้นก็หายไป เอ็นติตี้ แอททริบิวต์ของเอ็นติตี้ และโวลที่เชื่อมต่อกับเอ็นติตี้ที่ถูกเลือกก็จะถูกลบออก ดังแสดงในรูปที่ A44

### 6. การลบข้อความ

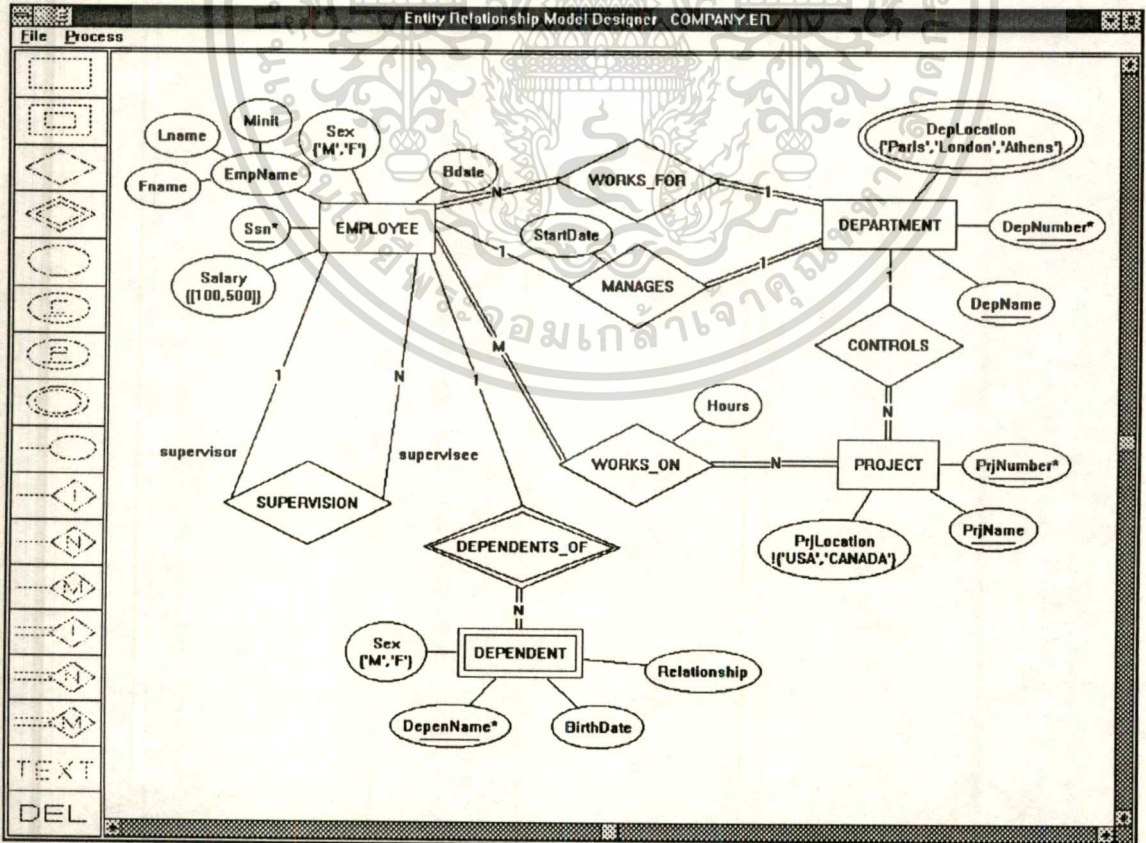
ผู้ใช้งานต้องเลือกเคอร์เซอร์เมา์ที่รูปหัวลูกศรสีดำไปยังข้อความที่ต้องการลบให้กดปุ่มซ้ายของเมา์หนึ่งครั้งจะเห็นว่าข้อความที่ต้องการลบที่นั้นเปลี่ยนสีเป็นดำและปรากฏกรอบข้อความเกิดขึ้นดังแสดงในรูปที่ A45 จากนั้นให้กดปุ่ม Yes ของกรอบข้อความเพื่อแสดงการยืนยันกรอบข้อความนั้นก็หายไปและข้อความที่ถูกเลือกก็จะถูกลบออก ดังแสดงในรูปที่ A46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่วารณิใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

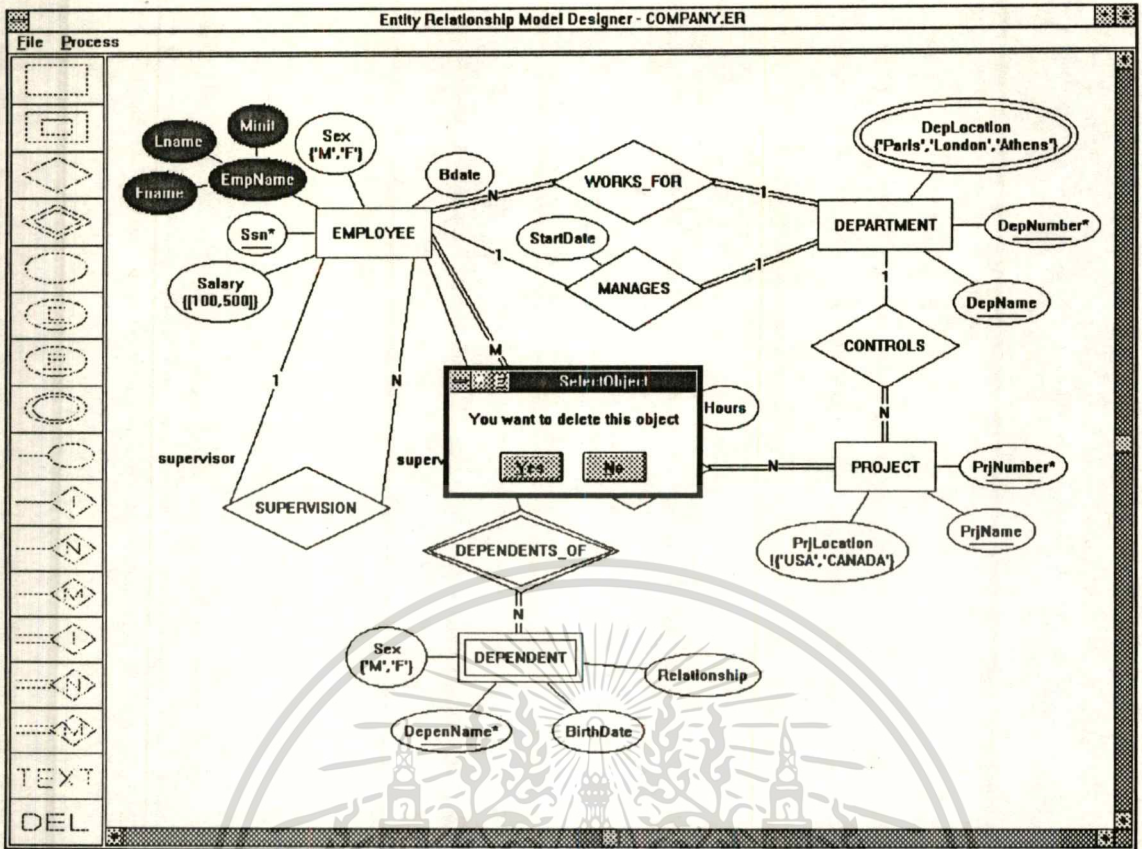


รูปที่ A35 แสดงการลบแอททริบิวต์

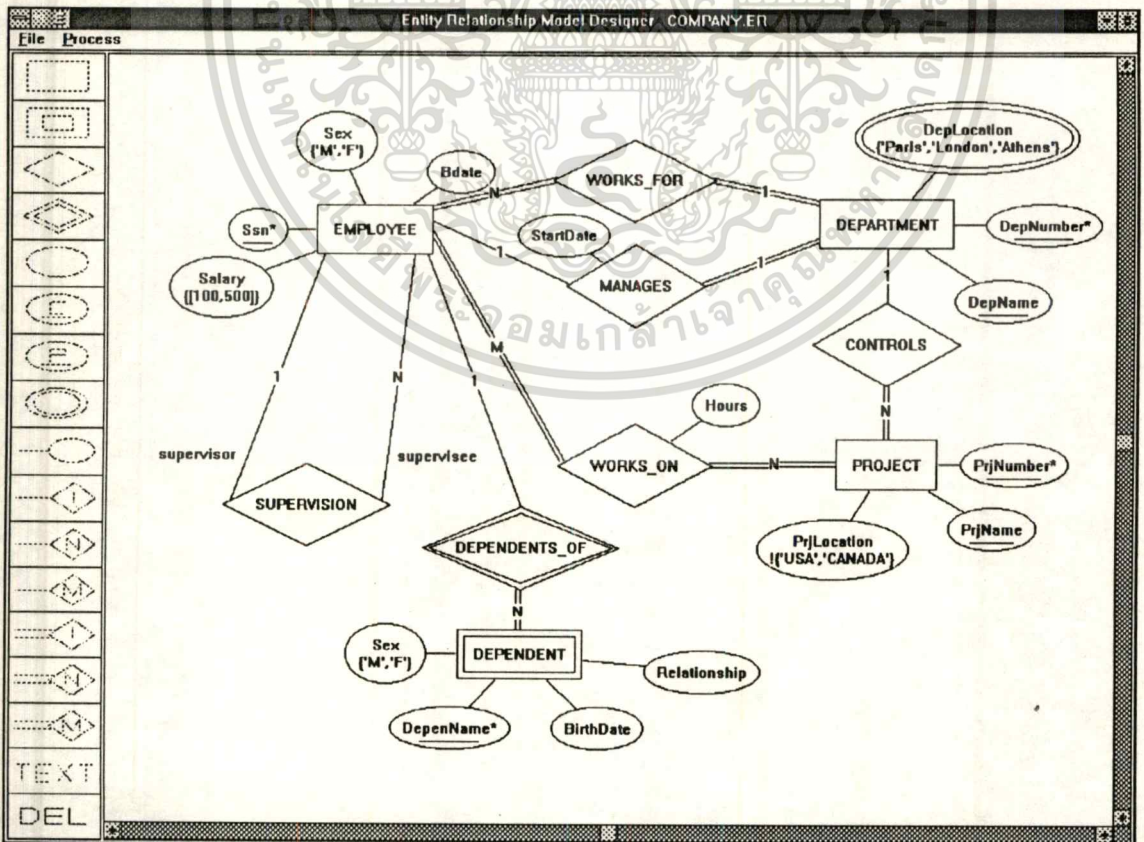


รูปที่ A36 แสดงแอททริบิวต์ Address ที่ถูกลบออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

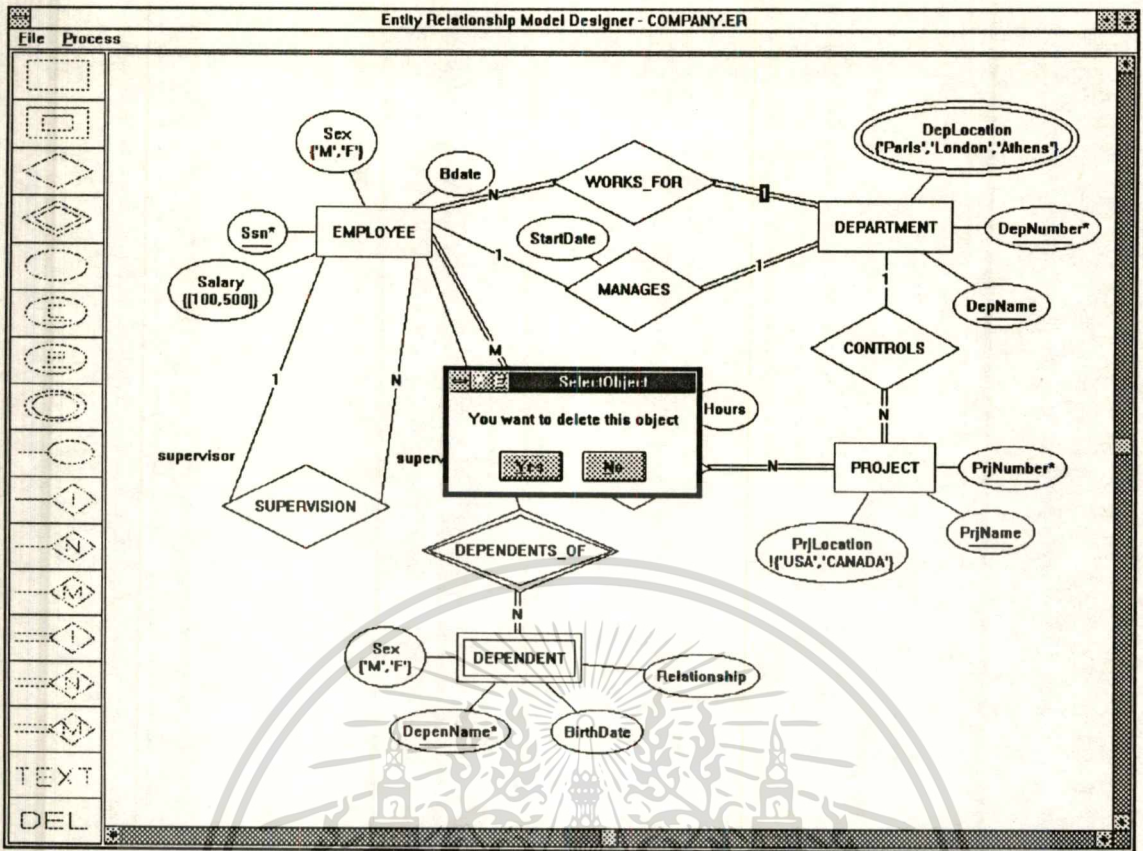


รูปที่ A37 แสดงการลบแอททริบิวต์ชนิดกลุ่ม

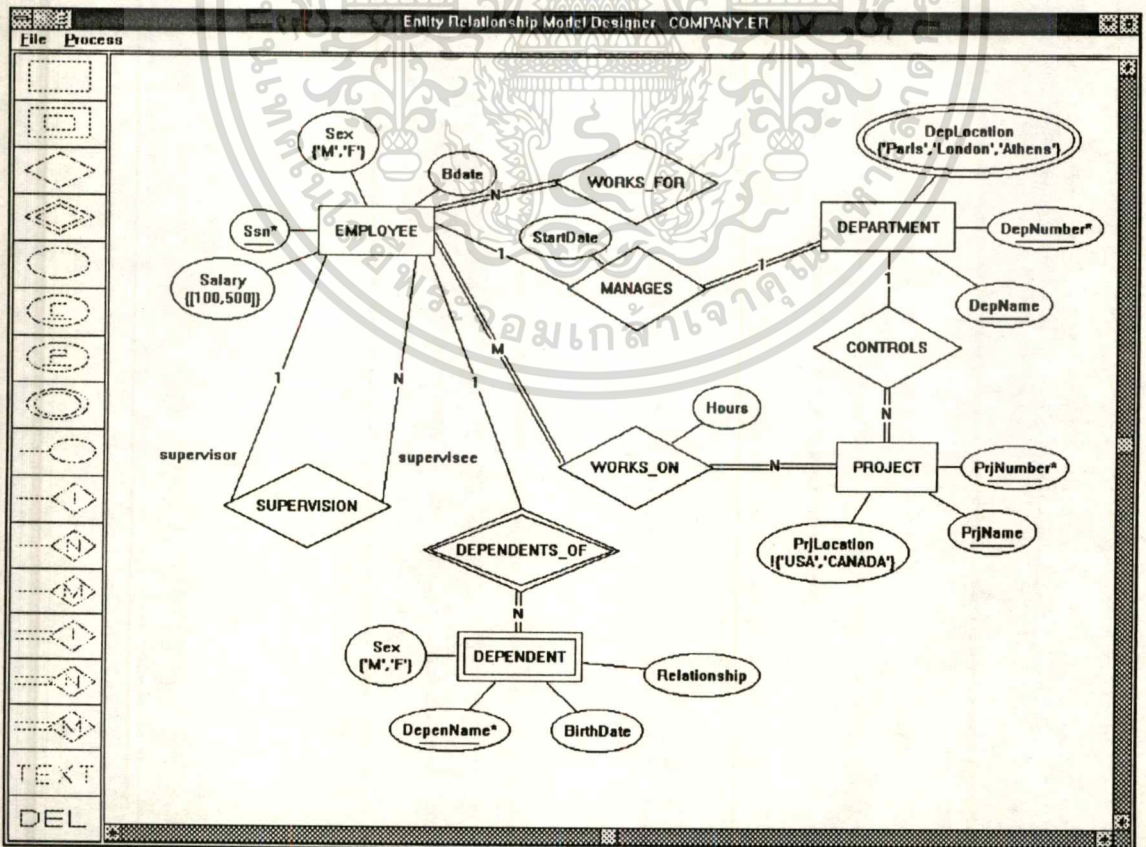


เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การแก้ไข เนื้อหา หรือการนำออกโดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



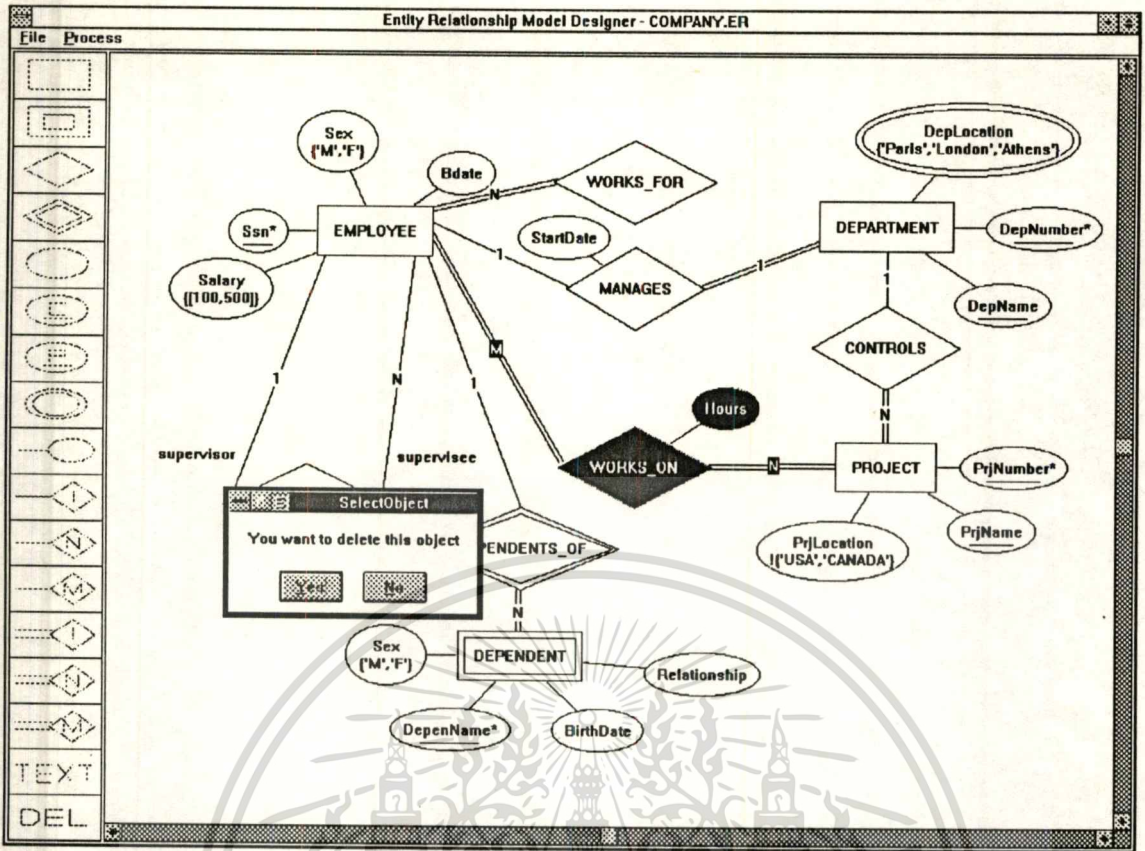
รูปที่ A39 แสดงการลบโรค



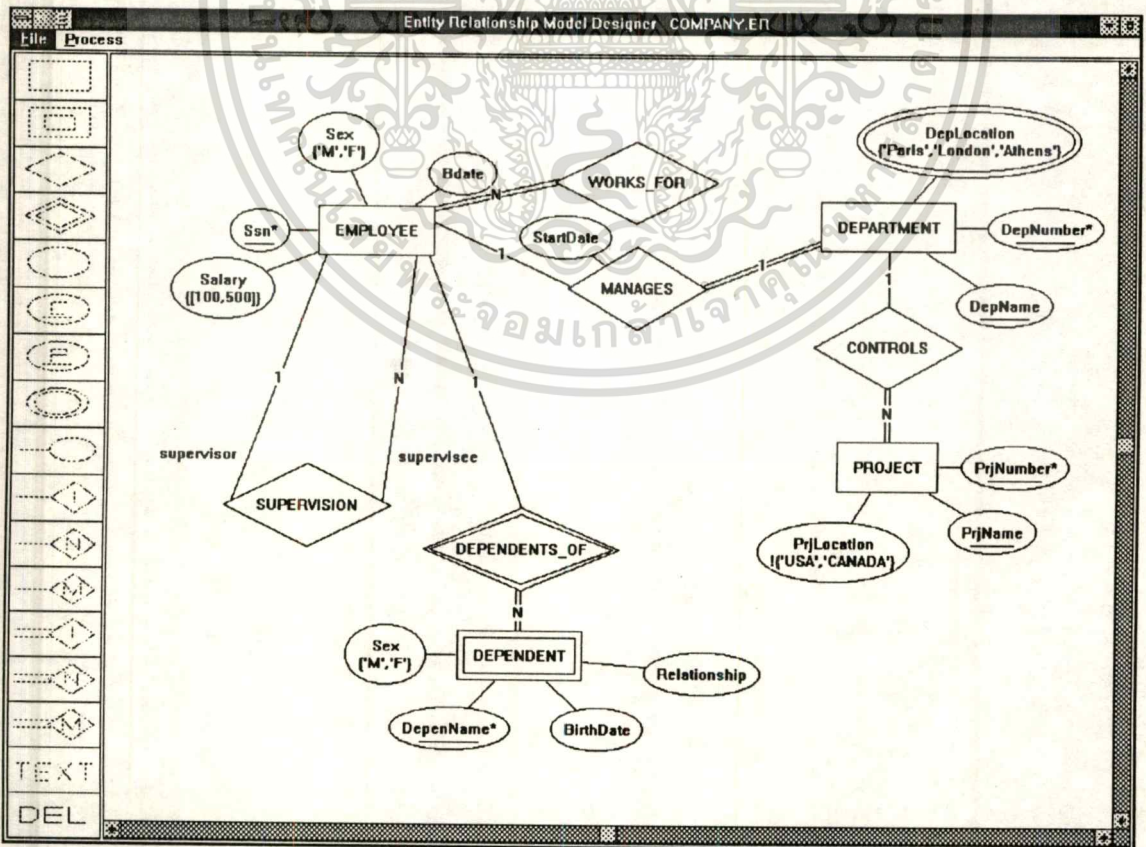
รูปที่ A40 แสดงโรคที่เชื่อมตอระหว่าง DEPARTMENT กับ WORKS\_FOR ที่ถูกลบออก

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า

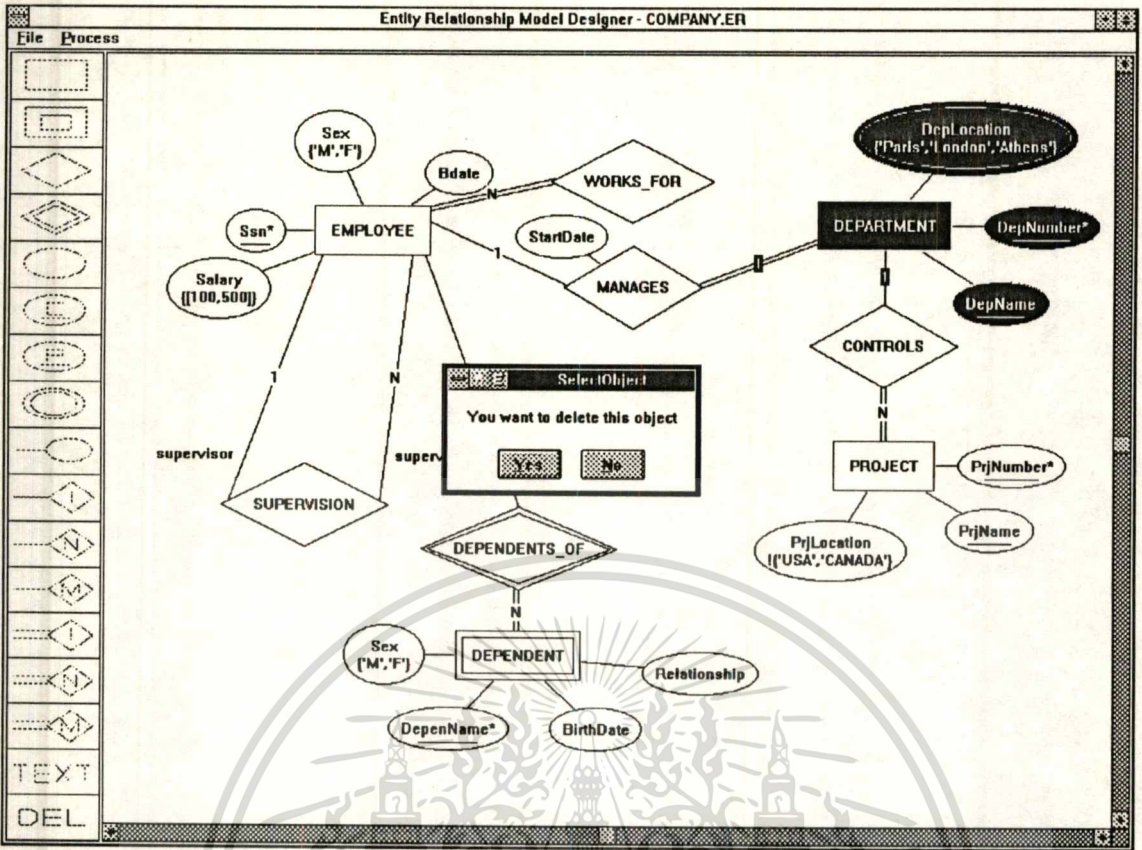
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



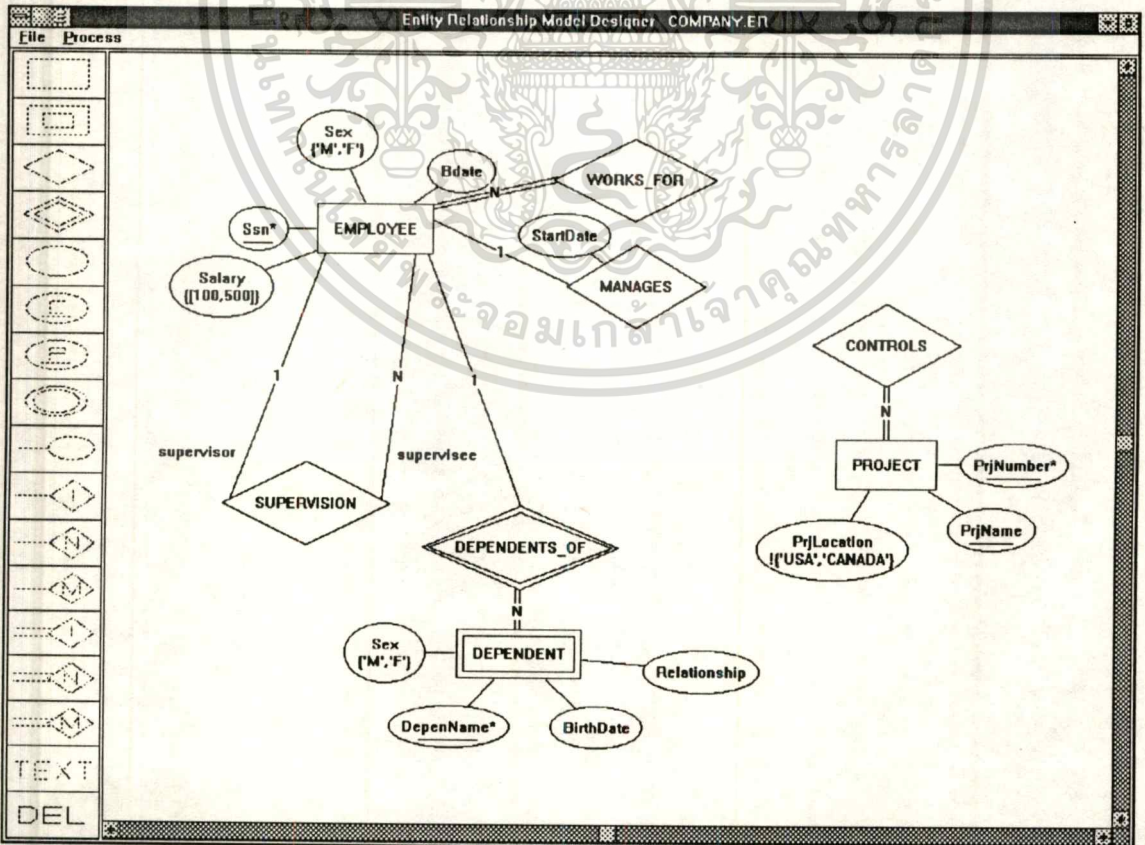
รูปที่ A41 แสดงการลบความเกี่ยวข้อง



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

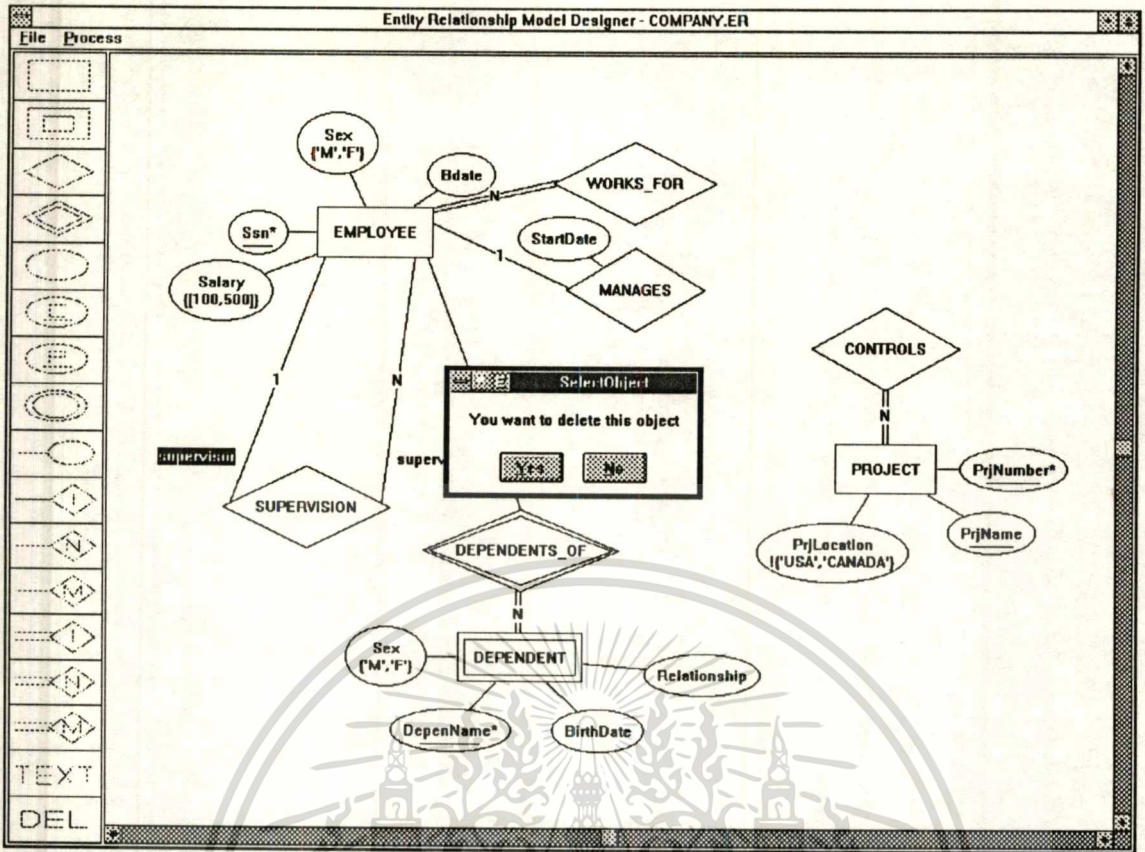


รูปที่ A43 แสดงการลบเอนิตตี้

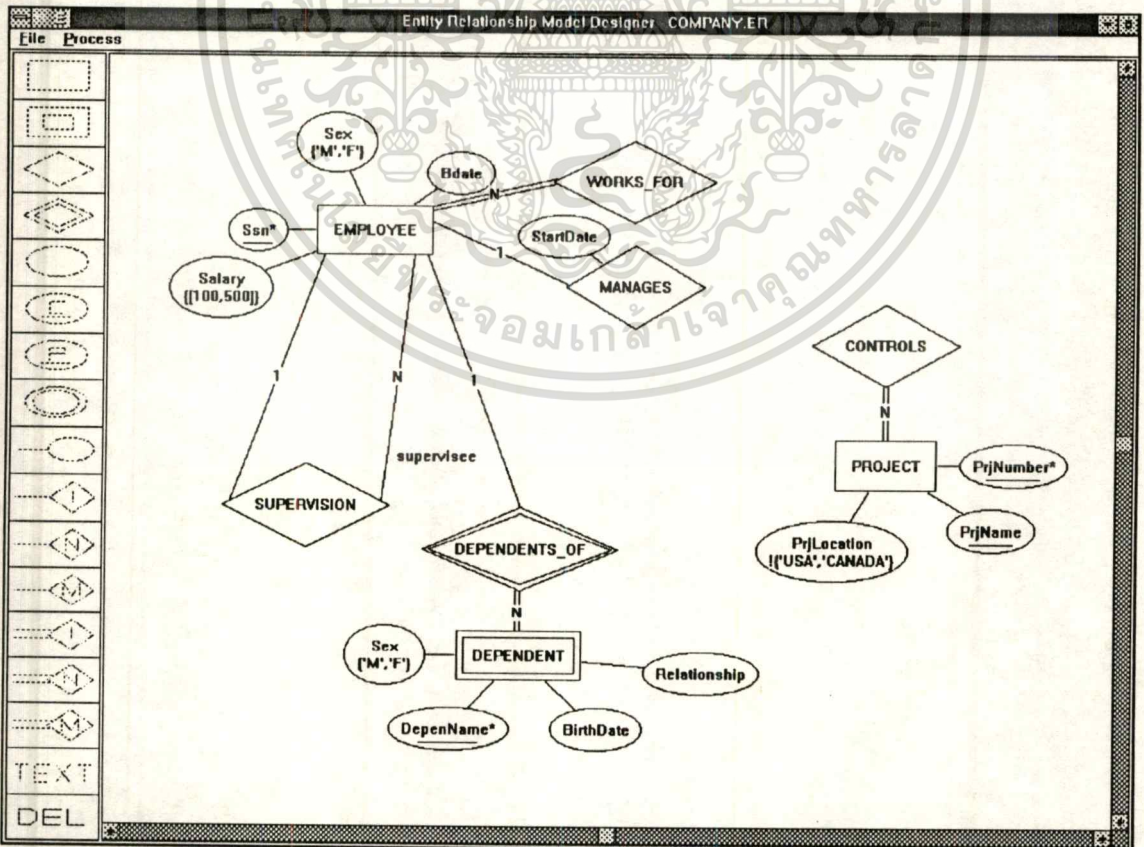


รูปที่ A44 แสดงเอนิตตี้ DEPARTMENT ที่ถูกลบออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ A45 แสดงการลบข้อความ



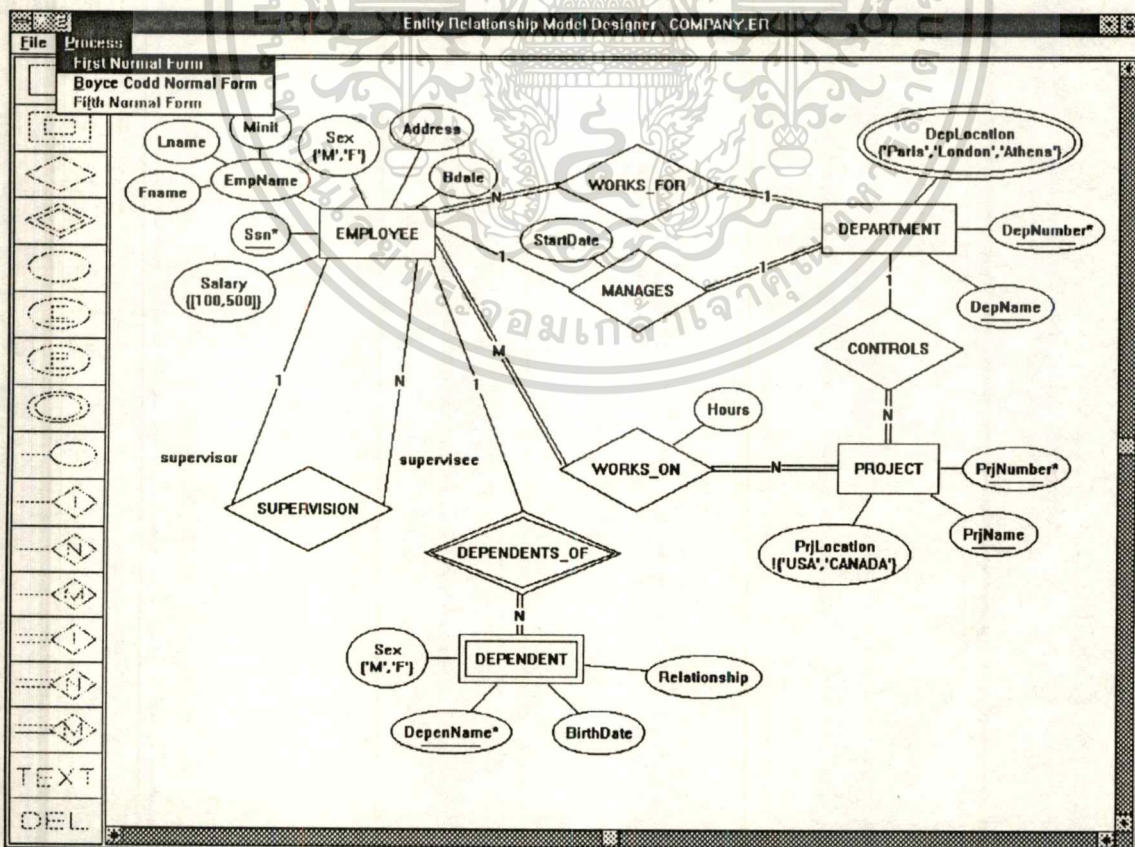
รูปที่ A46 แสดงข้อความ supervisor ที่ถูกลบออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การสร้างตารางฐานข้อมูลจากแบบจำลองอีอาร์

- การสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นแรก

เมื่อผู้ใช้งานได้วาดแบบจำลองอีอาร์เสร็จเป็นที่เรียบร้อยแล้วและต้องการที่จะแปลงแบบจำลองอีอาร์ที่ได้วาดขึ้นให้เป็นคำสั่งภาษาโครงสร้างที่ใช้ในการสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นแรกนั้นให้ผู้ใช้งานเลือกเมนู First Normal Form จากเมนูหลัก Process ดังแสดงในรูปที่ A47 เคอร์เซอร์เมา์ท์จะเปลี่ยนจากรูปลูกศรสีขาวเป็นรูปนาฬิกาทรายเพื่อให้ผู้ใช้งานทราบว่ากำลังทำการประมวลผลข้อมูลอยู่ให้รอและเมื่อเคอร์เซอร์เมา์ท์เปลี่ยนจากรูปนาฬิกาทรายกลับมาเป็นรูปลูกศรสีขาวเป็นดังเดิมแสดงว่าการประมวลผลข้อมูลได้เสร็จสิ้นลงแต่ถ้าในระหว่างการประมวลผลมีกรอบข้อความเกิดขึ้นแสดงว่าเกิดการผิดพลาดขึ้นที่แบบจำลองอีอาร์ให้จำคำอธิบายความผิดพลาดในกรอบข้อความนั้นเมื่อจำได้แล้วให้ปิดกรอบข้อความนั้นโดยกดปุ่ม OK และกลับไปแก้ไขแบบจำลองอีอาร์ใหม่ให้ถูกต้องตามข้อผิดพลาดที่อธิบายไว้ภายในกรอบข้อความและเมื่อแน่ใจว่าได้แก้ไขแบบจำลองอีอาร์ถูกต้องเรียบร้อยแล้วให้ปฏิบัติตามขั้นตอนที่กล่าวมาแล้วข้างต้นใหม่อีกครั้ง ผลลัพธ์ที่ได้จะเป็นคำสั่งในการสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นแรกและถูกเก็บลงในไฟล์ FIRST.SQL ดังแสดงในรูปที่ A48 สามารถนำไปใช้งานได้ทันที



รูปที่ A47 แสดงการเลือกเมนู First Normal Form จากเมนูหลัก Process

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น มิใช่สำหรับผู้ใดที่นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CREATE TABLE DEPARTMENT
(
    Ssn          NUMBER          NOT NULL,
    DepNumber    NUMBER          NOT NULL,
    DepName      CHAR(30)       NOT NULL,
    StartDate    DATE
);
CREATE TABLE DEPENDENT
(
    Sex          CHAR(1),
    DepenName    CHAR(30)       NOT NULL,
    BirthDate    DATE,
    Relationship  CHAR(30),
    Ssn          NUMBER          NOT NULL
);
CREATE TABLE DepLocation
(
    DepNumber    NUMBER          NOT NULL,
    DepLocation  CHAR(30)       NOT NULL
);
CREATE TABLE EMPLOYEE
(
    Fname        CHAR(1),
    Lname        CHAR(10),
    Minit        CHAR(1),
    Ssn          NUMBER          NOT NULL,
    Salary       NUMBER,
    Sex          CHAR(1),
    Address      CHAR(30),
    Bdate        DATE,
    DepNumber    NUMBER          NOT NULL,
    Ssn23        NUMBER
);
CREATE TABLE PROJECT
(
    DepNumber    NUMBER,
    PrjNumber    NUMBER          NOT NULL,
    PrjName      CHAR(30)       NOT NULL,
    PrjLocation  CHAR(30)
);
CREATE TABLE WORKS_ON
(
    Ssn          NUMBER          NOT NULL,
    PrjNumber    NUMBER          NOT NULL,
    Hours        DATE
);
CREATE UNIQUE INDEX INDEX_#1 ON EMPLOYEE (Ssn);
CREATE UNIQUE INDEX INDEX_#2 ON DEPARTMENT (DepNumber);
CREATE UNIQUE INDEX INDEX_#3 ON PROJECT (PrjNumber);
CREATE UNIQUE INDEX INDEX_#4 ON DEPENDENT (DepenName,Ssn);
CREATE UNIQUE INDEX INDEX_#5 ON DepLocation (DepNumber,DepLocation);
CREATE UNIQUE INDEX INDEX_#6 ON WORKS_ON (Ssn,PrjNumber);

CREATE VIEW DEPENDENT_VIEW AS SELECT *
FROM DEPENDENT
WHERE Sex IN ('F','M')
WITH CHECK OPTION;

CREATE VIEW DepLocation_VIEW AS SELECT *
FROM DepLocation
WHERE DepLocation IN ('Athens','London','Paris')
WITH CHECK OPTION;

CREATE VIEW EMPLOYEE_VIEW AS SELECT *
FROM EMPLOYEE
WHERE Sex IN ('F','M') AND ((Salary >= 100 AND Salary <= 500))
WITH CHECK OPTION;

CREATE VIEW PROJECT_VIEW AS SELECT *
FROM PROJECT
WHERE PrjLocation NOT IN ('CANADA','USA')
WITH CHECK OPTION;

```

รูปที่ A48 แสดงภาษาโครงสร้างที่ใช้ในการสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นแรก (FIRST.SQL)

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## - การสร้างตารางฐานข้อมูลในบรรทัดฐานชั้น Boyce/Codd

เมื่อผู้ใช้งานได้ตารางฐานข้อมูลในบรรทัดฐานชั้นแรกแล้วยังต้องการที่จะทำตารางฐานข้อมูลที่ได้นั้นให้สูงขึ้นอยู่ในบรรทัดฐานชั้น Boyce/Codd ให้ผู้ใช้งานเลือกเมนู Boyce Codd Normal Form จากเมนูหลัก Process ดังแสดงในรูปที่ A49 หลังจากที่ได้เลือกเมนูนี้แล้วจะปรากฏกรอบข้อความขึ้นดังแสดงในรูปที่ A50 ภายในกรอบข้อความจะประกอบด้วย

### 1. กรอบชื่อตาราง (TABLE NAME)

กรอบชื่อตารางจะเก็บชื่อของตารางที่ถูกสร้างขึ้นจากขั้นตอนการแปลงแบบจำลองอริให้เป็นตารางฐานข้อมูลในบรรทัดฐานชั้นแรกทั้งหมด จากรูปที่ A59 ตารางที่ถูกสร้างขึ้นมีอยู่ 6 ตารางได้แก่ DEPARTMENT DEPENDENT DepLocation EMPLOYEE PROJECT และ WORKS\_ON ผู้ใช้งานสามารถที่จะเลือกตารางใดก็ได้มาแก้ไขความสัมพันธ์ระหว่างแอททริบิวต์ (Functional Dependency) โดยการเลื่อนเมาท์ไปยังคอมโบบ็อกซ์ TABLE NAME แล้วกดปุ่มซ้ายของเมาท์หนึ่งครั้งที่สัญลักษณ์ลูกศรชี้ลง คอมโบบ็อกซ์จะแสดงรายชื่อของตารางทั้งหมดให้เห็น กดปุ่มซ้ายของเมาท์อีกหนึ่งครั้งบนชื่อตารางที่ต้องการแก้ไข จากตัวอย่างตารางที่ถูกเลือกมาแก้ไขได้แก่ตาราง WORKS\_ON

### 2. กรอบแอททริบิวต์ด้านซ้าย (Left Hand Side)

แอททริบิวต์ด้านซ้ายแสดงถึงแอททริบิวต์ทั้งหมดของตารางที่ถูกเลือกซึ่งจะเห็นว่าตาราง WORKS\_ON นั้นประกอบด้วยแอททริบิวต์ Hours, PrjNumber และ Ssn กรอบแอททริบิวต์ด้านซ้ายนี้มีไว้สำหรับแสดงความสัมพันธ์ระหว่างแอททริบิวต์ด้านซ้ายมือให้ผู้ใช้งานได้เห็นเพื่อช่วยในการได้เพิ่มเติมหรือแก้ไขความสัมพันธ์ระหว่างแอททริบิวต์ได้สะดวกขึ้น

### 3. กรอบแอททริบิวต์ด้านขวา (Right Hand Side)

แอททริบิวต์ด้านขวาแสดงถึงแอททริบิวต์ทั้งหมดของตารางที่ถูกเลือกซึ่งจะเห็นว่าตาราง WORKS\_ON นั้นประกอบด้วยแอททริบิวต์ Hours PrjNumber และ Ssn กรอบแอททริบิวต์ด้านขวานี้มีไว้สำหรับแสดงความสัมพันธ์ระหว่างแอททริบิวต์ด้านขวามือให้ผู้ใช้งานได้เห็นเพื่อช่วยในการได้เพิ่มเติมหรือแก้ไขความสัมพันธ์ระหว่างแอททริบิวต์ได้สะดวกขึ้น

### 4. ปุ่ม SELECT

ใช้ในการเปิดเคอร์เซอร์ของฐานข้อมูลความสัมพันธ์ระหว่างแอททริบิวต์ เมื่อผู้ใช้งานกดปุ่มนี้ ข้อมูลความสัมพันธ์ระหว่างแอททริบิวต์ทั้งหมดของตาราง WORKS\_ON ที่เลือกจะถูกเก็บลงในเคอร์เซอร์ของตัวจัดการฐานข้อมูล

### 5. ปุ่ม FETCH

ใช้ในการอ่านข้อมูลที่เก็บอยู่ในเคอร์เซอร์ของฐานข้อมูลความสัมพันธ์ระหว่างแอททริบิวต์ที่ได้ SELECT ไว้เมื่อผู้ใช้งานกดปุ่มนี้ข้อมูลของความสัมพันธ์ระหว่างแอททริบิวต์ของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญญาติให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง WORKS\_ON ก็จะถูกแสดงขึ้นดังแสดงในรูปที่ A51 เมื่อผู้ใช้งานกดปุ่ม FETCH นี้ซ้ำอีก ข้อมูลความสัมพันธ์ระหว่างแอททริบิวต์ของตาราง WORKS\_ON ก็จะถูกแสดงดังในรูปที่ A52 A53 จนเกิดกรอบข้อความว่า No More Data ดังแสดงในรูปที่ A54 แสดงว่าข้อมูลความสัมพันธ์ระหว่างแอททริบิวต์ของตาราง WORKS\_ON ในเคอร์เซอร์ของฐานข้อมูลได้ถูกแสดงจนหมดแล้ว ดังนั้นจากการที่ผู้ใช้งานกดปุ่ม FETCH จนเกิดกรอบข้อความว่า 'No More Data' ขึ้นสามารถอธิบายได้ว่าตาราง WORKS\_ON นั้นมีความสัมพันธ์ระหว่างแอททริบิวต์ดังต่อไปนี้คือ

- (PrjNumber, Ssn) → (Ssn) \* จากรูปที่ A51
- (PrjNumber, Ssn) → (PrjNumber) \* จากรูปที่ A52
- (PrjNumber, Ssn) → (Hours) \* จากรูปที่ A53

ความสัมพันธ์ระหว่างแอททริบิวต์ที่ถูกแสดงนี้ถูกสร้างขึ้นจากขั้นตอนการสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นแรก ดังนั้นถ้าผู้ใช้งานยังเห็นว่าไม่ถูกต้องก็สามารถทำการเปลี่ยนแปลงแก้ไขใหม่ได้ ปุ่ม FETCH นี้จะไม่สามารถกดได้ถ้าเคอร์เซอร์ของฐานข้อมูลความสัมพันธ์ระหว่างแอททริบิวต์ถูกปิดอยู่

#### 6. ปุ่ม DELETE

ใช้ในการลบความสัมพันธ์ระหว่างแอททริบิวต์ เมื่อผู้ใช้งานกดปุ่มนี้ความสัมพันธ์ระหว่างแอททริบิวต์ที่กำลังถูกแสดงอยู่ขณะนั้นก็จะถูกลบออกจากฐานข้อมูลแล้ว FETCH ข้อมูลความสัมพันธ์ระหว่างแอททริบิวต์ตัวถัดไปมาแสดง จากตัวอย่างให้ผู้ใช้งานลบความสัมพันธ์ระหว่างแอททริบิวต์ของตาราง WORKS\_ON ในขั้นตอนที่ 5 ทั้งให้หมด (เพื่อความถูกต้องของตัวอย่างที่ตามมาภายหลัง) แล้วกดปุ่ม SELECT เพื่อเปิดเคอร์เซอร์ของฐานข้อมูลความสัมพันธ์ระหว่างแอททริบิวต์ซ้ำอีกครั้งและกดปุ่ม FETCH ใหม่อีกครั้งจะเกิดกรอบข้อความว่า No More Data ทันทีแสดงว่าไม่มีความสัมพันธ์ระหว่างแอททริบิวต์ของตาราง WORKS\_ON เหลืออยู่อีกเลย

#### 7. ปุ่ม INSERT

ใช้ในการเพิ่มความสัมพันธ์ระหว่างแอททริบิวต์ เมื่อผู้ใช้งานกดปุ่มนี้กรอบข้อความแอททริบิวต์ด้านซ้ายและแอททริบิวต์ด้านขวาจะเคลียร์เป็นสีขาวให้ผู้ใช้งานเลื่อนเมาท์ไปยังกรอบข้อความความแอททริบิวต์ด้านซ้ายและเลือกแอททริบิวต์ที่ต้องการภายในกรอบความนั้นโดยการกดที่ปุ่มซ้ายของเมาท์ซึ่งการเลือกแอททริบิวต์นั้นสามารถเลือกได้มากกว่าหนึ่งแอททริบิวต์โดยแอททริบิวต์ที่ถูกเลือกนั้นจะมีแถบสีเกิดขึ้นหลังจากนั้นให้เลื่อนเมาท์ไปยังกรอบข้อความความแอททริบิวต์ด้านขวาและเลือกแอททริบิวต์โดยวิธีการเดียวกันกับการเลือกแอททริบิวต์ด้านซ้ายดังแสดงในรูปที่ A55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 8.ปุ่ม Confirm

ใช้ในการยืนยันการเพิ่มความสัมพันธ์ระหว่างแอททริบิวต์ เมื่อผู้ใช้งานกดปุ่มนี้ความสัมพันธ์ระหว่างแอททริบิวต์ที่เลือกในขั้นตอนที่ 7 ก็จะถูกเก็บลงบนฐานข้อมูล จากรูปที่ A55 A56 และ A57 แสดงการเพิ่มความสัมพันธ์ระหว่างแอททริบิวต์ดังต่อไปนี้คือ

### ตาราง WORKS\_ON

(Hours,PrjNumber) → (Ssn) \* จากรูปที่ A55

(Ssn) → (PrjNumber) \* จากรูปที่ A56

### ตาราง PROJECT

(DepNumber,PrjLocation,PrjName,PrjNumber) →

(DepNumber,PrjLocation,PrjName,PrjNumber) \* จากรูปที่ A57

## 9.ปุ่ม Cancel

ใช้ในการยกเลิกการเพิ่มความสัมพันธ์ระหว่างแอททริบิวต์ เมื่อผู้ใช้งานกดปุ่มนี้ความสัมพันธ์ระหว่างแอททริบิวต์ที่เลือกในขั้นตอนที่ 7 ก็จะไม่ถูกเก็บลงบนฐานข้อมูล

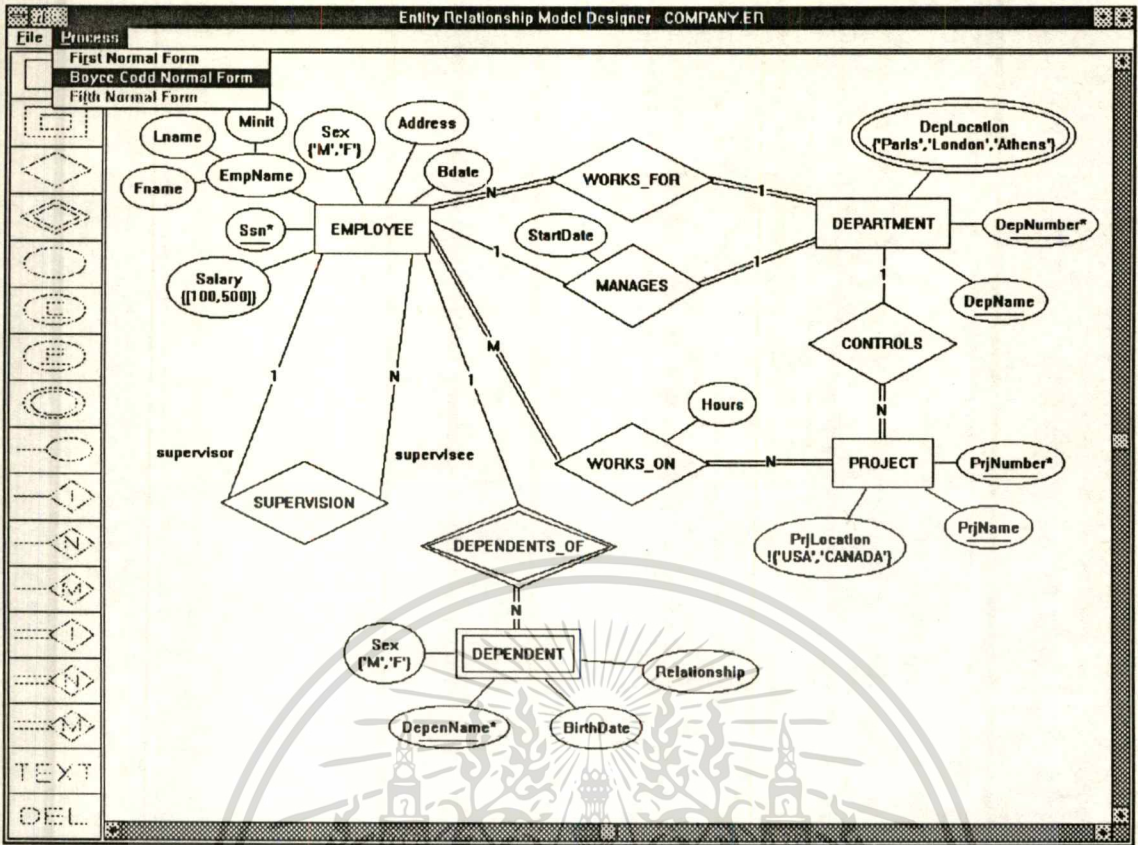
## 10.ปุ่ม RUN

ใช้ในการประมวลผลความสัมพันธ์ระหว่างแอททริบิวต์ เพื่อสร้างตารางฐานข้อมูลในบรรทัดฐานชั้น Boyce/Codd เมื่อผู้ใช้งานกดปุ่มนี้เคอร์เซอร์เมาท์จะเปลี่ยนจากรูปลูกศรสีขาวเป็นรูปนาฬิกาทรายเพื่อให้ผู้ใช้งานทราบว่ากำลังทำการประมวลผลข้อมูลอยู่ให้รอและเมื่อเคอร์เซอร์เมาท์เปลี่ยนจากรูปนาฬิกาทรายกลับมาเป็นรูปลูกศรสีขาวเป็นดังเดิมแสดงว่าการประมวลผลข้อมูลได้เสร็จสิ้นลงแต่ถ้าในระหว่างการประมวลผลมีกรอบข้อความเกิดขึ้นแสดงว่าเกิดการผิดพลาดขึ้นที่ความสัมพันธ์ระหว่างแอททริบิวต์ให้จำคำอธิบายความผิดพลาดในกรอบข้อความนั้นเมื่อจำได้แล้วให้ปิดกรอบข้อความนั้นโดยกดปุ่ม OK และกลับไปแก้ไขความสัมพันธ์ระหว่างแอททริบิวต์ใหม่ให้ถูกต้องตามข้อผิดพลาดที่อธิบายไว้ภายในกรอบข้อความและเมื่อแน่ใจว่าได้แก้ไขความสัมพันธ์ระหว่างแอททริบิวต์ถูกต้องเรียบร้อยแล้วให้ปฏิบัติตามขั้นตอนที่กล่าวมาแล้วข้างต้นใหม่อีกครั้ง ผลลัพธ์ที่ได้จะเป็นคำสั่งในการสร้างตารางฐานข้อมูลในบรรทัดฐานชั้น Boyce/Codd และถูกเก็บลงในไฟล์ BCNF.SQL ดังแสดงในรูปที่ A58 และสามารถนำไปใช้งานได้ทันทีและกรอบข้อความที่แสดงในรูปที่ A50 จะหายไปและกลับไปยังการวาดแผนภาพอีอาร์ตามเดิม

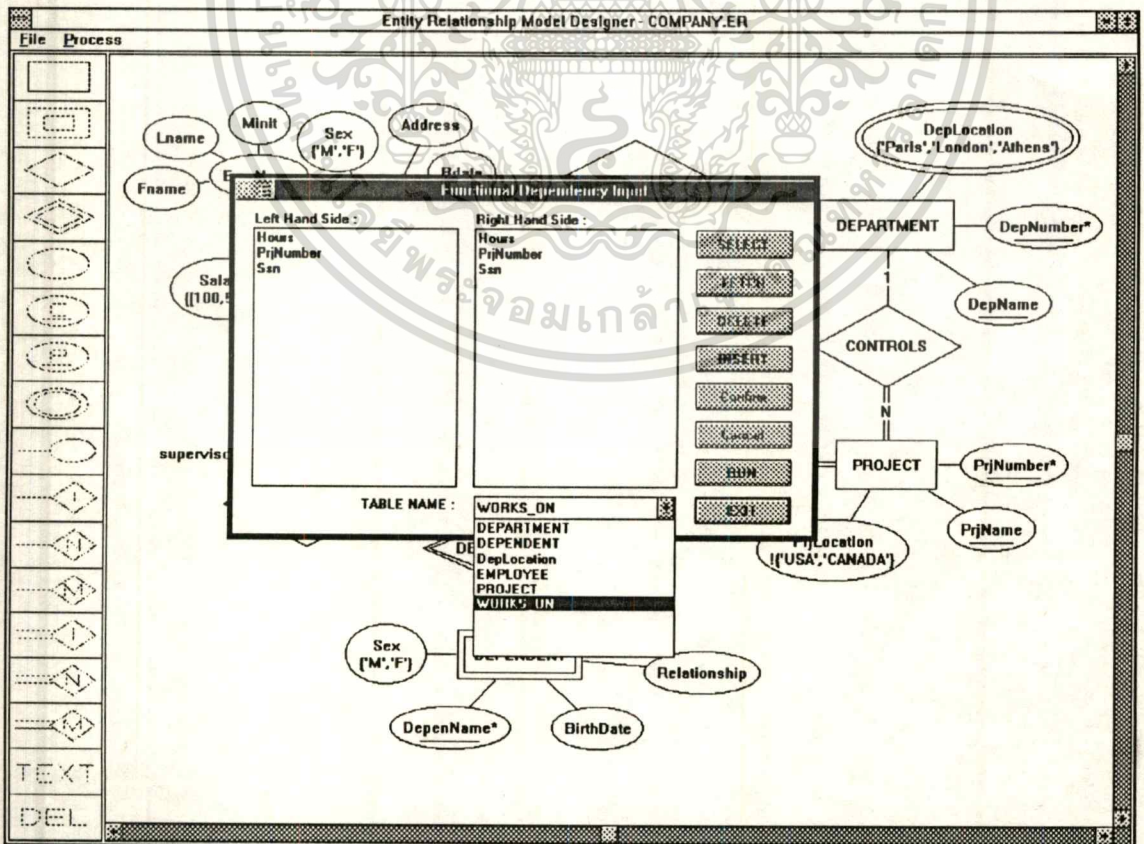
## 11.ปุ่ม EXIT

ใช้ในการออกจากการประมวลผลชั้น Boyce/Codd เมื่อผู้ใช้งานกดปุ่มนี้กรอบข้อความจะหายไปโดยไม่มีการประมวลผลใดๆเกิดขึ้นและกลับไปยังการวาดแผนภาพอีอาร์ตามเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

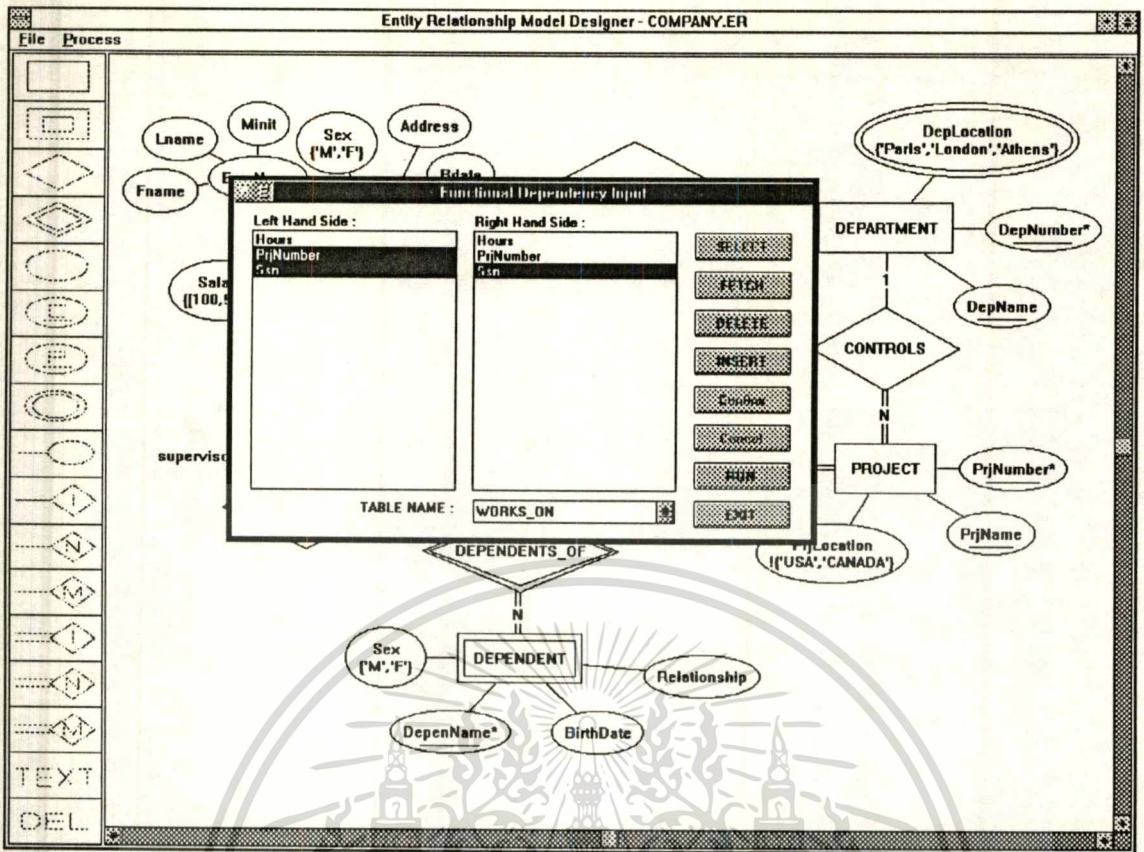


รูปที่ A49 แสดงการเลือกเมนู Boyce Codd Normal Form จากเมนูหลัก Process

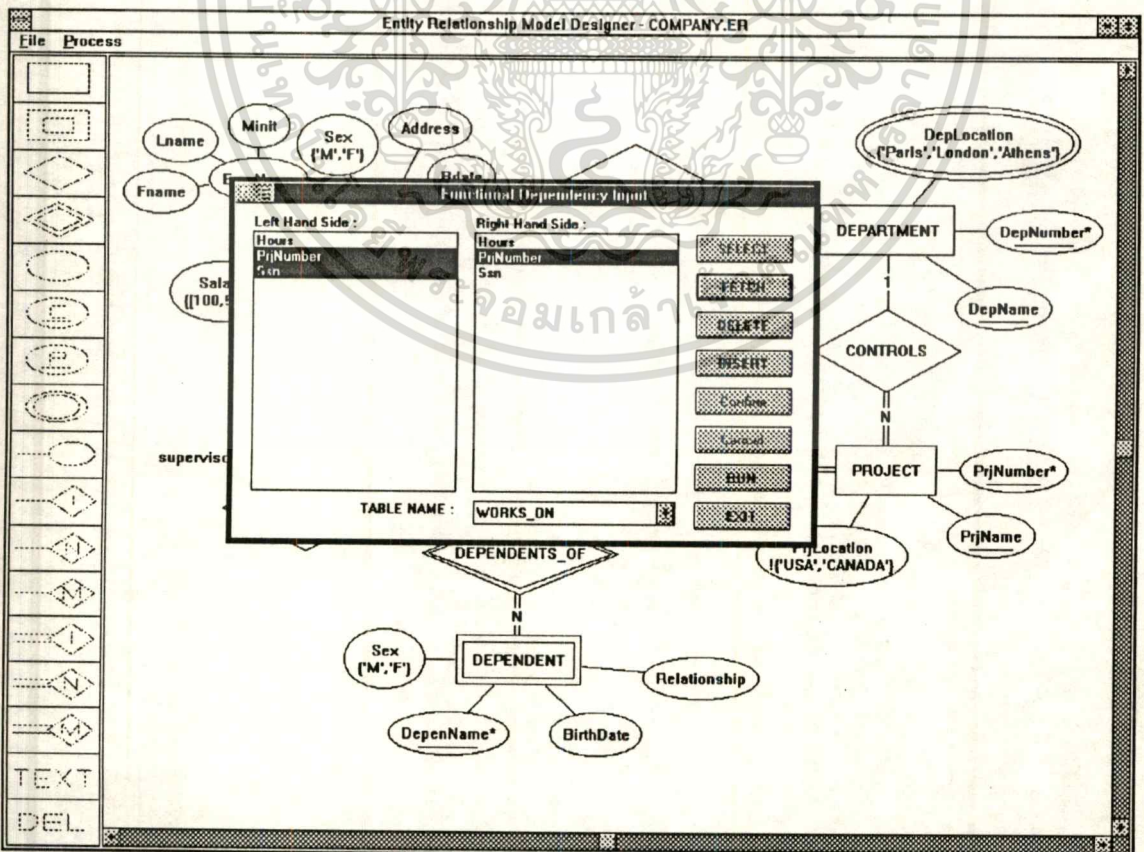


เอกสารนี้เป็นรูปที่ A50 แสดงกรอบข้อความของการประมวลผลขั้น Boyce/Codd Normal Form โดยใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

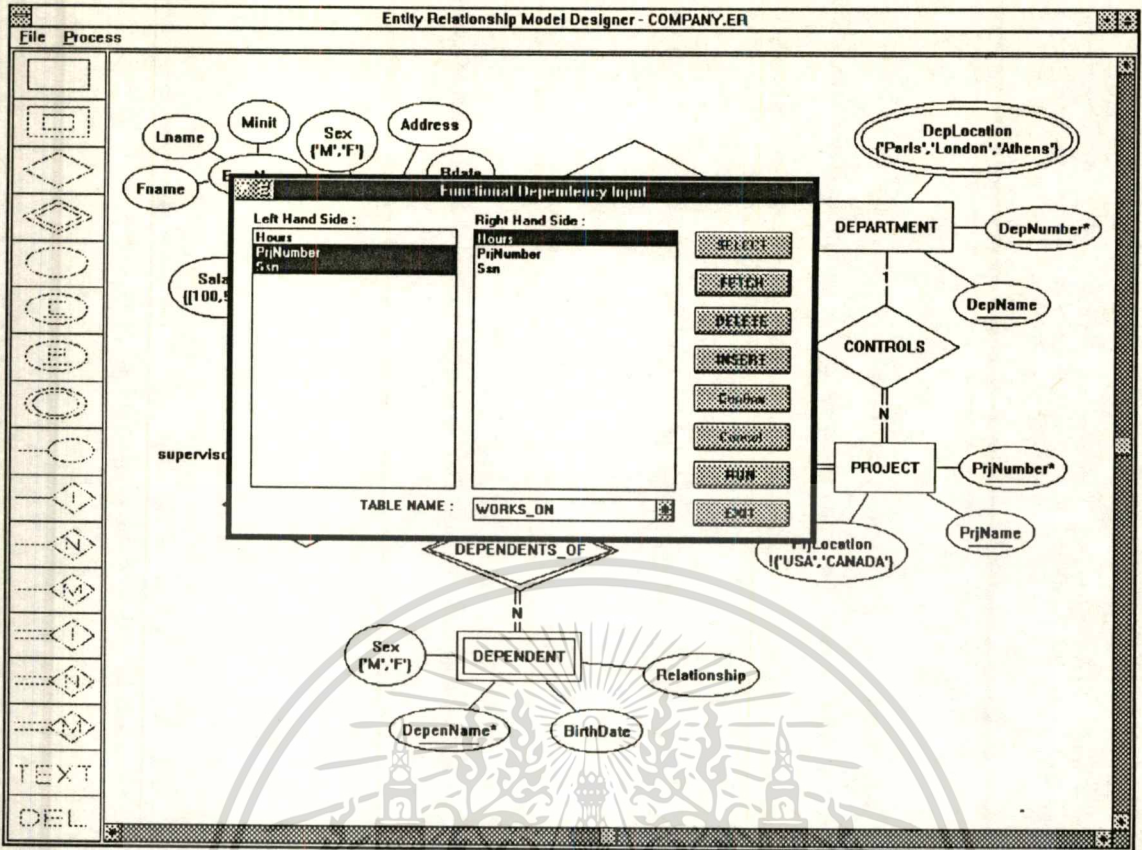


รูปที่ A51 แสดงความสัมพันธ์ระหว่างแอททริบิวต์ (PrjNumber,Ssn) → (Ssn) ของตาราง WORKS\_ON

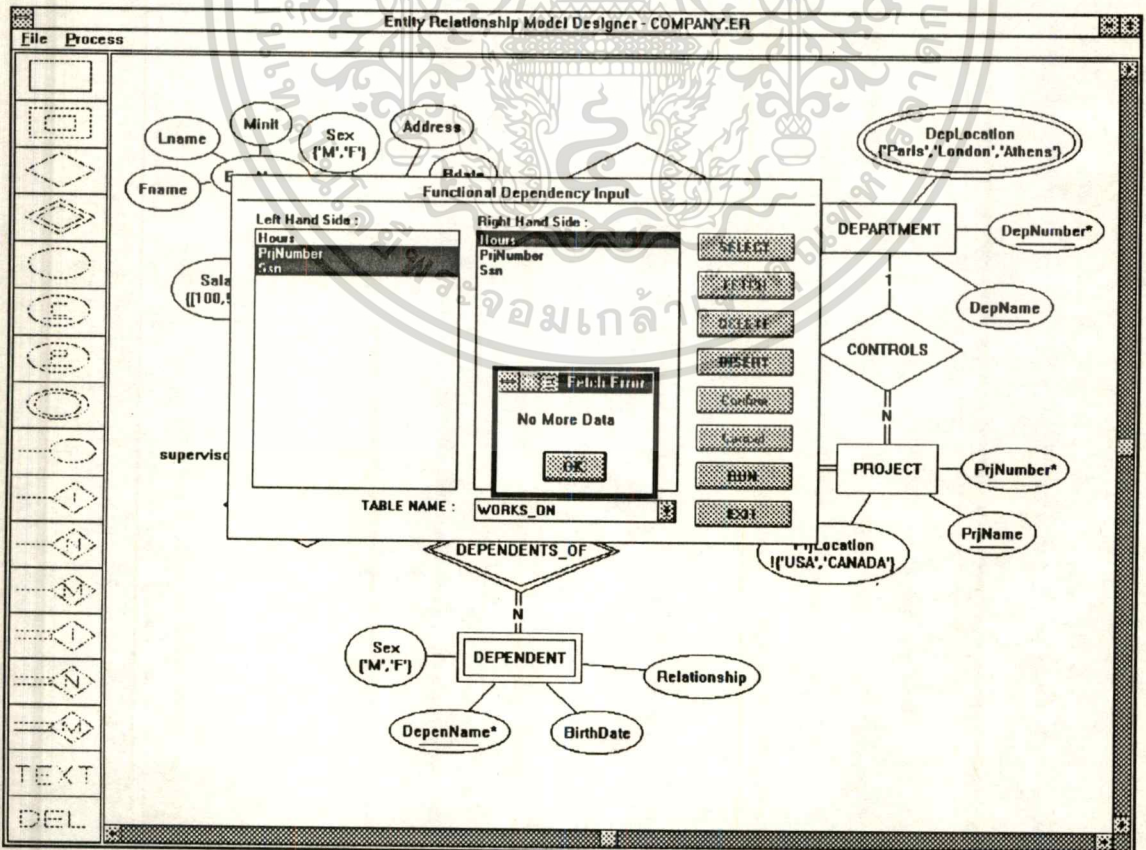


รูปที่ A52 แสดงความสัมพันธ์ระหว่างแอททริบิวต์ (PrjNumber,Ssn) → (PrjNumber) ของตาราง WORKS\_ON ด้านการค่า

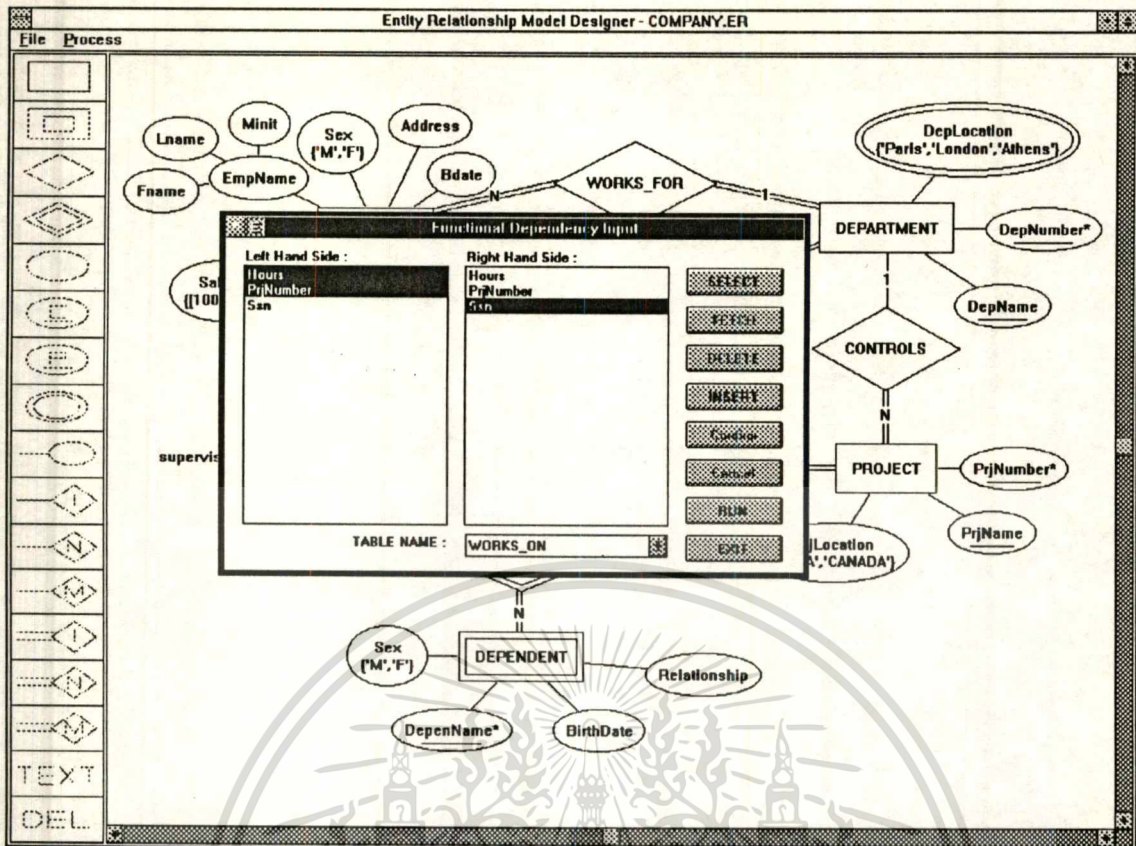
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



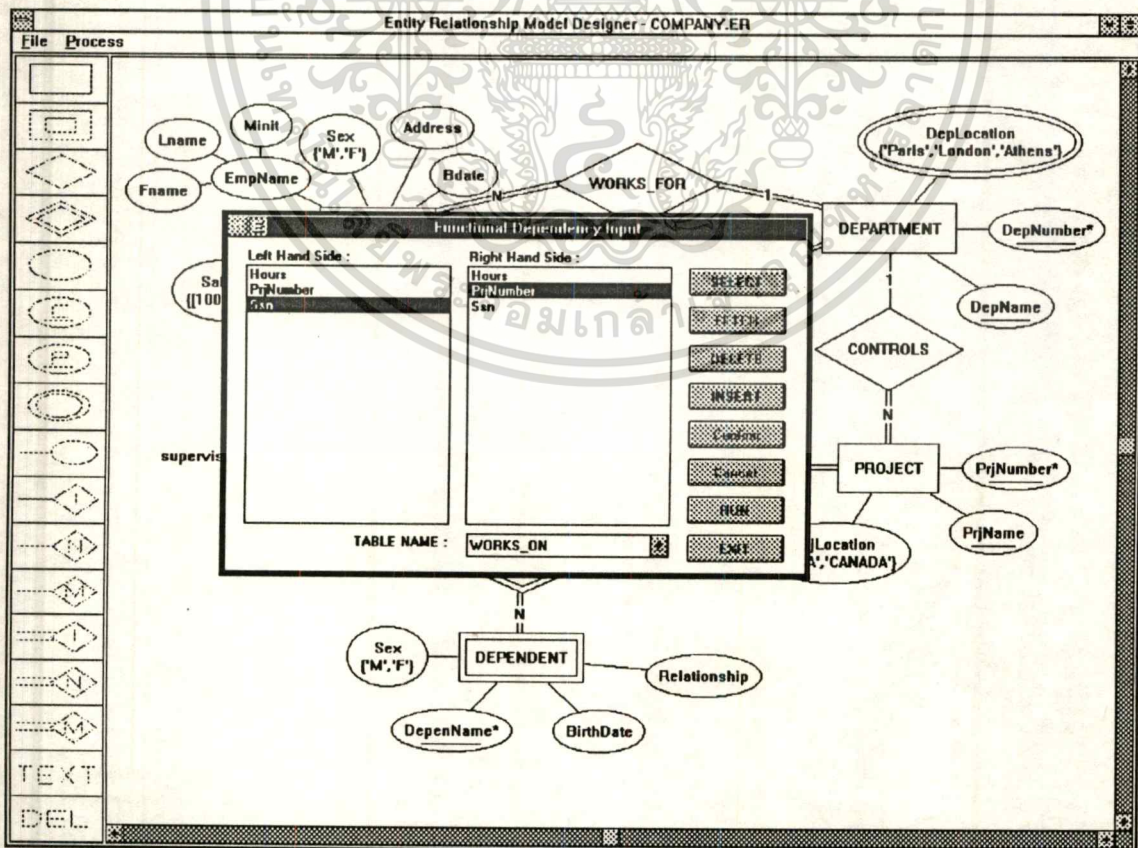
รูปที่ A53 แสดงความสัมพันธ์ระหว่างแอททริบิวต์ (PrjNumber,Ssn) → (Hours) ของตาราง WORKS\_ON



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และข้อมูลความสัมพันธ์ระหว่างแอททริบิวต์ของตาราง WORKS\_ON ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

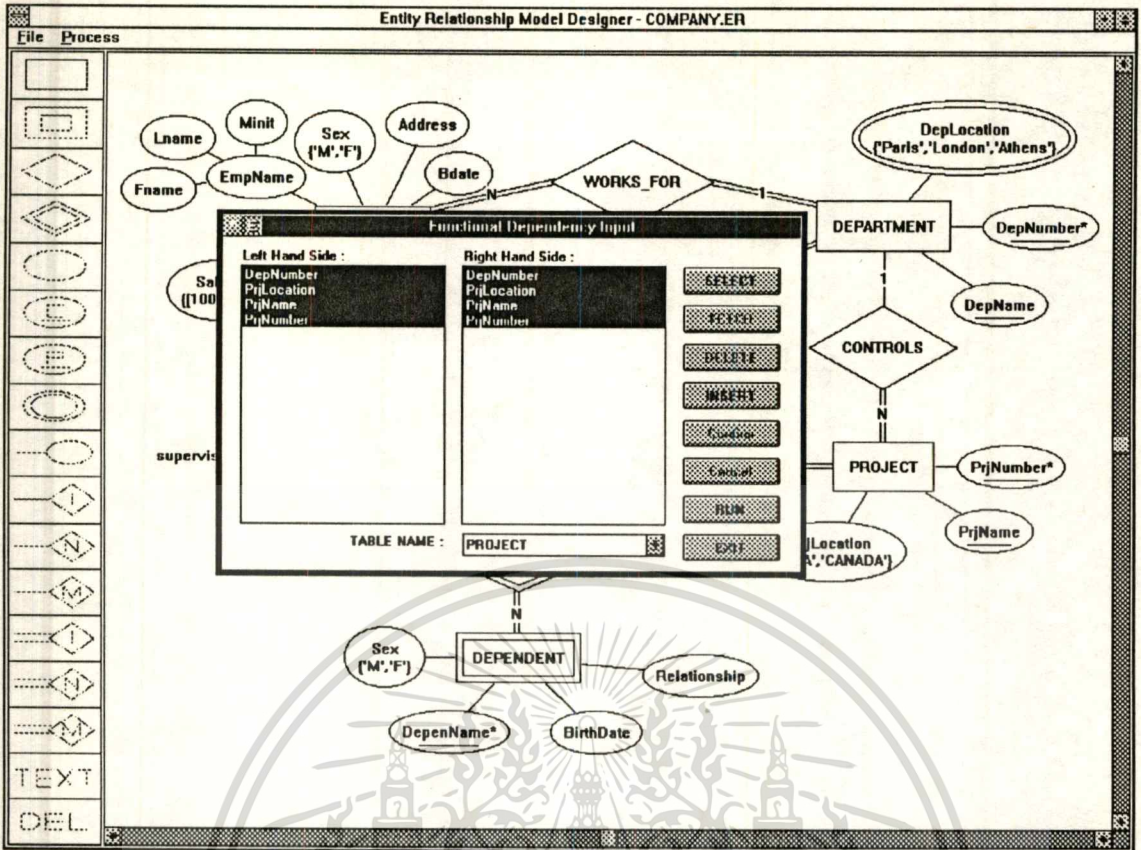


รูปที่ A55 แสดงการเพิ่มความสัมพันธ์ระหว่างแอททริบิวต์ (Hours, PrjNumber) → (Ssn) ลงในตาราง WORKS\_ON



รูปที่ A56 แสดงการเพิ่มความสัมพันธ์ระหว่างแอททริบิวต์ (Ssn) → (PrjNumber) ลงในตาราง WORKS\_ON

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลเนื้อหาใช้ระบบออนไลน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ A57 แสดงการเพิ่มความสัมพันธ์ระหว่างแอททริบิวต์ (DepNumber,PrjLocation,PrjName,PrjNumber) → (DepNumber,PrjLocation,PrjName,PrjNumber) ลงในตาราง PROJECT

```

CREATE TABLE DEPARTMENT
(
    Ssn          NUMBER,
    DepNumber   NUMBER          NOT NULL,
    DepName     CHAR(30),
    StartDate   DATE
);
CREATE TABLE DEPENDENT
(
    Sex         CHAR(1),
    DepenName   CHAR(30)       NOT NULL,
    BirthDate   DATE,
    Relationship CHAR(30),
    Ssn         NUMBER        NOT NULL
);
CREATE TABLE DepLocation
(
    DepNumber   NUMBER          NOT NULL,
    DepLocation CHAR(30)       NOT NULL
);
CREATE TABLE EMPLOYEE
(
    Fname       CHAR(1),
    Lname       CHAR(10),
    Minit       CHAR(1),
    Ssn         NUMBER          NOT NULL,
    Salary      NUMBER,
    Sex         CHAR(1),
    Address     CHAR(30),
    Bdate       DATE,
    DepNumber   NUMBER,
    Ssn23       NUMBER
);

```

ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(
    DepNumber      NUMBER      NOT NULL,
    PrjNumber      NUMBER      NOT NULL,
    PrjName        CHAR(30)    NOT NULL,
    PrjLocation    CHAR(30)    NOT NULL
);
CREATE TABLE WORKS_ON#1
(
    Ssn            NUMBER      NOT NULL,
    Hours         DATE        NOT NULL
);
CREATE TABLE WORKS_ON#2
(
    Ssn            NUMBER      NOT NULL,
    PrjNumber     NUMBER
);
CREATE UNIQUE INDEX INDEX_#1 ON DEPARTMENT (DepNumber) ;
CREATE UNIQUE INDEX INDEX_#2 ON DEPENDENT (DepenName,Ssn) ;
CREATE UNIQUE INDEX INDEX_#3 ON DepLocation (DepNumber,DepLocation) ;
CREATE UNIQUE INDEX INDEX_#4 ON EMPLOYEE (Ssn) ;
CREATE UNIQUE INDEX INDEX_#5 ON PROJECT (DepNumber,PrjNumber,PrjName,PrjLocation) ;
CREATE UNIQUE INDEX INDEX_#6 ON WORKS_ON#1 (Ssn,Hours) ;
CREATE UNIQUE INDEX INDEX_#7 ON WORKS_ON#2 (Ssn) ;

CREATE VIEW DEPENDENT_VIEW AS SELECT *
FROM DEPENDENT
WHERE Sex IN ('F','M')
WITH CHECK OPTION ;

CREATE VIEW DepLocation_VIEW AS SELECT *
FROM DepLocation
WHERE DepLocation IN ('Athens','London','Paris')
WITH CHECK OPTION ;

CREATE VIEW EMPLOYEE_VIEW AS SELECT *
FROM EMPLOYEE
WHERE Sex IN ('F','M') AND ((Salary >= 100 AND Salary <= 500))
WITH CHECK OPTION ;

CREATE VIEW PROJECT_VIEW AS SELECT *
FROM PROJECT
WHERE PrjLocation NOT IN ('CANADA','USA')
WITH CHECK OPTION ;

```

รูปที่ A58 แสดงภาษาโครงสร้างที่ใช้ในการสร้างตารางฐานข้อมูลในบรรทัดฐานชั้น Boyce/Codd (BCNF.SQL)

- การสร้างตารางฐานข้อมูลในบรรทัดฐานชั้นที่ห้า

เมื่อผู้ใช้งานได้ตารางฐานข้อมูลในบรรทัดฐานชั้น Boyce/Codd แล้วยังต้องการที่จะทำตารางฐานข้อมูลที่ได้นั้นให้สูงขึ้นไปอยู่ในบรรทัดฐานชั้นที่ห้าให้ผู้ใช้งานเลือกเมนู Fifth Normal Form จากเมนูหลัก Process ดังแสดงในรูปที่ A59 หลังจากที่ได้เลือกแล้วจะปรากฏกรอบข้อความขึ้นดังแสดงในรูปที่ A60 ภายในกรอบข้อความจะประกอบด้วย

1. กรอบชื่อตาราง (TABLE NAME)

กรอบชื่อตารางจะเก็บชื่อของตารางที่มีความเป็นไปได้ที่จะมีความสัมพันธ์เจตึแอบแฝงอยู่จากรูปที่ A60 ตารางที่มีความเป็นไปได้ที่จะมีความสัมพันธ์เจตึแอบแฝงอยู่มี 1 ตาราง คือ ตาราง PROJECT ผู้ใช้งานสามารถที่จะเลือกตาราง PROJECT มาเพิ่มเติมความสัมพันธ์เจตึดี (Join Dependency) ได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. แอททริบิวต์

แอททริบิวต์แสดงถึงแอททริบิวต์ทั้งหมดของตารางที่ถูกเลือกซึ่งจะเห็นว่าตาราง PROJECT นั้นประกอบด้วยแอททริบิวต์ DepNumber PrjLocation PrjName และ PrjNumber

## 3. ปุ่ม SELECT

ใช้ในการเปิดเคอร์เซอร์ของฐานข้อมูลความสัมพันธ์เจดี เมื่อผู้ใช้งานกดปุ่มนี้ ข้อมูลความสัมพันธ์เจดีทั้งหมดของตาราง PROJECT จะถูกเก็บลงในเคอร์เซอร์ของตัวจัดการฐานข้อมูล

## 4. ปุ่ม FETCH

ใช้ในการอ่านข้อมูลของฐานข้อมูลความสัมพันธ์เจดี เมื่อผู้ใช้งานกดปุ่มนี้หลังจากที่เคอร์เซอร์ของตัวจัดการฐานข้อมูลถูกเปิดขึ้นแล้วจากการกดปุ่ม SELECT ข้อมูลของความสัมพันธ์เจดีของตาราง PROJECT ก็จะถูกแสดงขึ้นดัง

## 5. ปุ่ม DELETE

ใช้ในการลบความสัมพันธ์เจดี เมื่อผู้ใช้งานกดปุ่มนี้ความสัมพันธ์เจดีที่กำลังถูกแสดงอยู่ขณะนั้นก็จะถูกลบออกจากฐานข้อมูลแล้ว FETCH ข้อมูลความสัมพันธ์เจดีตัวถัดไปมาแสดง

## 6. ปุ่ม INSERT

ใช้ในการเพิ่มความสัมพันธ์เจดี เมื่อผู้ใช้งานกดปุ่มนี้กรอบข้อความแอททริบิวต์จะเคลียร์เป็นสีขาวให้ผู้ใช้งานเลื่อนเมาท์ไปยังกรอบข้อความแอททริบิวต์และเลือกแอททริบิวต์ที่ต้องการภายในกรอบข้อความนั้นโดยการกดที่ปุ่มซ้ายของเมาท์ซึ่งการเลือกแอททริบิวต์นั้นสามารถเลือกได้มากกว่าหนึ่งแอททริบิวต์โดยแอททริบิวต์ที่ถูกเลือกนั้นจะมีแถบสีเกิดขึ้นดังแสดงในรูปที่ A61

## 7. ปุ่ม Confirm

ใช้ในการยืนยันการเพิ่มความสัมพันธ์เจดี เมื่อผู้ใช้งานกดปุ่มนี้ความสัมพันธ์เจดีที่เลือกในขั้นตอนที่ 6 ก็จะถูกเก็บลงบนฐานข้อมูล จากรูปที่ A61 A62 และ A63 แสดงการเพิ่มความสัมพันธ์เจดีดังต่อไปนี้คือ

### ตาราง PROJECT

JD\* ( (DepNumber,PrjLocation,PrjName),(DepNumber,PrjLocation,PrjName),(PrjName,PrjNumber) )

## 8. ปุ่ม Cancel

ใช้ในการยกเลิกการเพิ่มความสัมพันธ์เจดี เมื่อผู้ใช้งานกดปุ่มนี้ความสัมพันธ์เจดีที่เลือกในขั้นตอนที่ 7 ก็จะไม่ถูกเก็บลงบนฐานข้อมูล

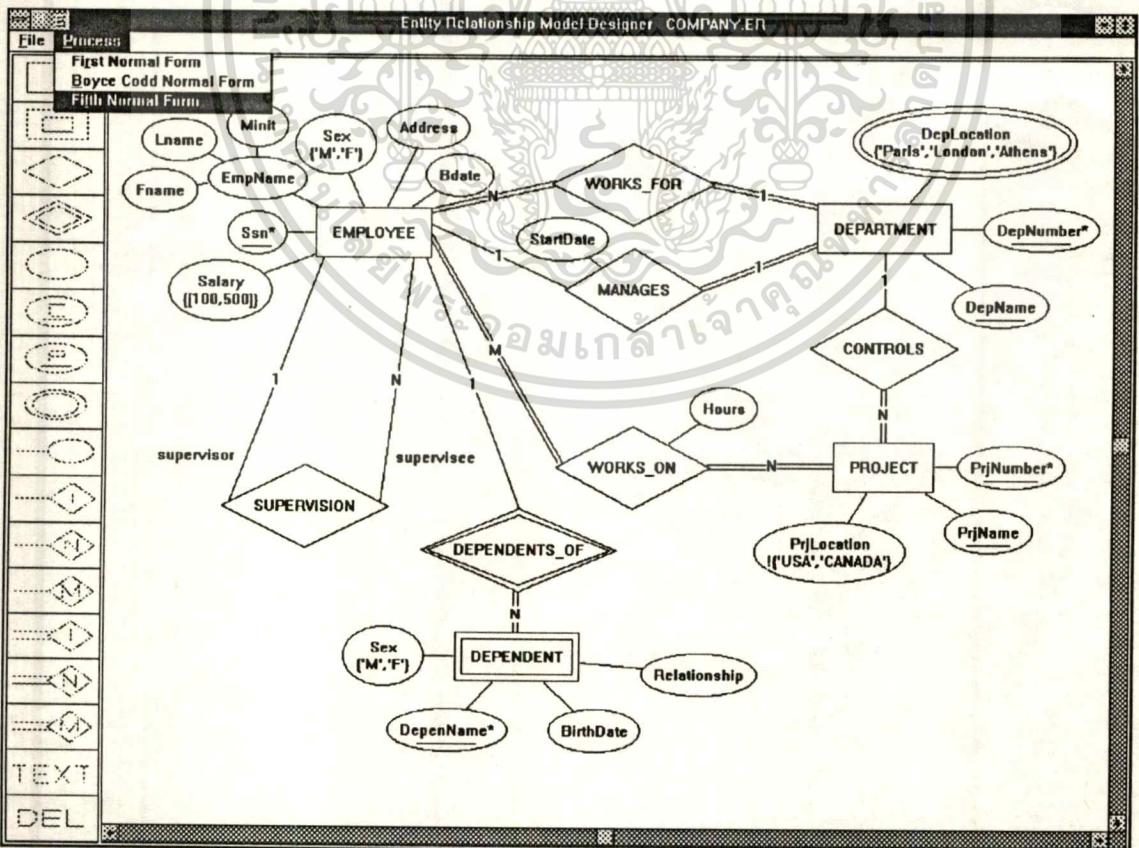
## 9. ปุ่ม RUN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ในการประมวลผลความสัมพันธ์เจตึ เพื่อสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นที่ห้า เมื่อผู้ใช้งานกดปุ่มนี้เคอร์เซอร์เมาท์จะเปลี่ยนจากรูปลูกศรสีขาวเป็นรูปนาฬิกาทรายเพื่อให้ผู้ใช้งานทราบว่ากำลังทำการประมวลผลข้อมูลอยู่ให้รอและเมื่อเคอร์เซอร์เมาท์เปลี่ยนจากรูปนาฬิกาทรายกลับมาเป็นรูปลูกศรสีขาวเป็นดังเดิมแสดงว่าการประมวลผลข้อมูลได้เสร็จสิ้นลงแต่ถ้าในระหว่างการประมวลผลมีรอบข้อความเกิดขึ้นแสดงว่าเกิดการผิดพลาดขึ้นที่ความสัมพันธ์เจตึให้จำคำอธิบายความผิดพลาดในรอบข้อความนั้นเมื่อจำได้แล้วให้ปิดกรอบข้อความนั้นโดยกดปุ่ม OK และกลับไปแก้ไขความสัมพันธ์เจตึใหม่ให้ถูกต้องตามข้อผิดพลาดที่อธิบายไว้ภายในกรอบข้อความและเมื่อแน่ใจว่าได้แก้ไขความสัมพันธ์เจตึถูกต้องเรียบร้อยแล้วให้ปฏิบัติตามขั้นตอนที่กล่าวมาแล้วข้างต้นใหม่อีกครั้ง ผลลัพธ์ที่ได้จะเป็นคำสั่งในการสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นที่ห้าและถูกเก็บลงในไฟล์ FIFTH.SQL ดังแสดงในรูปที่ A64 และสามารถนำไปใช้งานได้ทันทีและกรอบข้อความที่แสดงในรูปที่ A60 จะหายไปและกลับไปยังการวาดแผนภาพอีอาร์ตามเดิม

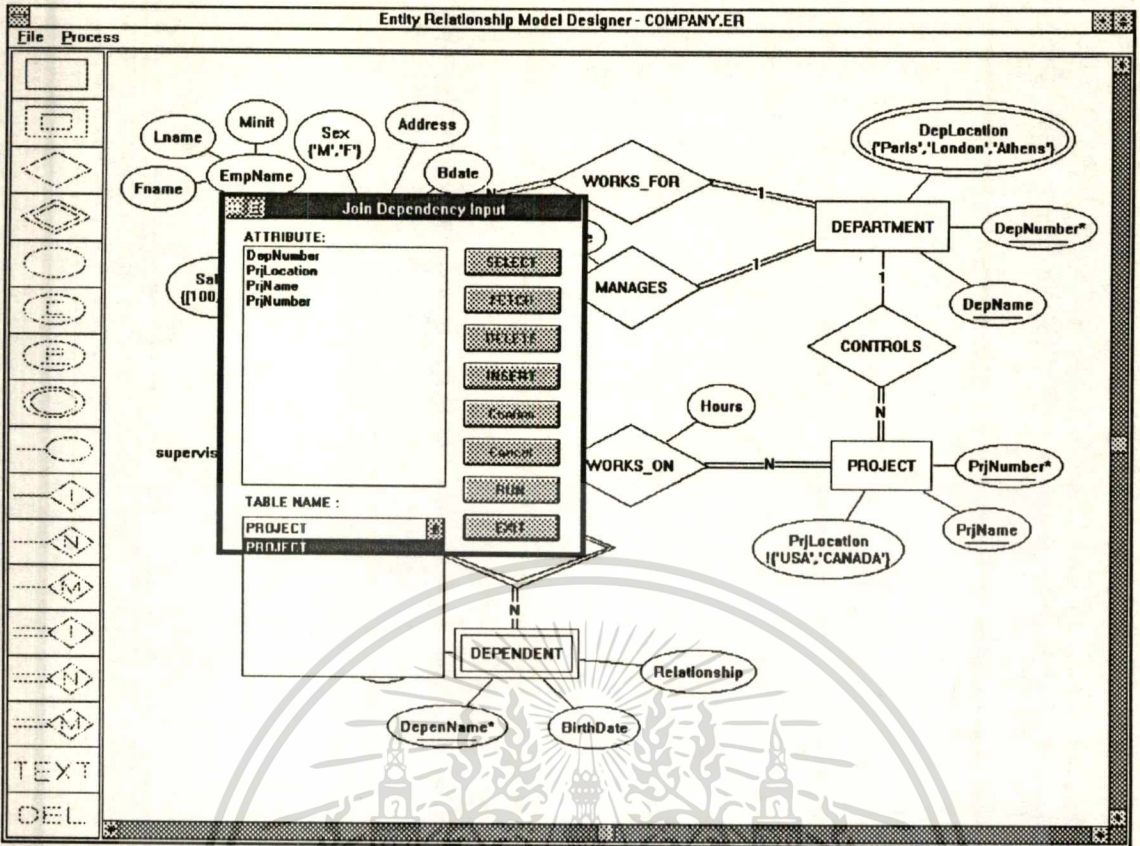
10.ปุ่ม EXIT

ใช้ในการออกจากการประมวลผลขั้นที่ห้า เมื่อผู้ใช้งานกดปุ่มนี้กรอบข้อความจะหายไปโดยไม่มีการประมวลผลใดๆเกิดขึ้นและกลับไปยังการวาดแผนภาพอีอาร์ตามเดิม

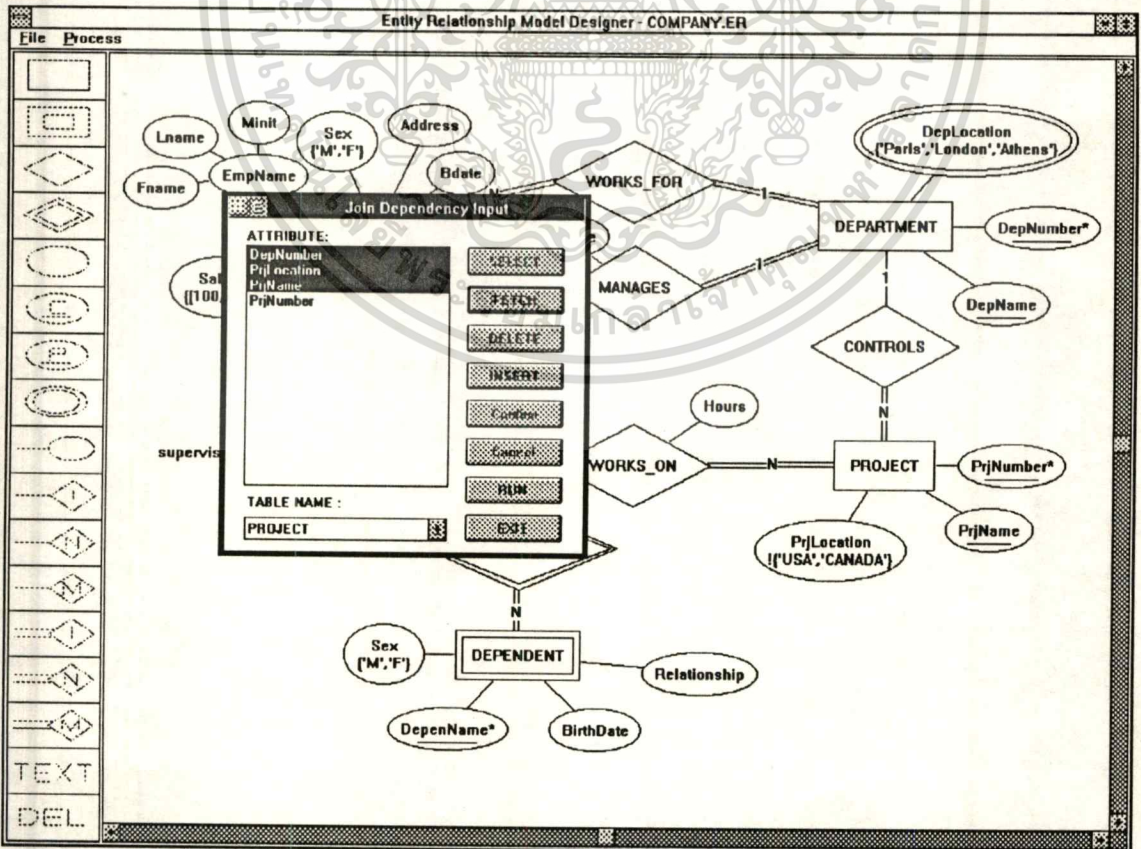


รูปที่ A59 แสดงการเลือกเมนู Fifth Normal Form จากเมนูหลัก Process

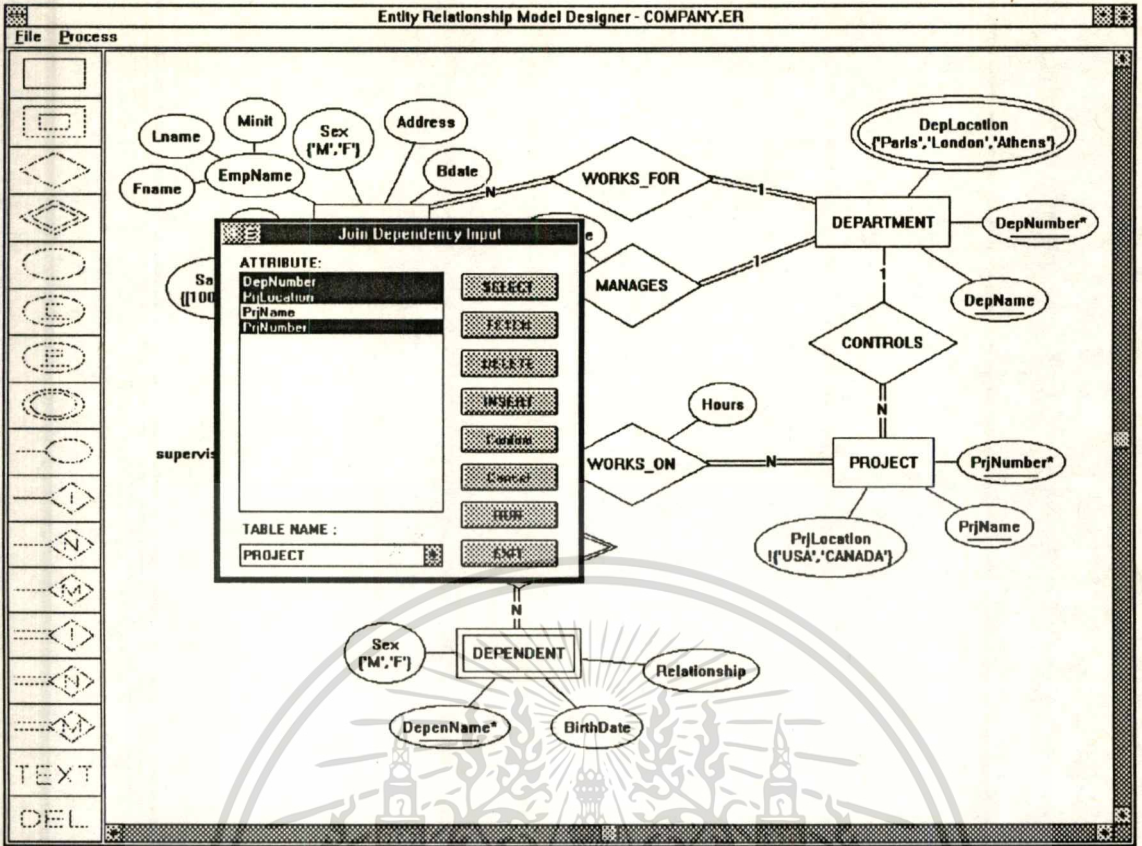
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



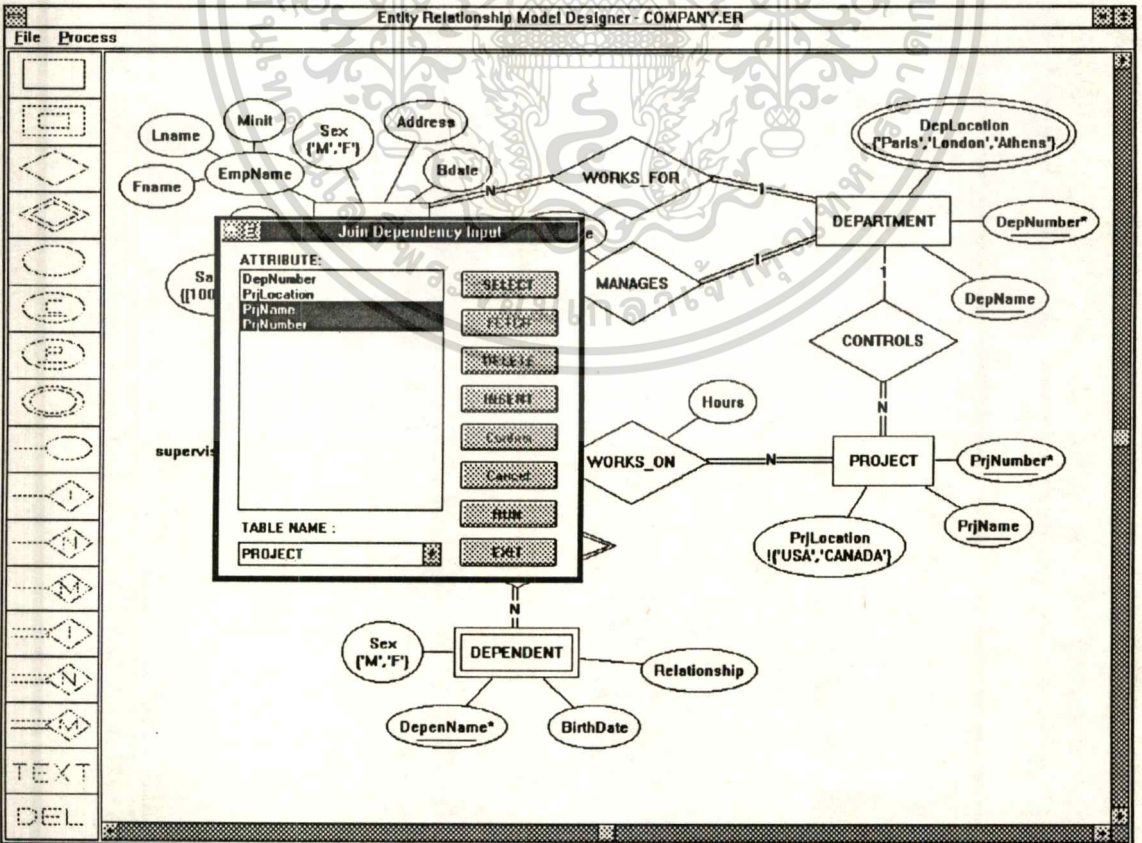
รูปที่ A60 แสดงกรอบข้อความของการประมวลผลขั้น Fifth Normal Form



รูปที่ A61 แสดงการเพิ่มความสัมพันธ์เจดี JD\*((DepNumber,PrjLocation,PrjName)) ลงในตาราง PROJECT ด้านการคำนวณ  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ A62 แสดงการเพิ่มความสัมพันธ์เจต JD\*((DepNumber,PrjLocation,PrjNumber) ลงในตาราง PROJECT



เอกสารนี้รูปที่ A63 แสดงการเพิ่มความสัมพันธ์เจต JD\*((PrjName,PrjNumber) ลงในตาราง PROJECT ระบุข้อดำเนินการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CREATE TABLE DEPARTMENT
(
    Ssn          NUMBER,
    DepNumber    NUMBER      NOT NULL,
    DepName      CHAR(30),
    StartDate    DATE
);
CREATE TABLE DEPENDENT
(
    Sex          CHAR(1),
    DepenName    CHAR(30)    NOT NULL,
    BirthDate    DATE,
    Relationship  CHAR(30),
    Ssn          NUMBER      NOT NULL
);
CREATE TABLE DepLocation
(
    DepNumber    NUMBER      NOT NULL,
    DepLocation  CHAR(30)    NOT NULL
);
CREATE TABLE EMPLOYEE
(
    Fname        CHAR(1),
    Lname        CHAR(10),
    Minit        CHAR(1),
    Ssn          NUMBER      NOT NULL,
    Salary       NUMBER,
    Sex          CHAR(1),
    Address      CHAR(30),
    Bdate        DATE,
    DepNumber    NUMBER,
    Ssn23        NUMBER
);
CREATE TABLE PROJECT#1
(
    DepNumber    NUMBER      NOT NULL,
    PrjName      CHAR(30)    NOT NULL,
    PrjLocation  CHAR(30)    NOT NULL
);
CREATE TABLE PROJECT#2
(
    DepNumber    NUMBER      NOT NULL,
    PrjNumber    NUMBER      NOT NULL,
    PrjLocation  CHAR(30)    NOT NULL
);
CREATE TABLE PROJECT#3
(
    PrjNumber    NUMBER      NOT NULL,
    PrjName      CHAR(30)    NOT NULL
);
CREATE TABLE WORKS_ON#1
(
    Ssn          NUMBER      NOT NULL,
    Hours        DATE        NOT NULL
);
CREATE TABLE WORKS_ON#2
(
    Ssn          NUMBER      NOT NULL,
    PrjNumber    NUMBER
);
CREATE UNIQUE INDEX INDEX_#1 ON DEPARTMENT (DepNumber) ;
CREATE UNIQUE INDEX INDEX_#2 ON DEPENDENT (DepenName,Ssn) ;
CREATE UNIQUE INDEX INDEX_#3 ON DepLocation (DepNumber,DepLocation) ;
CREATE UNIQUE INDEX INDEX_#4 ON EMPLOYEE (Ssn) ;
CREATE UNIQUE INDEX INDEX_#6 ON WORKS_ON#1 (Ssn,Hours) ;
CREATE UNIQUE INDEX INDEX_#7 ON WORKS_ON#2 (Ssn) ;
CREATE UNIQUE INDEX INDEX_#8 ON PROJECT#1 (DepNumber,PrjName,PrjLocation) ;
CREATE UNIQUE INDEX INDEX_#9 ON PROJECT#2 (DepNumber,PrjNumber,PrjLocation) ;
CREATE UNIQUE INDEX INDEX_#10 ON PROJECT#3 (PrjNumber,PrjName) ;

```

```

CREATE VIEW DEPENDENT_VIEW AS SELECT *
FROM DEPENDENT
WHERE Sex IN ('F','M')
WITH CHECK OPTION ;

```

เอาไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CREATE VIEW DepLocation_VIEW AS SELECT *
FROM DepLocation
WHERE DepLocation IN ('Athens','London','Paris')
WITH CHECK OPTION ;
```

```
CREATE VIEW EMPLOYEE_VIEW AS SELECT *
FROM EMPLOYEE
WHERE Sex IN ('F','M') AND ((Salary >= 100 AND Salary <= 500))
WITH CHECK OPTION ;
```

```
CREATE VIEW PROJECT#1_VIEW AS SELECT *
FROM PROJECT#1
WHERE PrjLocation NOT IN ('CANADA','USA')
WITH CHECK OPTION ;
```

```
CREATE VIEW PROJECT#2_VIEW AS SELECT *
FROM PROJECT#2
WHERE PrjLocation NOT IN ('CANADA','USA')
WITH CHECK OPTION ;
```

รูปที่ A64 แสดงภาษาโครงสร้างที่ใช้ในการสร้างตารางฐานข้อมูลในบรรทัดฐานขั้นที่ห้า (FIFTH.SQL)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ข โปรแกรมระบบ ER/5

```
.....
Filename: ERFIVEPC.MAK
.....
PCC = PROC INCLUDE=ORACLE6$PROCL IRECLEN=132 ORECLN=132 MAXOPENCURSORS=20 LTYPE=NONE
OBS_EXT = MAINWND.OBJ DRAWWND.OBJ TOOLWND.OBJ ERTOOL.OBJ PRTRFILE.OBJ PROCESS1.OBJ PROCESS2.OBJ
PROCESS3.OBJ PROCESS4.OBJ
CFLAGS = /nologo /G2 /W3 /AL /Od /Zpe /Gs
LFLAGS = /NOLOGO /NOB /PACKC:61440 /ALIGN:16 /ONERROR:NOEXE
LIBS = OLDNAMES LIBW LLIBCEW SQL13WIN.LIB COMMDLG.LIB OLECLI.LIB OLESVR.LIB SHELL.LIB
RCDEFINES = $(D_RCDEFINES)
!if !if exist MSVC.BND del MSVC.BND
endif
ERFIVE_DEP = C:\MSVCSAMPLES\ERFIVE\ERGLOBAL.INC\
C:\MSVCSAMPLES\ERFIVE\SQLPROTO.H
ERFIVE_RCDEP = C:\MSVCSAMPLES\ERFIVE\ERGLOBAL.INC\
C:\MSVCSAMPLES\ERFIVE\ERFIVEPC.IC0
all: ERFIVEPC.EXE
ERFIVEPC.C: ERFIVEPC.PC ERGLOBAL.INC
$(PCC) INAME=ERFIVEPC.PC
ERFIVEPC.OBJ: ERFIVEPC.C $(ERFIVE_DEP)
CL $(CFLAGS) /c ERFIVEPC.C
MAINWND.C: MAINWND.PC EREXTN.INC ERFIVEPC.PC
$(PCC) INAME=MAINWND.PC
MAINWND.OBJ: MAINWND.C $(ERFIVE_DEP)
CL $(CFLAGS) /c MAINWND.C
TOOLWND.C: TOOLWND.PC EREXTN.INC ERFIVEPC.PC
$(PCC) INAME=TOOLWND.PC
TOOLWND.OBJ: TOOLWND.C $(ERFIVE_DEP)
CL $(CFLAGS) /c TOOLWND.C
DRAWWND.C: DRAWWND.PC EREXTN.INC ERFIVEPC.PC
$(PCC) INAME=DRAWWND.PC
DRAWWND.OBJ: DRAWWND.C $(ERFIVE_DEP)
CL $(CFLAGS) /c DRAWWND.C
PROCESS2.C: PROCESS2.PC EREXTN.INC ERFIVEPC.PC
$(PCC) INAME=PROCESS2.PC
PROCESS2.OBJ: PROCESS2.C $(ERFIVE_DEP)
CL $(CFLAGS) /c PROCESS2.C
PROCESS3.C: PROCESS3.PC EREXTN.INC ERFIVEPC.PC
$(PCC) INAME=PROCESS3.PC
PROCESS3.OBJ: PROCESS3.C $(ERFIVE_DEP)
CL $(CFLAGS) /c PROCESS3.C
PROCESS4.C: PROCESS4.PC EREXTN.INC ERFIVEPC.PC
$(PCC) INAME=PROCESS4.PC
PROCESS4.OBJ: PROCESS4.C $(ERFIVE_DEP)
CL $(CFLAGS) /c PROCESS4.C
ERTOOL.C: ERTOOL.PC EREXTN.INC ERFIVEPC.PC
$(PCC) INAME=ERTOOL.PC
ERTOOL.OBJ: ERTOOL.C $(ERFIVE_DEP)
CL $(CFLAGS) /c ERTOOL.C
PRTRFILE.C: PRTRFILE.PC EREXTN.INC ERFIVEPC.PC
$(PCC) INAME=PRTRFILE.PC
PRTRFILE.OBJ: PRTRFILE.C $(ERFIVE_DEP)
CL $(CFLAGS) /c PRTRFILE.C
PROCESS1.C: PROCESS1.PC EREXTN.INC ERFIVEPC.PC
$(PCC) INAME=PROCESS1.PC
PROCESS1.OBJ: PROCESS1.C $(ERFIVE_DEP)
CL $(CFLAGS) /c PROCESS1.C
ERFIVEPC.RES: ERFIVEPC.RC $(ERFIVE_RCDEP)
RC /NOLOGO $(RCDEFINES) -r ERFIVEPC.RC

ERFIVEPC.EXE: ERFIVEPC.RES
ERFIVEPC.EXE: ERFIVEPC.OBJ $(OBS_EXT) ERFIVEPC.DEF
echo >NUL @<<ERFIVEPC.CRF
ERFIVEPC.OBJ +
$(OBS_EXT)
ERFIVEPC.EXE
nul
C:\MSVCLIB+ C:\MSVCMFCLIB+ C:\ORACLE6$PROLIB+
$(LIBS)
ERFIVEPC.DEF;
<<
LINK $(LFLAGS) @ERFIVEPC.CRF
RC $(RESFLAGS) ERFIVEPC.RES
.....
```

```
.....
Filename: ERCONST.INC
.....
#define MAXFILENAME 256 /* maximum length of file pathname */
#define MAXCUSTFILTER 40 /* maximum size of custom filter buffer */
#define ERROR 0
#define NOERROR 1
#define ON 0
#define OFF 1
#define NO_CONSTRAINT 0
#define CONSTRAINT_CHAR 1
#define CONSTRAINT_NUMBER 2
#define MAXROLE 10
#define MAXTOOL 17
#define R_ENTITY 0
#define W_ENTITY 1
#define N_RELATION 2
#define I_RELATION 3
#define N_ATTRIBUTE 4
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define K_ATTRIBUTE 5
#define P_ATTRIBUTE 6
#define MV_ATTRIBUTE 7
#define LINK 8
#define PARTIAL_1 9
#define PARTIAL_N 10
#define PARTIAL_M 11
#define TOTAL_1 12
#define TOTAL_N 13
#define TOTAL_M 14
#define TEXT 15
#define INDICATOR 16
#define NO_OBJECT 100
#define V_SCROLL_RANGE 1000
#define H_SCROLL_RANGE 1000
#define EDIT 100
#define MOVE 200
#define MIN 0
#define MAX 1
#define BORDER 6
#define H_OFFSET 15
#define W_OFFSET 15
#define ER 0
#define FIRST_NF 1
#define BCNF 2
#define FIFTH_NF 3
#define OUTPUT 4
typedef struct {
    HRGN hRgn ;
    int ObjectType ;
    char ObjectName[30] ;
    char Constraint[30] ;
    int MetaStartRec ;
    int MetaRecLength ;
    int Number ;
    HGLOBAL Next,Prev ;
} NODE ;

```

Filename: ERGLOBAL.INC

```

#include <windows.h>
#include "resource.h"
#include <commdlg.h>
#include <math.h>
#include <lo.h>
#include <ermo.h>
#include <stdio.h>
#include <stdlib.h>
#include <print.h>
#include <string.h>
#include "erconst.inc"
#include "erproto.inc"
#include "sqlproto.h"
#include <systypes.h>
#include <sysstat.h>
HANDLE hInst ;
HRGN hDisplayRgn ;
HRGN hEmptyRgn ;
HWND hMainWnd ;
HWND hToolWnd ;
HWND hDrawWnd ; /* handle to main window */
HPEN hOldPen ;
HPEN hBlackSolidPen ;
HPEN hGreenSolidPen ;
HPEN hBlueSolidPen ;
RECT MainWndRect ;
RECT ToolWndRect ;
RECT DrawWndRect ;
RECT ToolMatrix[MAXTOOL] ;
HCURSOR hOldCursor ;
HCURSOR hWaitCursor ;
HCURSOR hPencilCursor ;
HCURSOR hCrossCursor ;
HCURSOR hArrowCursor ;
HCURSOR hAsclCursor ;
HCURSOR hIndicatorCursor ;
POINT FromPoint,ToPoint ;
POINT FirstObjectPoint ;
POINT SecondObjectPoint ;
BOOL fNameFile ;
BOOL fPtInRgn ;
BOOL CursorOpen ;
int xPosition,yPosition ;
int HorzThumbPos ; /* Indicate Viewport X coordinate of DrawWnd */
int VertThumbPos ; /* Indicate Viewport Y coordinate of DrawWnd */
int SelectToolItem ; /* Indicate Tool Number Selected */
int CurrentMetaRec ;
int MetaRecCount ;
int ConstraintFlag ;
void (*ToolProc[]) () = {

```

```

R_ENTITY_PROC,
W_ENTITY_PROC,
N_RELATION_PROC,
I_RELATION_PROC,
N_ATTRIBUTE_PROC,
K_ATTRIBUTE_PROC,
P_ATTRIBUTE_PROC,
MV_ATTRIBUTE_PROC,
LINK_PROC,

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARTIAL\_1\_PROC,  
 PARTIAL\_N\_PROC,  
 PARTIAL\_M\_PROC,  
 TOTAL\_1\_PROC,  
 TOTAL\_N\_PROC,  
 TOTAL\_M\_PROC,  
 TEXT\_PROC,  
 INDICATOR\_PROC

```

};
struct {
    HDC hMetaDC;
} ErMetaFile;

HGLOBAL FIRST = NULL;
HGLOBAL LAST = NULL;
char NameBuffer[30];
char DataTypeBuffer[20];
char DomainBuffer[120];
char TempBuffer[120];
OPENFILENAME ofn;
char szFilterSpec [128] = "ER Files (*.ER)\0*.ER\0All Files (*.*)\0*. *";
char szFileName[MAXFILENAME];
char LinkListFileName[MAXFILENAME];
char OracleFileName[MAXFILENAME];
char szFileTitle[MAXFILENAME];
char PrintFileName[MAXFILENAME];
char szBaseWindowTitle[128] = "Entity Relationship Model Designer";
char szWindowTitle[80];
PRINTDLG pd;
char SaveFileName[30]; /* Common print dialog structure */
HDC hPr; /* handle for printer device context */
BOOL bAbort; /* FALSE if user cancels printing */
HWND hAbortDlgWnd;
FARPROC lpAbortDlg,lpAbortProc;
WORD IOStatus; /* result of file i/o */
int Status;
int nPageSize; /* vert. resolution of printer device */
BOOL FdFlag = FALSE;
BOOL JdFlag = FALSE;
BOOL ErrorFlag = NOERROR;
BOOL DisplayFlag = ON;
char NumberIndex[200];
char NumberTable[] = {
    '0','1','2','3','4','5','6','7',
    '8','9','(',')','(',')','T',
    '\t','\n','\r','\f','\b','\a'
};

struct {
    char Name[30];
    int Type;
} Entity,Relation;

struct {
    int Attr_Number;
    char Name[30];
    int Type;
    char DataType[20];
    char Entity_Name[30];
    char Relation_Name[30];
    int Composit_Number;
} Attr;

struct {
    int Role_Number;
    int Participation;
    int Cadinality;
    char Entity_Name[30];
    char Relation_Name[30];
} Role;

struct {
    int Attr_Number;
    char Domain_Name[30];
} cDomain;

struct {
    int Attr_Number;
    char Lower_Limit[2];
    char Min[30];
    char Max[30];
    char Upper_Limit[2];
} nDomain;

struct {
    int Attr_Number;
    char invert[2];
} cinvert;

struct {
    int Fd_Number;
    char Table_Name[30];
} Fd;

struct {
    int Fd_Number;
    int Attr_Number;
} LHS,RHS;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        int   Index_Number ;
        int   Attr_Number ;
    } Temp_Attr_of_Index,Attr_of_Index ;

struct
    {
        int   Index_Number ;
        char  Table_Name[30] ;
    } Temp_Table_of_Index,Table_of_Index ;

struct
    {
        char  Table_Name[30] ;
        int   Attr_Number ;
        char  Not_Null[2] ;
    } Temp_Attr_of_Table,Attr_of_Table ;

struct
    {
        int   Attr_Number ;
        char  Attr_Name[30] ;
        char  DataType[20] ;
    } Attr_Detail ;

```

Filename: EREXTERN.INC

```

#include <windows.h>
#include "resource.h"
#include <comdlg.h>
#include <math.h>
#include <io.h>
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include <print.h>
#include <string.h>
#include "erconst.inc"
#include "erproto.inc"
#include "sqlproto.h"
#include <sys/types.h>
#include <sys/stat.h>
extern HANDLE hInst;
extern HRGN hDisplayRgn ;
extern HRGN hEmptyRgn ;
extern HWND hMainWnd ;
extern HWND hToolWnd ;
extern HWND hDrawWnd ; /* handle to main window */
extern HPEN hOldPen ;
extern HPEN hBlackSolidPen ;
extern HPEN hGreenSolidPen ;
extern HPEN hBlueSolidPen ;
extern RECT MainWndRect ;
extern RECT ToolWndRect ;
extern RECT DrawWndRect ;
extern RECT ToolMatrix[MAXTOOL] ;
extern HCURSOR hOldCursor ;
extern HCURSOR hWaitCursor ;
extern HCURSOR hPencilCursor ;
extern HCURSOR hCrossCursor ;
extern HCURSOR hArrowCursor ;
extern HCURSOR hAsciiCursor ;
extern HCURSOR hIndicatorCursor ;
extern POINT FromPoint,ToPoint ;
extern POINT FirstObjectPoint ;
extern POINT SecondObjectPoint ;
extern BOOL fNameOfFile ;
extern BOOL fNameRgn ;
extern BOOL CursorOpen ;
extern int xPosition,yPosition ;
extern int HorizThumbPos ; /* Indicate Viewport X coordinate of DrawWnd */
extern int VertThumbPos ; /* Indicate Viewport Y coordinate of DrawWnd */
extern int SelectToolItem ; /* Indicate Tool Number Selected */
extern int CurrentMetaRec ;
extern int MetaRecCount ;
extern int ConstraintFlag ;
extern void (*ToolProc[MAXTOOL]) 0 ;

```

```

extern struct (
    HDC hMetaDC ;
) ErMetaFile ;

```

```

extern HGLOBAL FIRST ;
extern HGLOBAL LAST ;
extern char NameBuffer[30] ;
extern char DataTypeBuffer[20] ;
extern char DomainBuffer[120] ;
extern char TempBuffer[30] ;
extern OPENFILENAME ofn ;
extern char szFilterSpec [128] ;
extern char szFileName[MAXFILENAME] ;
extern char LinkListFileName[MAXFILENAME] ;
extern char OracleFileName[MAXFILENAME] ;
extern char szFileTitle[MAXFILENAME] ;
extern char PrintFileName[MAXFILENAME] ;
extern char szBaseWindowTitle[128] ;
extern char szWindowTitle[80] ;
extern PRINTDLG pd ; /* Common print dialog structure */
extern char SaveFileName[30] ; /* handle for printer device context */
extern HDC hPR ; /* FALSE if user cancels printing */
extern BOOL bAbort ;
extern HWND hAbortDlgWnd ;
extern FARPROC lpAbortProc ;

```

เอกสารนี้เป็นเอกสารที่... ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

extern WORD      IOStatus;          /* result of file I/O */
extern int       Status;
extern int       nPageSize;         /* vert. resolution of printer device */
extern BOOL      FdFlag;
extern BOOL      JdFlag;
extern BOOL      ErrorFlag;
extern BOOL      DisplayFlag;
extern char      NumberIndex[200];
extern char      NumberTable[24];

extern struct {
    char Name[30];
    int  Type;
} Entity,Relation;

extern struct {
    int  Attr_Number;
    char Name[30];
    int  Type;
    char DataType[20];
    char Entity_Name[30];
    char Relation_Name[30];
    int  Composit_Number;
} Attr;

extern struct {
    int  Role_Number;
    int  Participation;
    int  Cadinality;
    char Entity_Name[30];
    char Relation_Name[30];
} Role;

extern struct {
    int  Attr_Number;
    char Domain_Name[30];
} cDomain;

extern struct {
    int  Attr_Number;
    char Lower_Limit[2];
    char Min[30];
    char Max[30];
    char Upper_Limit[2];
} nDomain;

extern struct {
    int  Attr_Number;
    char Invert[2];
} cInvert;

extern struct {
    int  Fd_Number;
    char Table_Name[30];
} Fd;

extern struct {
    int  Fd_Number;
    int  Attr_Number;
} LHS,RHS;

extern struct {
    int  Index_Number;
    int  Attr_Number;
} Temp_Attr_of_Index,Attr_of_Index;

extern struct {
    int  Index_Number;
    char Table_Name[30];
} Temp_Table_of_Index,Table_of_Index;

extern struct {
    char Table_Name[30];
    int  Attr_Number;
    char Not_Null[2];
} Temp_Attr_of_Table,Attr_of_Table;

extern struct {
    int  Attr_Number;
    char Attr_Name[30];
    char DataType[20];
} Attr_Detail;

```

Filename: ERPROTO.INC

```

int  PASCAL WinMain(HANDLE, HANDLE, LPSTR, int);
int  FAR PASCAL EnumMetaFileProc (HDC,HANDLETABLE FAR*,METARECORD FAR*,int,BYTE FAR*);
int  FAR PASCAL UpdateMetaFileProc (HDC,HANDLETABLE FAR*,METARECORD FAR*,int,BYTE FAR*);
int  FAR PASCAL RegionMetaFileProc (HDC,HANDLETABLE FAR*,METARECORD FAR*,int,BYTE FAR*);
void CALLBACK LineDDAProc (int,int,LPARAM);
BOOL InitApplication(HANDLE);
BOOL InitInstance(HANDLE, int);
long FAR PASCAL __export DrawWndProc(HWND, UINT, WPARAM, LPARAM);
long FAR PASCAL __export ToolWndProc(HWND, UINT, WPARAM, LPARAM);
long FAR PASCAL __export MainWndProc(HWND, UINT, WPARAM, LPARAM);
BOOL FAR PASCAL __export About (HWND, unsigned, WORD, LONG);
BOOL FAR PASCAL __export FdProc (HWND, unsigned, WORD, LONG);
BOOL FAR PASCAL __export JdProc (HWND, unsigned, WORD, LONG);
BOOL FAR PASCAL __export rEntityDigProc (HWND, unsigned, WORD, LONG);

```

ขอสงวนลิขสิทธิ์ในเอกสารนี้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BOOL FAR PASCAL __export wEntityDlgProc (HWND, unsigned, WORD, LONG) ;
BOOL FAR PASCAL __export nRelationDlgProc (HWND, unsigned, WORD, LONG) ;
BOOL FAR PASCAL __export iRelationDlgProc (HWND, unsigned, WORD, LONG) ;
BOOL FAR PASCAL __export nAttributeDlgProc (HWND, unsigned, WORD, LONG) ;
BOOL FAR PASCAL __export kAttributeDlgProc (HWND, unsigned, WORD, LONG) ;
BOOL FAR PASCAL __export pAttributeDlgProc (HWND, unsigned, WORD, LONG) ;
BOOL FAR PASCAL __export mvAttributeDlgProc (HWND, unsigned, WORD, LONG) ;
BOOL FAR PASCAL __export TextDlgProc (HWND, unsigned, WORD, LONG) ;
void ENTITY_PROC (HWND, unsigned, WORD, LONG, int) ;
void R_ENTITY_PROC (HWND, unsigned, WORD, LONG) ;
void W_ENTITY_PROC (HWND, unsigned, WORD, LONG) ;
void RELATION_PROC (HWND, unsigned, WORD, LONG, int) ;
void N_RELATION_PROC (HWND, unsigned, WORD, LONG) ;
void I_RELATION_PROC (HWND, unsigned, WORD, LONG) ;
void ATTRIBUTE_PROC (HWND, unsigned, WORD, LONG, int) ;
void N_ATTRIBUTE_PROC (HWND, unsigned, WORD, LONG) ;
void K_ATTRIBUTE_PROC (HWND, unsigned, WORD, LONG) ;
void P_ATTRIBUTE_PROC (HWND, unsigned, WORD, LONG) ;
void MV_ATTRIBUTE_PROC (HWND, unsigned, WORD, LONG) ;
void LINK_PROC (HWND, unsigned, WORD, LONG) ;
void PARTIAL_1_PROC (HWND, unsigned, WORD, LONG) ;
void PARTIAL_N_PROC (HWND, unsigned, WORD, LONG) ;
void PARTIAL_M_PROC (HWND, unsigned, WORD, LONG) ;
void TOTAL_1_PROC (HWND, unsigned, WORD, LONG) ;
void TOTAL_N_PROC (HWND, unsigned, WORD, LONG) ;
void TOTAL_M_PROC (HWND, unsigned, WORD, LONG) ;
void ROLE_PROC (HWND, unsigned, WORD, LONG, LPSTR, int) ;
void TEXT_PROC (HWND, unsigned, WORD, LONG) ;
void INDICATOR_PROC (HWND, unsigned, WORD, LONG) ;

```

/\* Tool Function For Erfive Project \*/

```

void UpdateMetaFile (void) ;
void AdjustLineCenter (POINT *,POINT *) ;
void AdjustLineNormal (POINT *,POINT *) ;
void CreateDoubleLine (POINT,POINT,POINT *,POINT *,POINT *,POINT *) ;
int WhatObjectSelected (int,int) ;
void SetCursorCenterPos (HWND) ;
void CreateErMetaFile (BYTE FAR*) ;
void PlayErMetaFile (HDC) ;
void SaveErMetaFile (void) ;
void DeleteErMetaFile (void) ;
void ErrorMessage (LPSTR) ;
BOOL IsBlankString (BYTE FAR*) ;
BOOL IsValidPosition (HRGN) ;
BOOL DeleteLink (void) ;
BOOL DeleteNode (HGLOBAL) ;
BOOL AppendNode (HRGN,int,int,LPSTR,int,LPSTR) ;
void UpdateLink (HGLOBAL) ;
void SaveLink (void) ;
void LoadLink (void) ;
void CreateMetaFileRegion (void) ;
void CreateFileName (void) ;
void ChangeFileTitle (void) ;
BOOL CreateOracleTable (void) ;
BOOL DropOracleTable (void) ;
BOOL CreateProcessTable (void) ;
BOOL DropProcessTable (void) ;
BOOL LoadOracleTable (void) ;
BOOL SaveOracleTable (void) ;
BOOL ProcessOracleErrorCode (LPSTR) ;
BOOL LoginOracle (void) ;
BOOL LogoutOracle (void) ;
BOOL DeleteMetaFileObject (NODE *,HGLOBAL) ;
void InvertAttribute (int) ;
void InvertRole (int) ;
HGLOBAL IsObjectSelected (int,int) ;
HGLOBAL IsAttributeSelected (int) ;
HGLOBAL IsRoleSelected (int) ;
void DeleteMetaFileGroup (NODE *,HGLOBAL) ;
BOOL IsValidOracle (LPSTR,LPSTR,LPSTR) ;
BOOL IsCompleteER (void) ;
void CreateSqlTable (LPSTR) ;
BOOL ProcessFirst (void) ;
void ClearBuffer (LPSTR,int) ;
void BinaryMapiing (int,int,LPSTR) ;
void NaryMapping (void) ;
void IdentifyMapping (void) ;
void ManageComposit (void) ;
void ProcessBCNF (void) ;
void ClearListBox (HWND) ;
void ClearJdListBox (HWND) ;
BOOL CreateBCNFtable (void) ;
BOOL DropBCNFtable (void) ;
void IntegerMessage (int) ;
void SelectIndexListBox (HWND) ;
void SelectIndexJdListBox (HWND) ;
HDC PASCAL GetPrinterDC(void) ;
BOOL OpenViewFile (LPSTR) ;
BOOL PrintViewFile (void) ;
BOOL SaveViewFile (HWND) ;
void SetNewBuffer(HWND, HANDLE, PSTR) ;
BOOL PrintErMetaFile(void) ;
void WaitCursor(BOOL) ;
void MapBCNF (void) ;
BOOL GenBCNF (void) ;
BOOL GenThirdNF (void) ;
BOOL KillRedundantDependency (void) ;
void GenDefaultFD (void) ;
BOOL IsValidGroupFd (void) ;

```

เอกสารนี้เป็นลิขสิทธิ์การดำเนินงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BOOL      IsValidGroupJd (void) ;
void      SeparateFD (void) ;
void      FindRedundantFD (void) ;
void      MapThird (void) ;
void      FindBCNF (void) ;
void      CreateMetaTableBCNF (void) ;
void      ProcessFifth (void) ;
BOOL      GenFifth (void) ;
void      MapFifth (void) ;
BOOL      ValidStringChar (void) ;
BOOL      IsValidDomainCharBuffer (void) ;
BOOL      CheckDomainNumberBuffer (void) ;
BOOL      ValidNumberChar (char) ;
BOOL      ValidDomainNumberBuffer (void) ;
BOOL      IsDigit (char) ;
void      SaveStringConstraint (int) ;
void      SaveNumberConstraint (int) ;
void      DeleteOracleAttribute (int) ;
void      DisconnectOracle (void) ;
void      ConnectOracle (void) ;

```

Filename: ERFIVEPC.PC

```

#include "erglobal.inc"
int PASCAL WinMain(HANDLE hInstance, HANDLE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)
{

```

```

    MSG msg;

    if (hPrevInstance)
        if (!InitApplication(hInstance))
            return (FALSE);

    if (!InitInstance(hInstance, nCmdShow))
        return (FALSE);

    while (GetMessage(&msg, NULL, NULL, NULL))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return (msg.wParam);
}

```

```

BOOL InitApplication(HANDLE hInstance)
{

```

```

    WNDCLASS wc;

    wc.style = CS_VREDRAW | CS_HREDRAW ;
    wc.lpfnWndProc = MainWndProc;
    wc.cbClsExtra = 0;
    wc.cbWndExtra = 0;
    wc.hInstance = hInstance;
    wc.hIcon = LoadIcon(hInstance, "ERFIVEPC");
    wc.hCursor = NULL;
    wc.hbrBackground = GetStockObject(WHITE_BRUSH);
    wc.lpszMenuName = "ERFIVEMENU";
    wc.lpszClassName = "MainWClass";

    if (RegisterClass (&wc))
    {
        wc.style = NULL ;
        wc.lpfnWndProc = DrawWndProc ;
        wc.cbClsExtra = 0 ;
        wc.cbWndExtra = 0 ;
        wc.hInstance = hInstance ;
        wc.hIcon = LoadIcon(NULL, IDI_APPLICATION) ;
        wc.hCursor = NULL ;
        wc.hbrBackground = GetStockObject(WHITE_BRUSH) ;
        wc.lpszMenuName = NULL ;
        wc.lpszClassName = "DrawWClass" ;
        if (RegisterClass (&wc))
        {
            wc.style = NULL ;
            wc.lpfnWndProc = ToolWndProc ;
            wc.cbClsExtra = 0 ;
            wc.cbWndExtra = 0 ;
            wc.hInstance = hInstance ;
            wc.hIcon = LoadIcon(NULL, IDI_APPLICATION) ;
            wc.hCursor = LoadCursor (hInstance, "Toolcurs") ;
            wc.hbrBackground = GetStockObject(WHITE_BRUSH) ;
            wc.lpszMenuName = NULL ;
            wc.lpszClassName = "ToolWClass" ;
            if (RegisterClass (&wc))
                return TRUE ;
        }
    }
}

```

```

return FALSE ;
}

```

```

BOOL InitInstance(HANDLE hInstance, Int nCmdShow)
{

```

```

    int hFile ;
    OFSTRUCT OfStruct;

    hInst = hInstance ;
    hPencilCursor = LoadCursor (hInst, "PENCIL") ;
    hCrossCursor = LoadCursor (NULL, IDC_CROSS) ;
    hArrowCursor = LoadCursor (NULL, IDC_ARROW) ;
    hWaitCursor = LoadCursor (NULL, IDC_WAIT) ;

```

เอกสารนี้เป็นเอกสารเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

hAsciiCursor = LoadCursor (hInst,"ASCII");
hIndicatorCursor = LoadCursor (hInst,"INDICATE");
hMainWnd = CreateWindow (
    "MainWClass",
    szBaseWindowTitle,
    WS_OVERLAPPEDWINDOW,
    CW_USEDEFAULT,
    CW_USEDEFAULT,
    CW_USEDEFAULT,
    CW_USEDEFAULT,
    NULL,
    NULL,
    hInstance,
    NULL
);

if (hMainWnd)
    return (FALSE);

/* Create a child window */

hDrawWnd = CreateWindow
(
    "DrawWClass",NULL,WS_CHILD | WS_VISIBLE | WS_VSCROLL | WS_HSCROLL | WS_BORDER,
    0,0,0,hMainWnd,NULL,hInst,NULL
);
hToolWnd = CreateWindow
(
    "ToolWClass",NULL,WS_CHILD | WS_VISIBLE | WS_BORDER ,0,0,0,hMainWnd,NULL,hInst,NULL
);
if (hDrawWnd || hToolWnd)
{
    DestroyWindow (hMainWnd);
    return (NULL);
}

/* fill in non-variant fields of OPENFILENAME struct */

ofn.lStructSize = sizeof(OPENFILENAME);
ofn.hwndOwner = hMainWnd;
ofn.lpstrFilter = szFilterSpec;
ofn.lpstrCustomFilter = NULL;
ofn.nMaxCustFilter = 0;
ofn.nFilterIndex = 1;
ofn.lpstrFile = szFileName;
ofn.nMaxFile = MAXFILENAME;
ofn.lpstrInitialDir = NULL;
ofn.lpstrFileTitle = szFileTitle;
ofn.nMaxFileTitle = MAXFILENAME;
ofn.lpstrTitle = NULL;
ofn.lpstrDefExt = "ER";
ofn.Flags = 0;

/* fill in non-variant fields of PRINTDLG struct */

pd.lStructSize = sizeof(PRINTDLG);
pd.hwndOwner = hMainWnd;
pd.hDevMode = NULL;
pd.hDevNames = NULL;
pd.Flags = PD_RETURNDC | PD_NOSELECTION | PD_NOPAGENUMS;
pd.nCopies = 1;
SelectToolItem = 0;
CurrentMetaRec = 1;
ShowWindow(hMainWnd, nCmdShow);
UpdateWindow (hMainWnd);
hFile = OpenFile ("First.sql",&OfStruct,OF_CREATE);
if (hFile >=0) close (hFile);
hFile = OpenFile ("BCNF.sql",&OfStruct,OF_CREATE);
if (hFile >=0) close (hFile);
hFile = OpenFile ("Fifth.sql",&OfStruct,OF_CREATE);
if (hFile >=0) close (hFile);
return (TRUE);
}

```

Filename: MAINWND.PC

```

#include "erextem.inc"
long FAR PASCAL __export MainWndProc(hMainWnd, message, wParam, lParam)
HWND hMainWnd;
UINT message;
WPARAM wParam;
LPARAM lParam;
{
    FARPROC lpProcAbout;

    switch (message)
    {
        case WM_CREATE:
        {
            if (LoginOracle () == TRUE)
            {
                SetCapture (hToolWnd);
                hOldCursor = SetCursor (hWaitCursor);
                hEmptyRgn = CreateRectRgn(0, 0, 0, 0);
                hDisplayRgn = CreateRectRgn(0, 0, 0, 0);
                hBlackSolidPen = CreatePen (PS_SOLID,1,RGB (0,0,0));
                hGreenSolidPen = CreatePen (PS_SOLID,1,RGB (0,191,0));
                hBlueSolidPen = CreatePen (PS_SOLID,1,RGB (0,0,255));
                lstrcpy (szFileName,"NONAME.ER");
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CreateFileName 0 ;
Istrcpy (szWindowTitle, szBaseWindowTitle) ;
Istrcat (szWindowTitle, " - ") ;
Istrcat (szWindowTitle, szFileName) ;
SetWindowText (hMainWnd, szWindowTitle) ;
CreateErMetaFile (szFileName) ;
fNoNameFile = TRUE ;
SetCursor (hOldCursor) ;
EnableMenuItem (GetMenu (hMainWnd), IDM_FIFTH, MF_DISABLED|MF_GRAYED) ;
EnableMenuItem (GetMenu (hMainWnd), IDM_BCNF, MF_DISABLED|MF_GRAYED) ;
ReleaseCapture 0 ;
}
else
PostQuitMessage (0) ;
}
break ;
case WM_COMMAND:
{
switch (wParam)
{
case IDM_ABOUT:
{
lpProcAbout = MakeProcInstance (About, hInst) ;
DialogBox (hInst, "AboutBox", hMainWnd, lpProcAbout) ;
FreeProcInstance (lpProcAbout) ;
}
break ;
case IDM_OPEN:
{
HDC hDC ;
HMETAFILE hMF ;
BOOL ErFormat ;

Istrcpy (szFileName, ".ER") ;
if (!GetOpenFileName ((LPOpenFileName)&ofn))
return FALSE ;
SetCapture (hToolWnd) ;
hOldCursor = SetCursor (hWaitCursor) ;
hMF = GetMetaFile (szFileName) ;
hDC = GetDC (hDrawWnd) ;
Rectangle (hDC, DrawWndRect.left-1, DrawWndRect.top-1, DrawWndRect.right+1, DrawWndRect.bottom+1) ;
SetWindowOrg (hDC, HorzThumbPos, VertThumbPos) ;
ErFormat = PlayMetaFile (hDC, hMF) ;
ReleaseDC (hDrawWnd, hDC) ;
if (ErFormat != FALSE)
{
ErMetaFile.hMetaDC = CreateMetaFile (NULL) ;
PlayMetaFile (ErMetaFile.hMetaDC, hMF) ;
Istrcpy (szWindowTitle, szBaseWindowTitle) ;
Istrcat (szWindowTitle, " - ") ;
ChangeFileTitle 0 ;
Istrcat (szWindowTitle, szFileName) ;
SetWindowText (hMainWnd, szWindowTitle) ;
CreateFileName 0 ;
DeleteLink 0 ;
LoadLink 0 ;
LoadOracleTable 0 ;
if (FdFlag == TRUE)
EnableMenuItem (GetMenu (hMainWnd), IDM_BCNF, MF_ENABLED) ;
else
EnableMenuItem (GetMenu (hMainWnd), IDM_BCNF, MF_DISABLED|MF_GRAYED) ;
if (JdFlag == TRUE)
EnableMenuItem (GetMenu (hMainWnd), IDM_FIFTH, MF_ENABLED) ;
else
EnableMenuItem (GetMenu (hMainWnd), IDM_FIFTH, MF_DISABLED|MF_GRAYED) ;
}
else
ErrorMessage ("Don't ER Format File") ;
DeleteMetaFile (hMF) ;
SetCursor (hOldCursor) ;
ReleaseCapture 0 ;
}
break ;
case IDM_SAVEAS:
{
if (!GetSaveFileName ((LPOpenFileName)&ofn))
return FALSE ;
CreateFileName 0 ;
SaveErMetaFile 0 ;
SaveLink 0 ;
SaveOracleTable 0 ;
if (fNoNameFile == TRUE)
{
Istrcpy (szWindowTitle, szBaseWindowTitle) ;
Istrcat (szWindowTitle, " - ") ;
ChangeFileTitle 0 ;
Istrcat (szWindowTitle, szFileName) ;
SetWindowText (hMainWnd, szWindowTitle) ;
fNoNameFile = FALSE ;
}
}
break ;
case IDM_SAVE:
{
SaveErMetaFile 0 ;
SaveLink 0 ;
SaveOracleTable 0 ;
}
break ;
case IDM_PRINT:

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        Istrcpy (PrintFileName,szFileTitle);
        PrintErMetaFile ();
    }
    break;
case IDM_EXIT:
    DestroyWindow (hMainWnd);
    break;
case IDM_FIRST:
    {
        if (IsCompleteER () == FALSE)
            ErrorMessage ("Entity Relationship Model Error I");
        else
        {
            if (ProcessFirst () == FALSE)
                ErrorMessage ("Entity Relationship Model Error I");
            else
            {
                EnableMenuItem (GetMenu(hMainWnd),IDM_BCNF,MF_ENABLED);
                EnableMenuItem (GetMenu(hMainWnd),IDM_FIFTH,MF_DISABLED|MF_GRAYED);
            }
        }
    }
    break;
case IDM_BCNF:
    ProcessBCNF ();
    break;
case IDM_FIFTH:
    ProcessFifth ();
    break;
case IDM_LOGIN:
    ConnectOracle ();
    break;
case IDM_LOGOUT:
    DisconnectOracle ();
    break;
}
}
break;
case WM_SIZE:
{
    int xStart,yStart,xWidth,yHeight;

    GetClientRect(hMainWnd,&MainWndRect);
    xStart=2;
    yStart=2;
    xWidth=(MainWndRect.right-MainWndRect.left)/12 /* 1/10 */;
    yHeight=(MainWndRect.bottom-MainWndRect.top)-4;
    MoveWindow(hToolWnd,xStart,yStart,xWidth,yHeight,TRUE);
    xStart=xWidth+4;
    yStart=2;
    xWidth=MainWndRect.right-xStart-2;
    yHeight=MainWndRect.bottom-MainWndRect.top-4;
    MoveWindow(hDrawWnd,xStart,yStart,xWidth,yHeight,TRUE);
}
break;
case WM_ENDSESSION:
{
    if ((BOOL) wParam)
        DestroyWindow (hMainWnd);
}
break;
case WM_QUIT:
case WM_CLOSE:
case WM_DESTROY:
{
    DeleteObject (hDisplayRgn);
    DeleteObject (hEmptyRgn);
    DeleteObject (hBlackSolidPen);
    DeleteObject (hBlueSolidPen);
    DeleteObject (hGreenSolidPen);
    DeleteLink ();
    DeleteErMetaFile ();
    LogoutOracle ();
    PostQuitMessage (0);
}
break;
default:
    return (DefWindowProc (hMainWnd, message, wParam, lParam));
}
return (NULL);
}
}

```

Filename: TOOLWND.PC

```

#include "erxtem.inc"
void DrawToolItem ();
void SetToolItemNumber ();
long FAR PASCAL __export ToolWndProc(hToolWnd, message, wParam, lParam)
HWND hToolWnd;
UINT message;
WPARAM wParam;
LPARAM lParam;
{
    switch (message)
    {
        case WM_PAINT:
            {
                int xStart,yStart,xWidth,yHeight;
                DrawToolItem ();
            }
        case WM_DESTROY:
            DestroyWindow (hToolWnd);
            PostQuitMessage (0);
    }
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

HDC hDC ;
PAINTSTRUCT ps ;

GetClientRect (hToolWnd,&ToolWndRect) ;
ToolHight = (ToolWndRect.bottom-ToolWndRect.top)/MAXTOOL ;
Fragment = (ToolWndRect.bottom-ToolWndRect.top)%MAXTOOL ;
hDC = BeginPaint (hToolWnd,&ps) ;
hOldPen = SelectObject (hDC,hBlackSolidPen) ;
for (j=0;j<MAXTOOL;j++)
{
    ToolMetrix[j].left = ToolWndRect.left ;
    ToolMetrix[j].right = ToolWndRect.right ;
    if (Fragment>0)
    {
        ToolMetrix[j].top = ToolWndRect.top + (j*ToolHight) + 1 ;
        ToolMetrix[j].bottom = ToolWndRect.top + (ToolHight*(j+1)) + 1 + 1 ;
    }
    else
    {
        ToolMetrix[j].top = ToolWndRect.top + (j*ToolHight) + Fragment ;
        ToolMetrix[j].bottom = ToolWndRect.top + (ToolHight*(j+1)) + Fragment ;
    }
    if (j != MAXTOOL-1)
    {
        MoveTo (hDC,ToolMetrix[j].left,ToolMetrix[j].bottom) ;
        LineTo (hDC,ToolMetrix[j].right,ToolMetrix[j].bottom) ;
    }
}
DrawToolItem (hDC) ;
SelectObject (hDC,hOldPen) ;
EndPaint (hToolWnd,&ps) ;
}
break ;
case WM_LBUTTONDOWN:
{
    HDC hDC ;

    SetToolItemNumber (iParam) ;
    hDC = GetDC (hToolWnd) ;
    DrawToolItem (hDC) ;
    ReleaseDC (hToolWnd,hDC) ;
}
break ;
default
return (DefWindowProc(hToolWnd, message, wParam, lParam)) ;
}
return (NULL) ;
}
}

void SetToolItemNumber (LPARAM lParam)
{
    int MousePosition ;

    MousePosition = HIWORD (lParam) ;
    if ((MousePosition > ToolMetrix[R_ENTITTY].top) && (MousePosition < ToolMetrix[R_ENTITTY].bottom))
        SelectToolItem = R_ENTITTY ;
    if ((MousePosition > ToolMetrix[W_ENTITTY].top) && (MousePosition < ToolMetrix[W_ENTITTY].bottom))
        SelectToolItem = W_ENTITTY ;
    if ((MousePosition > ToolMetrix[N_RELATION].top) && (MousePosition < ToolMetrix[N_RELATION].bottom))
        SelectToolItem = N_RELATION ;
    if ((MousePosition > ToolMetrix[L_RELATION].top) && (MousePosition < ToolMetrix[L_RELATION].bottom))
        SelectToolItem = L_RELATION ;
    if ((MousePosition > ToolMetrix[N_ATTRIBUTE].top) && (MousePosition < ToolMetrix[N_ATTRIBUTE].bottom))
        SelectToolItem = N_ATTRIBUTE ;
    if ((MousePosition > ToolMetrix[K_ATTRIBUTE].top) && (MousePosition < ToolMetrix[K_ATTRIBUTE].bottom))
        SelectToolItem = K_ATTRIBUTE ;
    if ((MousePosition > ToolMetrix[P_ATTRIBUTE].top) && (MousePosition < ToolMetrix[P_ATTRIBUTE].bottom))
        SelectToolItem = P_ATTRIBUTE ;
    if ((MousePosition > ToolMetrix[MV_ATTRIBUTE].top) && (MousePosition < ToolMetrix[MV_ATTRIBUTE].bottom))
        SelectToolItem = MV_ATTRIBUTE ;
    if ((MousePosition > ToolMetrix[LINK].top) && (MousePosition < ToolMetrix[LINK].bottom))
        SelectToolItem = LINK ;
    if ((MousePosition > ToolMetrix[PARTIAL_1].top) && (MousePosition < ToolMetrix[PARTIAL_1].bottom))
        SelectToolItem = PARTIAL_1 ;
    if ((MousePosition > ToolMetrix[PARTIAL_N].top) && (MousePosition < ToolMetrix[PARTIAL_N].bottom))
        SelectToolItem = PARTIAL_N ;
    if ((MousePosition > ToolMetrix[PARTIAL_M].top) && (MousePosition < ToolMetrix[PARTIAL_M].bottom))
        SelectToolItem = PARTIAL_M ;
    if ((MousePosition > ToolMetrix[TOTAL_1].top) && (MousePosition < ToolMetrix[TOTAL_1].bottom))
        SelectToolItem = TOTAL_1 ;
    if ((MousePosition > ToolMetrix[TOTAL_N].top) && (MousePosition < ToolMetrix[TOTAL_N].bottom))
        SelectToolItem = TOTAL_N ;
    if ((MousePosition > ToolMetrix[TOTAL_M].top) && (MousePosition < ToolMetrix[TOTAL_M].bottom))
        SelectToolItem = TOTAL_M ;
    if ((MousePosition > ToolMetrix[TEXT].top) && (MousePosition < ToolMetrix[TEXT].bottom))
        SelectToolItem = TEXT ;
    if ((MousePosition > ToolMetrix[INDICATOR].top) && (MousePosition < ToolMetrix[INDICATOR].bottom))
        SelectToolItem = INDICATOR ;
}

void DrawToolItem (HDC hDC)
{
    int aVertices[] = {5,5} ;
    POINT Partial_1_Point [4] ;
    POINT Partial_N_Point [4] ;
    POINT Partial_M_Point [4] ;
    POINT Total_1_Point [4] ;
    POINT Total_N_Point [4] ;
    POINT Total_M_Point [4] ;
    POINT nRelationPoint [4] ;
}

```

เอกสารนี้เป็นเอกสารตัวอย่างสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

POINT    iRelationPoint [10] ;

hOldPen = SelectObject (hDC,hGreenSolidPen) ;
Rectangle
(
    hDC,
    (ToolMetric[R_ENTITY].right-ToolMetric[R_ENTITY].left)/8,
    (ToolMetric[R_ENTITY].bottom+ToolMetric[R_ENTITY].top)*5/8,
    (ToolMetric[R_ENTITY].right-ToolMetric[R_ENTITY].left)*5/8,
    (ToolMetric[R_ENTITY].bottom*5+ToolMetric[R_ENTITY].top)/8
);

Rectangle
(
    hDC,
    (ToolMetric[W_ENTITY].right-ToolMetric[W_ENTITY].left)/8,
    (ToolMetric[W_ENTITY].bottom+ToolMetric[W_ENTITY].top)*5/8,
    (ToolMetric[W_ENTITY].right-ToolMetric[W_ENTITY].left)*5/8,
    (ToolMetric[W_ENTITY].bottom*5+ToolMetric[W_ENTITY].top)/8
);

Rectangle
(
    hDC,
    (ToolMetric[V_ENTITY].right-ToolMetric[V_ENTITY].left)/3,
    (ToolMetric[V_ENTITY].bottom+ToolMetric[V_ENTITY].top)*2/3,
    (ToolMetric[V_ENTITY].right-ToolMetric[V_ENTITY].left)*2/3,
    (ToolMetric[V_ENTITY].bottom*2+ToolMetric[V_ENTITY].top)/3
);

nRelationPoint[0].x = (ToolMetric[N_RELATION].right-ToolMetric[N_RELATION].left)/8 ;
nRelationPoint[0].y = (ToolMetric[N_RELATION].bottom+ToolMetric[N_RELATION].top)/2 ;
nRelationPoint[1].x = (ToolMetric[N_RELATION].right-ToolMetric[N_RELATION].left)/2 ;
nRelationPoint[1].y = (ToolMetric[N_RELATION].bottom+ToolMetric[N_RELATION].top)*7/8 ;
nRelationPoint[2].x = (ToolMetric[N_RELATION].right-ToolMetric[N_RELATION].left)*7/8 ;
nRelationPoint[2].y = (ToolMetric[N_RELATION].bottom+ToolMetric[N_RELATION].top)/2 ;
nRelationPoint[3].x = (ToolMetric[N_RELATION].right-ToolMetric[N_RELATION].left)/2 ;
nRelationPoint[3].y = (ToolMetric[N_RELATION].bottom*7+ToolMetric[N_RELATION].top)/8 ;
Polygon (hDC,nRelationPoint,sizeof (nRelationPoint) / sizeof (POINT)) ;

iRelationPoint[0].x = (ToolMetric[I_RELATION].right-ToolMetric[I_RELATION].left)/8 ;
iRelationPoint[0].y = (ToolMetric[I_RELATION].bottom+ToolMetric[I_RELATION].top)/2 ;
iRelationPoint[1].x = (ToolMetric[I_RELATION].right-ToolMetric[I_RELATION].left)/2 ;
iRelationPoint[1].y = (ToolMetric[I_RELATION].bottom+ToolMetric[I_RELATION].top)*7/8 ;
iRelationPoint[2].x = (ToolMetric[I_RELATION].right-ToolMetric[I_RELATION].left)*7/8 ;
iRelationPoint[2].y = (ToolMetric[I_RELATION].bottom+ToolMetric[I_RELATION].top)/2 ;
iRelationPoint[3].x = (ToolMetric[I_RELATION].right-ToolMetric[I_RELATION].left)/2 ;
iRelationPoint[3].y = (ToolMetric[I_RELATION].bottom*7+ToolMetric[I_RELATION].top)/8 ;
iRelationPoint[4].x = (ToolMetric[I_RELATION].right-ToolMetric[I_RELATION].left)/8 ;
iRelationPoint[4].y = (ToolMetric[I_RELATION].bottom+ToolMetric[I_RELATION].top)/2 ;
iRelationPoint[5].x = (ToolMetric[I_RELATION].right-ToolMetric[I_RELATION].left)/4 ;
iRelationPoint[5].y = (ToolMetric[I_RELATION].bottom+ToolMetric[I_RELATION].top)/2 ;
iRelationPoint[6].x = (ToolMetric[I_RELATION].right-ToolMetric[I_RELATION].left)/2 ;
iRelationPoint[6].y = (ToolMetric[I_RELATION].bottom+ToolMetric[I_RELATION].top)*3/4 ;
iRelationPoint[7].x = (ToolMetric[I_RELATION].right-ToolMetric[I_RELATION].left)*3/4 ;
iRelationPoint[7].y = (ToolMetric[I_RELATION].bottom+ToolMetric[I_RELATION].top)/2 ;
iRelationPoint[8].x = (ToolMetric[I_RELATION].right-ToolMetric[I_RELATION].left)/2 ;
iRelationPoint[8].y = (ToolMetric[I_RELATION].bottom*3+ToolMetric[I_RELATION].top)/4 ;
iRelationPoint[9].x = (ToolMetric[I_RELATION].right-ToolMetric[I_RELATION].left)/4 ;
iRelationPoint[9].y = (ToolMetric[I_RELATION].bottom+ToolMetric[I_RELATION].top)/2 ;
PolyPolygon (hDC,iRelationPoint,aVertices,sizeof (aVertices)/sizeof (int)) ;

Ellipse
(
    hDC,
    (ToolMetric[N_ATTRIBUTE].right-ToolMetric[N_ATTRIBUTE].left)/8,
    (ToolMetric[N_ATTRIBUTE].bottom+ToolMetric[N_ATTRIBUTE].top)*5/8,
    (ToolMetric[N_ATTRIBUTE].right-ToolMetric[N_ATTRIBUTE].left)*5/8,
    (ToolMetric[N_ATTRIBUTE].bottom*5+ToolMetric[N_ATTRIBUTE].top)/8
);

Ellipse
(
    hDC,
    (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)/8,
    (ToolMetric[K_ATTRIBUTE].bottom+ToolMetric[K_ATTRIBUTE].top)*5/8,
    (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)*5/8,
    (ToolMetric[K_ATTRIBUTE].bottom*5+ToolMetric[K_ATTRIBUTE].top)/8
);

MoveTo
(
    hDC,
    (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)/3,
    (ToolMetric[K_ATTRIBUTE].bottom*2+ToolMetric[K_ATTRIBUTE].top)/3
);

LineTo
(
    hDC,
    (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)*2/3,
    (ToolMetric[K_ATTRIBUTE].bottom*2+ToolMetric[K_ATTRIBUTE].top)/3
);

MoveTo
(
    hDC,
    (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)*7/12,
    (ToolMetric[K_ATTRIBUTE].bottom+ToolMetric[K_ATTRIBUTE].top)*2/3
);

LineTo
(
    hDC,
    (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)*5/12,

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยขอนแก่น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        (ToolMetric[K_ATTRIBUTE].bottom+ToolMetric[K_ATTRIBUTE].top*2)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)*5/12,
        (ToolMetric[K_ATTRIBUTE].bottom*7+ToolMetric[K_ATTRIBUTE].top*5)/12
    );
    LineTo
    (
        hDC,
        (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)*7/12,
        (ToolMetric[K_ATTRIBUTE].bottom*7+ToolMetric[K_ATTRIBUTE].top*5)/12
    );
    Ellipse
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)/8,
        (ToolMetric[P_ATTRIBUTE].bottom+ToolMetric[P_ATTRIBUTE].top*5)/8,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)*5/8,
        (ToolMetric[P_ATTRIBUTE].bottom*5+ToolMetric[P_ATTRIBUTE].top)/8
    );
    MoveTo
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)/3,
        (ToolMetric[P_ATTRIBUTE].bottom*2+ToolMetric[P_ATTRIBUTE].top)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)*2/3,
        (ToolMetric[P_ATTRIBUTE].bottom*2+ToolMetric[P_ATTRIBUTE].top)/3
    );
    MoveTo
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)*5/12,
        (ToolMetric[P_ATTRIBUTE].bottom*7+ToolMetric[P_ATTRIBUTE].top*5)/12
    );
    LineTo
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)*5/12,
        (ToolMetric[P_ATTRIBUTE].bottom+ToolMetric[P_ATTRIBUTE].top*2)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)*7/12,
        (ToolMetric[P_ATTRIBUTE].bottom+ToolMetric[P_ATTRIBUTE].top*2)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)*7/12,
        (ToolMetric[P_ATTRIBUTE].bottom*11+ToolMetric[P_ATTRIBUTE].top*13)/24
    );
    LineTo
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)*5/12,
        (ToolMetric[P_ATTRIBUTE].bottom*11+ToolMetric[P_ATTRIBUTE].top*13)/24
    );
    Ellipse
    (
        hDC,
        (ToolMetric[MV_ATTRIBUTE].right-ToolMetric[MV_ATTRIBUTE].left)/8,
        (ToolMetric[MV_ATTRIBUTE].bottom+ToolMetric[MV_ATTRIBUTE].top*5)/8,
        (ToolMetric[MV_ATTRIBUTE].right-ToolMetric[MV_ATTRIBUTE].left)*5/8,
        (ToolMetric[MV_ATTRIBUTE].bottom*5+ToolMetric[MV_ATTRIBUTE].top)/8
    );
    Ellipse
    (
        hDC,
        (ToolMetric[MV_ATTRIBUTE].right-ToolMetric[MV_ATTRIBUTE].left)/4,
        (ToolMetric[MV_ATTRIBUTE].bottom+ToolMetric[MV_ATTRIBUTE].top*3)/4,
        (ToolMetric[MV_ATTRIBUTE].right-ToolMetric[MV_ATTRIBUTE].left)*3/4,
        (ToolMetric[MV_ATTRIBUTE].bottom*3+ToolMetric[MV_ATTRIBUTE].top)/4
    );
    Ellipse
    (
        hDC,
        (ToolMetric[LINK].right-ToolMetric[LINK].left)*5/12,
        (ToolMetric[LINK].bottom+ToolMetric[LINK].top*3)/4,
        (ToolMetric[LINK].right-ToolMetric[LINK].left)*11/12,
        (ToolMetric[LINK].bottom*3+ToolMetric[LINK].top)/4
    );
    MoveTo
    (
        hDC,
        (ToolMetric[LINK].right-ToolMetric[LINK].left)/12,
        (ToolMetric[LINK].bottom+ToolMetric[LINK].top)/2
    );
    LineTo
    (
        hDC,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        (ToolMetrix[LINK].right-ToolMetrix[LINK].left)*5/12,
        (ToolMetrix[LINK].bottom+ToolMetrix[LINK].top)/2
    );

Partial_1_Point[0].x = (ToolMetrix[PARTIAL_1].right*7+ToolMetrix[PARTIAL_1].left*9)/16 ;
Partial_1_Point[0].y = (ToolMetrix[PARTIAL_1].bottom+ToolMetrix[PARTIAL_1].top)/2 ;
Partial_1_Point[1].x = (ToolMetrix[PARTIAL_1].right*11+ToolMetrix[PARTIAL_1].left*5)/16 ;
Partial_1_Point[1].y = (ToolMetrix[PARTIAL_1].bottom*2+ToolMetrix[PARTIAL_1].top*10)/12 ;
Partial_1_Point[2].x = (ToolMetrix[PARTIAL_1].right*15+ToolMetrix[PARTIAL_1].left)/16 ;
Partial_1_Point[2].y = (ToolMetrix[PARTIAL_1].bottom+ToolMetrix[PARTIAL_1].top)/2 ;
Partial_1_Point[3].x = (ToolMetrix[PARTIAL_1].right*11+ToolMetrix[PARTIAL_1].left*5)/16 ;
Partial_1_Point[3].y = (ToolMetrix[PARTIAL_1].bottom*5+ToolMetrix[PARTIAL_1].top)/8 ;
Polygon (hDC,Partial_1_Point,sizeof (Partial_1_Point) / sizeof (POINT)) ;
MoveTo
(
    hDC,
    (ToolMetrix[PARTIAL_1].right+ToolMetrix[PARTIAL_1].left*15)/16,
    (ToolMetrix[PARTIAL_1].bottom+ToolMetrix[PARTIAL_1].top)/2
);
LineTo
(
    hDC,
    (ToolMetrix[PARTIAL_1].right*7+ToolMetrix[PARTIAL_1].left*15)/16,
    (ToolMetrix[PARTIAL_1].bottom+ToolMetrix[PARTIAL_1].top)/2
);
MoveTo
(
    hDC,
    (ToolMetrix[PARTIAL_1].right*11+ToolMetrix[PARTIAL_1].left*5)/16,
    (ToolMetrix[PARTIAL_1].bottom*9+ToolMetrix[PARTIAL_1].top*15)/24
);
LineTo
(
    hDC,
    (ToolMetrix[PARTIAL_1].right*11+ToolMetrix[PARTIAL_1].left*5)/16,
    (ToolMetrix[PARTIAL_1].bottom*15+ToolMetrix[PARTIAL_1].top*9)/24
);

Partial_N_Point[0].x = (ToolMetrix[PARTIAL_N].right*7+ToolMetrix[PARTIAL_N].left*9)/16 ;
Partial_N_Point[0].y = (ToolMetrix[PARTIAL_N].bottom+ToolMetrix[PARTIAL_N].top)/2 ;
Partial_N_Point[1].x = (ToolMetrix[PARTIAL_N].right*11+ToolMetrix[PARTIAL_N].left*5)/16 ;
Partial_N_Point[1].y = (ToolMetrix[PARTIAL_N].bottom*2+ToolMetrix[PARTIAL_N].top*10)/12 ;
Partial_N_Point[2].x = (ToolMetrix[PARTIAL_N].right*15+ToolMetrix[PARTIAL_N].left)/16 ;
Partial_N_Point[2].y = (ToolMetrix[PARTIAL_N].bottom+ToolMetrix[PARTIAL_N].top)/2 ;
Partial_N_Point[3].x = (ToolMetrix[PARTIAL_N].right*11+ToolMetrix[PARTIAL_N].left*5)/16 ;
Partial_N_Point[3].y = (ToolMetrix[PARTIAL_N].bottom*5+ToolMetrix[PARTIAL_N].top)/8 ;
Polygon (hDC,Partial_N_Point,sizeof (Partial_N_Point) / sizeof (POINT)) ;
MoveTo
(
    hDC,
    (ToolMetrix[PARTIAL_N].right+ToolMetrix[PARTIAL_N].left*15)/16,
    (ToolMetrix[PARTIAL_N].bottom+ToolMetrix[PARTIAL_N].top)/2
);
LineTo
(
    hDC,
    (ToolMetrix[PARTIAL_N].right*7+ToolMetrix[PARTIAL_N].left*15)/16,
    (ToolMetrix[PARTIAL_N].bottom+ToolMetrix[PARTIAL_N].top)/2
);
MoveTo
(
    hDC,
    (ToolMetrix[PARTIAL_N].right*5+ToolMetrix[PARTIAL_N].left*3)/8,
    (ToolMetrix[PARTIAL_N].bottom*15+ToolMetrix[PARTIAL_N].top*9)/24
);
LineTo
(
    hDC,
    (ToolMetrix[PARTIAL_N].right*5+ToolMetrix[PARTIAL_N].left*3)/8,
    (ToolMetrix[PARTIAL_N].bottom*9+ToolMetrix[PARTIAL_N].top*15)/24
);
LineTo
(
    hDC,
    (ToolMetrix[PARTIAL_N].right*3+ToolMetrix[PARTIAL_N].left)/4,
    (ToolMetrix[PARTIAL_N].bottom*15+ToolMetrix[PARTIAL_N].top*9)/24
);
LineTo
(
    hDC,
    (ToolMetrix[PARTIAL_N].right*3+ToolMetrix[PARTIAL_N].left)/4,
    (ToolMetrix[PARTIAL_N].bottom*9+ToolMetrix[PARTIAL_N].top*15)/24
);

Partial_M_Point[0].x = (ToolMetrix[PARTIAL_M].right*7+ToolMetrix[PARTIAL_M].left*9)/16 ;
Partial_M_Point[0].y = (ToolMetrix[PARTIAL_M].bottom+ToolMetrix[PARTIAL_M].top)/2 ;
Partial_M_Point[1].x = (ToolMetrix[PARTIAL_M].right*11+ToolMetrix[PARTIAL_M].left*5)/16 ;
Partial_M_Point[1].y = (ToolMetrix[PARTIAL_M].bottom*2+ToolMetrix[PARTIAL_M].top*10)/12 ;
Partial_M_Point[2].x = (ToolMetrix[PARTIAL_M].right*15+ToolMetrix[PARTIAL_M].left)/16 ;
Partial_M_Point[2].y = (ToolMetrix[PARTIAL_M].bottom+ToolMetrix[PARTIAL_M].top)/2 ;
Partial_M_Point[3].x = (ToolMetrix[PARTIAL_M].right*11+ToolMetrix[PARTIAL_M].left*5)/16 ;
Partial_M_Point[3].y = (ToolMetrix[PARTIAL_M].bottom*5+ToolMetrix[PARTIAL_M].top)/8 ;
Polygon (hDC,Partial_M_Point,sizeof (Partial_M_Point) / sizeof (POINT)) ;
MoveTo
(
    hDC,
    (ToolMetrix[PARTIAL_M].right+ToolMetrix[PARTIAL_M].left*15)/16,
    (ToolMetrix[PARTIAL_M].bottom+ToolMetrix[PARTIAL_M].top)/2
);

```

เอกสารนี้ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LineTo
(
    hDC,
    (ToolMetric[PARTIAL_M].right*7+ToolMetric[PARTIAL_M].left*15)/16,
    (ToolMetric[PARTIAL_M].bottom+ToolMetric[PARTIAL_M].top)/2
);

MoveTo
(
    hDC,
    (ToolMetric[PARTIAL_M].right*19+ToolMetric[PARTIAL_M].left*13)/32,
    (ToolMetric[PARTIAL_M].bottom*15+ToolMetric[PARTIAL_M].top*9)/24
);
LineTo
(
    hDC,
    (ToolMetric[PARTIAL_M].right*19+ToolMetric[PARTIAL_M].left*13)/32,
    (ToolMetric[PARTIAL_M].bottom*9+ToolMetric[PARTIAL_M].top*15)/24
);
LineTo
(
    hDC,
    (ToolMetric[PARTIAL_M].right*11+ToolMetric[PARTIAL_M].left*5)/16,
    (ToolMetric[PARTIAL_M].bottom*15+ToolMetric[PARTIAL_M].top*9)/24
);
LineTo
(
    hDC,
    (ToolMetric[PARTIAL_M].right*25+ToolMetric[PARTIAL_M].left*7)/32,
    (ToolMetric[PARTIAL_M].bottom*9+ToolMetric[PARTIAL_M].top*15)/24
);
LineTo
(
    hDC,
    (ToolMetric[PARTIAL_M].right*25+ToolMetric[PARTIAL_M].left*7)/32,
    (ToolMetric[PARTIAL_M].bottom*15+ToolMetric[PARTIAL_M].top*9)/24
);

Total_1_Point[0].x = (ToolMetric[TOTAL_1].right*7+ToolMetric[TOTAL_1].left*9)/16 ;
Total_1_Point[0].y = (ToolMetric[TOTAL_1].bottom+ToolMetric[TOTAL_1].top)/2 ;
Total_1_Point[1].x = (ToolMetric[TOTAL_1].right*11+ToolMetric[TOTAL_1].left*5)/16 ;
Total_1_Point[1].y = (ToolMetric[TOTAL_1].bottom*2+ToolMetric[TOTAL_1].top*10)/12 ;
Total_1_Point[2].x = (ToolMetric[TOTAL_1].right*15+ToolMetric[TOTAL_1].left)/16 ;
Total_1_Point[2].y = (ToolMetric[TOTAL_1].bottom+ToolMetric[TOTAL_1].top)/2 ;
Total_1_Point[3].x = (ToolMetric[TOTAL_1].right*11+ToolMetric[TOTAL_1].left*5)/16 ;
Total_1_Point[3].y = (ToolMetric[TOTAL_1].bottom*5+ToolMetric[TOTAL_1].top)/6 ;
Polygon (hDC,Total_1_Point,sizeof (Total_1_Point) / sizeof (POINT));
MoveTo
(
    hDC,
    (ToolMetric[TOTAL_1].right+ToolMetric[TOTAL_1].left*15)/16,
    (ToolMetric[TOTAL_1].bottom*5+ToolMetric[TOTAL_1].top*7)/12
);
LineTo
(
    hDC,
    (ToolMetric[TOTAL_1].right+ToolMetric[TOTAL_1].left)/2,
    (ToolMetric[TOTAL_1].bottom*5+ToolMetric[TOTAL_1].top*7)/12
);
MoveTo
(
    hDC,
    (ToolMetric[TOTAL_1].right+ToolMetric[TOTAL_1].left*15)/16,
    (ToolMetric[TOTAL_1].bottom*7+ToolMetric[TOTAL_1].top*5)/12
);
LineTo
(
    hDC,
    (ToolMetric[TOTAL_1].right+ToolMetric[TOTAL_1].left)/2,
    (ToolMetric[TOTAL_1].bottom*7+ToolMetric[TOTAL_1].top*5)/12
);
MoveTo
(
    hDC,
    (ToolMetric[TOTAL_1].right*11+ToolMetric[TOTAL_1].left*5)/16,
    (ToolMetric[TOTAL_1].bottom*9+ToolMetric[TOTAL_1].top*15)/24
);
LineTo
(
    hDC,
    (ToolMetric[TOTAL_1].right*11+ToolMetric[TOTAL_1].left*5)/16,
    (ToolMetric[TOTAL_1].bottom*15+ToolMetric[TOTAL_1].top*9)/24
);

Total_N_Point[0].x = (ToolMetric[TOTAL_N].right*7+ToolMetric[TOTAL_N].left*9)/16 ;
Total_N_Point[0].y = (ToolMetric[TOTAL_N].bottom+ToolMetric[TOTAL_N].top)/2 ;
Total_N_Point[1].x = (ToolMetric[TOTAL_N].right*11+ToolMetric[TOTAL_N].left*5)/16 ;
Total_N_Point[1].y = (ToolMetric[TOTAL_N].bottom*2+ToolMetric[TOTAL_N].top*10)/12 ;
Total_N_Point[2].x = (ToolMetric[TOTAL_N].right*15+ToolMetric[TOTAL_N].left)/16 ;
Total_N_Point[2].y = (ToolMetric[TOTAL_N].bottom+ToolMetric[TOTAL_N].top)/2 ;
Total_N_Point[3].x = (ToolMetric[TOTAL_N].right*11+ToolMetric[TOTAL_N].left*5)/16 ;
Total_N_Point[3].y = (ToolMetric[TOTAL_N].bottom*5+ToolMetric[TOTAL_N].top)/6 ;
Polygon (hDC,Total_N_Point,sizeof (Total_N_Point) / sizeof (POINT));
MoveTo
(
    hDC,
    (ToolMetric[TOTAL_N].right+ToolMetric[TOTAL_N].left*15)/16,
    (ToolMetric[TOTAL_N].bottom*5+ToolMetric[TOTAL_N].top*7)/12
);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(
    hDC,
    (ToolMetric[TOTAL_N].right+ToolMetric[TOTAL_N].left)/2,
    (ToolMetric[TOTAL_N].bottom*5+ToolMetric[TOTAL_N].top*7)/12
);
MoveTo
(
    hDC,
    (ToolMetric[TOTAL_N].right+ToolMetric[TOTAL_N].left*15)/16,
    (ToolMetric[TOTAL_N].bottom*7+ToolMetric[TOTAL_N].top*5)/12
);
LineTo
(
    hDC,
    (ToolMetric[TOTAL_N].right+ToolMetric[TOTAL_N].left)/2,
    (ToolMetric[TOTAL_N].bottom*7+ToolMetric[TOTAL_N].top*5)/12
);
MoveTo
(
    hDC,
    (ToolMetric[TOTAL_N].right*5+ToolMetric[TOTAL_N].left*3)/8,
    (ToolMetric[TOTAL_N].bottom*15+ToolMetric[TOTAL_N].top*9)/24
);
LineTo
(
    hDC,
    (ToolMetric[TOTAL_N].right*5+ToolMetric[TOTAL_N].left*3)/8,
    (ToolMetric[TOTAL_N].bottom*9+ToolMetric[TOTAL_N].top*15)/24
);
LineTo
(
    hDC,
    (ToolMetric[TOTAL_N].right*3+ToolMetric[TOTAL_N].left)/4,
    (ToolMetric[TOTAL_N].bottom*15+ToolMetric[TOTAL_N].top*9)/24
);
LineTo
(
    hDC,
    (ToolMetric[TOTAL_N].right*3+ToolMetric[TOTAL_N].left)/4,
    (ToolMetric[TOTAL_N].bottom*9+ToolMetric[TOTAL_N].top*15)/24
);
);
Total_M_Point[0].x = (ToolMetric[TOTAL_M].right*7+ToolMetric[TOTAL_M].left*9)/16 ;
Total_M_Point[0].y = (ToolMetric[TOTAL_M].bottom+ToolMetric[TOTAL_M].top)/2 ;
Total_M_Point[1].x = (ToolMetric[TOTAL_M].right*11+ToolMetric[TOTAL_M].left*5)/16 ;
Total_M_Point[1].y = (ToolMetric[TOTAL_M].bottom*2+ToolMetric[TOTAL_M].top*10)/12 ;
Total_M_Point[2].x = (ToolMetric[TOTAL_M].right*15+ToolMetric[TOTAL_M].left)/16 ;
Total_M_Point[2].y = (ToolMetric[TOTAL_M].bottom+ToolMetric[TOTAL_M].top)/2 ;
Total_M_Point[3].x = (ToolMetric[TOTAL_M].right*11+ToolMetric[TOTAL_M].left*5)/16 ;
Total_M_Point[3].y = (ToolMetric[TOTAL_M].bottom*5+ToolMetric[TOTAL_M].top)/8 ;
Polygon (hDC,Total_M_Point,sizeof (Total_M_Point) / sizeof (POINT)) ;
MoveTo
(
    hDC,
    (ToolMetric[TOTAL_M].right+ToolMetric[TOTAL_M].left*15)/16,
    (ToolMetric[TOTAL_M].bottom*5+ToolMetric[TOTAL_M].top*7)/12
);
LineTo
(
    hDC,
    (ToolMetric[TOTAL_M].right+ToolMetric[TOTAL_M].left)/2,
    (ToolMetric[TOTAL_M].bottom*5+ToolMetric[TOTAL_M].top*7)/12
);
MoveTo
(
    hDC,
    (ToolMetric[TOTAL_M].right+ToolMetric[TOTAL_M].left*15)/16,
    (ToolMetric[TOTAL_M].bottom*7+ToolMetric[TOTAL_M].top*5)/12
);
LineTo
(
    hDC,
    (ToolMetric[TOTAL_M].right+ToolMetric[TOTAL_M].left)/2,
    (ToolMetric[TOTAL_M].bottom*7+ToolMetric[TOTAL_M].top*5)/12
);
MoveTo
(
    hDC,
    (ToolMetric[TOTAL_M].right*19+ToolMetric[TOTAL_M].left*13)/32,
    (ToolMetric[TOTAL_M].bottom*15+ToolMetric[TOTAL_M].top*9)/24
);
LineTo
(
    hDC,
    (ToolMetric[TOTAL_M].right*19+ToolMetric[TOTAL_M].left*13)/32,
    (ToolMetric[TOTAL_M].bottom*9+ToolMetric[TOTAL_M].top*15)/24
);
LineTo
(
    hDC,
    (ToolMetric[TOTAL_M].right*11+ToolMetric[TOTAL_M].left*5)/16,
    (ToolMetric[TOTAL_M].bottom*15+ToolMetric[TOTAL_M].top*9)/24
);
LineTo
(
    hDC,
    (ToolMetric[TOTAL_M].right*25+ToolMetric[TOTAL_M].left*7)/32,
    (ToolMetric[TOTAL_M].bottom*9+ToolMetric[TOTAL_M].top*15)/24
);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LineTo
(
    hDC,
    (ToolMetric[TOTAL_M].right*25+ToolMetric[TOTAL_M].left*7)/32,
    (ToolMetric[TOTAL_M].bottom*15+ToolMetric[TOTAL_M].top*9)/24
);
MoveTo
(
    hDC,
    (ToolMetric[TEXT].right+ToolMetric[TEXT].left*12)/13,
    (ToolMetric[TEXT].bottom*6+ToolMetric[TEXT].top*10)/16
);
LineTo
(
    hDC,
    (ToolMetric[TEXT].right*3+ToolMetric[TEXT].left*10)/13,
    (ToolMetric[TEXT].bottom*6+ToolMetric[TEXT].top*10)/16
);
MoveTo
(
    hDC,
    (ToolMetric[TEXT].right*2+ToolMetric[TEXT].left*11)/13,
    (ToolMetric[TEXT].bottom*6+ToolMetric[TEXT].top*10)/16
);
LineTo
(
    hDC,
    (ToolMetric[TEXT].right*2+ToolMetric[TEXT].left*11)/13,
    (ToolMetric[TEXT].bottom*12+ToolMetric[TEXT].top*4)/16
);
MoveTo
(
    hDC,
    (ToolMetric[TEXT].right*6+ToolMetric[TEXT].left*7)/13,
    (ToolMetric[TEXT].bottom*6+ToolMetric[TEXT].top*10)/16
);
LineTo
(
    hDC,
    (ToolMetric[TEXT].right*4+ToolMetric[TEXT].left*9)/13,
    (ToolMetric[TEXT].bottom*6+ToolMetric[TEXT].top*10)/16
);
LineTo
(
    hDC,
    (ToolMetric[TEXT].right*4+ToolMetric[TEXT].left*9)/13,
    (ToolMetric[TEXT].bottom*12+ToolMetric[TEXT].top*4)/16
);
LineTo
(
    hDC,
    (ToolMetric[TEXT].right*6+ToolMetric[TEXT].left*7)/13,
    (ToolMetric[TEXT].bottom*12+ToolMetric[TEXT].top*4)/16
);
MoveTo
(
    hDC,
    (ToolMetric[TEXT].right*4+ToolMetric[TEXT].left*9)/13,
    (ToolMetric[TEXT].bottom*9+ToolMetric[TEXT].top*7)/16
);
LineTo
(
    hDC,
    (ToolMetric[TEXT].right*6+ToolMetric[TEXT].left*7)/13,
    (ToolMetric[TEXT].bottom*9+ToolMetric[TEXT].top*7)/16
);
MoveTo
(
    hDC,
    (ToolMetric[TEXT].right*7+ToolMetric[TEXT].left*6)/13,
    (ToolMetric[TEXT].bottom*6+ToolMetric[TEXT].top*10)/16
);
LineTo
(
    hDC,
    (ToolMetric[TEXT].right*9+ToolMetric[TEXT].left*4)/13,
    (ToolMetric[TEXT].bottom*12+ToolMetric[TEXT].top*4)/16
);
MoveTo
(
    hDC,
    (ToolMetric[TEXT].right*9+ToolMetric[TEXT].left*4)/13,
    (ToolMetric[TEXT].bottom*6+ToolMetric[TEXT].top*10)/16
);
LineTo
(
    hDC,
    (ToolMetric[TEXT].right*7+ToolMetric[TEXT].left*6)/13,
    (ToolMetric[TEXT].bottom*12+ToolMetric[TEXT].top*4)/16
);
MoveTo
(
    hDC,
    (ToolMetric[TEXT].right*10+ToolMetric[TEXT].left*3)/13,
    (ToolMetric[TEXT].bottom*6+ToolMetric[TEXT].top*10)/16
);
LineTo
(
    hDC,

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        (ToolMetric[TEXT].right*12+ToolMetric[TEXT].left)/13,
        (ToolMetric[TEXT].bottom*6+ToolMetric[TEXT].top*10)/16
    );
    MoveTo
    (
        hDC,
        (ToolMetric[TEXT].right*11+ToolMetric[TEXT].left*2)/13,
        (ToolMetric[TEXT].bottom*6+ToolMetric[TEXT].top*10)/16
    );
    LineTo
    (
        hDC,
        (ToolMetric[TEXT].right*11+ToolMetric[TEXT].left*2)/13,
        (ToolMetric[TEXT].bottom*12+ToolMetric[TEXT].top*4)/16
    );
    MoveTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*4+ToolMetric[INDICATOR].left*11)/15,
        (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top*2)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*2+ToolMetric[INDICATOR].left*13)/15,
        (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top*2)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*2+ToolMetric[INDICATOR].left*13)/15,
        (ToolMetric[INDICATOR].bottom*2+ToolMetric[INDICATOR].top)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*4+ToolMetric[INDICATOR].left*11)/15,
        (ToolMetric[INDICATOR].bottom*2+ToolMetric[INDICATOR].top)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right+ToolMetric[INDICATOR].left*2)/3,
        (ToolMetric[INDICATOR].bottom*7+ToolMetric[INDICATOR].top*5)/12
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right+ToolMetric[INDICATOR].left*2)/3,
        (ToolMetric[INDICATOR].bottom*5+ToolMetric[INDICATOR].top*7)/12
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*4+ToolMetric[INDICATOR].left*11)/15,
        (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top*2)/3
    );
    MoveTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*3+ToolMetric[INDICATOR].left*2)/5,
        (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top*2)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*2+ToolMetric[INDICATOR].left*3)/5,
        (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top*2)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*2+ToolMetric[INDICATOR].left*3)/5,
        (ToolMetric[INDICATOR].bottom*2+ToolMetric[INDICATOR].top)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*3+ToolMetric[INDICATOR].left*2)/5,
        (ToolMetric[INDICATOR].bottom*2+ToolMetric[INDICATOR].top)/3
    );
    MoveTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*2+ToolMetric[INDICATOR].left*3)/5,
        (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top)/2
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*3+ToolMetric[INDICATOR].left*2)/5,
        (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top)/2
    );
    MoveTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*2+ToolMetric[INDICATOR].left)/3,
        (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top*2)/3
    );

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LineTo
(
    hDC,
    (ToolMetric[INDICATOR].right*2+ToolMetric[INDICATOR].left)/3,
    (ToolMetric[INDICATOR].bottom*2+ToolMetric[INDICATOR].top)/3
);
LineTo
(
    hDC,
    (ToolMetric[INDICATOR].right*13+ToolMetric[INDICATOR].left*2)/15,
    (ToolMetric[INDICATOR].bottom*2+ToolMetric[INDICATOR].top)/3
);
SelectObject (hDC,hOldPen);
switch (SelectToolItem)
(
    case R_ENTITY:
    (
        hOldPen = SelectObject (hDC,hBlueSolidPen);
        Rectangle
        (
            hDC,
            (ToolMetric[R_ENTITY].right-ToolMetric[R_ENTITY].left)/6,
            (ToolMetric[R_ENTITY].bottom+ToolMetric[R_ENTITY].top*5)/6,
            (ToolMetric[R_ENTITY].right-ToolMetric[R_ENTITY].left)*5/6,
            (ToolMetric[R_ENTITY].bottom*5+ToolMetric[R_ENTITY].top)/6
        );
        SelectObject (hDC,hOldPen);
    )
    break;
    case W_ENTITY:
    (
        hOldPen = SelectObject (hDC,hBlueSolidPen);
        Rectangle
        (
            hDC,
            (ToolMetric[W_ENTITY].right-ToolMetric[W_ENTITY].left)/6,
            (ToolMetric[W_ENTITY].bottom+ToolMetric[W_ENTITY].top*5)/6,
            (ToolMetric[W_ENTITY].right-ToolMetric[W_ENTITY].left)*5/6,
            (ToolMetric[W_ENTITY].bottom*5+ToolMetric[W_ENTITY].top)/6
        );
        Rectangle
        (
            hDC,
            (ToolMetric[W_ENTITY].right-ToolMetric[W_ENTITY].left)/3,
            (ToolMetric[W_ENTITY].bottom+ToolMetric[W_ENTITY].top*2)/3,
            (ToolMetric[W_ENTITY].right-ToolMetric[W_ENTITY].left)*2/3,
            (ToolMetric[W_ENTITY].bottom*2+ToolMetric[W_ENTITY].top)/3
        );
        SelectObject (hDC,hOldPen);
    )
    break;
    case N_RELATION:
    (
        hOldPen = SelectObject (hDC,hBlueSolidPen);
        Polygon (hDC,nRelationPoint,sizeof (nRelationPoint) / sizeof (POINT));
        SelectObject (hDC,hOldPen);
    )
    break;
    case L_RELATION:
    (
        hOldPen = SelectObject (hDC,hBlueSolidPen);
        PolyPolygon (hDC,lRelationPoint,aVertices,sizeof (aVertices)/sizeof (int));
        SelectObject (hDC,hOldPen);
    )
    break;
    case N_ATTRIBUTE:
    (
        hOldPen = SelectObject (hDC,hBlueSolidPen);
        Ellipse
        (
            hDC,
            (ToolMetric[N_ATTRIBUTE].right-ToolMetric[N_ATTRIBUTE].left)/6,
            (ToolMetric[N_ATTRIBUTE].bottom+ToolMetric[N_ATTRIBUTE].top*5)/6,
            (ToolMetric[N_ATTRIBUTE].right-ToolMetric[N_ATTRIBUTE].left)*5/6,
            (ToolMetric[N_ATTRIBUTE].bottom*5+ToolMetric[N_ATTRIBUTE].top)/6
        );
        SelectObject (hDC,hOldPen);
    )
    break;
    case K_ATTRIBUTE:
    (
        hOldPen = SelectObject (hDC,hBlueSolidPen);
        Ellipse
        (
            hDC,
            (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)/6,
            (ToolMetric[K_ATTRIBUTE].bottom+ToolMetric[K_ATTRIBUTE].top*5)/6,
            (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)*5/6,
            (ToolMetric[K_ATTRIBUTE].bottom*5+ToolMetric[K_ATTRIBUTE].top)/6
        );
        MoveTo
        (
            hDC,
            (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)/3,
            (ToolMetric[K_ATTRIBUTE].bottom*2+ToolMetric[K_ATTRIBUTE].top)/3
        );
        LineTo
        (
            hDC,

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)*2/3,
        (ToolMetric[K_ATTRIBUTE].bottom*2+ToolMetric[K_ATTRIBUTE].top)/3
    );
    MoveTo
    (
        hDC,
        (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)*7/12,
        (ToolMetric[K_ATTRIBUTE].bottom+ToolMetric[K_ATTRIBUTE].top*2)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)*5/12,
        (ToolMetric[K_ATTRIBUTE].bottom+ToolMetric[K_ATTRIBUTE].top*2)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)*5/12,
        (ToolMetric[K_ATTRIBUTE].bottom*7+ToolMetric[K_ATTRIBUTE].top*5)/12
    );
    LineTo
    (
        hDC,
        (ToolMetric[K_ATTRIBUTE].right-ToolMetric[K_ATTRIBUTE].left)*7/12,
        (ToolMetric[K_ATTRIBUTE].bottom*7+ToolMetric[K_ATTRIBUTE].top*5)/12
    );
    SelectObject (hDC,hOldPen);
}
break;
case P_ATTRIBUTE:
{
    hOldPen = SelectObject (hDC,hBlueSolidPen);
    Ellipse
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)/6,
        (ToolMetric[P_ATTRIBUTE].bottom+ToolMetric[P_ATTRIBUTE].top*5)/6,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)*5/6,
        (ToolMetric[P_ATTRIBUTE].bottom*5+ToolMetric[P_ATTRIBUTE].top)/6
    );
    MoveTo
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)/3,
        (ToolMetric[P_ATTRIBUTE].bottom*2+ToolMetric[P_ATTRIBUTE].top)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)*2/3,
        (ToolMetric[P_ATTRIBUTE].bottom*2+ToolMetric[P_ATTRIBUTE].top)/3
    );
    MoveTo
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)*5/12,
        (ToolMetric[P_ATTRIBUTE].bottom*7+ToolMetric[P_ATTRIBUTE].top*5)/12
    );
    LineTo
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)*5/12,
        (ToolMetric[P_ATTRIBUTE].bottom+ToolMetric[P_ATTRIBUTE].top*2)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)*7/12,
        (ToolMetric[P_ATTRIBUTE].bottom+ToolMetric[P_ATTRIBUTE].top*2)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)*7/12,
        (ToolMetric[P_ATTRIBUTE].bottom*11+ToolMetric[P_ATTRIBUTE].top*13)/24
    );
    LineTo
    (
        hDC,
        (ToolMetric[P_ATTRIBUTE].right-ToolMetric[P_ATTRIBUTE].left)*5/12,
        (ToolMetric[P_ATTRIBUTE].bottom*11+ToolMetric[P_ATTRIBUTE].top*13)/24
    );
    SelectObject (hDC,hOldPen);
}
break;
case MV_ATTRIBUTE:
{
    hOldPen = SelectObject (hDC,hBlueSolidPen);
    Ellipse
    (
        hDC,
        (ToolMetric[MV_ATTRIBUTE].right-ToolMetric[MV_ATTRIBUTE].left)/6,
        (ToolMetric[MV_ATTRIBUTE].bottom+ToolMetric[MV_ATTRIBUTE].top*5)/6,
        (ToolMetric[MV_ATTRIBUTE].right-ToolMetric[MV_ATTRIBUTE].left)*5/6,
        (ToolMetric[MV_ATTRIBUTE].bottom*5+ToolMetric[MV_ATTRIBUTE].top)/6
    );
    Ellipse
    (
        hDC,

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        (ToolMetric[MV_ATTRIBUTE].right-ToolMetric[MV_ATTRIBUTE].left)/4,
        (ToolMetric[MV_ATTRIBUTE].bottom+ToolMetric[MV_ATTRIBUTE].top*3)/4,
        (ToolMetric[MV_ATTRIBUTE].right-ToolMetric[MV_ATTRIBUTE].left)*3/4,
        (ToolMetric[MV_ATTRIBUTE].bottom*3+ToolMetric[MV_ATTRIBUTE].top)/4
    );
    SelectObject (hDC,hOldPen);
}
break;
case LINK:
{
    hOldPen = SelectObject (hDC,hBlueSolidPen);
    Ellipse
    (
        hDC,
        (ToolMetric[LINK].right-ToolMetric[LINK].left)*5/12,
        (ToolMetric[LINK].bottom+ToolMetric[LINK].top*3)/4,
        (ToolMetric[LINK].right-ToolMetric[LINK].left)*11/12,
        (ToolMetric[LINK].bottom*3+ToolMetric[LINK].top)/4
    );
    MoveTo
    (
        hDC,
        (ToolMetric[LINK].right-ToolMetric[LINK].left)/12,
        (ToolMetric[LINK].bottom+ToolMetric[LINK].top)/2
    );
    LineTo
    (
        hDC,
        (ToolMetric[LINK].right-ToolMetric[LINK].left)*5/12,
        (ToolMetric[LINK].bottom+ToolMetric[LINK].top)/2
    );
    SelectObject (hDC,hOldPen);
}
break;
case PARTIAL_1:
{
    hOldPen = SelectObject (hDC,hBlueSolidPen);
    Polygon (hDC,Partial_1_Point,sizeof (Partial_1_Point) / sizeof (POINT));
    MoveTo
    (
        hDC,
        (ToolMetric[PARTIAL_1].right+ToolMetric[PARTIAL_1].left*15)/16,
        (ToolMetric[PARTIAL_1].bottom+ToolMetric[PARTIAL_1].top)/2
    );
    LineTo
    (
        hDC,
        (ToolMetric[PARTIAL_1].right*7+ToolMetric[PARTIAL_1].left*15)/16,
        (ToolMetric[PARTIAL_1].bottom+ToolMetric[PARTIAL_1].top)/2
    );
    MoveTo
    (
        hDC,
        (ToolMetric[PARTIAL_1].right*11+ToolMetric[PARTIAL_1].left*5)/16,
        (ToolMetric[PARTIAL_1].bottom*9+ToolMetric[PARTIAL_1].top*15)/24
    );
    LineTo
    (
        hDC,
        (ToolMetric[PARTIAL_1].right*11+ToolMetric[PARTIAL_1].left*5)/16,
        (ToolMetric[PARTIAL_1].bottom*15+ToolMetric[PARTIAL_1].top*9)/24
    );
    SelectObject (hDC,hOldPen);
}
break;
case PARTIAL_N:
{
    hOldPen = SelectObject (hDC,hBlueSolidPen);
    Polygon (hDC,Partial_N_Point,sizeof (Partial_N_Point) / sizeof (POINT));
    MoveTo
    (
        hDC,
        (ToolMetric[PARTIAL_N].right+ToolMetric[PARTIAL_N].left*15)/16,
        (ToolMetric[PARTIAL_N].bottom+ToolMetric[PARTIAL_N].top)/2
    );
    LineTo
    (
        hDC,
        (ToolMetric[PARTIAL_N].right*7+ToolMetric[PARTIAL_N].left*15)/16,
        (ToolMetric[PARTIAL_N].bottom+ToolMetric[PARTIAL_N].top)/2
    );
    MoveTo
    (
        hDC,
        (ToolMetric[PARTIAL_N].right*5+ToolMetric[PARTIAL_N].left*3)/8,
        (ToolMetric[PARTIAL_N].bottom*15+ToolMetric[PARTIAL_N].top*9)/24
    );
    LineTo
    (
        hDC,
        (ToolMetric[PARTIAL_N].right*5+ToolMetric[PARTIAL_N].left*3)/8,
        (ToolMetric[PARTIAL_N].bottom*9+ToolMetric[PARTIAL_N].top*15)/24
    );
    LineTo
    (
        hDC,
        (ToolMetric[PARTIAL_N].right*3+ToolMetric[PARTIAL_N].left)/4,
        (ToolMetric[PARTIAL_N].bottom*15+ToolMetric[PARTIAL_N].top*9)/24
    );
    LineTo

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    (
        hDC,
        (ToolMatrix[PARTIAL_M].right*3+ToolMatrix[PARTIAL_M].left)/4,
        (ToolMatrix[PARTIAL_M].bottom*9+ToolMatrix[PARTIAL_M].top*15)/24
    );
    SelectObject (hDC,hOldPen);
}
break;
case PARTIAL_M:
{
    hOldPen = SelectObject (hDC,hBlueSolidPen);
    Polygon (hDC,Partial_M_Point,sizeof (Partial_M_Point) / sizeof (POINT));
    MoveTo
    (
        hDC,
        (ToolMatrix[PARTIAL_M].right+ToolMatrix[PARTIAL_M].left*15)/16,
        (ToolMatrix[PARTIAL_M].bottom+ToolMatrix[PARTIAL_M].top)/2
    );
    LineTo
    (
        hDC,
        (ToolMatrix[PARTIAL_M].right*7+ToolMatrix[PARTIAL_M].left*15)/16,
        (ToolMatrix[PARTIAL_M].bottom+ToolMatrix[PARTIAL_M].top)/2
    );
    MoveTo
    (
        hDC,
        (ToolMatrix[PARTIAL_M].right*19+ToolMatrix[PARTIAL_M].left*13)/32,
        (ToolMatrix[PARTIAL_M].bottom*15+ToolMatrix[PARTIAL_M].top*9)/24
    );
    LineTo
    (
        hDC,
        (ToolMatrix[PARTIAL_M].right*19+ToolMatrix[PARTIAL_M].left*13)/32,
        (ToolMatrix[PARTIAL_M].bottom*9+ToolMatrix[PARTIAL_M].top*15)/24
    );
    LineTo
    (
        hDC,
        (ToolMatrix[PARTIAL_M].right*11+ToolMatrix[PARTIAL_M].left*5)/16,
        (ToolMatrix[PARTIAL_M].bottom*15+ToolMatrix[PARTIAL_M].top*9)/24
    );
    LineTo
    (
        hDC,
        (ToolMatrix[PARTIAL_M].right*25+ToolMatrix[PARTIAL_M].left*7)/32,
        (ToolMatrix[PARTIAL_M].bottom*9+ToolMatrix[PARTIAL_M].top*15)/24
    );
    LineTo
    (
        hDC,
        (ToolMatrix[PARTIAL_M].right*25+ToolMatrix[PARTIAL_M].left*7)/32,
        (ToolMatrix[PARTIAL_M].bottom*15+ToolMatrix[PARTIAL_M].top*9)/24
    );
    SelectObject (hDC,hOldPen);
}
break;
case TOTAL_1:
{
    hOldPen = SelectObject (hDC,hBlueSolidPen);
    Polygon (hDC,TOTAL_1_Point,sizeof (TOTAL_1_Point) / sizeof (POINT));
    MoveTo
    (
        hDC,
        (ToolMatrix[TOTAL_1].right+ToolMatrix[TOTAL_1].left*15)/16,
        (ToolMatrix[TOTAL_1].bottom*5+ToolMatrix[TOTAL_1].top*7)/12
    );
    LineTo
    (
        hDC,
        (ToolMatrix[TOTAL_1].right+ToolMatrix[TOTAL_1].left)/2,
        (ToolMatrix[TOTAL_1].bottom*5+ToolMatrix[TOTAL_1].top*7)/12
    );
    MoveTo
    (
        hDC,
        (ToolMatrix[TOTAL_1].right+ToolMatrix[TOTAL_1].left*15)/16,
        (ToolMatrix[TOTAL_1].bottom*7+ToolMatrix[TOTAL_1].top*5)/12
    );
    LineTo
    (
        hDC,
        (ToolMatrix[TOTAL_1].right+ToolMatrix[TOTAL_1].left)/2,
        (ToolMatrix[TOTAL_1].bottom*7+ToolMatrix[TOTAL_1].top*5)/12
    );
    MoveTo
    (
        hDC,
        (ToolMatrix[TOTAL_1].right*11+ToolMatrix[TOTAL_1].left*5)/16,
        (ToolMatrix[TOTAL_1].bottom*9+ToolMatrix[TOTAL_1].top*15)/24
    );
    LineTo
    (
        hDC,
        (ToolMatrix[TOTAL_1].right*11+ToolMatrix[TOTAL_1].left*5)/16,
        (ToolMatrix[TOTAL_1].bottom*15+ToolMatrix[TOTAL_1].top*9)/24
    );
    SelectObject (hDC,hOldPen);
}
break;

```

เอกสารนี้เก็บเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case TOTAL_N:
{
    hOldPen = SelectObject (hDC,hBlueSolidPen);
    Polygon (hDC,Total_N_Point,sizeof (Total_N_Point) / sizeof (POINT));
    MoveTo
    (
        hDC,
        (ToolMetric[TOTAL_N].right+ToolMetric[TOTAL_N].left*15)/16,
        (ToolMetric[TOTAL_N].bottom*5+ToolMetric[TOTAL_N].top*7)/12
    );
    LineTo
    (
        hDC,
        (ToolMetric[TOTAL_N].right+ToolMetric[TOTAL_N].left)/2,
        (ToolMetric[TOTAL_N].bottom*5+ToolMetric[TOTAL_N].top*7)/12
    );
    MoveTo
    (
        hDC,
        (ToolMetric[TOTAL_N].right+ToolMetric[TOTAL_N].left*15)/16,
        (ToolMetric[TOTAL_N].bottom*7+ToolMetric[TOTAL_N].top*5)/12
    );
    LineTo
    (
        hDC,
        (ToolMetric[TOTAL_N].right+ToolMetric[TOTAL_N].left)/2,
        (ToolMetric[TOTAL_N].bottom*7+ToolMetric[TOTAL_N].top*5)/12
    );
    MoveTo
    (
        hDC,
        (ToolMetric[TOTAL_N].right*5+ToolMetric[TOTAL_N].left*3)/8,
        (ToolMetric[TOTAL_N].bottom*15+ToolMetric[TOTAL_N].top*9)/24
    );
    LineTo
    (
        hDC,
        (ToolMetric[TOTAL_N].right*5+ToolMetric[TOTAL_N].left*3)/8,
        (ToolMetric[TOTAL_N].bottom*9+ToolMetric[TOTAL_N].top*15)/24
    );
    LineTo
    (
        hDC,
        (ToolMetric[TOTAL_N].right*3+ToolMetric[TOTAL_N].left)/4,
        (ToolMetric[TOTAL_N].bottom*15+ToolMetric[TOTAL_N].top*9)/24
    );
    LineTo
    (
        hDC,
        (ToolMetric[TOTAL_N].right*3+ToolMetric[TOTAL_N].left)/4,
        (ToolMetric[TOTAL_N].bottom*9+ToolMetric[TOTAL_N].top*15)/24
    );
    SelectObject (hDC,hOldPen);
}
break;
case TOTAL_M:
{
    hOldPen = SelectObject (hDC,hBlueSolidPen);
    Polygon (hDC,Total_M_Point,sizeof (Total_M_Point) / sizeof (POINT));
    MoveTo
    (
        hDC,
        (ToolMetric[TOTAL_M].right+ToolMetric[TOTAL_M].left*15)/16,
        (ToolMetric[TOTAL_M].bottom*5+ToolMetric[TOTAL_M].top*7)/12
    );
    LineTo
    (
        hDC,
        (ToolMetric[TOTAL_M].right+ToolMetric[TOTAL_M].left)/2,
        (ToolMetric[TOTAL_M].bottom*5+ToolMetric[TOTAL_M].top*7)/12
    );
    MoveTo
    (
        hDC,
        (ToolMetric[TOTAL_M].right+ToolMetric[TOTAL_M].left*15)/16,
        (ToolMetric[TOTAL_M].bottom*7+ToolMetric[TOTAL_M].top*5)/12
    );
    LineTo
    (
        hDC,
        (ToolMetric[TOTAL_M].right+ToolMetric[TOTAL_M].left)/2,
        (ToolMetric[TOTAL_M].bottom*7+ToolMetric[TOTAL_M].top*5)/12
    );
    MoveTo
    (
        hDC,
        (ToolMetric[TOTAL_M].right*19+ToolMetric[TOTAL_M].left*13)/32,
        (ToolMetric[TOTAL_M].bottom*15+ToolMetric[TOTAL_M].top*9)/24
    );
    LineTo
    (
        hDC,
        (ToolMetric[TOTAL_M].right*19+ToolMetric[TOTAL_M].left*13)/32,
        (ToolMetric[TOTAL_M].bottom*9+ToolMetric[TOTAL_M].top*15)/24
    );
    LineTo
    (
        hDC,
        (ToolMetric[TOTAL_M].right*11+ToolMetric[TOTAL_M].left*5)/16,
        (ToolMetric[TOTAL_M].bottom*15+ToolMetric[TOTAL_M].top*9)/24
    );
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ห้ามมิให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

        (ToolMetric[TEXT].right*7+ToolMetric[TEXT].left*6)/13,
        (ToolMetric[TEXT].bottom*12+ToolMetric[TEXT].top*4)/16
    );
    MoveTo
    (
        hDC,
        (ToolMetric[TEXT].right*10+ToolMetric[TEXT].left*3)/13,
        (ToolMetric[TEXT].bottom*6+ToolMetric[TEXT].top*10)/16
    );
    LineTo
    (
        hDC,
        (ToolMetric[TEXT].right*12+ToolMetric[TEXT].left)/13,
        (ToolMetric[TEXT].bottom*6+ToolMetric[TEXT].top*10)/16
    );
    MoveTo
    (
        hDC,
        (ToolMetric[TEXT].right*11+ToolMetric[TEXT].left*2)/13,
        (ToolMetric[TEXT].bottom*6+ToolMetric[TEXT].top*10)/16
    );
    LineTo
    (
        hDC,
        (ToolMetric[TEXT].right*11+ToolMetric[TEXT].left*2)/13,
        (ToolMetric[TEXT].bottom*12+ToolMetric[TEXT].top*4)/16
    );
    SelectObject (hDC,hOldPen) ;
}
break ;
case INDICATOR:
{
    hOldPen = SelectObject (hDC,hBlueSolidPen) ;
    MoveTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*4+ToolMetric[INDICATOR].left*11)/15,
        (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top*2)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*2+ToolMetric[INDICATOR].left*13)/15,
        (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top*2)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*2+ToolMetric[INDICATOR].left*13)/15,
        (ToolMetric[INDICATOR].bottom*2+ToolMetric[INDICATOR].top)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*4+ToolMetric[INDICATOR].left*11)/15,
        (ToolMetric[INDICATOR].bottom*2+ToolMetric[INDICATOR].top)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right+ToolMetric[INDICATOR].left*2)/3,
        (ToolMetric[INDICATOR].bottom*7+ToolMetric[INDICATOR].top*5)/12
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right+ToolMetric[INDICATOR].left*2)/3,
        (ToolMetric[INDICATOR].bottom*5+ToolMetric[INDICATOR].top*7)/12
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*4+ToolMetric[INDICATOR].left*11)/15,
        (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top*2)/3
    );
    MoveTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*3+ToolMetric[INDICATOR].left*2)/5,
        (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top*2)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*2+ToolMetric[INDICATOR].left*3)/5,
        (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top*2)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*2+ToolMetric[INDICATOR].left*3)/5,
        (ToolMetric[INDICATOR].bottom*2+ToolMetric[INDICATOR].top)/3
    );
    LineTo
    (
        hDC,
        (ToolMetric[INDICATOR].right*3+ToolMetric[INDICATOR].left*2)/5,
        (ToolMetric[INDICATOR].bottom*2+ToolMetric[INDICATOR].top)/3
    );
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(
    hDC,
    (ToolMetric[INDICATOR].right*2+ToolMetric[INDICATOR].left*3)/5,
    (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top)/2
);
LineTo
(
    hDC,
    (ToolMetric[INDICATOR].right*3+ToolMetric[INDICATOR].left*2)/5,
    (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top)/2
);
MoveTo
(
    hDC,
    (ToolMetric[INDICATOR].right*2+ToolMetric[INDICATOR].left)/3,
    (ToolMetric[INDICATOR].bottom+ToolMetric[INDICATOR].top*2)/3
);
LineTo
(
    hDC,
    (ToolMetric[INDICATOR].right*2+ToolMetric[INDICATOR].left)/3,
    (ToolMetric[INDICATOR].bottom*2+ToolMetric[INDICATOR].top)/3
);
LineTo
(
    hDC,
    (ToolMetric[INDICATOR].right*13+ToolMetric[INDICATOR].left*2)/15,
    (ToolMetric[INDICATOR].bottom*2+ToolMetric[INDICATOR].top)/3
);
SelectObject (hDC,hOldPen);
}
break;
}
}

```

```

.....
Filename: DRAWWND.PC
.....
#include "erextern.inc"
EXEC SQL BEGIN DECLARE SECTION ;
  VARCHAR Entity_Name[30];
  VARCHAR Relation_Name[30];
  VARCHAR Attr_Name[30];
  VARCHAR DataType[20];
  int Type;
  int Role_Number;
  int Attr_Number;
  int Composit_Number;
  int Index_Number;
EXEC SQL END DECLARE SECTION ;
EXEC SQL INCLUDE SQLCA ;

long FAR PASCAL __export DrawWndProc(hDrawWnd,message,wParam,lParam)
HWND hDrawWnd;
UINT message;
WPARAM wParam;
LPARAM lParam;
{
  switch (message)
  {
    case WM_SETCURSOR:
    {
      switch (LOWORD (lParam))
      {
        case HTVSCROLL:
        case HTHSCROLL:
        {
          SetCursor (hArrowCursor);
        }
        break;
        default:
        {
          switch (SelectToolItem)
          {
            case TEXT:
            case R_ENTITY:
            case W_ENTITY:
            case N_RELATION:
            case I_RELATION:
            case N_ATTRIBUTE:
            case K_ATTRIBUTE:
            case P_ATTRIBUTE:
            case MV_ATTRIBUTE:
            {
              SetCursor (hAsciiCursor);
            }
            break;
            case INDICATOR:
            {
              SetCursor (hIndicatorCursor);
            }
            break;
            case PARTIAL_1:
            case PARTIAL_N:
            case PARTIAL_M:
            case TOTAL_1:
            case TOTAL_N:
            case TOTAL_M:
            case UNK:
          }
        }
      }
    }
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

if (VertThumbPos != GetScrollPos (hDrawWnd,SB_VERT))
{
    SetScrollPos (hDrawWnd,SB_VERT,VertThumbPos,TRUE);
    InvalidateRect (hDrawWnd,NULL,1);
    UpdateWindow (hDrawWnd);
}
}
break;
case WM_PAINT:
{
    HDC hDC;

    hDC = GetDC (hDrawWnd);
    Rectangle (hDC,DrawWndRect.left-1,DrawWndRect.top-1,DrawWndRect.right+1,DrawWndRect.bottom+1);
    PlayErMetaFile (hDC);
    ValidateRect (hDrawWnd,NULL);
    ReleaseDC (hDrawWnd,hDC);
    if (DisplayFlag == OFF)
    {
        hDC = GetDC (hDrawWnd);
        OffsetRgn (hDisplayRgn,-HorzThumbPos,-VertThumbPos);
        SelectObject (hDC,hDisplayRgn);
        InvertRgn (hDC,hDisplayRgn);
        OffsetRgn (hDisplayRgn,+HorzThumbPos,+VertThumbPos);
        ReleaseDC (hDrawWnd,hDC);
    }
}
break;
default:
(*ToolProc [SelectToolItem]) (hDrawWnd,message,wParam,lParam);
return (DefWindowProc(hDrawWnd, message, wParam, lParam));
}
}

void R_ENTITY_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)
{
    ENTITY_PROC (hDrawWnd,message,wParam,lParam,R_ENTITY);
}

void W_ENTITY_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)
{
    ENTITY_PROC (hDrawWnd,message,wParam,lParam,W_ENTITY);
}

void ENTITY_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam,int Entity_Type)
{
    static RECT      Frame;
    static DWORD    dwExtent;
    static int      Flag = EDIT;
    static int      Lenght;
    static char     EntityBoxName[20];
    FARPROC        lpProc_Entity;

    switch (message)
    {
        case WM_LBUTTONDOWN:
        {
            HDC      hDC;
            HRGN     hRgn;

            hDC = GetDC (hDrawWnd);
            SetROP2 (hDC,R2_NOT);
            SelectObject (hDC,GetStockObject(NULL_BRUSH));
            if (Flag == EDIT)
            {
                if (Entity_Type == R_ENTITY)
                {
                    lpProc_Entity = MakeProclnstance (rEntityDlgProc, hInst);
                    lstrcpy (EntityBoxName,"EntityBox");
                    Lenght = 2;
                }
                else
                {
                    lpProc_Entity = MakeProclnstance (wEntityDlgProc, hInst);
                    lstrcpy (EntityBoxName,"wEntityBox");
                    Lenght = 3;
                }
            }
            for (;;)
            {
                if (DialogBox(hInst,EntityBoxName, hMainWnd, lpProc_Entity) == 0)
                    break;
                if (IsBlankString (NameBuffer) == FALSE)
                {
                    lstrcpy (Entity_Name,arr,NameBuffer);
                    Entity_Name.len = lstrlen (Entity_Name.arr);
                    Type = Entity_Type;
                    EXEC SQL INSERT INTO ENTITY (ENTITY_NAME,ENTITY_TYPE)
                    VALUES (:Entity_Name,:Type);
                    if (ProcessOracleErrorCode ("INSERT ENTITY") == FALSE)
                    {
                        lstrcpy (NameBuffer,"");
                        continue;
                    }
                }
                SetCursorCenterPos (hDrawWnd);
                dwExtent = GetTextExtent (hDC,NameBuffer,lstrlen (NameBuffer));
                SetRect
                {
                    &Frame,
                    (DrawWndRect.right-DrawWndRect.left)/2 - (LOWORD (dwExtent))/2 - H_OFFSET,
                    (DrawWndRect.bottom-DrawWndRect.top)/2 - (HIWORD (dwExtent))/2 - W_OFFSET,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ห้ามเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร

ไม่ว่ากรณีใดๆก็ตาม ห้ามทำซ้ำหรือดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

    }
    else if (NextY >= DrawWndRect.bottom)
    {
        if (VertThumbPos >= V_SCROLL_RANGE)
            VertThumbPos = V_SCROLL_RANGE;
        else
        {
            VertThumbPos++;
            ThumbPosChange = TRUE;
        }
    }
    hDC = GetDC (hDrawWnd);
    SelectObject (hDC,GetStockObject(NULL_BRUSH));
    SetROP2 (hDC,R2_NOT);
    Rectangle (hDC,Frame.left,Frame.top,Frame.right,Frame.bottom);
    if (ThumbPosChange)
    {
        SetScrollPos (hDrawWnd,SB_VERT,VertThumbPos,TRUE);
        SetScrollPos (hDrawWnd,SB_HORZ,HorzThumbPos,TRUE);
        InvalidateRect (hDrawWnd,NULL,1);
        UpdateWindow (hDrawWnd);
    }
    SetRect
    (
        &Frame,NextX - (LOWORD (dwExtent))/2 - H_OFFSET,NextY - (HIWORD (dwExtent))/2 - W_OFFSET,
        NextX + (LOWORD (dwExtent))/2 + H_OFFSET,NextY + (HIWORD (dwExtent))/2 + W_OFFSET
    );
    Rectangle (hDC,Frame.left,Frame.top,Frame.right,Frame.bottom);
    ReleaseDC (hDrawWnd,hDC);
}
}
break;
}
}
}

```

```

BOOL FAR PASCAL _export rEntityDlgProc (HWND hDlg,unsigned message,WORD wParam,LONG lParam)

```

```

{
    switch (message)
    {
        case WM_INITDIALOG:
        {
            SetDlgItemText (hDlg, IDC_EDIT,NULL);
            return (TRUE);
        }
        case WM_COMMAND:
        {
            switch (wParam)
            {
                case IDOK:
                {
                    GetDlgItemText (hDlg, IDC_EDIT,NameBuffer,30);
                    EndDialog (hDlg,TRUE);
                    return (TRUE);
                }
                case IDCANCEL:
                {
                    EndDialog (hDlg,FALSE);
                    return (FALSE);
                }
            }
        }
    }
    return (FALSE);
}

```

```

BOOL FAR PASCAL _export wEntityDlgProc (HWND hDlg,unsigned message,WORD wParam,LONG lParam)

```

```

{
    switch (message)
    {
        case WM_INITDIALOG:
        {
            SetDlgItemText (hDlg, IDC_EDIT,NULL);
            return (TRUE);
        }
        case WM_COMMAND:
        {
            switch (wParam)
            {
                case IDOK:
                {
                    GetDlgItemText (hDlg, IDC_EDIT,NameBuffer,30);
                    EndDialog (hDlg,TRUE);
                    return (TRUE);
                }
                case IDCANCEL:
                {
                    EndDialog (hDlg,FALSE);
                    return (FALSE);
                }
            }
        }
    }
    return (FALSE);
}

```

```

void N_RELATION_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)

```

```

{
    RELATION_PROC (hDrawWnd,message,wParam,lParam,N_RELATION);
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ที่โรงเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void I_RELATION_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)
{
    RELATION_PROC (hDrawWnd,message,wParam,lParam,I_RELATION);
}

void RELATION_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam, int Relation_Type)
{
    static POINT      RelationPoint [4];
    static POINT      TempPoint [4];
    static RECT       Frame;
    static DWORD      dwExtent;
    static int        Flag = EDIT;
    static char       RelationBoxName[20];
    static int        Lenght;
    FARPROC           lpProc_Relation;

    switch (message)
    {
        case WM_LBUTTONDOWN:
        {
            HDC      hDC;
            HRGN     hRgn;

            hDC = GetDC (hDrawWnd);
            SetROP2 (hDC, R2_NOT);
            SelectObject (hDC, GetStockObject(NULL_BRUSH));
            if (Flag == EDIT)
            {
                if (Relation_Type == N_RELATION)
                {
                    lpProc_Relation = MakeProcInstance (nRelationDlgProc, hInst);
                    lstrcpy (RelationBoxName, "nRelationBox");
                    Lenght = 2;
                }
                else
                {
                    lpProc_Relation = MakeProcInstance (iRelationDlgProc, hInst);
                    lstrcpy (RelationBoxName, "iRelationBox");
                    Lenght = 3;
                }
                for (;;)
                {
                    if (DialogBox(hInst,RelationBoxName, hMainWnd, lpProc_Relation) == 0)
                        break;
                    if (IsBlankString (NameBuffer) == FALSE)
                    {
                        lstrcpy (Relation_Name.arr, NameBuffer);
                        Relation_Name.len = strlen (Relation_Name.arr);
                        Type = Relation_Type;
                        EXEC SQL INSERT INTO RELATION (RELATION_NAME, RELATION_TYPE)
                        VALUES (:Relation_Name, :Type);
                        if (ProcessOracleErrorCode ("INSERT RELATION") == FALSE)
                        {
                            lstrcpy (NameBuffer, "");
                            continue;
                        }
                        SetCursorCenterPos (hDrawWnd);
                        dwExtent = GetTextExtent (hDC, NameBuffer, strlen (NameBuffer));
                        SetRect
                        (
                            &Frame,
                            (DrawWndRect.right-DrawWndRect.left)/2 - (LOWORD (dwExtent))/2 - H_OFFSET*2,
                            (DrawWndRect.bottom-DrawWndRect.top)/2 - (HIWORD (dwExtent))/2 - W_OFFSET*2,
                            (DrawWndRect.right-DrawWndRect.left)/2 + (LOWORD (dwExtent))/2 + H_OFFSET*2,
                            (DrawWndRect.bottom-DrawWndRect.top)/2 + (HIWORD (dwExtent))/2 + W_OFFSET*2
                        );
                        RelationPoint[0].x = Frame.left;
                        RelationPoint[0].y = (Frame.bottom-Frame.top)/2 + Frame.top;
                        RelationPoint[1].x = (Frame.right-Frame.left)/2 + Frame.left;
                        RelationPoint[1].y = Frame.top;
                        RelationPoint[2].x = Frame.right;
                        RelationPoint[2].y = (Frame.bottom-Frame.top)/2 + Frame.top;
                        RelationPoint[3].x = (Frame.right-Frame.left)/2 + Frame.left;
                        RelationPoint[3].y = Frame.bottom;
                        Polygon (hDC, RelationPoint, sizeof (RelationPoint) / sizeof (POINT));
                        Flag = MOVE;
                        break;
                    }
                }
                FreeProcInstance(lpProc_Relation);
            }
            else
            {
                TempPoint[0].x = Frame.left+HorzThumbPos;
                TempPoint[0].y = (Frame.bottom-Frame.top)/2 + Frame.top+VertThumbPos;
                TempPoint[1].x = (Frame.right-Frame.left)/2 + Frame.left+HorzThumbPos;
                TempPoint[1].y = Frame.top+VertThumbPos;
                TempPoint[2].x = Frame.right+HorzThumbPos;
                TempPoint[2].y = (Frame.bottom-Frame.top)/2 + Frame.top+VertThumbPos;
                TempPoint[3].x = (Frame.right-Frame.left)/2 + Frame.left+HorzThumbPos;
                TempPoint[3].y = Frame.bottom+VertThumbPos;
                hRgn = CreatePolygonRgn (TempPoint, sizeof (TempPoint) / sizeof (POINT), WINDING);
                if (IsValidPosition (hRgn) == TRUE)
                {
                    Frame.left += HorzThumbPos;
                    Frame.top += VertThumbPos;
                    Frame.right += HorzThumbPos;
                    Frame.bottom += VertThumbPos;
                    RelationPoint[0].x = Frame.left;
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RelationPoint[0].y = (Frame.bottom-Frame.top)/2 + Frame.top;
RelationPoint[1].x = (Frame.right-Frame.left)/2 + Frame.left;
RelationPoint[1].y = Frame.top;
RelationPoint[2].x = Frame.right;
RelationPoint[2].y = (Frame.bottom-Frame.top)/2 + Frame.top;
RelationPoint[3].x = (Frame.right-Frame.left)/2 + Frame.left;
RelationPoint[3].y = Frame.bottom;
if (AppendNode (hRgn,Length,Relation_Type,NameBuffer,-1,"") == TRUE)
{
    Polygon (ErMetaFile.hMetaDC,RelationPoint,sizeof (RelationPoint) / sizeof (POINT)) ;
    if (Relation_Type == I_RELATION)
    {
        Frame.left += BORDER ;
        Frame.top += BORDER ;
        Frame.right -= BORDER ;
        Frame.bottom -= BORDER ;
        RelationPoint[0].x = Frame.left ;
        RelationPoint[0].y = (Frame.bottom-Frame.top)/2 + Frame.top;
        RelationPoint[1].x = (Frame.right-Frame.left)/2 + Frame.left;
        RelationPoint[1].y = Frame.top;
        RelationPoint[2].x = Frame.right;
        RelationPoint[2].y = (Frame.bottom-Frame.top)/2 + Frame.top;
        RelationPoint[3].x = (Frame.right-Frame.left)/2 + Frame.left;
        RelationPoint[3].y = Frame.bottom;
        Polygon (ErMetaFile.hMetaDC,RelationPoint,sizeof (RelationPoint) / sizeof (POINT)) ;
    }
    TextOut
    (
        ErMetaFile.hMetaDC,(Frame.right+Frame.left)/2 - (LOWORD (dwExtent))/2,
        (Frame.bottom+Frame.top)/2 - (HIWORD (dwExtent))/2,NameBuffer,Isbrien (NameBuffer)
    );
    InvalidateRect (hDrawWnd,NULL,1);
    UpdateWindow (hDrawWnd);
    Flag = EDIT ;
    EXEC SQL COMMIT WORK ;
}
else
{
    DeleteObject (hRgn) ;
    EXEC SQL ROLLBACK WORK ;
}
ReleaseCapture () ;
}
else
{
    Flag = MOVE ;
    DeleteObject (hRgn) ;
}
}
ReleaseDC (hDrawWnd,hDC) ;
}
break ;
case WM_MOUSEMOVE:
{
    HDC hDC ;
    Int NextX,NextY ;
    BOOL ThumbPosChange ;

    if (Flag == MOVE)
    {
        SetCapture (hDrawWnd) ;
        NextX = LOWORD (iParam) ;
        NextY = HIWORD (iParam) ;
        ThumbPosChange = FALSE ;
        if (NextX < DrawWndRect.left)
        {
            if (HorzThumbPos < 0)
                HorzThumbPos = 0 ;
            else
            {
                HorzThumbPos-- ;
                ThumbPosChange = TRUE ;
            }
        }
        else if (NextX >= DrawWndRect.right)
        {
            if (HorzThumbPos >= H_SCROLL_RANGE)
                HorzThumbPos = H_SCROLL_RANGE ;
            else
            {
                HorzThumbPos++ ;
                ThumbPosChange = TRUE ;
            }
        }
        if (NextY < DrawWndRect.top)
        {
            if (VertThumbPos < 0)
                VertThumbPos = 0 ;
            else
            {
                VertThumbPos-- ;
                ThumbPosChange = TRUE ;
            }
        }
        else if (NextY >= DrawWndRect.bottom)
        {
            if (VertThumbPos >= V_SCROLL_RANGE)
                VertThumbPos = V_SCROLL_RANGE ;
            else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        VertThumbPos++;
        ThumbPosChange = TRUE;
    }
}
hDC = GetDC (hDrawWnd);
SelectObject (hDC,GetStockObject(NULL_BRUSH));
SetROP2 (hDC,R2_NOT);
Polygon (hDC,RelationPoint,sizeof (RelationPoint) / sizeof (POINT));
if (ThumbPosChange)
{
    SetScrollPos (hDrawWnd,SB_VERT,VertThumbPos,TRUE);
    SetScrollPos (hDrawWnd,SB_HORZ,HorzThumbPos,TRUE);
    InvalidateRect (hDrawWnd,NULL,1);
    UpdateWindow (hDrawWnd);
}
SetRect
(
    &Frame,NextX - (LOWORD (dwExtent))/2 - H_OFFSET*2,NextY - (HIWORD (dwExtent))/2 - W_OFFSET*2,
    NextX + (LOWORD (dwExtent))/2 + H_OFFSET*2,NextY + (HIWORD (dwExtent))/2 + W_OFFSET*2
);
RelationPoint[0].x = Frame.left;
RelationPoint[0].y = (Frame.bottom-Frame.top)/2 + Frame.top;
RelationPoint[1].x = (Frame.right-Frame.left)/2 + Frame.left;
RelationPoint[1].y = Frame.top;
RelationPoint[2].x = Frame.right;
RelationPoint[2].y = (Frame.bottom-Frame.top)/2 + Frame.top;
RelationPoint[3].x = (Frame.right-Frame.left)/2 + Frame.left;
RelationPoint[3].y = Frame.bottom;
Polygon (hDC,RelationPoint,sizeof (RelationPoint) / sizeof (POINT));
ReleaseDC (hDrawWnd,hDC);
}
}
break;
}
}
}

```

```

BOOL FAR PASCAL _export nRelationDlgProc (HWND hDlg,unsigned message,WORD wParam,LPARAM lParam)
{
    switch (message)
    {
        case WM_INITDIALOG:
        {
            SetDlgItemText (hDlg, IDC_EDIT,NULL);
            return (TRUE);
        }
        case WM_COMMAND:
        {
            switch (wParam)
            {
                case IDOK:
                {
                    GetDlgItemText (hDlg, IDC_EDIT,NameBuffer,30);
                    EndDialog (hDlg,TRUE);
                    return (TRUE);
                }
                case IDCANCEL:
                {
                    EndDialog (hDlg,FALSE);
                    return (FALSE);
                }
            }
        }
    }
    return (FALSE);
}

```

```

BOOL FAR PASCAL _export lRelationDlgProc (HWND hDlg,unsigned message,WORD wParam,LPARAM lParam)
{
    switch (message)
    {
        case WM_INITDIALOG:
        {
            SetDlgItemText (hDlg, IDC_EDIT,NULL);
            return (TRUE);
        }
        case WM_COMMAND:
        {
            switch (wParam)
            {
                case IDOK:
                {
                    GetDlgItemText (hDlg, IDC_EDIT,NameBuffer,30);
                    EndDialog (hDlg,TRUE);
                    return (TRUE);
                }
                case IDCANCEL:
                {
                    EndDialog (hDlg,FALSE);
                    return (FALSE);
                }
            }
        }
    }
    return (FALSE);
}

```

```

void N_ATTRIBUTE_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LPARAM lParam)
{
    ATTRIBUTE_PROC (hDrawWnd,message,wParam,lParam,N_ATTRIBUTE);
}

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void K_ATTRIBUTE_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)
{
    ATTRIBUTE_PROC (hDrawWnd,message,wParam,lParam,K_ATTRIBUTE);
}
void P_ATTRIBUTE_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)
{
    ATTRIBUTE_PROC (hDrawWnd,message,wParam,lParam,P_ATTRIBUTE);
}
void MV_ATTRIBUTE_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)
{
    ATTRIBUTE_PROC (hDrawWnd,message,wParam,lParam,MV_ATTRIBUTE);
}

void ATTRIBUTE_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam, int Attr_Type)
{
    static int      Width;
    static int      Height;
    static RECT     Frame;
    static DWORD    cwExtent;
    static DWORD    dwExtent;
    static int      Flag = EDIT;
    static char     DialogBoxName[20];
    static int      Length;
    FARPROC        lpProc_Attribute;

    switch (message)
    {
        case WM_LBUTTONDOWN:
        {
            HDC      hDC;
            HRGN     hRgn;

            hDC = GetDC (hDrawWnd);
            SetROP2 (hDC, R2_NOT);
            SelectObject (hDC, GetStockObject(NULL_BRUSH));
            if (Flag == EDIT)
            {
                switch (Attr_Type)
                {
                    case N_ATTRIBUTE:
                    {
                        lpProc_Attribute = MakeProcInstance (nAttributeDlgProc, hInst);
                        lstrcpy (DialogBoxName, "nAttributeBox");
                        Length = 5;
                    }
                    break;
                    case K_ATTRIBUTE:
                    {
                        lpProc_Attribute = MakeProcInstance (kAttributeDlgProc, hInst);
                        lstrcpy (DialogBoxName, "kAttributeBox");
                        Length = 7;
                    }
                    break;
                    case P_ATTRIBUTE:
                    {
                        lpProc_Attribute = MakeProcInstance (pAttributeDlgProc, hInst);
                        lstrcpy (DialogBoxName, "pAttributeBox");
                        Length = 7;
                    }
                    break;
                    case MV_ATTRIBUTE:
                    {
                        lpProc_Attribute = MakeProcInstance (mvAttributeDlgProc, hInst);
                        lstrcpy (DialogBoxName, "mvAttributeBox");
                        Length = 8;
                    }
                    break;
                }
            }
            for (;;)
            {
                if (DialogBox(hInst, DialogBoxName, hMainWnd, lpProc_Attribute) == 0)
                    break;
                if ((IsBlankString (NameBuffer) == FALSE) && (IsBlankString (DataTypeBuffer) == FALSE))
                {
                    if (IsValidOracle (NameBuffer, DataTypeBuffer, DomainBuffer) == TRUE)
                    {
                        EXEC SQL SELECT MAX(ATTR_NUMBER) INTO :Attr_Number FROM ATTR;
                        lstrcpy (Attr_Name.arr, NameBuffer);
                        Attr_Name.len = lstrlen (Attr_Name.arr);
                        lstrcpy (DataType.arr, DataTypeBuffer);
                        DataType.len = lstrlen (DataType.arr);
                        Type = Attr_Type;
                        Attr_Number++;
                        EXEC SQL INSERT INTO ATTR
                        (
                            ATTR_NUMBER,ATTR_NAME,ATTR_TYPE,ATTR_DATATYPE,
                            ENTITY_NAME,RELATION_NAME,COMPOSIT_NUMBER
                        )
                        VALUES (:Attr_Number,:Attr_Name,:Type,:DataType,null,null,-1);
                        if (ProcessOracleErrorCode ("INSERT N_ATTRIBUTE") == FALSE)
                        {
                            continue;
                        }
                        SetCursorCenterPos (hDrawWnd);
                        if (Attr_Type == P_ATTRIBUTE)
                            lstrcat (NameBuffer, "");
                        dwExtent = GetTextExtent (hDC, NameBuffer, lstrlen (NameBuffer));
                        cwExtent = GetTextExtent (hDC, DomainBuffer, lstrlen (DomainBuffer));
                        if (IsBlankString (DomainBuffer) == FALSE)
                    }
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (ConstraintFlag == CONSTRAINT_CHAR)
            SaveStringConstraint (Attr_Number);
        if (ConstraintFlag == CONSTRAINT_NUMBER)
            SaveNumberConstraint (Attr_Number);
        Height = HIWORD (dwExtent) + HIWORD (cwExtent);
        Width = (LOWORD (cwExtent) > LOWORD (dwExtent)) ?
            LOWORD (cwExtent) : LOWORD (dwExtent);
    }
    else
    {
        Height = HIWORD (dwExtent);
        Width = LOWORD (dwExtent);
    }
    if (Attr_Type == MV_ATTRIBUTE)
    {
        Height += BORDER;
        Width += BORDER;
    }
    else
    {
        Height -= BORDER;
        Width -= BORDER;
    }
    SetRect
    (
        &Frame, (DrawWndRect.right-DrawWndRect.left)/2 - Width/2 - H_OFFSET,
        (DrawWndRect.bottom-DrawWndRect.top)/2 - Height/2 - W_OFFSET,
        (DrawWndRect.right-DrawWndRect.left)/2 + Width/2 + H_OFFSET,
        (DrawWndRect.bottom-DrawWndRect.top)/2 + Height/2 + W_OFFSET
    );
    Ellipse (hDC, Frame.left, Frame.top, Frame.right, Frame.bottom);
    Flag = MOVE;
    break;
}
}
}
FreeProcInstance (pProc_Attribute);
}
else
{
    hRgn = CreateEllipticRgn
    (
        Frame.left+HorzThumbPos, Frame.top +VertThumbPos,
        Frame.right+HorzThumbPos, Frame.bottom+VertThumbPos
    );
    if (IsValidPosition (hRgn) == TRUE)
    {
        Frame.left += HorzThumbPos;
        Frame.top += VertThumbPos;
        Frame.right += HorzThumbPos;
        Frame.bottom += VertThumbPos;
        if (AppendNode (hRgn, Length, Attr_Type, NameBuffer, Attr_Number, DomainBuffer) == TRUE)
        {
            MoveTo (ErMetaFile.hMetaDC, 0, 0);
            LineTo (ErMetaFile.hMetaDC, 0, 0);
            Ellipse (ErMetaFile.hMetaDC, Frame.left, Frame.top, Frame.right, Frame.bottom);
            if (Attr_Type == MV_ATTRIBUTE)
            {
                Ellipse
                (
                    ErMetaFile.hMetaDC, Frame.left+BORDER, Frame.top+BORDER,
                    Frame.right-BORDER, Frame.bottom-BORDER
                );
            }
            if (IsBlankString (DomainBuffer) == FALSE)
            {
                TextOut
                (
                    ErMetaFile.hMetaDC, (Frame.right+Frame.left)/2 - LOWORD (cwExtent)/2,
                    (Frame.bottom+Frame.top)/2, DomainBuffer, lstrlen (DomainBuffer)
                );
            }
            TextOut
            (
                ErMetaFile.hMetaDC, (Frame.right+Frame.left)/2 - LOWORD (dwExtent)/2,
                (Frame.bottom+Frame.top)/2 - HIWORD (dwExtent), NameBuffer, lstrlen (NameBuffer)
            );
        }
        else
        {
            TextOut
            (
                ErMetaFile.hMetaDC, (Frame.right+Frame.left)/2 - LOWORD (cwExtent)/2,
                (Frame.bottom+Frame.top)/2 - HIWORD (cwExtent)/2, DomainBuffer, lstrlen (DomainBuffer)
            );
            TextOut
            (
                ErMetaFile.hMetaDC, (Frame.right+Frame.left)/2 - LOWORD (dwExtent)/2,
                (Frame.bottom+Frame.top)/2 - HIWORD (dwExtent)/2, NameBuffer, lstrlen (NameBuffer)
            );
        }
    }
    if (Attr_Type == P_ATTRIBUTE || Attr_Type == K_ATTRIBUTE)
    {
        MoveTo
        (
            ErMetaFile.hMetaDC, (Frame.right-Frame.left)*5/20 + Frame.left,
            (Frame.bottom-Frame.top)*8/10 + Frame.top
        );
        LineTo
        (
            ErMetaFile.hMetaDC, (Frame.right-Frame.left)*15/20 + Frame.left,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคคลที่อนุญาตเท่านั้น ไม่แนะนำให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        (Frame.bottom-Frame.top)*8/10 + Frame.top
    );
    }
    InvalidateRect (hDrawWnd,NULL,1);
    UpdateWindow (hDrawWnd);
    Flag = EDIT;
    EXEC SQL COMMIT WORK;
}
else
{
    DeleteObject (hRgn);
    EXEC SQL ROLLBACK WORK;
}
ReleaseCapture ();
}
else
{
    Flag = MOVE;
    DeleteObject (hRgn);
}
}
ReleaseDC (hDrawWnd,hDC);
}
break;
case WM_MOUSEMOVE:
{
    HDC hDC;
    int NextX,NextY;
    BOOL ThumbPosChange;

    if (Flag == MOVE)
    {
        SetCapture (hDrawWnd);
        NextX = LOWORD (lParam);
        NextY = HIWORD (lParam);
        ThumbPosChange = FALSE;
        if (NextX < DrawWndRect.left)
        {
            if (HorzThumbPos < 0)
                HorzThumbPos = 0;
            else
            {
                HorzThumbPos--;
                ThumbPosChange = TRUE;
            }
        }
        else if (NextX >= DrawWndRect.right)
        {
            if (HorzThumbPos >= H_SCROLL_RANGE)
                HorzThumbPos = H_SCROLL_RANGE;
            else
            {
                HorzThumbPos++;
                ThumbPosChange = TRUE;
            }
        }
        if (NextY < DrawWndRect.top)
        {
            if (VertThumbPos < 0)
                VertThumbPos = 0;
            else
            {
                VertThumbPos--;
                ThumbPosChange = TRUE;
            }
        }
        else if (NextY >= DrawWndRect.bottom)
        {
            if (VertThumbPos >= V_SCROLL_RANGE)
                VertThumbPos = V_SCROLL_RANGE;
            else
            {
                VertThumbPos++;
                ThumbPosChange = TRUE;
            }
        }
        hDC = GetDC (hDrawWnd);
        SelectObject (hDC,GetStockObject(NULL_BRUSH));
        SetROP2 (hDC,R2_NOT);
        Ellipse (hDC,Frame.left,Frame.top,Frame.right,Frame.bottom);
        if (ThumbPosChange)
        {
            SetScrollPos (hDrawWnd,SB_VERT,VertThumbPos,TRUE);
            SetScrollPos (hDrawWnd,SB_HORZ,HorzThumbPos,TRUE);
            InvalidateRect (hDrawWnd,NULL,1);
            UpdateWindow (hDrawWnd);
        }
        SetRect
        (
            &Frame,NextX - Width/2 - H_OFFSET,NextY - Height/2 - W_OFFSET,
            NextX + Width/2 + H_OFFSET,NextY + Height/2 + W_OFFSET
        );
        Ellipse (hDC,Frame.left,Frame.top,Frame.right,Frame.bottom);
        ReleaseDC (hDrawWnd,hDC);
    }
}
break;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BOOL FAR PASCAL _export nAttributeDlgProc (HWND hDlg, unsigned message, WORD wParam, LONG lParam)
{
    switch (message)
    {
        case WM_INITDIALOG:
        {
            SetDlgItemText (hDlg, IDC_NAME, NULL) ;
            SetDlgItemText (hDlg, IDC_DATATYPE, NULL) ;
            SetDlgItemText (hDlg, IDC_DOMAIN, NULL) ;
            lstrcpy (NameBuffer, "");
            lstrcpy (DataTypeBuffer, "");
            lstrcpy (DomainBuffer, "");
            return (TRUE) ;
        }
        case WM_COMMAND:
        {
            switch (wParam)
            {
                case IDOK:
                {
                    GetDlgItemText (hDlg, IDC_NAME, NameBuffer, 30) ;
                    GetDlgItemText (hDlg, IDC_DATATYPE, DataTypeBuffer, 20) ;
                    GetDlgItemText (hDlg, IDC_DOMAIN, DomainBuffer, 120) ;
                    AnsiUpper (DataTypeBuffer) ;
                    EndDialog (hDlg, TRUE) ;
                    return (TRUE) ;
                }
                case IDCANCEL:
                {
                    EndDialog (hDlg, FALSE) ;
                    return (FALSE) ;
                }
            }
        }
    }
    return (FALSE) ;
}

```

```

BOOL FAR PASCAL _export kAttributeDlgProc (HWND hDlg, unsigned message, WORD wParam, LONG lParam)
{
    switch (message)
    {
        case WM_INITDIALOG:
        {
            SetDlgItemText (hDlg, IDC_NAME, NULL) ;
            SetDlgItemText (hDlg, IDC_DATATYPE, NULL) ;
            SetDlgItemText (hDlg, IDC_DOMAIN, NULL) ;
            lstrcpy (NameBuffer, "");
            lstrcpy (DataTypeBuffer, "");
            lstrcpy (DomainBuffer, "");
            return (TRUE) ;
        }
        case WM_COMMAND:
        {
            switch (wParam)
            {
                case IDOK:
                {
                    GetDlgItemText (hDlg, IDC_NAME, NameBuffer, 30) ;
                    GetDlgItemText (hDlg, IDC_DATATYPE, DataTypeBuffer, 20) ;
                    GetDlgItemText (hDlg, IDC_DOMAIN, DomainBuffer, 120) ;
                    AnsiUpper (DataTypeBuffer) ;
                    EndDialog (hDlg, TRUE) ;
                    return (TRUE) ;
                }
                case IDCANCEL:
                {
                    EndDialog (hDlg, FALSE) ;
                    return (FALSE) ;
                }
            }
        }
    }
    return (FALSE) ;
}

```

```

BOOL FAR PASCAL _export pAttributeDlgProc (HWND hDlg, unsigned message, WORD wParam, LONG lParam)
{
    switch (message)
    {
        case WM_INITDIALOG:
        {
            SetDlgItemText (hDlg, IDC_NAME, NULL) ;
            SetDlgItemText (hDlg, IDC_DATATYPE, NULL) ;
            SetDlgItemText (hDlg, IDC_DOMAIN, NULL) ;
            lstrcpy (NameBuffer, "");
            lstrcpy (DataTypeBuffer, "");
            lstrcpy (DomainBuffer, "");
            return (TRUE) ;
        }
        case WM_COMMAND:
        {
            switch (wParam)
            {
                case IDOK:
                {
                    GetDlgItemText (hDlg, IDC_NAME, NameBuffer, 30) ;
                    GetDlgItemText (hDlg, IDC_DATATYPE, DataTypeBuffer, 20) ;
                    GetDlgItemText (hDlg, IDC_DOMAIN, DomainBuffer, 120) ;
                    AnsiUpper (DataTypeBuffer) ;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสำนักงานคณะกรรมการกฤษฎีกา ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        EndDialog (hDlg,TRUE);
        return (TRUE);
    }
    case IDCANCEL:
    {
        EndDialog (hDlg,FALSE);
        return (FALSE);
    }
}
}
return (FALSE);
}

```

BOOL FAR PASCAL \_export mvAttributeDlgProc (HWND hDlg, unsigned message, WORD wParam, LONG lParam)

```

{
    switch (message)
    {
        case WM_INITDIALOG:
        {
            SetDlgItemText (hDlg, IDC_NAME, NULL);
            SetDlgItemText (hDlg, IDC_DATATYPE, NULL);
            SetDlgItemText (hDlg, IDC_DOMAIN, NULL);
            strcpy (NameBuffer, "");
            strcpy (DataTypeBuffer, "");
            strcpy (DomainBuffer, "");
            return (TRUE);
        }
        case WM_COMMAND:
        {
            switch (wParam)
            {
                case IDOK:
                {
                    GetDlgItemText (hDlg, IDC_NAME, NameBuffer, 30);
                    GetDlgItemText (hDlg, IDC_DATATYPE, DataTypeBuffer, 20);
                    GetDlgItemText (hDlg, IDC_DOMAIN, DomainBuffer, 120);
                    AnsiUpper (DataTypeBuffer);
                    EndDialog (hDlg, TRUE);
                    return (TRUE);
                }
                case IDCANCEL:
                {
                    EndDialog (hDlg, FALSE);
                    return (FALSE);
                }
            }
        }
    }
    return (FALSE);
}

```

void LINK\_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)

```

{
    static int fFromPoint = EDIT;
    static int fToPoint = EDIT;

    switch (message)
    {
        case WM_LBUTTONDOWN:
        {
            int Object;
            static POINT Point;

            Point.x = LOWORD (lParam) + HorzThumbPos;
            Point.y = HIWORD (lParam) + VertThumbPos;
            Object = WhatObjectSelected (Point.x, Point.y);
            switch (Object)
            {
                case N_ATTRIBUTE:
                {
                    if (fFromPoint == EDIT)
                    {
                        FromPoint.x = Point.x;
                        FromPoint.y = Point.y;
                        FirstObjectPoint.x = Point.x;
                        FirstObjectPoint.y = Point.y;
                        fFromPoint = N_ATTRIBUTE;
                    }
                    else
                    {
                        ToPoint.x = Point.x;
                        ToPoint.y = Point.y;
                        SecondObjectPoint.x = Point.x;
                        SecondObjectPoint.y = Point.y;
                        fToPoint = N_ATTRIBUTE;
                    }
                }
                break;
                case MV_ATTRIBUTE:
                case P_ATTRIBUTE:
                case K_ATTRIBUTE:
                {
                    if (fFromPoint == EDIT)
                    {
                        FromPoint.x = Point.x;
                        FromPoint.y = Point.y;
                        FirstObjectPoint.x = Point.x;
                        FirstObjectPoint.y = Point.y;
                        fFromPoint = K_ATTRIBUTE;
                    }
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    if (fFromPoint == N_ATTRIBUTE)
    {
        ToPoint.x = Point.x;
        ToPoint.y = Point.y;
        SecondObjectPoint.x = Point.x;
        SecondObjectPoint.y = Point.y;
        fToPoint = K_ATTRIBUTE;
    }
}
break;
case R_ENTITY:
case W_ENTITY:
{
    ToPoint.x = Point.x;
    ToPoint.y = Point.y;
    SecondObjectPoint.x = Point.x;
    SecondObjectPoint.y = Point.y;
    fToPoint = R_ENTITY;
}
break;
case N_RELATION:
case I_RELATION:
{
    ToPoint.x = Point.x;
    ToPoint.y = Point.y;
    SecondObjectPoint.x = Point.x;
    SecondObjectPoint.y = Point.y;
    fToPoint = N_RELATION;
}
break;
}
if (
    ((fFromPoint == N_ATTRIBUTE && fToPoint == R_ENTITY) ||
     (fFromPoint == K_ATTRIBUTE && fToPoint == R_ENTITY) ||
     (fFromPoint == MV_ATTRIBUTE && fToPoint == R_ENTITY) ||
     (fFromPoint == N_ATTRIBUTE && fToPoint == N_RELATION) ||
     (fFromPoint == N_ATTRIBUTE && fToPoint == N_ATTRIBUTE) ||
     (fFromPoint == N_ATTRIBUTE && fToPoint == K_ATTRIBUTE) ||
     (fFromPoint == N_ATTRIBUTE && fToPoint == MV_ATTRIBUTE)
    )
)
{
    NODE *Link,*Attr;
    HGLOBAL hLinkMem,hAttrMem;

    hLinkMem = IsObjectSelected (SecondObjectPoint.x,SecondObjectPoint.y);
    hAttrMem = IsObjectSelected (FirstObjectPoint.x,FirstObjectPoint.y);
    Link = (NODE*) GlobalLock (hLinkMem);
    Attr = (NODE*) GlobalLock (hAttrMem);
    switch (Link->ObjectType)
    {
        case R_ENTITY:
        case W_ENTITY:
        {
            Istrcpy (Entity_Name.arr,Link->ObjectName);
            Entity_Name.len = Istrlen (Entity_Name.arr);
            Attr_Number = Attr->Number;
            EXEC SQL UPDATE ATTR SET
            ENTITY_NAME=:Entity_Name,RELATION_NAME=:null,COMPOSIT_NUMBER=:1
            WHERE ATTR_NUMBER=:Attr_Number;
            if (ProcessOracleErrorCode ("UPDATE ATTRIBUTE ENTITY") == FALSE)
                break;
            EXEC SQL COMMIT WORK;
            if (ProcessOracleErrorCode ("UPDATE ATTRIBUTE ENTITY COMMIT") == FALSE)
                break;
            fFromPoint = EDIT;
            fToPoint = EDIT;
            AdjustLineCenter (&fFromPoint,&fToPoint);
            UpdateMetaFile ();
            InvalidateRect (hDrawWnd,NULL,1);
            UpdateWindow (hDrawWnd);
        }
        break;
        case N_RELATION:
        case I_RELATION:
        {
            Istrcpy (Relation_Name.arr,Link->ObjectName);
            Relation_Name.len = Istrlen (Relation_Name.arr);
            Attr_Number = Attr->Number;
            EXEC SQL UPDATE ATTR SET
            ENTITY_NAME=:null,RELATION_NAME=:Relation_Name,COMPOSIT_NUMBER=:1
            WHERE ATTR_NUMBER=:Attr_Number;
            if (ProcessOracleErrorCode ("UPDATE ATTRIBUTE RELATION") == FALSE)
                break;
            EXEC SQL COMMIT WORK;
            if (ProcessOracleErrorCode ("UPDATE ATTRIBUTE RELATION COMMIT") == FALSE)
                break;
            fFromPoint = EDIT;
            fToPoint = EDIT;
            AdjustLineCenter (&fFromPoint,&fToPoint);
            UpdateMetaFile ();
            InvalidateRect (hDrawWnd,NULL,1);
            UpdateWindow (hDrawWnd);
        }
        break;
        case N_ATTRIBUTE:
        case K_ATTRIBUTE:
        case P_ATTRIBUTE:
        case MV_ATTRIBUTE:
    }
}

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Composit_Number = Link->Number ;
Attr_Number = Attr->Number ;
EXEC SQL UPDATE ATTR SET
ENTITY_NAME=null,RELATION_NAME=null,COMPOSIT_NUMBER=:Composit_Number
WHERE ATTR_NUMBER=:Attr_Number;
if (ProcessOracleErrorCode ("UPDATE ATTRIBUTE COMPOSIT") == FALSE)
break ;
EXEC SQL COMMIT WORK ;
if (ProcessOracleErrorCode ("UPDATE ATTRIBUTE COMPOSIT COMMIT") == FALSE)
break ;
fFromPoint = EDIT ;
fToPoint = EDIT ;
AdjustLineCenter (&FromPoint,&ToPoint) ;
UpdateMetaFile () ;
InvalidateRect (hDrawWnd,NULL,1) ;
UpdateWindow (hDrawWnd) ;
}
break ;
}
GlobalUnlock (hLinkMem) ;
GlobalUnlock (hAttrMem) ;
}
break ;
}
}
)

```

```
void UpdateMetaFile (void)
```

```
{
NODE *Temp ;
HGLOBAL hMem ;
MFENUMPROC lpUpdateMetaProc ;
HMETAFILE hMF ;
```

```

hMem = IsObjectSelected (FirstObjectPoint.x,FirstObjectPoint.y) ;
Temp = (NODE*) GlobalLock (hMem) ;
hMF = CloseMetaFile (ErMetaFile.hMetaDC) ;
MetaRecCount = 0 ;
ErMetaFile.hMetaDC = CreateMetaFile (NULL) ;
lpUpdateMetaProc = (MFENUMPROC) MakeProcInstance ((FARPROC) UpdateMetaFileProc,hInst) ;
EnumMetaFile (ErMetaFile.hMetaDC,hMF,lpUpdateMetaProc,(LPARAM) Temp) ;
DeleteMetaFile (hMF) ;
FreeProcInstance ((FARPROC) lpUpdateMetaProc) ;
GlobalUnlock (hMem) ;
}

```

```
int FAR PASCAL UpdateMetaFileProc
```

```
{
HDC hMetaDC,HANDLETABLE FAR* lpHTable, METARECORD FAR* lpMFR,int cObj, BYTE FAR* lpClientData
}

```

```
{
NODE *Temp ;
```

```

MetaRecCount ++ ;
Temp = (NODE FAR*) lpClientData ;
if (MetaRecCount == Temp->MetaStartRec)
{
lpMFR -> rdParm [0] = FromPoint.y ;
lpMFR -> rdParm [1] = FromPoint.x ;
}
if (MetaRecCount == (Temp->MetaStartRec)+1)
{
lpMFR -> rdParm [0] = ToPoint.y ;
lpMFR -> rdParm [1] = ToPoint.x ;
}
PlayMetaFileRecord (hMetaDC,lpHTable,lpMFR,cObj) ;
return (TRUE) ;
}

```

```
void ROLE_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam,LPSTR Cadinality,int Length)
```

```
{
static int fFromPoint = EDIT ;
static int fToPoint = EDIT ;

switch (message)
{
case WM_LBUTTONDOWN:
{
int Object ;
POINT Point ;

Point.x = LOWORD (lParam) + HorzThumbPos ;
Point.y = HIWORD (lParam) + VertThumbPos ;
Object = WhatObjectSelected (Point.x,Point.y) ;
switch (Object)
{
case R_ENTITY:
case W_ENTITY:
{
FromPoint.x = Point.x ;
FromPoint.y = Point.y ;
FirstObjectPoint.x = Point.x ;
FirstObjectPoint.y = Point.y ;
fFromPoint = TRUE ;
}
break ;
case N_RELATION:
case I_RELATION:
{
ToPoint.x = Point.x ;

```

เอกสารนี้เป็นเอกสารที่... ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        ToPoint.y = Point.y ;
        SecondObjectPoint.x = Point.x ;
        SecondObjectPoint.y = Point.y ;
        fToPoint = TRUE ;
    }
    break ;
}
}
if (fFromPoint == TRUE && fToPoint == TRUE)
{
    static    DWORD    dwExtent ;
    HDC      hDC ;
    HRGN     hRgn ;
    POINT    BeginPoint1,BeginPoint2,EndPoint1,EndPoint2 ;

    fToPoint = EDIT ;
    fFromPoint = EDIT ;
    hDC = GetDC (hDrawWnd) ;
    dwExtent = GetTextExtent (hDC,Cadinality,Istrlen (Cadinality)) ;
    ReleaseDC (hDrawWnd,hDC) ;
    if (Partial == TOTAL_1)
        CreateDoubleLine (FromPoint,ToPoint,&BeginPoint1,&EndPoint1,&BeginPoint2,&EndPoint2) ;
    else
        AdjustLineNormal (&FromPoint,&ToPoint) ;
    hRgn = CreateRectRgn
    (
        (ToPoint.x+FromPoint.x)/2 - (LOWORD (dwExtent))/2,(ToPoint.y+FromPoint.y)/2 - (HIWORD (dwExtent))/2,
        (ToPoint.x+FromPoint.x)/2 + (LOWORD (dwExtent))/2,(ToPoint.y+FromPoint.y)/2 + (HIWORD (dwExtent))/2
    );
    if (IsValidPosition (hRgn) == TRUE)
    {
        NODE    *Entity,*Relation ;
        HGLOBAL  hEntityMem,hRelationMem ;

        EXEC SQL SELECT MAX (ROLE_NUMBER) INTO :Role_Number FROM ROLE ;
        if (ProcessOracleErrorCode ("Select Count ROLE") == FALSE)
            break ;
        if (Istrcmp (Cadinality,"1") == 0) Type = 0 ;
        if (Istrcmp (Cadinality,"N") == 0) Type = 1 ;
        if (Istrcmp (Cadinality,"M") == 0) Type = 1 ;
        hEntityMem = IsObjectSelected (FirstObjectPoint.x,FirstObjectPoint.y) ;
        hRelationMem = IsObjectSelected (SecondObjectPoint.x,SecondObjectPoint.y) ;
        Entity = (NODE*) GlobalLock (hEntityMem) ;
        Relation = (NODE*) GlobalLock (hRelationMem) ;
        Istrcpy (Entity_Name.arr,Entity->ObjectName) ;
        Entity_Name.len = Istrlen (Entity_Name.arr) ;
        Istrcpy (Relation_Name.arr,Relation->ObjectName) ;
        Relation_Name.len = Istrlen (Relation_Name.arr) ;
        Role_Number++ ;
        EXEC SQL INSERT INTO ROLE
        (ROLE_NUMBER,PARTICIPATION,CADINALITY,ENTITY_NAME,RELATION_NAME)
        VALUES (:Role_Number,:Partial,:Type,:Entity_Name,:Relation_Name) ;
        if (ProcessOracleErrorCode ("Insert T_ROLE") == FALSE)
            break ;
        GlobalUnlock (hEntityMem) ;
        GlobalUnlock (hRelationMem) ;
        if (AppendNode (hRgn,Length,Partial,Cadinality,Role_Number,"") == TRUE)
        {
            if (Length == 5)
            {
                MoveTo (ErMetaFile.hMetaDC,BeginPoint1.x,BeginPoint1.y) ;
                LineTo (ErMetaFile.hMetaDC,EndPoint1.x,EndPoint1.y) ;
                MoveTo (ErMetaFile.hMetaDC,BeginPoint2.x,BeginPoint2.y) ;
                LineTo (ErMetaFile.hMetaDC,EndPoint2.x,EndPoint2.y) ;
            }
            else
            {
                MoveTo (ErMetaFile.hMetaDC,FromPoint.x,FromPoint.y) ;
                LineTo (ErMetaFile.hMetaDC,ToPoint.x,ToPoint.y) ;
            }
        }
        TextOut
        (
            ErMetaFile.hMetaDC,(ToPoint.x+FromPoint.x)/2 - (LOWORD (dwExtent))/2,
            (ToPoint.y+FromPoint.y)/2 - (HIWORD (dwExtent))/2,Cadinality,Istrlen (Cadinality)
        );
        InvalidateRect(hDrawWnd,NULL,1);
        UpdateWindow(hDrawWnd);
        EXEC SQL COMMIT WORK ;
    }
    else
    {
        DeleteObject (hRgn) ;
        EXEC SQL ROLLBACK WORK ;
    }
}
else
    DeleteObject (hRgn) ;
}
break ;
}
}
}

void PARTIAL_1_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)
{
    Partial = PARTIAL_1 ;
    ROLE_PROC (hDrawWnd,message,wParam,lParam,"1",3) ;
}
void PARTIAL_N_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)
{
    Partial = PARTIAL_1 ;

```

```

    ROLE_PROC (hDrawWnd,message,wParam,lParam,"N",3);
}
void PARTIAL_M_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)
{
    Partial = PARTIAL_1;
    ROLE_PROC (hDrawWnd,message,wParam,lParam,"M",3);
}
void TOTAL_1_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)
{
    Partial = TOTAL_1;
    ROLE_PROC (hDrawWnd,message,wParam,lParam,"1",5);
}
void TOTAL_N_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)
{
    Partial = TOTAL_1;
    ROLE_PROC (hDrawWnd,message,wParam,lParam,"N",5);
}
void TOTAL_M_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)
{
    Partial = TOTAL_1;
    ROLE_PROC (hDrawWnd,message,wParam,lParam,"M",5);
}

void TEXT_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)
{
    static RECT    Frame;
    static DWORD  dwExtent;
    static int     Flag = EDIT;
    FARPROC       lpProc_TEXT;

    switch (message)
    {
        case WM_LBUTTONDOWN:
        {
            HDC      hDC;
            HRGN     hRgn;

            hDC = GetDC (hDrawWnd);
            SetROP2 (hDC,R2_NOT);
            SelectObject (hDC,GetStockObject(NULL_BRUSH));
            if (Flag == EDIT)
            {
                Istropy (NameBuffer,"");
                lpProc_TEXT = MakeProcInstance (TextDlgProc, hInst);
                for (;;)
                {
                    if (DialogBox(hInst, "TextBox", hMainWnd, lpProc_TEXT) == 0)
                        break;
                    if (IsBlankString (NameBuffer) == FALSE)
                    {
                        SetCursorCenterPos (hDrawWnd);
                        dwExtent = GetTextExtent (hDC,NameBuffer,Istrien (NameBuffer));
                        SetRect
                        (
                            &Frame,
                            (DrawWndRect.right-DrawWndRect.left)/2 - (LOWORD (dwExtent))/2,
                            (DrawWndRect.bottom-DrawWndRect.top)/2 - (HIWORD (dwExtent))/2,
                            (DrawWndRect.right-DrawWndRect.left)/2 + (LOWORD (dwExtent))/2,
                            (DrawWndRect.bottom-DrawWndRect.top)/2 + (HIWORD (dwExtent))/2
                        );
                        Rectangle (hDC,Frame.left,Frame.top,Frame.right,Frame.bottom);
                        Flag = MOVE;
                        break;
                    }
                }
                FreeProcInstance(lpProc_TEXT);
            }
        }
        else
        {
            hRgn = CreateRectRgn
            (
                Frame.left+HorzThumbPos,Frame.top+VertThumbPos,
                Frame.right+HorzThumbPos,Frame.bottom+VertThumbPos
            );
            if (IsValidPosition (hRgn) == TRUE)
            {
                if (AppendNode (hRgn,1,TEXT,NameBuffer,-1,"") == TRUE)
                {
                    TextOut
                    (
                        ErMetaFile,hMetaDC,
                        (Frame.right+Frame.left)/2 + HorzThumbPos - (LOWORD (dwExtent))/2,
                        (Frame.bottom+Frame.top)/2 + VertThumbPos - (HIWORD (dwExtent))/2,
                        NameBuffer,Istrien (NameBuffer)
                    );
                    InvalidateRect (hDrawWnd,NULL,1);
                    UpdateWindow (hDrawWnd);
                    Flag = EDIT;
                }
                else
                    DeleteObject (hRgn);
                ReleaseCapture ();
            }
            else
            {
                Flag = MOVE;
                DeleteObject (hRgn);
            }
        }
    }
}
ReleaseDC (hDrawWnd,hDC);

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

        EndDialog (hDlg,FALSE);
        return (FALSE);
    }
}
return (FALSE);
}

```

```

int xxx;
void INDICATOR_PROC (HWND hDrawWnd, unsigned message, WORD wParam, LONG lParam)
{
    int DeleteFlag;

    EXEC SQL DECLARE C5 CURSOR FOR
    SELECT ATTR_NUMBER
    FROM ATTR
    WHERE ENTITY_NAME = :Entity_Name;

    EXEC SQL DECLARE C8 CURSOR FOR
    SELECT ROLE_NUMBER
    FROM ROLE
    WHERE ENTITY_NAME = :Entity_Name;

    EXEC SQL DECLARE C11 CURSOR FOR
    SELECT ATTR_NUMBER
    FROM ATTR
    WHERE RELATION_NAME = :Relation_Name;

    EXEC SQL DECLARE C12 CURSOR FOR
    SELECT ROLE_NUMBER
    FROM ROLE
    WHERE RELATION_NAME = :Relation_Name;

    EXEC SQL DECLARE C21 CURSOR FOR
    SELECT ATTR_NUMBER
    FROM ATTR
    CONNECT BY PRIOR ATTR_NUMBER = COMPOSIT_NUMBER
    START WITH ATTR_NUMBER = :Attr_Number;

    switch (message)
    {
        case WM_LBUTTONDOWN:
        {
            POINT Point;
            HGLOBAL hMem;
            NODE *Temp;

            Point.x = LOWORD (lParam) + HorzThumbPos;
            Point.y = HIWORD (lParam) + VertThumbPos;
            if ((hMem = IsObjectSelected (Point.x,Point.y)) != NULL)
            {
                Temp = (NODE*) GlobalLock (hMem);
                switch (Temp->ObjectType)
                {
                    case R_ENTITY:
                    case W_ENTITY:
                    {
                        CombineRgn (hDisplayRgn,hEmptyRgn,hEmptyRgn,RGN_AND);
                        CombineRgn (hDisplayRgn,hDisplayRgn,Temp->hRgn,RGN_OR);
                        Istrcpy (Entity_Name.ar,Temp->ObjectName);
                        Entity_Name.len = lstrlen (Entity_Name.ar);
                        EXEC SQL OPEN C5;
                        EXEC SQL WHENEVER NOT FOUND GOTO quit5;
                        for (;;)
                        {
                            EXEC SQL FETCH C5 INTO :Attr_Number;
                            EXEC SQL OPEN C21;
                            EXEC SQL WHENEVER NOT FOUND GOTO quit24;
                            for (;;)
                            {
                                EXEC SQL FETCH C21 INTO :Composit_Number;
                                InvertAttribute (Composit_Number);
                            }
                        }
                        quit24: EXEC SQL CLOSE C21;
                    }
                }
                quit5: EXEC SQL CLOSE C5;
                EXEC SQL OPEN C8;
                EXEC SQL WHENEVER NOT FOUND GOTO quit6;
                for (;;)
                {
                    EXEC SQL FETCH C8 INTO :Role_Number;
                    InvertRole (Role_Number);
                }
                quit6: EXEC SQL CLOSE C8;
                DisplayFlag = OFF;
                InvalidateRect (hDrawWnd,NULL,1);
                UpdateWindow (hDrawWnd);
                DeleteFlag = MessageBox (hDrawWnd,"You want to delete this object","SelectObject",MB_YESNO);
                DisplayFlag = ON;
                if (DeleteFlag == IDYES)
                {
                    EXEC SQL OPEN C5;
                    EXEC SQL WHENEVER NOT FOUND GOTO quit9;
                    for (;;)
                    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ **NODE** ใช้งาน \*Node: การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

HGLOBAL hAttrMem ;

EXEC SQL FETCH C5 INTO :Attr_Number ;
EXEC SQL OPEN C21 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit26 ;
for (;;)
{
    EXEC SQL FETCH C21 INTO :Composit_Number ;
    DeleteOracleAttribute (Composit_Number) ;
    hAttrMem = IsAttributeSelected (Composit_Number) ;
    Node = (NODE*) GlobalLock (hAttrMem) ;
    DeleteMetaFileGroup (Node,hAttrMem) ;
    GlobalUnlock (hAttrMem) ;
}
quit26:
EXEC SQL CLOSE C21 ;
}
quit9:
EXEC SQL CLOSE C5 ;
EXEC SQL OPEN C6 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit10 ;
for (;;)
{
    NODE *Node ;
    HGLOBAL hRoleMem ;

    EXEC SQL FETCH C6 INTO :Role_Number ;
    EXEC SQL DELETE FROM ROLE WHERE ROLE_NUMBER = :Role_Number ;
    if (ProcessOracleErrorCode ("Delete Role Tuple") == FALSE)
        EXEC SQL ROLLBACK WORK ;
    else
        EXEC SQL COMMIT WORK ;
    hRoleMem = IsRoleSelected (Role_Number) ;
    Node = (NODE*) GlobalLock (hRoleMem) ;
    DeleteMetaFileGroup (Node,hRoleMem) ;
    GlobalUnlock (hRoleMem) ;
}
quit10:
EXEC SQL CLOSE C6 ;
EXEC SQL DELETE FROM ENTITY WHERE ENTITY_NAME = :Entity_Name ;
if (ProcessOracleErrorCode ("Delete Entity Tuple") == FALSE) ;
DeleteMetaFileGroup (Temp,hMem) ;
}
InvalidateRect (hDrawWnd,NULL,1) ;
UpdateWindow (hDrawWnd) ;
}
break ;
case N_RELATION:
case _RELATION:
{
    CombineRgn (hDisplayRgn,hEmptyRgn,Temp->hRgn,RGN_OR) ;
    Istropy (Relation_Name,arr,Temp->ObjectName) ;
    Relation_Name.len = Istrien (Relation_Name.arr) ;
    EXEC SQL OPEN C11 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit11 ;
    for (;;)
    {
        EXEC SQL FETCH C11 INTO :Attr_Number ;
        EXEC SQL OPEN C21 ;
        EXEC SQL WHENEVER NOT FOUND GOTO quit27 ;
        for (;;)
        {
            EXEC SQL FETCH C21 INTO :Composit_Number ;
            InvertAttribute (Composit_Number) ;
        }
    }
quit27:
EXEC SQL CLOSE C21 ;
}
quit11:
EXEC SQL CLOSE C11 ;
EXEC SQL OPEN C12 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit12 ;
for (;;)
{
    EXEC SQL FETCH C12 INTO :Role_Number ;
    InvertRole (Role_Number) ;
}
quit12:
EXEC SQL CLOSE C12 ;
DisplayFlag = OFF ;
InvalidateRect (hDrawWnd,NULL,1) ;
UpdateWindow (hDrawWnd) ;
DeleteFlag = MessageBox (hDrawWnd,"You want to delete this object","SelectObject",MB_YESNO) ;
DisplayFlag = ON ;
if (DeleteFlag == IDYES)
{
    EXEC SQL OPEN C11 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit15 ;
    for (;;)
    {
        NODE *Node ;
        HGLOBAL hAttrMem ;

        EXEC SQL FETCH C11 INTO :Attr_Number ;
        EXEC SQL OPEN C21 ;
        EXEC SQL WHENEVER NOT FOUND GOTO quit29 ;
        for (;;)
        {
            EXEC SQL FETCH C21 INTO :Composit_Number ;
            DeleteOracleAttribute (Composit_Number) ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคลากรภายในเท่านั้น ไม่ควรเผยแพร่สู่สาธารณะโดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

        }
        break;
    }
    GlobalUnlock (hMem);
}
break;
}
}

void DeleteMetaFileGroup (NODE *Temp,HGLOBAL hMem)
{
    HMETAFILE    hMF;
    MFENUMPROC   lpEnumMetaProc;

    xxx = 0;
    MetaRecCount = 0;
    hMF = CloseMetaFile (ErMetaFile,hMetaDC);
    ErMetaFile.hMetaDC = CreateMetaFile (NULL);
    lpEnumMetaProc = (MFENUMPROC) MakeProcInstance ((FARPROC) EnumMetaFileProc,hInst);
    EnumMetaFile (ErMetaFile,hMetaDC,hMF,lpEnumMetaProc,(LPARAM) Temp);
    DeleteMetaFile (hMF);
    FreeProcInstance ((FARPROC) lpEnumMetaProc);
    DeleteNode (hMem);
}

```

```

BOOL DeleteMetaFileObject (NODE *Temp,HGLOBAL hMem)
{
    int    ReturnFlag,DeleteFlag;
    MFENUMPROC   lpEnumMetaProc;

    CombineRgn (hDisplayRgn,hEmptyRgn,Temp->hRgn,RGN_OR);
    DisplayFlag = OFF;
    InvalidateRect (hDrawWnd,NULL,1);
    UpdateWindow (hDrawWnd);
    DeleteFlag = MessageBox (hDrawWnd,"You want to delete this object","SelectObject",MB_YESNO);
    DisplayFlag = ON;
    ReturnFlag = FALSE;
    if (DeleteFlag == IDYES)
    {
        HMETAFILE hMF;

        xxx = 0;
        MetaRecCount = 0;
        ReturnFlag = TRUE;
        hMF = CloseMetaFile (ErMetaFile,hMetaDC);
        ErMetaFile.hMetaDC = CreateMetaFile (NULL);
        lpEnumMetaProc = (MFENUMPROC) MakeProcInstance ((FARPROC) EnumMetaFileProc,hInst);
        EnumMetaFile (ErMetaFile,hMetaDC,hMF,lpEnumMetaProc,(LPARAM) Temp);
        DeleteMetaFile (hMF);
        FreeProcInstance ((FARPROC) lpEnumMetaProc);
        DeleteNode (hMem);
    }
    InvalidateRect (hDrawWnd,NULL,1);
    UpdateWindow (hDrawWnd);
    return (ReturnFlag);
}

```

```

int FAR PASCAL EnumMetaFileProc
(
    HDC hMetaDC,HANDLETABLE FAR* lpHTable,METARECORD FAR* lpMFR,int cObj,BYTE FAR* lpClientData
)
{
    NODE *Temp;

    MetaRecCount++;
    Temp = (NODE FAR*) lpClientData;
    if (xxx == 0)
    {
        if (MetaRecCount == Temp->MetaStartRec)
            xxx++;
        else
            PlayMetaFileRecord (hMetaDC,lpHTable,lpMFR,cObj);
    }
    else
    {
        if (xxx < Temp->MetaRecLength)
            xxx++;
        else
        {
            xxx = 0;
            PlayMetaFileRecord (hMetaDC,lpHTable,lpMFR,cObj);
        }
    }
    return (TRUE);
}

```

```

BOOL FAR PASCAL __export About(HWND hDlg,unsigned message,WORD wParam,LPARAM lParam)
{
    switch (message)
    {
        case WM_INITDIALOG:
            return (TRUE);
        case WM_COMMAND:
            {
                if (wParam == IDOK || wParam == IDCANCEL)
                {
                    EndDialog (hDlg, TRUE);
                    return (TRUE);
                }
            }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    break;
}
return (FALSE);
}

```

Filename: *PRTFLE.PC*

```

#include "orextem.inc"
char szNull[] = "";
BOOL PrintErMetaFile()
{
    HDC hDC,hMemoryDC ;
    HBITMAP hOldBitmap,hBitmap ;
    RECT Rect;

    hOldCursor = SetCursor (hWaitCursor) ;
    hPr = GetPrinterDC () ;
    if (!hPr)
        return (NULL) ;
    hDC = GetDC (hDrawWnd) ;
    GetClientRect (hDrawWnd, &Rect) ;
    hMemoryDC = CreateCompatibleDC (hDC) ;
    hBitmap = CreateDiscardableBitmap (hDC,Rect.right,Rect.bottom) ;
    hOldBitmap = SelectObject (hMemoryDC,hBitmap) ;
    BitBlt (hMemoryDC,0,0,Rect.right,Rect.bottom,hDC,0,0,SRCCOPY) ;
    ReleaseDC (hDrawWnd,hDC) ;
    if (Escape(hPr, STARTDOC, 25, "Entity Relationship Model", (LPSTR) NULL) < 0)
    {
        MessageBox (hDrawWnd, "Unable to start print job",NULL, MB_OK | MB_ICONHAND) ;
        DeleteDC (hPr) ;
    }
    EnableWindow (hDrawWnd, FALSE) ;
    SetCursor (hOldCursor) ;
    StretchBlt
    (
        hPr,0,0,GetDeviceCaps(hPr, HORZRES),GetDeviceCaps(hPr, VERTRES),hMemoryDC,0,0,Rect.right,Rect.bottom,SRCCOPY
    );
    SelectObject (hMemoryDC,hOldBitmap) ;
    DeleteObject (hBitmap) ;
    DeleteDC (hMemoryDC) ;
    if (Status >= 0 && !bAbort)
    {
        Escape (hPr, NEWFRAME, 0, 0L, 0L) ;
        Escape (hPr, ENDDOC, 0, 0L, 0L) ;
    }
    EnableWindow (hDrawWnd, TRUE) ;
    DeleteDC (hPr) ;
}

HDC PASCAL GetPrinterDC()
{
    char msgbuf[128];
    LPSTR pch;
    LPSTR pchFile;
    LPSTR pchPort;

    if (!GetProfileString ("windows", "device", szNull, msgbuf, sizeof(msgbuf)))
        return NULL;
    for (pch = msgbuf; *pch && *pch != ';'; pch = AnsiNext (pch) )
    if (*pch)
        *pch++ = 0;
    while (*pch && *pch <= ' ')
        pch=AnsiNext (pch) ;
    pchFile = pch;
    while (*pch && *pch != ':' && *pch > ' ')
        pch = AnsiNext(pch) ;
    if (*pch)
        *pch++ = 0;
    while (*pch && (*pch <= ' ' || *pch == ','))
        pch = AnsiNext(pch) ;
    pchPort = pch;
    while (*pch && *pch > ' ')
        pch = AnsiNext (pch) ;
    *pch = 0;
    return CreateDC (pchFile, msgbuf, pchPort, NULL) ;
}

```

Filename: *PROCESS1.PC*

```

#include "orextem.inc"
EXEC SQL BEGIN DECLARE SECTION ;
VARCHAR Entity_Name[30] ;
VARCHAR Relation_Name[30] ;
VARCHAR Attr_Name[30] ;
VARCHAR Temp_Name[30] ;
VARCHAR DataType[20] ;
VARCHAR Table_Name[30] ;
VARCHAR Not_Null[2] ;
VARCHAR Domain_Name[30] ;
VARCHAR Invert[2] ;
VARCHAR Lower_Limit[2] ;
VARCHAR Upper_Limit[2] ;
VARCHAR Min[30] ;
VARCHAR Max[30] ;
int Type ;
int Role_Number ;
int Attr_Number ;

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int      Temp_Number ;
int      Composit_Number ;
int      Index_Number ;
int      Partial ;
int      Cadinal ;
short    iEntity ;
short    iRelation ;
short    iComposit ;
int      OraCount ;
EXEC SQL END DECLARE SECTION ;
EXEC SQL INCLUDE SQLCA ;

```

```
void BinaryMapping (int One,int Many,LPSTR ch)
```

```

{
    char itoaBuff[10];

    EXEC SQL DECLARE C43 CURSOR FOR
    SELECT  ATTR_NUMBER,ATTR_NAME,ATTR_DATATYPE
    FROM    TEMP_ATTR
    WHERE   (
        ATTR_TYPE = 6 AND
        ENTITY_NAME IN
        (
            SELECT  ENTITY_NAME
            FROM    TEMP_ROLE
            WHERE   ROLE_NUMBER = :Role_Number
        )
    ) OR RELATION_NAME = :Relation_Name ;

    EXEC SQL DECLARE C92 CURSOR FOR
    SELECT  DOMAIN_NAME
    FROM    C_CONSTRAINT
    WHERE   ATTR_NUMBER = :Temp_Number ;

    EXEC SQL DECLARE C93 CURSOR FOR
    SELECT  LOWER_LIMIT,MIN,MAX,UPPER_LIMIT
    FROM    N_CONSTRAINT
    WHERE   ATTR_NUMBER = :Temp_Number ;

    Role_Number = Many ;
    EXEC SQL WHENEVER NOT FOUND CONTINUE ;
    EXEC SQL
    SELECT  ENTITY_NAME INTO :Entity_Name
    FROM    TEMP_ROLE
    WHERE   ROLE_NUMBER = :Role_Number ;
    Role_Number = One ;
    EXEC SQL OPEN C43 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit43 ;
    for (,;)
    {
        EXEC SQL FETCH C43 INTO :Attr_Number,:Attr_Name,:DataType ;
        Attr_Name.arr[Attr_Name.len] = '0' ;
        DataType.arr[DataType.len] = '0' ;
        EXEC SQL
        SELECT  COUNT(*) INTO :OraCount
        FROM    TEMP_ATTR
        WHERE   ATTR_NUMBER = :Attr_Number AND RELATION_NAME = :Relation_Name ;
        if (OraCount == 0)
            Istrcpy (Not_Null.arr,ch) ;
        else
            Istrcpy (Not_Null.arr,"N") ;
        Not_Null.len = Istrlen (Not_Null.arr) ;
        EXEC SQL INSERT INTO ATTR_OF_TABLE (TABLE_NAME,ATTR_NUMBER,NOT_NULL)
        VALUES (:Entity_Name,:Attr_Number,:Not_Null) ;
        if (sqlca.sqlcode < 0)
            /* Recursive relationship */
            {
                Temp_Number = Attr_Number ;
                EXEC SQL SELECT MAX(ATTR_NUMBER) INTO :Attr_Number FROM TEMP_ATTR ;
                Attr_Number++ ;
                itoa (Attr_Number,itoaBuff,10) ;
                Istrcat (Attr_Name.arr,itoaBuff) ;
                Attr_Name.len = Istrlen (Attr_Name.arr) ;
                EXEC SQL INSERT INTO TEMP_ATTR
                (ATTR_NUMBER,ATTR_NAME,ATTR_TYPE,ATTR_DATATYPE,ENTITY_NAME,RELATION_NAME,COMPOSIT_NUMBER)
                VALUES (:Attr_Number,:Attr_Name,4,:DataType,:Entity_Name,null,-1) ;
                EXEC SQL INSERT INTO ATTR_OF_TABLE (TABLE_NAME,ATTR_NUMBER,NOT_NULL)
                VALUES (:Entity_Name,:Attr_Number,:Not_Null) ;
                EXEC SQL WHENEVER NOT FOUND CONTINUE ;
                EXEC SQL
                SELECT  COUNT (*),INVERT INTO :OraCount,:invert
                FROM    INVERT
                WHERE   ATTR_NUMBER = :Temp_Number ;
                if (OraCount > 0)
                {
                    EXEC SQL INSERT INTO C_CONSTRAINT (ATTR_NUMBER,DOMAIN_NAME)
                    VALUES (:Attr_Number,:Domain_Name) ;
                    EXEC SQL OPEN C92 ;
                    EXEC SQL WHENEVER NOT FOUND GOTO quit92 ;
                    for (,;)
                    {
                        EXEC SQL FETCH C92 INTO :Domain_Name ;
                        Domain_Name.arr[Domain_Name.len] = '0' ;
                        EXEC SQL INSERT INTO C_CONSTRAINT (ATTR_NUMBER,DOMAIN_NAME)
                        VALUES (:Attr_Number,:Domain_Name) ;
                    }
                }
                quit92:
                EXEC SQL CLOSE C92 ;
                EXEC SQL INSERT INTO INVERT (ATTR_NUMBER,INVERT) VALUES (:Attr_Number,:invert) ;
            }
        EXEC SQL OPEN C93 ;
    }
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามทำซ้ำโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EXEC SQL WHENEVER NOT FOUND GOTO quit93 ;
for (;;)
{
    EXEC SQL FETCH C93 INTO :Lower_Limit,:Min,:Max,:Upper_Limit ;
    Lower_Limit.arr[Lower_Limit.len] = '0' ;
    Min.arr[Min.len] = '0' ;
    Max.arr[Max.len] = '0' ;
    Upper_Limit.arr[Upper_Limit.len] = '0' ;
    EXEC SQL INSERT INTO N_CONSTRAINT (ATTR_NUMBER,LOWER_LIMIT,MIN,MAX,UPPER_LIMIT)
    VALUES (:Attr_Number,:Lower_Limit,:Min,:Max,:Upper_Limit) ;
}
quit93:
EXEC SQL CLOSE C93 ;
}
EXEC SQL WHENEVER NOT FOUND CONTINUE ;
EXEC SQL
SELECT  ATTR_NAME,ATTR_DATATYPE INTO :Attr_Name,:DataType
FROM    TEMP_ATTR
WHERE   ATTR_NUMBER = :Attr_Number ;
Attr_Name.arr[Attr_Name.len] = '0' ;
DataType.arr[DataType.len] = '0' ;
EXEC SQL INSERT INTO ATTR_DETAIL (ATTR_NUMBER,ATTR_NAME,DATATYPE)
VALUES (:Attr_Number,:Attr_Name,:DataType) ;
}
quit43:
EXEC SQL CLOSE C43 ;
}

void NaryMapping (void)
{
    char ltoaBuff[10] ;

    EXEC SQL DECLARE C25 CURSOR FOR
    SELECT  DOMAIN_NAME
    FROM    C_CONSTRAINT
    WHERE   ATTR_NUMBER = :Temp_Number ;

    EXEC SQL DECLARE C28 CURSOR FOR
    SELECT  LOWER_LIMIT,MIN,MAX,UPPER_LIMIT
    FROM    N_CONSTRAINT
    WHERE   ATTR_NUMBER = :Temp_Number ;

    EXEC SQL DECLARE C27 CURSOR FOR
    SELECT  ENTITY_NAME
    FROM    ROLE
    WHERE   RELATION_NAME = :Relation_Name ;

    EXEC SQL DECLARE C28 CURSOR FOR
    SELECT  ATTR_NUMBER,ATTR_NAME,ATTR_DATATYPE
    FROM    TEMP_ATTR
    WHERE   ATTR_TYPE = 6 AND ENTITY_NAME = :Entity_Name ;

    EXEC SQL DECLARE C29 CURSOR FOR
    SELECT  ATTR_NUMBER,ATTR_NAME,ATTR_DATATYPE
    FROM    TEMP_ATTR
    WHERE   RELATION_NAME = :Relation_Name ;

    EXEC SQL SELECT COUNT(*) INTO :Index_Number FROM TABLE_OF_INDEX ;
    Index_Number++ ;
    EXEC SQL INSERT INTO TABLE_OF_INDEX (INDEX_NUMBER,TABLE_NAME) VALUES (:Index_Number,:Relation_Name) ;
    EXEC SQL OPEN C27 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit27 ;
    for (;;)
    {
        EXEC SQL FETCH C27 INTO :Entity_Name ;
        Entity_Name.arr[Entity_Name.len] = '0' ;
        EXEC SQL OPEN C28 ;
        EXEC SQL WHENEVER NOT FOUND GOTO quit28 ;
        for (;;)
        {
            EXEC SQL FETCH C28 INTO :Attr_Number,:Attr_Name,:DataType ;
            Attr_Name.arr[Attr_Name.len] = '0' ;
            DataType.arr[DataType.len] = '0' ;
            EXEC SQL INSERT INTO ATTR_OF_INDEX (INDEX_NUMBER,ATTR_NUMBER) VALUES (:Index_Number,:Attr_Number) ;
            if (sqlca.sqlcode < 0) /* Recursive relationship */
            {
                Temp_Number = Attr_Number ;
                EXEC SQL SELECT MAX(ATTR_NUMBER) INTO :Attr_Number FROM TEMP_ATTR ;
                Attr_Number++ ;
                ltoa (Attr_Number,ltoaBuff,10) ;
                lstrcat (Attr_Name,arr,ltoaBuff) ;
                Attr_Name.len = lstrlen (Attr_Name.arr) ;
                EXEC SQL INSERT INTO TEMP_ATTR
                (
                    ATTR_NUMBER,ATTR_NAME,ATTR_TYPE,ATTR_DATATYPE,
                    ENTITY_NAME,RELATION_NAME,COMPOSIT_NUMBER
                )
                VALUES (:Attr_Number,:Attr_Name,6,:DataType,:Relation_Name,null,-1) ;
                EXEC SQL WHENEVER NOT FOUND CONTINUE ;
                EXEC SQL
                SELECT  COUNT (*),INVERT INTO :OraCount,:Invert
                FROM    INVERT
                WHERE   ATTR_NUMBER = :Temp_Number ;
                if (OraCount > 0)
                {
                    EXEC SQL OPEN C25 ;
                    EXEC SQL WHENEVER NOT FOUND GOTO quit25 ;
                    for (;;)
                    {
                        EXEC SQL FETCH C25 INTO :Domain_Name ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้เพื่อการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Domain_Name.arr[Domain_Name.len] = '\0';
EXEC SQL INSERT INTO C_CONSTRAINT (ATTR_NUMBER,DOMAIN_NAME)
VALUES (:Attr_Number,:Domain_Name);
}
quit25:
EXEC SQL CLOSE C25;
EXEC SQL INSERT INTO INVERT (ATTR_NUMBER,INVERT) VALUES (:Attr_Number,:Invert);
}
EXEC SQL OPEN C26;
EXEC SQL WHENEVER NOT FOUND GOTO quit26;
for (;;)
{
EXEC SQL FETCH C26 INTO :Lower_Limit,:Min,:Max,:Upper_Limit;
Lower_Limit.arr[Lower_Limit.len] = '\0';
Min.arr[Min.len] = '\0';
Max.arr[Max.len] = '\0';
Upper_Limit.arr[Upper_Limit.len] = '\0';
EXEC SQL INSERT INTO N_CONSTRAINT (ATTR_NUMBER,LOWER_LIMIT,MIN,MAX,UPPER_LIMIT)
VALUES (:Attr_Number,:Lower_Limit,:Min,:Max,:Upper_Limit);
}
quit26:
EXEC SQL CLOSE C26;
}
EXEC SQL INSERT INTO ATTR_OF_INDEX (INDEX_NUMBER,ATTR_NUMBER) VALUES (:Index_Number,:Attr_Number);
EXEC SQL INSERT INTO ATTR_OF_TABLE (TABLE_NAME,ATTR_NUMBER,NOT_NULL)
VALUES (:Relation_Name,:Attr_Number,'Y');
EXEC SQL
SELECT ATTR_NAME,ATTR_DATATYPE INTO :Attr_Name,:DataType
FROM TEMP_ATTR
WHERE ATTR_NUMBER = :Attr_Number;
Attr_Name.arr[Attr_Name.len] = '\0';
DataType.arr[DataType.len] = '\0';
EXEC SQL INSERT INTO ATTR_DETAIL (ATTR_NUMBER,ATTR_NAME,DATATYPE)
VALUES (:Attr_Number,:Attr_Name,:DataType);
}
quit28:
EXEC SQL CLOSE C28;
}
quit27:
EXEC SQL CLOSE C27;
EXEC SQL OPEN C29;
EXEC SQL WHENEVER NOT FOUND GOTO quit29;
for (;;)
{
EXEC SQL FETCH C29 INTO :Attr_Number,:Attr_Name,:DataType;
Attr_Name.arr[Attr_Name.len] = '\0';
DataType.arr[DataType.len] = '\0';
EXEC SQL INSERT INTO ATTR_OF_TABLE (TABLE_NAME,ATTR_NUMBER,NOT_NULL)
VALUES (:Relation_Name,:Attr_Number,'N');
EXEC SQL
SELECT ATTR_NAME,ATTR_DATATYPE INTO :Attr_Name,:DataType
FROM TEMP_ATTR
WHERE ATTR_NUMBER = :Attr_Number;
Attr_Name.arr[Attr_Name.len] = '\0';
DataType.arr[DataType.len] = '\0';
EXEC SQL INSERT INTO ATTR_DETAIL (ATTR_NUMBER,ATTR_NAME,DATATYPE)
VALUES (:Attr_Number,:Attr_Name,:DataType);
}
quit29:
EXEC SQL CLOSE C29;
}
BOOL ProcessFirst (void)
{
int hFile,ErrorFlag,count,ErrorCount;
char LineBuffer[100];
char ItoaBuff[10];
OFSTRUCT OfStruct;

EXEC SQL DECLARE C34 CURSOR FOR
SELECT ENTITY_NAME
FROM TEMP_ENTITY
WHERE ENTITY_TYPE = 1;

EXEC SQL DECLARE C35 CURSOR FOR
SELECT ATTR_NAME,ATTR_DATATYPE
FROM TEMP_ATTR
WHERE
(
ATTR_TYPE = 6 AND
ENTITY_NAME IN
(
SELECT DISTINCT ENTITY_NAME
FROM TEMP_ROLE
WHERE RELATION_NAME IN
(
SELECT RELATION_NAME
FROM TEMP_ROLE
WHERE ENTITY_NAME = :Entity_Name
INTERSECT
SELECT RELATION_NAME
FROM TEMP_RELATION
WHERE RELATION_TYPE = 3
)
)
INTERSECT
SELECT ENTITY_NAME
FROM TEMP_ENTITY
WHERE ENTITY_TYPE = 0
)
)
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EXEC SQL DECLARE C36 CURSOR FOR
SELECT ENTITY_NAME
FROM TEMP_ENTITY ;

EXEC SQL DECLARE C37 CURSOR FOR
SELECT ATTR_NUMBER,ATTR_NAME,ATTR_TYPE,ATTR_DATATYPE
FROM TEMP_ATTR
WHERE ENTITY_NAME = :Entity_Name AND ATTR_TYPE != 7 ;

EXEC SQL DECLARE C38 CURSOR FOR
SELECT ATTR_NUMBER,ATTR_NAME,ATTR_DATATYPE
FROM TEMP_ATTR
WHERE ENTITY_NAME = :Entity_Name AND ATTR_TYPE = 7 ; /* Multivalue Attribute */

EXEC SQL DECLARE C40 CURSOR FOR
SELECT ATTR_NUMBER
FROM TEMP_ATTR
WHERE ENTITY_NAME = :Entity_Name AND ATTR_TYPE = 6 ; /* Primary Key Attribute */

EXEC SQL DECLARE C41 CURSOR FOR
SELECT RELATION_NAME
FROM TEMP_RELATION
WHERE RELATION_TYPE != 3 ;

EXEC SQL DECLARE C42 CURSOR FOR
SELECT ROLE_NUMBER,PATICIPATION,CADINALITY
FROM TEMP_ROLE
WHERE RELATION_NAME = :Relation_Name ;

hFile = OpenFile ("Output.msg",&OfStruct,OF_CREATE) ;
if (hFile < 0)
    return (FALSE) ;
SetCapture (hDrawWnd) ;
hOldCursor = SetCursor (hWaitCursor) ;
ErrorCount = 0 ;
ErrorFlag = TRUE ;
if (CreateProcessTable () == FALSE)
{
    ErrorCount++ ;
    itoa (ErrorCount,itoaBuff,10) ;
    ClearBuffer (LineBuffer,100) ;
    Istrcpy (LineBuffer,"Error No. ") ;
    Istrcat (LineBuffer,itoaBuff) ;
    Istrcat (LineBuffer," ") ;
    Istrcat (LineBuffer,"Can't create process table ") ;
    write (hFile,LineBuffer,Istrlen(LineBuffer)) ;
    close (hFile) ;
    ErrorFlag = FALSE ;
    return (ErrorFlag) ;
}
do
{
    EXEC SQL OPEN C34 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit34 ;
    for (;;)
    {
        EXEC SQL FETCH C34 INTO :Entity_Name ;
        Entity_Name.arr[Entity_Name.len] = '\0' ;
        count = 0 ;
        EXEC SQL OPEN C35 ;
        EXEC SQL WHENEVER NOT FOUND GOTO quit35 ;
        for (;;)
        {
            count++ ;
            EXEC SQL FETCH C35 INTO :Attr_Name,:DataType ;
            Attr_Name.arr[Attr_Name.len] = '\0' ;
            DataType.arr[DataType.len] = '\0' ;
            EXEC SQL SELECT MAX(ATTR_NUMBER) INTO :Attr_Number FROM TEMP_ATTR ;
            Attr_Number++ ;
            EXEC SQL INSERT INTO TEMP_ATTR
            (
                ATTR_NUMBER,ATTR_NAME,ATTR_TYPE,ATTR_DATATYPE,
                ENTITY_NAME,RELATION_NAME,COMPOSIT_NUMBER
            )
            VALUES (:Attr_Number,:Attr_Name,6,:DataType,:Entity_Name,null,-1) ;
            if (ProcessOracleErrorCode ("INSERT TEMP ATTRIBUTE") == FALSE)
            {
                ErrorCount++ ;
                itoa (ErrorCount,itoaBuff,10) ;
                ClearBuffer (LineBuffer,100) ;
                Istrcpy (LineBuffer,"Error No. ") ;
                Istrcat (LineBuffer,itoaBuff) ;
                Istrcat (LineBuffer," ") ;
                Istrcat (LineBuffer,"Can't Insert temp_attr process table ") ;
                write (hFile,LineBuffer,Istrlen(LineBuffer)) ;
                ErrorFlag = FALSE ;
            }
        }
    }
quit35:
    EXEC SQL CLOSE C35 ;
    if (count > 0)
    {
        EXEC SQL UPDATE TEMP_ENTITY SET ENTITY_TYPE = 0 WHERE ENTITY_NAME = :Entity_Name ;
        if (ProcessOracleErrorCode ("UPDATE TEMP ENTITY") == FALSE)
        {
            ErrorCount++ ;
            itoa (ErrorCount,itoaBuff,10) ;
            ClearBuffer (LineBuffer,100) ;
            Istrcpy (LineBuffer,"Error No. ") ;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Istrcat (LineBuffer,itoaBuff);
        Istrcat (LineBuffer,"");
        Istrcat (LineBuffer,"Can't update temp_entity process table");
        write (hFile,LineBuffer,Istrlen(LineBuffer));
        ErrorFlag = FALSE;
    }
}
quit34:
EXEC SQL CLOSE C34;
EXEC SQL COMMIT WORK;
EXEC SQL SELECT COUNT (*) INTO :OraCount FROM TEMP_ENTITY WHERE ENTITY_TYPE = 1;
} while (OraCount > 0);
close (hFile);
/* Map Attribute of Identifies Relationship */
IdentifyMapping ();
/* Rule 1. Each Entity Create One Table */
EXEC SQL OPEN C36;
EXEC SQL WHENEVER NOT FOUND GOTO quit36;
for (;;)
{
    EXEC SQL SELECT COUNT(*) INTO :Index_Number FROM TABLE_OF_INDEX;
    EXEC SQL FETCH C36 INTO :Entity_Name;
    Entity_Name.arr[Entity_Name.len] = '\0';
    Index_Number++;
    EXEC SQL INSERT INTO TABLE_OF_INDEX (INDEX_NUMBER,TABLE_NAME) VALUES (:Index_Number,:Entity_Name);
    EXEC SQL OPEN C37;
    EXEC SQL WHENEVER NOT FOUND GOTO quit37;
    for (;;)
    {
        EXEC SQL FETCH C37 INTO :Attr_Number,:Attr_Name,:Type,:DataType;
        Attr_Name.arr[Attr_Name.len] = '\0';
        DataType.arr[DataType.len] = '\0';
        if (Type == N_ATTRIBUTE)
            Istrcpy (Not_Null.arr,"N");
        else
            Istrcpy (Not_Null.arr,"Y");
        Not_Null.len = Istrlen (Not_Null.arr);
        EXEC SQL INSERT INTO ATTR_OF_TABLE (TABLE_NAME,ATTR_NUMBER,NOT_NULL)
        VALUES (:Entity_Name,:Attr_Number,Not_Null);
        EXEC SQL INSERT INTO ATTR_DETAIL (ATTR_NUMBER,ATTR_NAME,DATATYPE)
        VALUES (:Attr_Number,:Attr_Name,:DataType);
        if (Type == P_ATTRIBUTE)
        {
            EXEC SQL INSERT INTO ATTR_OF_INDEX (INDEX_NUMBER,ATTR_NUMBER)
            VALUES (:Index_Number,:Attr_Number);
        }
    }
}
quit37:
EXEC SQL CLOSE C37;
}
quit36:
EXEC SQL CLOSE C36;
/* Rule 2. Each Multivalue Attribute Create One Table */
EXEC SQL OPEN C38;
EXEC SQL WHENEVER NOT FOUND GOTO quit39;
for (;;)
{
    EXEC SQL FETCH C36 INTO :Entity_Name;
    Entity_Name.arr[Entity_Name.len] = '\0';
    EXEC SQL OPEN C38;
    EXEC SQL WHENEVER NOT FOUND GOTO quit38;
    for (;;)
    {
        EXEC SQL FETCH C38 INTO :Attr_Number,:Attr_Name,:DataType;
        Attr_Name.arr[Attr_Name.len] = '\0';
        DataType.arr[DataType.len] = '\0';
        Istrcpy (Not_Null.arr,"Y");
        Not_Null.len = Istrlen (Not_Null.arr);
        EXEC SQL SELECT COUNT(*) INTO :Index_Number FROM TABLE_OF_INDEX;
        Index_Number++;
        EXEC SQL INSERT INTO TABLE_OF_INDEX (INDEX_NUMBER,TABLE_NAME) VALUES (:Index_Number,:Attr_Name);
        EXEC SQL INSERT INTO ATTR_OF_TABLE (TABLE_NAME,ATTR_NUMBER,NOT_NULL)
        VALUES (:Attr_Name,:Attr_Number,Not_Null);
        EXEC SQL INSERT INTO ATTR_DETAIL (ATTR_NUMBER,ATTR_NAME,DATATYPE)
        VALUES (:Attr_Number,:Attr_Name,:DataType);
        EXEC SQL INSERT INTO ATTR_OF_INDEX (INDEX_NUMBER,ATTR_NUMBER) VALUES (:Index_Number,:Attr_Number);
        EXEC SQL OPEN C40;
        EXEC SQL WHENEVER NOT FOUND GOTO quit40;
        for (;;)
        {
            EXEC SQL FETCH C40 INTO :Attr_Number;
            EXEC SQL INSERT INTO ATTR_OF_TABLE (TABLE_NAME,ATTR_NUMBER,NOT_NULL)
            VALUES (:Attr_Name,:Attr_Number,Not_Null);
            EXEC SQL
            SELECT ATTR_NAME,ATTR_DATATYPE INTO :Temp_Name,:DataType
            FROM TEMP_ATTR
            WHERE ATTR_NUMBER = :Attr_Number;
            Temp_Name.arr[Temp_Name.len] = '\0';
            DataType.arr[DataType.len] = '\0';
            EXEC SQL INSERT INTO ATTR_DETAIL (ATTR_NUMBER,ATTR_NAME,DATATYPE)
            VALUES (:Attr_Number,:Temp_Name,:DataType);
            EXEC SQL INSERT INTO ATTR_OF_INDEX (INDEX_NUMBER,ATTR_NUMBER)
            VALUES (:Index_Number,:Attr_Number);
        }
    }
}
quit40:
EXEC SQL CLOSE C40;
}
quit39:
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EXEC SQL CLOSE C37 ;
quit39:
EXEC SQL CLOSE C36 ;

/* Map Relationship */

EXEC SQL OPEN C41 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit41 ;
for (;;)
{
EXEC SQL FETCH C41 INTO :Relation_Name ;
Relation_Name.arr[Relation_Name.len] = '\0' ;
EXEC SQL WHENEVER NOT FOUND CONTINUE ;
EXEC SQL
SELECT COUNT (*) INTO :OraCount
FROM TEMP_ROLE
WHERE RELATION_NAME = :Relation_Name ;
if (OraCount == 2)
{
int i ;
struct (int N,P,C) Role[2] ;

i = 0 ;
EXEC SQL OPEN C42 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit42 ;
for (;;)
{
EXEC SQL FETCH C42 INTO :Role_Number,:Partial,:Cadinal ;
Role[i].N = Role_Number ;
Role[i].P = Partial ;
Role[i].C = Cadinal ;
i++ ;
}
quit42:
EXEC SQL CLOSE C42 ;
if (Role[0].C == 0 && Role[0].P == 0 && Role[1].C == 0 && Role[1].P == 0) BinaryMapping (Role[0].N,Role[1].N,"N") ; /* 1P:1P */
if (Role[0].C == 0 && Role[0].P == 0 && Role[1].C == 0 && Role[1].P == 1) BinaryMapping (Role[0].N,Role[1].N,"Y") ; /* 1P:1T */
if (Role[0].C == 0 && Role[0].P == 1 && Role[1].C == 0 && Role[1].P == 0) BinaryMapping (Role[0].N,Role[1].N,"Y") ; /* 1T:1P */
if (Role[0].C == 0 && Role[0].P == 1 && Role[1].C == 0 && Role[1].P == 1) BinaryMapping (Role[0].N,Role[1].N,"Y") ; /* 1T:1T */
if (Role[0].C == 0 && Role[0].P == 0 && Role[1].C == 1 && Role[1].P == 0) BinaryMapping (Role[0].N,Role[1].N,"N") ; /* 1P:NP */
if (Role[0].C == 0 && Role[0].P == 0 && Role[1].C == 1 && Role[1].P == 1) BinaryMapping (Role[0].N,Role[1].N,"Y") ; /* 1P:NT */
if (Role[0].C == 0 && Role[0].P == 1 && Role[1].C == 1 && Role[1].P == 0) BinaryMapping (Role[0].N,Role[1].N,"N") ; /* 1T:NP */
if (Role[0].C == 0 && Role[0].P == 1 && Role[1].C == 1 && Role[1].P == 1) BinaryMapping (Role[0].N,Role[1].N,"Y") ; /* 1T:NT */
if (Role[0].C == 1 && Role[0].P == 0 && Role[1].C == 0 && Role[1].P == 0) BinaryMapping (Role[1].N,Role[0].N,"N") ; /* NP:1P */
if (Role[0].C == 1 && Role[0].P == 0 && Role[1].C == 0 && Role[1].P == 1) BinaryMapping (Role[1].N,Role[0].N,"Y") ; /* NP:1T */
if (Role[0].C == 1 && Role[0].P == 1 && Role[1].C == 0 && Role[1].P == 0) BinaryMapping (Role[1].N,Role[0].N,"N") ; /* NT:1P */
if (Role[0].C == 1 && Role[0].P == 1 && Role[1].C == 0 && Role[1].P == 1) BinaryMapping (Role[1].N,Role[0].N,"Y") ; /* NT:1T */
if (Role[0].C == 1 && Role[1].C == 1) NaryMapping 0 ; /* N:N */
}
else
{
NaryMapping 0 ;
}
}
quit41:
EXEC SQL CLOSE C41 ;
EXEC SQL COMMIT WORK ;
ManageComposit () ;
CreateSqlTable ("First.sql") ;
GenDefaultFD () ;
DropProcessTable () ;
SetCursor (hOldCursor) ;
ReleaseCapture () ;
return (ErrorFlag) ;
}

BOOL IsCompleteER (void)
{
int hFile,count,ErrorCount ;
char LineBuffer[100] ;
char ltoaBuff[10] ;
OFSTRUCT OfStruct ;

EXEC SQL DECLARE C17 CURSOR FOR
SELECT ATTR_NAME,ENTITY_NAME,RELATION_NAME,COMPOSIT_NUMBER
FROM ATTR ;

EXEC SQL DECLARE C18 CURSOR FOR
SELECT RELATION_NAME,RELATION_TYPE
FROM RELATION ;

EXEC SQL DECLARE C19 CURSOR FOR
SELECT ENTITY_NAME
FROM ENTITY ;

hFile = OpenFile ("Output.msg",&OfStruct,OF_CREATE) ;
if (hFile < 0)
return (FALSE) ;

SetCapture (hDrawWnd) ;
hOldCursor = SetCursor (hWaitCursor) ;
EXEC SQL OPEN C17 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit17 ;
count = 0 ;
ErrorCount = 0 ;
ErrorFlag = NOERROR ;
for (;;)
{
EXEC SQL FETCH C17 INTO :Attr_Name,:Entity_Name,:Entity,:Relation_Name,:Relation,:Composit_Number ;

```

เอกสารนี้ใช้ฟรี กรุณาแจ้งที่มาของเอกสารนี้ถ้าไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If (iEntity == -1 && iRelation == -1 && Composit_Number == -1)
{
    Attr_Name.arr[Attr_Name.len] = '\0';
    ErrorCount++;
    itoa (ErrorCount, itoaBuff, 10);
    ClearBuffer (LineBuffer, 100);
    Istrcpy (LineBuffer, "Error No. ");
    Istrcat (LineBuffer, itoaBuff);
    Istrcat (LineBuffer, " ");
    Istrcat (LineBuffer, Attr_Name.arr);
    Istrcat (LineBuffer, " Disconnect ");
    LineBuffer[strlen(LineBuffer)] = (char) 0x0d;
    LineBuffer[strlen(LineBuffer)] = (char) 0x0a;
    write (hFile, LineBuffer, strlen(LineBuffer));
    ErrorFlag = ERROR;
}
count++;
}
quit17:
EXEC SQL CLOSE C17;
if (count==0)
{
    ErrorCount++;
    itoa (ErrorCount, itoaBuff, 10);
    ClearBuffer (LineBuffer, 100);
    Istrcpy (LineBuffer, "Error No. ");
    Istrcat (LineBuffer, itoaBuff);
    Istrcat (LineBuffer, " ");
    Istrcat (LineBuffer, "Attribute Not Found");
    LineBuffer[strlen(LineBuffer)] = (char) 0x0d;
    LineBuffer[strlen(LineBuffer)] = (char) 0x0a;
    write (hFile, LineBuffer, strlen(LineBuffer));
    ErrorFlag = ERROR;
}
EXEC SQL OPEN C18;
EXEC SQL WHENEVER NOT FOUND GOTO quit18;
count = 0;
for (;;)
{
    EXEC SQL FETCH C18 INTO :Relation_Name,:Type;
    EXEC SQL
    SELECT      COUNT(*) INTO :OraCount
    FROM        ROLE
    WHERE       RELATION_NAME = :Relation_Name;
    Relation_Name.arr[Relation_Name.len] = '\0';
    if (OraCount < 2)
    {
        ErrorCount++;
        itoa (ErrorCount, itoaBuff, 10);
        ClearBuffer (LineBuffer, 100);
        Istrcpy (LineBuffer, "Error No. ");
        Istrcat (LineBuffer, itoaBuff);
        Istrcat (LineBuffer, " ");
        Istrcpy (LineBuffer, Relation_Name.arr);
        Istrcat (LineBuffer, " Uncomplete ");
        LineBuffer[strlen(LineBuffer)] = (char) 0x0d;
        LineBuffer[strlen(LineBuffer)] = (char) 0x0a;
        write (hFile, LineBuffer, strlen(LineBuffer));
        ErrorFlag = ERROR;
    }
    if (Type == _RELATION)
    {
        EXEC SQL
        SELECT      COUNT(*) INTO :OraCount
        FROM        ROLE
        WHERE
        (
            CADINALITY IN (1,2) AND
            PATICIPATION = 1 AND
            RELATION_NAME = :Relation_Name AND
            ENTITY_NAME NOT IN
            (
                SELECT      ENTITY_NAME
                FROM        ENTITY
                WHERE       ENTITY_TYPE = 0
            )
        );
        if (OraCount != 1)
        {
            ErrorCount++;
            itoa (ErrorCount, itoaBuff, 10);
            ClearBuffer (LineBuffer, 100);
            Istrcpy (LineBuffer, "Error No. ");
            Istrcat (LineBuffer, itoaBuff);
            Istrcat (LineBuffer, " ");
            Istrcpy (LineBuffer, Relation_Name.arr);
            Istrcat (LineBuffer, " (Weak entity type not found OR Role invalid used)");
            LineBuffer[strlen(LineBuffer)] = (char) 0x0d;
            LineBuffer[strlen(LineBuffer)] = (char) 0x0a;
            write (hFile, LineBuffer, strlen(LineBuffer));
            ErrorFlag = ERROR;
        }
        EXEC SQL
        SELECT      COUNT(*) INTO :OraCount
        FROM        ROLE
        WHERE
        (
            CADINALITY IN (1,2) AND
            PATICIPATION = 0 AND
            RELATION_NAME = :Relation_Name

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    );
    if (OraCount > 0)
    {
        ErrorCount++;
        itoa (ErrorCount, itoaBuff, 10);
        ClearBuffer (LineBuffer, 100);
        Istrcpy (LineBuffer, "Error No. ");
        Istrcat (LineBuffer, itoaBuff);
        Istrcat (LineBuffer, " ");
        Istrcpy (LineBuffer, Relation_Name.arr);
        Istrcat (LineBuffer, " (Role invalid used)");
        LineBuffer[istrlen(LineBuffer)] = (char) 0x0d;
        LineBuffer[istrlen(LineBuffer)] = (char) 0x0a;
        write (hFile, LineBuffer, istrlen(LineBuffer));
        ErrorFlag = ERROR;
    }
    count++;
}
quit18:
EXEC SQL CLOSE C18;
if (count == 0)
{
    ErrorCount++;
    itoa (ErrorCount, itoaBuff, 10);
    ClearBuffer (LineBuffer, 100);
    Istrcpy (LineBuffer, "Error No. ");
    Istrcat (LineBuffer, itoaBuff);
    Istrcat (LineBuffer, " ");
    Istrcat (LineBuffer, "Relationship Type Not Found");
    LineBuffer[istrlen(LineBuffer)] = (char) 0x0d;
    LineBuffer[istrlen(LineBuffer)] = (char) 0x0a;
    write (hFile, LineBuffer, istrlen(LineBuffer));
    ErrorFlag = ERROR;
}
count = 0;
EXEC SQL OPEN C19;
EXEC SQL WHENEVER NOT FOUND GOTO quit19;
for (;;)
{
    EXEC SQL FETCH C19 INTO :Entity_Name;
    Entity_Name.arr[Entity_Name.len] = '\0';
    EXEC SQL
    SELECT COUNT(*) INTO :OraCount
    FROM ATTR
    WHERE ATTR_TYPE = 8 AND ENTITY_NAME = :Entity_Name;
    if (OraCount == 0)
    {
        ErrorCount++;
        itoa (ErrorCount, itoaBuff, 10);
        ClearBuffer (LineBuffer, 100);
        Istrcpy (LineBuffer, "Error No. ");
        Istrcat (LineBuffer, itoaBuff);
        Istrcat (LineBuffer, " ");
        Entity_Name.arr[Entity_Name.len] = '\0';
        Istrcat (LineBuffer, Entity_Name.arr);
        Istrcat (LineBuffer, " Primary key Attribute Not Found");
        LineBuffer[istrlen(LineBuffer)] = (char) 0x0d;
        LineBuffer[istrlen(LineBuffer)] = (char) 0x0a;
        write (hFile, LineBuffer, istrlen(LineBuffer));
        ErrorFlag = ERROR;
    }
    if (OraCount > 1)
    {
        ErrorCount++;
        itoa (ErrorCount, itoaBuff, 10);
        ClearBuffer (LineBuffer, 100);
        Istrcpy (LineBuffer, "Error No. ");
        Istrcat (LineBuffer, itoaBuff);
        Istrcat (LineBuffer, " ");
        Entity_Name.arr[Entity_Name.len] = '\0';
        Istrcat (LineBuffer, Entity_Name.arr);
        Istrcat (LineBuffer, " Primary key Attribute more than one");
        LineBuffer[istrlen(LineBuffer)] = (char) 0x0d;
        LineBuffer[istrlen(LineBuffer)] = (char) 0x0a;
        write (hFile, LineBuffer, istrlen(LineBuffer));
        ErrorFlag = ERROR;
    }
    count++;
}
quit19:
EXEC SQL CLOSE C19;
if (count == 0)
{
    ErrorCount++;
    itoa (ErrorCount, itoaBuff, 10);
    ClearBuffer (LineBuffer, 100);
    Istrcpy (LineBuffer, "Error No. ");
    Istrcat (LineBuffer, itoaBuff);
    Istrcat (LineBuffer, " ");
    Istrcat (LineBuffer, "Entity Type Not Found");
    LineBuffer[istrlen(LineBuffer)] = (char) 0x0d;
    LineBuffer[istrlen(LineBuffer)] = (char) 0x0a;
    write (hFile, LineBuffer, istrlen(LineBuffer));
    ErrorFlag = ERROR;
}
ClearBuffer (LineBuffer, 100);
if (ErrorFlag == NOERROR)
    Istrcpy (LineBuffer, "Entity Relationship Model Complete");
else

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Istrcpy (LineBuffer,"Entity Relationship Model Error !");
        write (hFile,LineBuffer,Istrlen(LineBuffer));
        SetCursor (hOldCursor);
        ReleaseCapture ();
        close (hFile);
        return (ErrorFlag);
}

```

```

void CreateSqlTable (LPSTR FileName)

```

```

{
    int         i,hFile,count;
    OFSTRUCT   OfStruct;
    char       TableBuffer[200],itoaBuff[8];

    EXEC SQL   DECLARE C30 CURSOR FOR
    SELECT     DISTINCT TABLE_NAME
    FROM       ATTR_OF_TABLE;

    EXEC SQL   DECLARE C31 CURSOR FOR
    SELECT     ATTR_NUMBER,NOT_NULL
    FROM       ATTR_OF_TABLE
    WHERE      TABLE_NAME = :Table_Name;

    EXEC SQL   DECLARE C32 CURSOR FOR
    SELECT     INDEX_NUMBER,TABLE_NAME
    FROM       TABLE_OF_INDEX;

    EXEC SQL   DECLARE C33 CURSOR FOR
    SELECT     ATTR_NUMBER
    FROM       ATTR_OF_INDEX
    WHERE      INDEX_NUMBER = :Index_Number;

    EXEC SQL   DECLARE C94 CURSOR FOR
    SELECT     DISTINCT TABLE_NAME
    FROM       ATTR_OF_TABLE
    WHERE      ATTR_NUMBER IN
    (
        SELECT   ATTR_NUMBER
        FROM     C_CONSTRAINT
        UNION
        SELECT   ATTR_NUMBER
        FROM     N_CONSTRAINT
    );

    EXEC SQL   DECLARE C95 CURSOR FOR
    SELECT     DISTINCT ATTR_NUMBER
    FROM       C_CONSTRAINT
    WHERE      ATTR_NUMBER IN
    (
        SELECT   ATTR_NUMBER
        FROM     ATTR_OF_TABLE
        WHERE    TABLE_NAME = :Table_Name
    );

    EXEC SQL   DECLARE C96 CURSOR FOR
    SELECT     DOMAIN_NAME
    FROM       C_CONSTRAINT
    WHERE      ATTR_NUMBER = :Attr_Number;

    EXEC SQL   DECLARE C97 CURSOR FOR
    SELECT     DISTINCT ATTR_NUMBER
    FROM       N_CONSTRAINT
    WHERE      ATTR_NUMBER IN
    (
        SELECT   ATTR_NUMBER
        FROM     ATTR_OF_TABLE
        WHERE    TABLE_NAME = :Table_Name
    );

    EXEC SQL   DECLARE C98 CURSOR FOR
    SELECT     LOWER_LIMIT,MIN,MAX,UPPER_LIMIT
    FROM       N_CONSTRAINT
    WHERE      ATTR_NUMBER = :Attr_Number;

    hFile = OpenFile (FileName,&OfStruct,OF_CREATE);
    if (hFile >=0)
    {
        EXEC SQL OPEN C30;
        EXEC SQL WHENEVER NOT FOUND GOTO quit30;
        for(;;)
        {
            EXEC SQL FETCH C30 INTO :Table_Name;
            Table_Name.arr[Table_Name.len] = '\0';
            ClearBuffer (TableBuffer,200);
            Istrcpy (TableBuffer,"CREATE TABLE ");
            Istrcat (TableBuffer,Table_Name.arr);
            TableBuffer[Istrlen(TableBuffer)] = (char) 0x0d;
            TableBuffer[Istrlen(TableBuffer)] = (char) 0x0a;
            Istrcat (TableBuffer,"(");
            TableBuffer[Istrlen(TableBuffer)] = (char) 0x0d;
            TableBuffer[Istrlen(TableBuffer)] = (char) 0x0a;
            write (hFile,TableBuffer,Istrlen(TableBuffer));
            EXEC SQL
            SELECT     COUNT(*) INTO :OraCount
            FROM       ATTR_OF_TABLE
            WHERE      TABLE_NAME = :Table_Name;
            EXEC SQL OPEN C31;
            EXEC SQL WHENEVER NOT FOUND GOTO quit31;
            count = 0;
            for (i=0; i<OraCount; i++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

count++;
EXEC SQL FETCH C31 INTO :Attr_Number,:Not_Null ;
EXEC SQL
SELECT  ATTR_NAME,DATATYPE INTO :Attr_Name,:DataType
FROM    ATTR_DETAIL
WHERE   ATTR_NUMBER = :Attr_Number ;
Attr_Name.arr[Attr_Name.len] = '\0' ;
ClearBuffer (TableBuffer,200) ;
TableBuffer[istrlen(TableBuffer)] = '\t' ;
Istrcat (TableBuffer,Attr_Name.arr) ;
TableBuffer[istrlen(TableBuffer)] = '\t' ;
TableBuffer[istrlen(TableBuffer)] = '\t' ;
DataType.arr[DataType.len] = '\0' ;
Not_Null.arr[Not_Null.len] = '\0' ;
Istrcat (TableBuffer,DataType.arr) ;
if (Not_Null.arr[0] == '\Y')
{
    TableBuffer[istrlen(TableBuffer)] = '\t' ;
    Istrcat (TableBuffer,"NOT NULL") ;
}
if (count<OraCount)
    Istrcat (TableBuffer,",") ;
TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
write (hFile,TableBuffer,istrlen(TableBuffer)) ;
}
quit31:
EXEC SQL CLOSE C31 ;
ClearBuffer (TableBuffer,200) ;
Istrcat (TableBuffer,",") ;
TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
write (hFile,TableBuffer,istrlen(TableBuffer)) ;
}
quit30:
EXEC SQL CLOSE C30 ;
EXEC SQL OPEN C32 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit32 ;
for (;;)
{
    EXEC SQL FETCH C32 INTO :Index_Number,:Table_Name ;
    Table_Name.arr[Table_Name.len] = '\0' ;
    ClearBuffer (TableBuffer,200) ;
    Istrcpy (TableBuffer,"CREATE UNIQUE INDEX INDEX_#") ;
    itoa (Index_Number,itoaBuff,10) ;
    Istrcat (TableBuffer,itoaBuff) ;
    Istrcat (TableBuffer," ON ") ;
    Istrcat (TableBuffer,Table_Name.arr) ;
    Istrcat (TableBuffer," ()") ;
    EXEC SQL
    SELECT  COUNT(*) INTO :OraCount
    FROM    ATTR_OF_INDEX
    WHERE   INDEX_NUMBER = :Index_Number ;
    EXEC SQL OPEN C33 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit33 ;
    count = 0 ;
    for (;;)
    {
        count++;
        EXEC SQL FETCH C33 INTO :Attr_Number ;
        EXEC SQL
        SELECT  ATTR_NAME INTO :Attr_Name
        FROM    ATTR_DETAIL
        WHERE   ATTR_NUMBER = :Attr_Number ;
        Attr_Name.arr[Attr_Name.len] = '\0' ;
        Istrcat (TableBuffer,Attr_Name.arr) ;
        if (count<OraCount)
            Istrcat (TableBuffer,",") ;
    }
}
quit33:
EXEC SQL CLOSE C33 ;
Istrcat (TableBuffer,",") ;
TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
write (hFile,TableBuffer,istrlen(TableBuffer)) ;
}
quit32:
EXEC SQL CLOSE C32 ;
ClearBuffer (TableBuffer,200) ;
TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
write (hFile,TableBuffer,istrlen(TableBuffer)) ;
EXEC SQL OPEN C94 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit94 ;
for (;;)
{
    EXEC SQL FETCH C94 INTO :Table_Name ;
    Table_Name.arr[Table_Name.len] = '\0' ;
    ClearBuffer (TableBuffer,200) ;
    Istrcpy (TableBuffer,"CREATE VIEW ") ;
    Istrcat (TableBuffer,Table_Name.arr) ;
    Istrcat (TableBuffer," VIEW AS SELECT ") ;
    TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
    TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
    write (hFile,TableBuffer,istrlen(TableBuffer)) ;
    ClearBuffer (TableBuffer,200) ;
    Istrcpy (TableBuffer,"FROM ") ;
    Istrcat (TableBuffer,Table_Name.arr) ;
    TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
write (hFile,TableBuffer,istrlen(TableBuffer));
ClearBuffer (TableBuffer,200);
Istrcpy (TableBuffer,"WHERE ");
TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
write (hFile,TableBuffer,istrlen(TableBuffer));
EXEC SQL WHENEVER NOT FOUND CONTINUE ;
EXEC SQL
SELECT      COUNT (DISTINCT (ATTR_NUMBER)) INTO :Temp_Number
FROM        C_CONSTRAINT
WHERE       ATTR_NUMBER IN
(
        SELECT      ATTR_NUMBER
        FROM        ATTR_OF_TABLE
        WHERE       TABLE_NAME = :Table_Name
);
if (Temp_Number > 0)
{
    EXEC SQL OPEN C95 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit95 ;
    i = 0 ;
    for (;;)
    {
        i++ ;
        EXEC SQL FETCH C95 INTO :Attr_Number ;
        EXEC SQL WHENEVER NOT FOUND CONTINUE ;
        EXEC SQL
        SELECT      ATTR_NAME INTO :Attr_Name
        FROM        ATTR_DETAIL
        WHERE       ATTR_NUMBER = :Attr_Number ;
        Attr_Name.arr[Attr_Name.len] = '\0' ;
        ClearBuffer (TableBuffer,200) ;
        Istrcpy (TableBuffer,Attr_Name.arr) ;
        EXEC SQL WHENEVER NOT FOUND CONTINUE ;
        EXEC SQL
        SELECT      INVERT INTO :invert
        FROM        INVERT
        WHERE       ATTR_NUMBER = :Attr_Number ;
        Invert.arr[invert.len] = '\0' ;
        if (Invert.arr[0] == '\N')
            Istrcat (TableBuffer," IN" ) ;
        else
            Istrcat (TableBuffer," NOT IN" ) ;
        TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
        TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
        write (hFile,TableBuffer,istrlen(TableBuffer));
        ClearBuffer (TableBuffer,200) ;
        Istrcpy (TableBuffer,"") ;
        TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
        TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
        write (hFile,TableBuffer,istrlen(TableBuffer));
        EXEC SQL WHENEVER NOT FOUND CONTINUE ;
        EXEC SQL
        SELECT      COUNT(*) INTO :OraCount
        FROM        C_CONSTRAINT
        WHERE       ATTR_NUMBER = :Attr_Number ;
        EXEC SQL OPEN C98 ;
        EXEC SQL WHENEVER NOT FOUND GOTO quit96 ;
        count = 0 ;
        for (;;)
        {
            count++ ;
            EXEC SQL FETCH C98 INTO :Domain_Name ;
            Domain_Name.arr[Domain_Name.len] = '\0' ;
            ClearBuffer (TableBuffer,200) ;
            TableBuffer[istrlen(TableBuffer)] = '\t' ;
            Istrcat (TableBuffer,"") ;
            Istrcat (TableBuffer,Domain_Name.arr) ;
            Istrcat (TableBuffer,"") ;
            if (count<OraCount)
                Istrcat (TableBuffer,"") ;
            TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
            TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
            write (hFile,TableBuffer,istrlen(TableBuffer));
        }
    }
quit96:
    EXEC SQL CLOSE C98 ;
    ClearBuffer (TableBuffer,200) ;
    Istrcpy (TableBuffer,"") ;
    TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
    TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
    write (hFile,TableBuffer,istrlen(TableBuffer));
    if (i<Temp_Number)
    {
        ClearBuffer (TableBuffer,200) ;
        Istrcpy (TableBuffer,"AND") ;
        TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
        TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
        write (hFile,TableBuffer,istrlen(TableBuffer));
    }
}
quit95:
    EXEC SQL CLOSE C95 ;
}
EXEC SQL WHENEVER NOT FOUND CONTINUE ;
EXEC SQL
SELECT      COUNT (DISTINCT (ATTR_NUMBER)) INTO :index_Number
FROM        N_CONSTRAINT
WHERE       ATTR_NUMBER IN

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ในการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        (
            SELECT  ATTR_NUMBER
            FROM    ATTR_OF_TABLE
            WHERE   TABLE_NAME = :Table_Name
        );
    If (Index_Number > 0)
    {
        If (Temp_Number > 0)
        {
            ClearBuffer (TableBuffer,200) ;
            Istrcpy (TableBuffer,"AND") ;
            TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
            TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
            write (hFile,TableBuffer,istrlen(TableBuffer)) ;
        }
        EXEC SQL OPEN C97;
        EXEC SQL WHENEVER NOT FOUND GOTO quit97 ;
        i = 0 ;
        for (;;)
        {
            i++;
            EXEC SQL FETCH C97 INTO :Attr_Number ;
            EXEC SQL
            SELECT  ATTR_NAME INTO :Attr_Name
            FROM    ATTR_DETAIL
            WHERE   ATTR_NUMBER = :Attr_Number ;
            Attr_Name.arr[Attr_Name.len] = '\0' ;
            EXEC SQL
            SELECT  COUNT(*) INTO :OraCount
            FROM    N_CONSTRAINT
            WHERE   ATTR_NUMBER = :Attr_Number ;
            ClearBuffer (TableBuffer,200) ;
            Istrcpy (TableBuffer,"(") ;
            TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
            TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
            write (hFile,TableBuffer,istrlen(TableBuffer)) ;
            EXEC SQL OPEN C98 ;
            EXEC SQL WHENEVER NOT FOUND GOTO quit98 ;
            count = 0 ;
            for (;;)
            {
                count++ ;
                EXEC SQL FETCH C98 INTO :Lower_Limit,Min,Max,Upper_Limit ;
                Lower_Limit.arr[Lower_Limit.len] = '\0' ;
                Min.arr[Min.len] = '\0' ;
                Max.arr[Max.len] = '\0' ;
                Upper_Limit.arr[Upper_Limit.len] = '\0' ;
                if (Min.arr[0] == '<' || Max.arr[0] == '>')
                {
                    ClearBuffer (TableBuffer,200) ;
                    TableBuffer[istrlen(TableBuffer)] = '\t' ;
                    Istrcat (TableBuffer,"(") ;
                    Istrcat (TableBuffer,Attr_Name.arr) ;
                    if (Min.arr[0] == '<')
                    {
                        {
                            if (Lower_Limit.arr[0] == 'I')
                                Istrcat (TableBuffer," < ") ;
                            else
                                Istrcat (TableBuffer," <=") ;
                            Istrcat (TableBuffer,Max.arr) ;
                        }
                    }
                    else
                    {
                        if (Upper_Limit.arr[0] == 'I')
                            Istrcat (TableBuffer," > ") ;
                        else
                            Istrcat (TableBuffer," >=") ;
                        Istrcat (TableBuffer,Min.arr) ;
                    }
                    Istrcat (TableBuffer,")") ;
                }
                else
                {
                    ClearBuffer (TableBuffer,200) ;
                    TableBuffer[istrlen(TableBuffer)] = '\t' ;
                    Istrcat (TableBuffer,"(") ;
                    Istrcat (TableBuffer,Attr_Name.arr) ;
                    if (Lower_Limit.arr[0] == 'I')
                        Istrcat (TableBuffer," >=") ;
                    else
                        Istrcat (TableBuffer," > ") ;
                    Istrcat (TableBuffer,Min.arr) ;
                    Istrcat (TableBuffer," AND ") ;
                    Istrcat (TableBuffer,Attr_Name.arr) ;
                    if (Upper_Limit.arr[0] == 'I')
                        Istrcat (TableBuffer," <=") ;
                    else
                        Istrcat (TableBuffer," < ") ;
                    Istrcat (TableBuffer,Max.arr) ;
                    Istrcat (TableBuffer,")") ;
                }
            }
            if (count<OraCount)
                Istrcat (TableBuffer," OR") ;
            TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
            TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
            write (hFile,TableBuffer,istrlen(TableBuffer)) ;
        }
    }
quit98:
EXEC SQL CLOSE C98 ;
ClearBuffer (TableBuffer,200) ;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์... งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Istrcpy (TableBuffer,"");
TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
write (hFile,TableBuffer,istrlen(TableBuffer));
if (i<Index_Number)
{
    ClearBuffer (TableBuffer,200);
    Istrcpy (TableBuffer,"AND");
    TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
    TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
    write (hFile,TableBuffer,istrlen(TableBuffer));
}
}
quit97:
EXEC SQL CLOSE C97 ;
}
ClearBuffer (TableBuffer,200);
Istrcpy (TableBuffer,"WITH CHECK OPTION ;");
TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
TableBuffer[istrlen(TableBuffer)] = (char) 0x0d ;
TableBuffer[istrlen(TableBuffer)] = (char) 0x0a ;
write (hFile,TableBuffer,istrlen(TableBuffer));
}
quit94:
EXEC SQL CLOSE C94 ;
close (hFile);
}
}

```

Filename: PROCESS2.PC

```

#include "erextern.inc"
EXEC SQL BEGIN DECLARE SECTION ;
VARCHAR Attr_Name[30];
VARCHAR Temp_Name[30];
VARCHAR DataType[20];
VARCHAR Table_Name[30];
VARCHAR LHS_Name[30];
VARCHAR RHS_Name[30];
VARCHAR Not_Null[2];
int Type;
int Attr_Number;
int Composit_Number;
int Index_Number;
int Fd_Number;
int RHS_Number;
int LHS_Number;
int Temp_Number;
int OraCount;
VARCHAR OraImmediate[132];
EXEC SQL END DECLARE SECTION;
EXEC SQL INCLUDE SQLCA;

void ManageComposit (void)
{
    char itoaBuff[10];

    EXEC SQL DECLARE C47 CURSOR FOR
    SELECT ATTR_NUMBER,ATTR_TYPE
    FROM TEMP_ATTR
    WHERE COMPOSIT_NUMBER = -1;

    EXEC SQL DECLARE C48 CURSOR FOR
    SELECT ATTR_NUMBER,ATTR_NAME,ATTR_DATATYPE
    FROM COMPOSIT
    WHERE ATTR_NUMBER NOT IN
    (
        SELECT COMPOSIT_NUMBER
        FROM COMPOSIT
    );

    EXEC SQL DECLARE C49 CURSOR FOR
    SELECT TABLE_NAME
    FROM ATTR_OF_TABLE
    WHERE ATTR_NUMBER = :Attr_Number;

    EXEC SQL DECLARE C50 CURSOR FOR
    SELECT INDEX_NUMBER
    FROM ATTR_OF_INDEX
    WHERE ATTR_NUMBER = :Attr_Number;

    EXEC SQL OPEN C47 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit47 ;
    for (;)
    {
        EXEC SQL FETCH C47 INTO :Attr_Number,:Type ;
        EXEC SQL
        SELECT COUNT (*) INTO :OraCount
        FROM TEMP_ATTR
        CONNECT BY PRIOR ATTR_NUMBER = COMPOSIT_NUMBER
        START WITH ATTR_NUMBER = :Attr_Number ;
        if (OraCount > 1)
        {
            Istrcpy (OraImmediate.ara, "CREATE VIEW COMPOSIT AS SELECT * FROM TEMP_ATTR ");
            Istrcat (OraImmediate.ara, "CONNECT BY PRIOR ATTR_NUMBER = COMPOSIT_NUMBER ");
            Istrcat (OraImmediate.ara, "START WITH ATTR_NUMBER = ");
            itoa (Attr_Number,itoaBuff,10);
            Istrcat (OraImmediate.ara,itoaBuff);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Oralmmediate.len = strlen (Oralmmediate.arr);
EXEC SQL EXECUTE IMMEDIATE :Oralmmediate;
if (ProcessOracleErrorCode ("Create View Composit") == FALSE);
EXEC SQL OPEN C48;
EXEC SQL WHENEVER NOT FOUND GOTO quit48;
for (;;)
{
    EXEC SQL FETCH C48 INTO :Composit_Number,:Attr_Name,:DataType;
    Attr_Name.arr[Attr_Name.len] = '\0';
    DataType.arr[DataType.len] = '\0';
    EXEC SQL INSERT INTO ATTR_DETAIL (ATTR_NUMBER,ATTR_NAME,DATATYPE)
    VALUES (:Composit_Number,:Attr_Name,:DataType);
    EXEC SQL OPEN C49;
    EXEC SQL WHENEVER NOT FOUND GOTO quit49;
    for (;;)
    {
        EXEC SQL FETCH C49 INTO :Table_Name;
        Table_Name.arr[Table_Name.len] = '\0';
        if (Type == N_ATTRIBUTE || Type == K_ATTRIBUTE)
            Istrcpy (Not_Null.arr,"N");
        else
            Istrcpy (Not_Null.arr,"Y");
        Not_Null.len = strlen (Not_Null.arr);
        EXEC SQL INSERT INTO ATTR_OF_TABLE (TABLE_NAME,ATTR_NUMBER,NOT_NULL)
        VALUES (:Table_Name,:Composit_Number,:Not_Null);
    }
quit49:
EXEC SQL CLOSE C49;
EXEC SQL OPEN C50;
EXEC SQL WHENEVER NOT FOUND GOTO quit50;
for (;;)
{
    EXEC SQL FETCH C50 INTO :Index_Number;
    EXEC SQL INSERT INTO ATTR_OF_INDEX (INDEX_NUMBER,ATTR_NUMBER)
    VALUES (:Index_Number,:Composit_Number);
}
quit50:
EXEC SQL CLOSE C50;
}
quit48:
EXEC SQL CLOSE C48;
EXEC SQL DROP VIEW COMPOSIT;
if (ProcessOracleErrorCode ("Drop View Composit") == FALSE);
EXEC SQL COMMIT WORK;
EXEC SQL DELETE FROM ATTR_OF_DETAIL WHERE ATTR_NUMBER = :Attr_Number;
EXEC SQL DELETE FROM ATTR_OF_TABLE WHERE ATTR_NUMBER = :Attr_Number;
if (Type == P_ATTRIBUTE || Type == MV_ATTRIBUTE)
    EXEC SQL DELETE FROM ATTR_OF_INDEX WHERE ATTR_NUMBER = :Attr_Number;
}
quit47:
EXEC SQL CLOSE C47;
}
void ProcessBCNF (void)
{
    FARPROC lpProcFd;

    SetCapture (hDrawWnd);
    hOldCursor = SetCursor (hWaitCursor);
    SetCursor (hOldCursor);
    ReleaseCapture ();
    lpProcFd = MakeProcInstance (FdProc, hInst);
    DialogBox (hInst, "FdBox", hMainWnd, lpProcFd);
    FreeProcInstance (lpProcFd);
}
void GenDefaultFD ()
{
    EXEC SQL DECLARE C54 CURSOR FOR
    SELECT INDEX_NUMBER,TABLE_NAME
    FROM TABLE_OF_INDEX;

    EXEC SQL DECLARE C55 CURSOR FOR
    SELECT ATTR_NUMBER
    FROM ATTR_DETAIL
    WHERE ATTR_NUMBER IN
    (
        SELECT ATTR_NUMBER
        FROM ATTR_OF_INDEX
        WHERE INDEX_NUMBER = :Index_Number
    );

    EXEC SQL DECLARE C56 CURSOR FOR
    SELECT ATTR_NUMBER
    FROM ATTR_DETAIL
    WHERE ATTR_NUMBER IN
    (
        SELECT ATTR_NUMBER
        FROM ATTR_OF_TABLE
        WHERE TABLE_NAME = :Table_Name
    );

    EXEC SQL WHENEVER NOT FOUND CONTINUE;
    EXEC SQL DELETE FD;
    if (ProcessOracleErrorCode ("DELETE * FROM FD") == FALSE);
    EXEC SQL DELETE LHS;
    if (ProcessOracleErrorCode ("DELETE * FROM LHS") == FALSE);
    EXEC SQL DELETE RHS;
    if (ProcessOracleErrorCode ("DELETE * FROM RHS") == FALSE);
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EXEC SQL DELETE TEMP_FD ;
if (ProcessOracleErrorCode ("DELETE * FROM TEMP_FD") == FALSE) ;
EXEC SQL DELETE TEMP_LHS ;
if (ProcessOracleErrorCode ("DELETE * FROM TEMP_LHS") == FALSE) ;
EXEC SQL DELETE TEMP_RHS ;
if (ProcessOracleErrorCode ("DELETE * FROM TEMP_RHS") == FALSE) ;
EXEC SQL DELETE TEMP_ATTR_OF_INDEX ;
if (ProcessOracleErrorCode ("DELETE * FROM TEMP_ATTR_OF_INDEX") == FALSE) ;
EXEC SQL DELETE TEMP_TABLE_OF_INDEX ;
if (ProcessOracleErrorCode ("DELETE * FROM TEMP_TABLE_OF_INDEX") == FALSE) ;
EXEC SQL DELETE TEMP_ATTR_OF_TABLE ;
if (ProcessOracleErrorCode ("DELETE * FROM TEMP_ATTR_OF_TABLE") == FALSE) ;
EXEC SQL INSERT INTO TEMP_ATTR_OF_INDEX SELECT * FROM ATTR_OF_INDEX ;
if (ProcessOracleErrorCode ("INSERT INTO TEMP_ATTR_OF_INDEX") == FALSE) ;
EXEC SQL INSERT INTO TEMP_TABLE_OF_INDEX SELECT * FROM TABLE_OF_INDEX ;
if (ProcessOracleErrorCode ("INSERT INTO TEMP_TABLE_OF_INDEX") == FALSE) ;
EXEC SQL INSERT INTO TEMP_ATTR_OF_TABLE SELECT * FROM ATTR_OF_TABLE ;
if (ProcessOracleErrorCode ("INSERT INTO TEMP_ATTR_OF_TABLE") == FALSE) ;
Fd_Number = 1 ;
EXEC SQL OPEN C54 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit54 ;
for (;;)
{
    EXEC SQL FETCH C54 INTO :Index_Number, :Table_Name ;
    Table_Name.ar[Table_Name.len] = '\0' ;
    EXEC SQL OPEN C56 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit56 ;
    for (;;)
    {
        EXEC SQL FETCH C56 INTO :RHS_Number ;
        EXEC SQL INSERT INTO RHS (FD_NUMBER,RHS_NUMBER) VALUES (:Fd_Number,:RHS_Number) ;
        EXEC SQL OPEN C55 ;
        EXEC SQL WHENEVER NOT FOUND GOTO quit55 ;
        for (;;)
        {
            EXEC SQL FETCH C55 INTO :LHS_Number ;
            EXEC SQL INSERT INTO LHS (FD_NUMBER,LHS_NUMBER) VALUES (:Fd_Number,:LHS_Number) ;
        }
        quit55: EXEC SQL CLOSE C55 ;
        EXEC SQL INSERT INTO FD (FD_NUMBER,TABLE_NAME) VALUES (:Fd_Number,:Table_Name) ;
        Fd_Number++ ;
    }
    quit56: EXEC SQL CLOSE C56 ;
}
quit54: EXEC SQL CLOSE C54 ;
EXEC SQL WHENEVER NOT FOUND CONTINUE ;
EXEC SQL DELETE ATTR_OF_INDEX ;
if (ProcessOracleErrorCode ("DELETE * FROM ATTR_OF_INDEX") == FALSE) ;
EXEC SQL DELETE TABLE_OF_INDEX ;
if (ProcessOracleErrorCode ("DELETE * FROM TABLE_OF_INDEX") == FALSE) ;
EXEC SQL DELETE ATTR_OF_TABLE ;
if (ProcessOracleErrorCode ("DELETE * FROM ATTR_OF_TABLE") == FALSE) ;
}
BOOL FAR PASCAL __export FdProc (HWND hDlg, unsigned message, WORD wParam, LONG lParam)
{
    EXEC SQL DECLARE C45 CURSOR FOR
    SELECT DISTINCT TABLE_NAME
    FROM TEMP_ATTR_OF_TABLE ;

    EXEC SQL DECLARE C51 CURSOR FOR
    SELECT FD_NUMBER
    FROM FD
    WHERE TABLE_NAME = :Table_Name ;

    switch (message)
    {
        case WM_INITDIALOG:
        {
            CursorOpen = FALSE ;
            EXEC SQL OPEN C45 ;
            EXEC SQL WHENEVER NOT FOUND GOTO quit45 ;
            for (;;)
            {
                EXEC SQL FETCH C45 INTO :Table_Name ;
                Table_Name.ar[Table_Name.len] = '\0' ;
                SendDlgItemMessage (hDlg, IDC_TNAME, CB_ADDSTRING, 0, (LPARAM) ((LPSTR) Table_Name.ar)) ;
            }
            quit45: EXEC SQL CLOSE C45 ;
            SendDlgItemMessage (hDlg, IDC_TNAME, CB_SELECTSTRING, 0, (LPARAM) ((LPSTR) Table_Name.ar)) ;
            Table_Name.len = lstrlen (Table_Name.ar) ;
            ClearListBox (hDlg) ;
            return (TRUE) ;
        }
        case WM_COMMAND:
        {
            switch (wParam)
            {
                case IDC_EXIT:
                {
                    if (CursorOpen == TRUE)
                    {
                        CursorOpen = FALSE ;
                        EXEC SQL CLOSE C51 ;
                    }
                    EndDialog (hDlg, TRUE) ;
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่คัดลอกมาเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return (TRUE);
    }
    break;
    case IDC_RUN:
    {
        if (CursorOpen == TRUE)
        {
            CursorOpen = FALSE;
            EXEC SQL CLOSE C51;
        }
        SetCapture (hDrawWnd);
        hOldCursor = SetCursor (hWaitCursor);
        if (GenBCNF () == NOERROR)
        {
            SetCursor (hOldCursor);
            ReleaseCapture ();
            EndDialog (hDlg, TRUE);
            EnableMenuItem (GetMenu(hMainWnd),IDM_FIFTH, MF_ENABLED);
            return (TRUE);
        }
        else
        {
            SetCursor (hOldCursor);
            ReleaseCapture ();
            Istrcpy (Table_Name.arr,Temp_Name.arr);
            Table_Name.len = strlen (Table_Name.arr);
            ClearListBox (hDlg);
        }
    }
    break;
    case IDC_SELECT:
    {
        CursorOpen = TRUE;
        EXEC SQL OPEN C51;
        EnableWindow (GetDlgItem (hDlg,IDC_FETCH),TRUE);
        EnableWindow (GetDlgItem (hDlg,IDC_SELECT),FALSE);
    }
    break;
    case IDC_DELETE:
    {
        EXEC SQL DELETE FROM LHS WHERE FD_NUMBER = :Fd_Number;
        EXEC SQL DELETE FROM RHS WHERE FD_NUMBER = :Fd_Number;
        EXEC SQL DELETE FROM FD WHERE FD_NUMBER = :Fd_Number;
    }
    case IDC_FETCH:
    {
        EXEC SQL WHENEVER NOT FOUND CONTINUE;
        EXEC SQL FETCH C51 INTO :Fd_Number;
        if (sqlca.sqlcode == 1403)
        {
            CursorOpen = FALSE;
            EXEC SQL CLOSE C51;
            EnableWindow (GetDlgItem (hDlg,IDC_FETCH),FALSE);
            EnableWindow (GetDlgItem (hDlg,IDC_DELETE),FALSE);
            EnableWindow (GetDlgItem (hDlg,IDC_SELECT),TRUE);
            MessageBox(GetFocus (), "No More Data", "Fetch Error", MB_OK);
            ClearListBox (hDlg);
        }
        else
        {
            EnableWindow (GetDlgItem (hDlg,IDC_DELETE),TRUE);
            SelectIndexListBox (hDlg);
        }
    }
    break;
    case IDC_INSERT:
    {
        if (CursorOpen == TRUE)
        {
            CursorOpen = FALSE;
            EXEC SQL CLOSE C51;
        }
        ClearListBox (hDlg);
        EnableWindow (GetDlgItem (hDlg,IDC_CONFIRM),TRUE);
        EnableWindow (GetDlgItem (hDlg,IDC_CANCEL),TRUE);
        EnableWindow (GetDlgItem (hDlg,IDC_SELECT),FALSE);
        EnableWindow (GetDlgItem (hDlg,IDC_FETCH),FALSE);
        EnableWindow (GetDlgItem (hDlg,IDC_DELETE),FALSE);
        EnableWindow (GetDlgItem (hDlg,IDC_INSERT),FALSE);
        EnableWindow (GetDlgItem (hDlg,IDC_RUN),FALSE);
        EnableWindow (GetDlgItem (hDlg,IDC_EXIT),FALSE);
    }
    break;
    case IDC_CANCEL:
    {
        EnableWindow (GetDlgItem (hDlg,IDC_CONFIRM),FALSE);
        EnableWindow (GetDlgItem (hDlg,IDC_CANCEL),FALSE);
        EnableWindow (GetDlgItem (hDlg,IDC_SELECT),TRUE);
        EnableWindow (GetDlgItem (hDlg,IDC_FETCH),FALSE);
        EnableWindow (GetDlgItem (hDlg,IDC_DELETE),FALSE);
        EnableWindow (GetDlgItem (hDlg,IDC_INSERT),TRUE);
        EnableWindow (GetDlgItem (hDlg,IDC_RUN),TRUE);
        EnableWindow (GetDlgItem (hDlg,IDC_EXIT),TRUE);
        ClearListBox (hDlg);
    }
    break;
    case IDC_CONFIRM:
    {
        HLOCAL hMem;
        int i;

```

เอกสารนี้เป็นเอกสารที่  
 1. วัตถุประสงค์: ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 2. หน่วยงาน: วิทยาลัยพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
 3. หมายเหตุ: ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DWORD index,dwIndex,cItems,LcItems,RcItems;
```

```
EnableWindow (GetDlgItem (hDlg, IDC_CONFIRM), FALSE);
EnableWindow (GetDlgItem (hDlg, IDC_CANCEL), FALSE);
EnableWindow (GetDlgItem (hDlg, IDC_SELECT), TRUE);
EnableWindow (GetDlgItem (hDlg, IDC_FETCH), FALSE);
EnableWindow (GetDlgItem (hDlg, IDC_DELETE), FALSE);
EnableWindow (GetDlgItem (hDlg, IDC_INSERT), TRUE);
EnableWindow (GetDlgItem (hDlg, IDC_RUN), TRUE);
EnableWindow (GetDlgItem (hDlg, IDC_EXIT), TRUE);
LcItems = SendDlgItemMessage(hDlg, IDC_LHS, LB_GETSELCOUNT, 0, 0);
RcItems = SendDlgItemMessage(hDlg, IDC_RHS, LB_GETSELCOUNT, 0, 0);
if (LcItems == 0 || RcItems == 0)
{
    ErrorMessage ("LHS or RHS no selection");
    break;
}
EXEC SQL SELECT MAX(FD_NUMBER) INTO :Fd_Number FROM FD;
Fd_Number++;
EXEC SQL INSERT INTO FD (FD_NUMBER, TABLE_NAME) VALUES (:Fd_Number, :Table_Name);
cItems = SendDlgItemMessage(hDlg, IDC_LHS, LB_GETSELCOUNT, 0, 0);
if ((hMem = LocalAlloc (LMEM_MOVEABLE, sizeof (int)* (int) cItems)) != NULL)
{
    lpItems = (int*) LocalLock (hMem);
    dwIndex = SendDlgItemMessage
    (
        hDlg, IDC_LHS, LB_GETSELEITEMS, (WPARAM) cItems, (LPARAM) (int FAR*) lpItems
    );
    for (index=0; index<cItems; index++)
    {
        LHS_Number = NumberIndex((WPARAM)lpItems[(int) index]);
        EXEC SQL INSERT INTO LHS (FD_NUMBER, LHS_NUMBER)
        VALUES(:Fd_Number, :LHS_Number);
        EXEC SQL COMMIT WORK;
    }
    LocalUnlock (hMem);
    LocalFree (hMem);
}
cItems = SendDlgItemMessage(hDlg, IDC_RHS, LB_GETSELCOUNT, 0, 0);
if ((hMem = LocalAlloc (LMEM_MOVEABLE, sizeof (int)* (int) cItems)) != NULL)
{
    lpItems = (int*) LocalLock (hMem);
    dwIndex = SendDlgItemMessage
    (
        hDlg, IDC_RHS, LB_GETSELEITEMS, (WPARAM) cItems, (LPARAM) (int FAR*) lpItems
    );
    for (index=0; index<cItems; index++)
    {
        RHS_Number = NumberIndex((WPARAM)lpItems[(int) index]);
        EXEC SQL INSERT INTO RHS (FD_NUMBER, RHS_NUMBER)
        VALUES(:Fd_Number, :RHS_Number);
        EXEC SQL COMMIT WORK;
    }
    LocalUnlock (hMem);
    LocalFree (hMem);
}
}
break;
case IDC_TNAME:
{
    switch (HIWORD (lParam))
    {
        case CBN_SELENDOK:
        {
            DWORD dwIndex;

            dwIndex = SendDlgItemMessage(hDlg, IDC_TNAME, CB_GETCURSEL, 0, 0);
            if (dwIndex != CB_ERR)
            {
                SendDlgItemMessage
                (
                    hDlg, IDC_TNAME, CB_GETLBTEXT, (WPARAM) dwIndex,
                    (LPARAM) ((LPSTR) Table_Name.arr)
                );
                Table_Name.len = strlen (Table_Name.arr);
            }
            if (CursorOpen == TRUE)
            {
                CursorOpen = FALSE;
                EXEC SQL CLOSE CS1;
            }
            ClearListBox (hDlg);
            EnableWindow (GetDlgItem (hDlg, IDC_CONFIRM), FALSE);
            EnableWindow (GetDlgItem (hDlg, IDC_CANCEL), FALSE);
            EnableWindow (GetDlgItem (hDlg, IDC_SELECT), TRUE);
            EnableWindow (GetDlgItem (hDlg, IDC_FETCH), FALSE);
            EnableWindow (GetDlgItem (hDlg, IDC_DELETE), FALSE);
            EnableWindow (GetDlgItem (hDlg, IDC_INSERT), TRUE);
            EnableWindow (GetDlgItem (hDlg, IDC_RUN), TRUE);
            EnableWindow (GetDlgItem (hDlg, IDC_EXIT), TRUE);
        }
    }
}
break;
}
return (FALSE);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void ClearListBox (HWND hDlg)
{
    int i;

    EXEC SQL DECLARE C48 CURSOR FOR
    SELECT ATTR_NUMBER,ATTR_NAME
    FROM ATTR_DETAIL
    WHERE ATTR_NUMBER IN
        (
            SELECT ATTR_NUMBER
            FROM TEMP_ATTR_OF_TABLE
            WHERE TABLE_NAME = :Table_Name
        )
    ORDER BY ATTR_NAME ;

    i = 0;

    SendDlgItemMessage (hDlg, IDC_LHS, LB_RESETCONTENT, 0, 0);
    SendDlgItemMessage (hDlg, IDC_RHS, LB_RESETCONTENT, 0, 0);
    EXEC SQL OPEN C48 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit48 ;
    for(;;)
    {
        EXEC SQL FETCH C48 INTO :Attr_Number,:Attr_Name ;
        Numberindex[++] = Attr_Number ;
        Attr_Name.arr[Attr_Name.len] = '\0' ;
        SendDlgItemMessage (hDlg, IDC_LHS, LB_ADDSTRING, 0, (LPARAM) ((LPSTR) Attr_Name.arr)) ;
        SendDlgItemMessage (hDlg, IDC_RHS, LB_ADDSTRING, 0, (LPARAM) ((LPSTR) Attr_Name.arr)) ;
    }
quit48:
    EXEC SQL CLOSE C48 ;
}

void SelectIndex.ListBox (HWND hDlg)
{
    EXEC SQL DECLARE C52 CURSOR FOR
    SELECT ATTR_NAME
    FROM ATTR_DETAIL
    WHERE ATTR_NUMBER IN
        (
            SELECT LHS_NUMBER
            FROM LHS
            WHERE FD_NUMBER = :Fd_Number
        )
    );

    EXEC SQL DECLARE C53 CURSOR FOR
    SELECT ATTR_NAME
    FROM ATTR_DETAIL
    WHERE ATTR_NUMBER IN
        (
            SELECT RHS_NUMBER
            FROM RHS
            WHERE FD_NUMBER = :Fd_Number
        )
    );

    ClearListBox (hDlg) ;
    EXEC SQL OPEN C52 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit52;
    for (;;)
    {
        DWORD dwIndex;

        EXEC SQL FETCH C52 INTO :LHS_Name ;
        LHS_Name.arr[LHS_Name.len] = '\0' ;
        dwIndex = SendDlgItemMessage (hDlg, IDC_LHS, LB_FINDSTRING, 0, (LPARAM) ((LPSTR) LHS_Name.arr)) ;
        SendDlgItemMessage (hDlg, IDC_LHS, LB_SETSEL, TRUE, (LPARAM) MAKELPARAM(dwIndex, 0)) ;
    }
quit52:
    EXEC SQL CLOSE C52 ;
    EXEC SQL OPEN C53 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit53;
    for (;;)
    {
        DWORD dwIndex;

        EXEC SQL FETCH C53 INTO :RHS_Name ;
        RHS_Name.arr[RHS_Name.len] = '\0' ;
        dwIndex = SendDlgItemMessage (hDlg, IDC_RHS, LB_FINDSTRING, 0, (LPARAM) ((LPSTR) RHS_Name.arr)) ;
        SendDlgItemMessage (hDlg, IDC_RHS, LB_SETSEL, TRUE, (LPARAM) MAKELPARAM(dwIndex, 0)) ;
    }
quit53:
    EXEC SQL CLOSE C53 ;
}

BOOL GenBCNF (void)
{
    EXEC SQL WHENEVER NOT FOUND CONTINUE ;
    EXEC SQL INSERT INTO TEMP_FD SELECT * FROM FD ;
    if (ProcessOracleErrorCode (" INSERT TEMP_FD ") == FALSE) ;
    EXEC SQL INSERT INTO TEMP_LHS SELECT * FROM LHS ;
    if (ProcessOracleErrorCode (" INSERT TEMP_LHS ") == FALSE) ;
    EXEC SQL INSERT INTO TEMP_RHS SELECT * FROM RHS ;
    if (ProcessOracleErrorCode (" INSERT TEMP_RHS ") == FALSE) ;
    ErrorFlag = NOERROR ;
    if (IsValidGroupFd () == NOERROR)
    {
        SeparateFD 0 ;
        KillRedundantDependency 0 ;
        GenThirdNF 0 ;
    }
}

```

/\* if A -> XY then A -> X and A -> Y \*/

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        MapBCNF ();
    }
    else
        ErrorFlag = ERROR ;
    return (ErrorFlag) ;
}

void SeparateFD (void)
{
    EXEC SQL DECLARE C59 CURSOR FOR
    SELECT      FD_NUMBER
    FROM        TEMP_RHS
    GROUP BY    FD_NUMBER
    HAVING      COUNT (*) > 1 ;

    EXEC SQL DECLARE C60 CURSOR FOR
    SELECT      RHS_NUMBER
    FROM        TEMP_RHS
    WHERE       FD_NUMBER = :Fd_Number ;

    EXEC SQL DECLARE C61 CURSOR FOR
    SELECT      LHS_NUMBER
    FROM        TEMP_LHS
    WHERE       FD_NUMBER = :Fd_Number ;

    EXEC SQL OPEN C59 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit59 ;
    for (;;)
    {
        EXEC SQL FETCH C59 INTO :Fd_Number ;
        EXEC SQL
        SELECT   TABLE_NAME INTO :Table_Name
        FROM     TEMP_FD
        WHERE    FD_NUMBER = :Fd_Number ;
        Table_Name.arr[Table_Name.len] = '\0' ;
        EXEC SQL OPEN C60 ;
        EXEC SQL WHENEVER NOT FOUND GOTO quit60 ;
        for (;;)
        {
            EXEC SQL FETCH C60 INTO :RHS_Number ;
            EXEC SQL SELECT MAX (FD_NUMBER) INTO :OraCount FROM TEMP_FD ;
            OraCount++ ;
            EXEC SQL INSERT INTO TEMP_FD (FD_NUMBER, TABLE_NAME) VALUES (:OraCount, :Table_Name) ;
            if (ProcessOracleErrorCode ("INSERT NEW TEMP_FD") == FALSE) ;
            EXEC SQL INSERT INTO TEMP_RHS (FD_NUMBER, RHS_NUMBER) VALUES (:OraCount, :RHS_Number) ;
            if (ProcessOracleErrorCode ("INSERT NEW TEMP_RHS") == FALSE) ;
            EXEC SQL OPEN C61 ;
            EXEC SQL WHENEVER NOT FOUND GOTO quit61 ;
            for (;;)
            {
                EXEC SQL FETCH C61 INTO :LHS_Number ;
                EXEC SQL INSERT INTO TEMP_LHS (FD_NUMBER, LHS_NUMBER) VALUES (:OraCount, :LHS_Number) ;
                if (ProcessOracleErrorCode ("INSERT NEW TEMP_LHS") == FALSE) ;
            }
            quit61:
            EXEC SQL CLOSE C61 ;
        }
        quit60:
        EXEC SQL CLOSE C60 ;
        EXEC SQL DELETE FROM TEMP_LHS WHERE FD_NUMBER = :Fd_Number ;
        EXEC SQL DELETE FROM TEMP_RHS WHERE FD_NUMBER = :Fd_Number ;
        EXEC SQL DELETE FROM TEMP_FD WHERE FD_NUMBER = :Fd_Number ;
    }
    quit59:
    EXEC SQL CLOSE C59 ;
    EXEC SQL COMMIT WORK ;
}

BOOL IsValidGroupFd ()
{
    char buffer[100] ;

    EXEC SQL DECLARE C57 CURSOR FOR
    SELECT   TABLE_NAME
    FROM     TEMP_TABLE_OF_INDEX ;

    ErrorFlag = NOERROR ;
    Istrcpy (Temp_Name.arr, Table_Name.arr) ;
    Temp_Name.len = Istrlen (Temp_Name.arr) ;
    EXEC SQL
    CREATE   VIEW TEMP (FD_NUMBER, ATTR_NUMBER) AS
    SELECT   FD_NUMBER, LHS_NUMBER
    FROM     TEMP_LHS
    UNION
    SELECT   FD_NUMBER, RHS_NUMBER
    FROM     TEMP_RHS ;
    EXEC SQL OPEN C57 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit57 ;
    for (;;)
    {
        EXEC SQL FETCH C57 INTO :Table_Name ;
        Table_Name.arr[Table_Name.len] = '\0' ;
        EXEC SQL
        SELECT   COUNT (DISTINCT ATTR_NUMBER) INTO :OraCount
        FROM     TEMP_ATTR_OF_TABLE
        WHERE    TABLE_NAME = :Table_Name ;
        EXEC SQL
        SELECT   COUNT (DISTINCT ATTR_NUMBER) INTO :Temp_Number
        FROM     TEMP
        WHERE    FD_NUMBER IN

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        (
            SELECT  FD_NUMBER
            FROM    TEMP_FD
            WHERE   TABLE_NAME = :Table_Name
        );

    if (OraCount != Temp_Number)
    {
        ErrorFlag = ERROR ;
        Istrcpy (buffer," FD not cover all attribute in table ");
        Istrcat (buffer,Table_Name.arn) ;
        ErrorMessage (buffer);
    }
}
quit57:
EXEC SQL CLOSE C57 ;
EXEC SQL DROP VIEW TEMP ;
return (ErrorFlag);
}

BOOL KillRedundantDependency ()
{
    EXEC SQL  DECLARE C62 CURSOR FOR
    SELECT    TABLE_NAME
    FROM      TEMP_TABLE_OF_INDEX ;

    EXEC SQL OPEN C62 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit62 ;
    for (;;)
    {
        EXEC SQL FETCH C62 INTO :Table_Name ;
        Table_Name.arn[Table_Name.len] = '\0' ;
        FindRedundantFD () ;
    }
quit62:
EXEC SQL CLOSE C62 ;
EXEC SQL WHENEVER NOT FOUND CONTINUE ;
EXEC SQL DELETE KILL ;
if (ProcessOracleErrorCode ("DELETE KILL") == FALSE)
    return (FALSE);
return (TRUE);
}

void FindRedundantFD ()
{
    int    ChangeSetT ;

    EXEC SQL  DECLARE C63 CURSOR FOR
    SELECT    FD_NUMBER
    FROM      TEMP_FD
    WHERE     TABLE_NAME = :Table_Name ;

    EXEC SQL  DECLARE C64 CURSOR FOR
    SELECT    LHS_NUMBER
    FROM      TEMP_LHS
    WHERE     FD_NUMBER = :Fd_Number ;

    EXEC SQL  DECLARE C65 CURSOR FOR
    SELECT    DISTINCT RHS_NUMBER
    FROM      TEMP_RHS
    WHERE     FD_NUMBER IN
    (
        (
            SELECT  FD_NUMBER
            FROM    TEMP_LHS
            WHERE   LHS_NUMBER IN
            (
                SELECT  DISTINCT ATTR_NUMBER
                FROM    KILL
            )
        )
        MINUS
        SELECT  FD_NUMBER
        FROM    TEMP_LHS
        WHERE   LHS_NUMBER NOT IN
        (
            SELECT  DISTINCT ATTR_NUMBER
            FROM    KILL
        )
    )
    INTERSECT
    SELECT  FD_NUMBER
    FROM    TEMP_FD
    WHERE   TABLE_NAME = :Table_Name AND FD_NUMBER != :Fd_Number
);

EXEC SQL OPEN C63 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit63 ;
for (;;)
{
    EXEC SQL FETCH C63 INTO :Fd_Number ;
    EXEC SQL OPEN C64 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit64 ;
    for (;;)
    {
        EXEC SQL FETCH C64 INTO :LHS_Number ;
        EXEC SQL INSERT INTO KILL (ATTR_NUMBER) VALUES (:LHS_Number) ;
    }
quit64:
EXEC SQL CLOSE C64 ;
EXEC SQL COMMIT WORK ;
}

```

เอกสารนี้เป็นเอกสารราชการของสำนักงานคณะกรรมการการอุดมศึกษา ห้ามมิให้เผยแพร่โดยไม่ได้รับอนุญาตจากผู้อนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EXEC SQL
SELECT   RHS_NUMBER INTO :RHS_Number
FROM     TEMP_RHS
WHERE    FD_NUMBER = :Fd_Number ;
Temp_Number = RHS_Number ;
do
{
    ChangeSet = FALSE ;
    EXEC SQL OPEN C65 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit65 ;
    for (,;)
    {
        EXEC SQL FETCH C65 INTO :Attr_Number ;
        EXEC SQL
        SELECT   COUNT(*) INTO :OraCount
        FROM     KILL
        WHERE    ATTR_NUMBER = :Attr_Number ;
        if (OraCount == 0)
        {
            EXEC SQL INSERT INTO KILL (ATTR_NUMBER) VALUES (:Attr_Number) ;
            EXEC SQL COMMIT WORK ;
            ChangeSet = TRUE ;
        }
        EXEC SQL
        SELECT   COUNT(*) INTO :OraCount
        FROM     KILL
        WHERE    ATTR_NUMBER = :Temp_Number ;
        if (OraCount > 0)
        {
            EXEC SQL DELETE FROM TEMP_LHS WHERE FD_NUMBER = :Fd_Number ;
            EXEC SQL DELETE FROM TEMP_RHS WHERE FD_NUMBER = :Fd_Number ;
            EXEC SQL DELETE FROM TEMP_FD WHERE FD_NUMBER = :Fd_Number ;
            EXEC SQL COMMIT WORK ;
            ChangeSet = FALSE ;
            break ;
        }
    }
quit65:
    EXEC SQL CLOSE C65 ;
} while (ChangeSet == TRUE) ;
EXEC SQL DELETE FROM KILL ;
EXEC SQL COMMIT WORK ;
}
quit63:
EXEC SQL CLOSE C63 ;
}

```

Filename: *PROCESS3.PC*

```

#include "erextem.inc"
EXEC SQL BEGIN DECLARE SECTION ;
VARCHAR Attr_Name[30] ;
VARCHAR Temp_Name[30] ;
VARCHAR DataType[20] ;
VARCHAR Table_Name[30] ;
VARCHAR Not_Null[2] ;
VARCHAR Domain_Name[30] ;
VARCHAR Invert[2] ;
VARCHAR Lower_Limit[2] ;
VARCHAR Upper_Limit[2] ;
VARCHAR Min[30] ;
VARCHAR Max[30] ;
int Role_Number ;
int Attr_Number ;
int Index_Number ;
int Fd_Number ;
int RHS_Number ;
int LHS_Number ;
int Temp_Number ;
int OraCount ;
VARCHAR OraImmediate[132] ;
EXEC SQL END DECLARE SECTION ;
EXEC SQL INCLUDE SQLCA ;

```

void MapThird (void)

```

{
    int i ;

    EXEC SQL DECLARE C66 CURSOR FOR
    SELECT   FD_NUMBER
    FROM     LHS_VIEW
    GROUP BY FD_NUMBER
    HAVING   COUNT(*) = :Temp_Number ;

    EXEC SQL DECLARE C67 CURSOR FOR
    SELECT   FD_NUMBER
    FROM     LHS_VIEW
    WHERE    LHS_NUMBER IN
    (
        SELECT   LHS_NUMBER
        FROM     LHS_VIEW
        WHERE    FD_NUMBER = :Fd_Number
    )
    GROUP BY FD_NUMBER
    HAVING   COUNT(*) = :Temp_Number
    MINUS
    SELECT   FD_NUMBER
    FROM     LHS_VIEW
    WHERE    LHS_NUMBER NOT IN

```

เอกสารนี้เป็นเอกสารที่ให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        (
            SELECT  LHS_NUMBER
            FROM    LHS_VIEW
            WHERE   FD_NUMBER = :Fd_Number
        )
    MINUS
    SELECT  FD_NUMBER
    FROM    LHS_VIEW
    WHERE   FD_NUMBER = :Fd_Number;

EXEC SQL
SELECT  MAX(COUNT(*)) INTO :OraCount
FROM    LHS_VIEW
GROUP BY FD_NUMBER;
for (i=1;i<=OraCount;i++)
{
    Temp_Number = i;
    EXEC SQL OPEN C66;
    EXEC SQL WHENEVER NOT FOUND GOTO quit66;
    for(;;)
    {
        EXEC SQL FETCH C66 INTO :Fd_Number;
        EXEC SQL OPEN C67;
        EXEC SQL WHENEVER NOT FOUND GOTO quit67;
        for (;;)
        {
            EXEC SQL FETCH C67 INTO :Role_Number;
            EXEC SQL UPDATE TEMP_RHS SET FD_NUMBER = :Fd_Number WHERE FD_NUMBER = :Role_Number;
            if (ProcessOracleErrorCode ("UPDATE RHS") == FALSE);
            EXEC SQL COMMIT WORK;
        }
    }
    quit67: EXEC SQL CLOSE C67;
}
quit66: EXEC SQL CLOSE C66;
}
}

BOOL GenThirdNF (void)
{
    int  count;

    EXEC SQL DECLARE C68 CURSOR FOR
    SELECT  DISTINCT TABLE_NAME
    FROM    TEMP_FD;

    EXEC SQL DECLARE C80 CURSOR FOR
    SELECT  DISTINCT TABLE_NAME
    FROM    FD;

    EXEC SQL DECLARE C81 CURSOR FOR
    SELECT  LHS_NUMBER
    FROM    LHS
    WHERE   FD_NUMBER IN
    (
        SELECT  FD_NUMBER
        FROM    FD
        WHERE   TABLE_NAME = :Table_Name
    )
    UNION
    SELECT  RHS_NUMBER
    FROM    RHS
    WHERE   FD_NUMBER IN
    (
        SELECT  FD_NUMBER
        FROM    FD
        WHERE   TABLE_NAME = :Table_Name
    )
    MINUS
    SELECT  LHS_NUMBER
    FROM    TEMP_LHS
    WHERE   FD_NUMBER IN
    (
        SELECT  FD_NUMBER
        FROM    TEMP_FD
        WHERE   TABLE_NAME = :Table_Name
    )
    MINUS
    SELECT  RHS_NUMBER
    FROM    TEMP_RHS
    WHERE   FD_NUMBER IN
    (
        SELECT  FD_NUMBER
        FROM    TEMP_FD
        WHERE   TABLE_NAME = :Table_Name
    )
    );

    EXEC SQL DECLARE C82 CURSOR FOR
    SELECT  LHS_NUMBER
    FROM    TEMP_LHS
    WHERE   FD_NUMBER = :Temp_Number;

    EXEC SQL OPEN C68;
    EXEC SQL WHENEVER NOT FOUND GOTO quit68;
    for (;;)
    {
        EXEC SQL FETCH C68 INTO :Table_Name;
        Table_Name.arr[Table_Name.len] = '\0';
        Istrcpy (Oralmediate.arr, "CREATE VIEW LHS_VIEW AS SELECT * FROM TEMP_LHS");
    }
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Istrcat (Oralmmediate.arr, "WHERE FD_NUMBER IN ");
Istrcat (Oralmmediate.arr, "( SELECT FD_NUMBER FROM TEMP_FD WHERE TABLE_NAME = ") ;
Istrcat (Oralmmediate.arr, Table_Name.arr) ;
Istrcat (Oralmmediate.arr, " ) ) ;
Oralmmediate.len = Istrlen (Oralmmediate.arr) ;
EXEC SQL EXECUTE IMMEDIATE :Oralmmediate ;
if (ProcessOracleErrorCode ("Create View LHS_VIEW") == FALSE) return FALSE ;
MapThird 0 ;
EXEC SQL DROP VIEW LHS_VIEW ;
EXEC SQL COMMIT WORK ;
}
quit88:
EXEC SQL CLOSE C68 ;
EXEC SQL DELETE TEMP_FD WHERE FD_NUMBER NOT IN ( SELECT DISTINCT FD_NUMBER FROM TEMP_RHS ) ;
EXEC SQL DELETE TEMP_LHS WHERE FD_NUMBER NOT IN ( SELECT DISTINCT FD_NUMBER FROM TEMP_RHS ) ;
EXEC SQL OPEN C80 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit80 ;
for (;;)
{
EXEC SQL FETCH C80 INTO :Table_Name ;
Table_Name.arr[Table_Name.len] = '\0' ;
EXEC SQL WHENEVER NOT FOUND CONTINUE ;
EXEC SQL
SELECT      MAX(FD_NUMBER) INTO :Fd_Number
FROM        TEMP_FD ;
Fd_Number++ ;
count = 0 ;
EXEC SQL OPEN C81 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit81 ;
for (;;)
{
EXEC SQL FETCH C81 INTO :Attr_Number ;
EXEC SQL INSERT INTO TEMP_LHS (FD_NUMBER,LHS_NUMBER) VALUES (:Fd_Number,:Attr_Number) ;
EXEC SQL INSERT INTO TEMP_RHS (FD_NUMBER,RHS_NUMBER) VALUES (:Fd_Number,:Attr_Number) ;
count++ ;
}
quit81:
EXEC SQL CLOSE C81 ;
if (count > 0)
{
Temp_Number = Fd_Number - 1 ;
EXEC SQL INSERT INTO TEMP_FD (FD_NUMBER, TABLE_NAME) VALUES (:Fd_Number, :Table_Name) ;
EXEC SQL OPEN C82 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit82 ;
for (;;)
{
EXEC SQL FETCH C82 INTO :Attr_Number ;
EXEC SQL INSERT INTO TEMP_LHS (FD_NUMBER,LHS_NUMBER) VALUES (:Fd_Number,:Attr_Number) ;
EXEC SQL INSERT INTO TEMP_RHS (FD_NUMBER,RHS_NUMBER) VALUES (:Fd_Number,:Attr_Number) ;
}
quit82:
EXEC SQL CLOSE C82 ;
}
}
quit80:
EXEC SQL CLOSE C80 ;
EXEC SQL COMMIT WORK ;
return TRUE ;
}

void MapBCNF (void)
{
EXEC SQL DECLARE C70 CURSOR FOR
SELECT  DISTINCT TABLE_NAME
FROM    TEMP_FD ;

EXEC SQL DECLARE C71 CURSOR FOR
SELECT  FD_NUMBER
FROM    TEMP_FD
WHERE   TABLE_NAME = :Table_Name ;

EXEC SQL OPEN C70 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit70 ;
for (;;)
{
EXEC SQL FETCH C70 INTO :Table_Name ;
Table_Name.arr[Table_Name.len] = '\0' ;
EXEC SQL OPEN C71 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit71 ;
for (;;)
{
EXEC SQL FETCH C71 INTO :Fd_Number ;
FindBCNF 0 ;
}
quit71:
EXEC SQL CLOSE C71 ;
}
quit70:
EXEC SQL CLOSE C70 ;
CreateMetaTableBCNF 0 ;
}

void FindBCNF (void)
{
EXEC SQL DECLARE C69 CURSOR FOR
SELECT  FD_NUMBER
FROM    TEMP
WHERE   ATTR_NUMBER IN
( SELECT

```

เอกสารนี้เป็นเอกสารที่รวบรวมขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        FROM      TEMP_LHS
        WHERE     FD_NUMBER = :Fd_Number
        UNION
        SELECT    RHS_NUMBER
        FROM      TEMP_RHS
        WHERE     FD_NUMBER = :Fd_Number
    )
GROUP BY FD_NUMBER
HAVING   COUNT(FD_NUMBER) = :Temp_Number
MINUS
SELECT   FD_NUMBER
FROM     TEMP
WHERE    ATTR_NUMBER NOT IN
(
        SELECT    LHS_NUMBER
        FROM      TEMP_LHS
        WHERE     FD_NUMBER = :Fd_Number
        UNION
        SELECT    RHS_NUMBER
        FROM      TEMP_RHS
        WHERE     FD_NUMBER = :Fd_Number
    )
MINUS
SELECT   FD_NUMBER
FROM     TEMP_FD
WHERE    FD_NUMBER = :Fd_Number ;

EXEC SQL DECLARE C72 CURSOR FOR
SELECT   FD_NUMBER
FROM     TEMP
WHERE    ATTR_NUMBER IN
(
        SELECT    LHS_NUMBER
        FROM      TEMP_LHS
        WHERE     FD_NUMBER = :Fd_Number
        UNION
        SELECT    RHS_NUMBER
        FROM      TEMP_RHS
        WHERE     FD_NUMBER = :Fd_Number
    )
GROUP BY FD_NUMBER
HAVING   COUNT(FD_NUMBER) = :Temp_Number
INTERSECT
SELECT   FD_NUMBER
FROM     TEMP_RHS
WHERE    RHS_NUMBER IN
(
        SELECT    LHS_NUMBER
        FROM      TEMP_LHS
        WHERE     FD_NUMBER = :Fd_Number
    )
MINUS
SELECT   FD_NUMBER
FROM     TEMP_FD
WHERE    FD_NUMBER = :Fd_Number ;

EXEC SQL DECLARE C73 CURSOR FOR
SELECT   LHS_NUMBER
FROM     TEMP_LHS
WHERE    FD_NUMBER = :Fd_Number AND
LHS_NUMBER NOT IN
(
        SELECT    LHS_NUMBER
        FROM      TEMP_LHS
        WHERE     FD_NUMBER = :Role_Number
    )
);

EXEC SQL CREATE VIEW TEMP (FD_NUMBER,ATTR_NUMBER) AS
SELECT   FD_NUMBER,LHS_NUMBER
FROM     TEMP_LHS
UNION
SELECT   FD_NUMBER,RHS_NUMBER
FROM     TEMP_RHS ;
if (ProcessOracleErrorCode ("CREATE VIEW TEMP") == FALSE) ;
EXEC SQL WHENEVER NOT FOUND CONTINUE ;
EXEC SQL
SELECT   COUNT(*) INTO :Temp_Number
FROM     TEMP
WHERE    FD_NUMBER = :Fd_Number ;
EXEC SQL OPEN C69 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit69 ;
for ( ;;)
(
    EXEC SQL FETCH C69 INTO :Fd_Number ;
    EXEC SQL DELETE TEMP_FD WHERE FD_NUMBER = :Fd_Number ;
    EXEC SQL DELETE TEMP_LHS WHERE FD_NUMBER = :Fd_Number ;
    EXEC SQL DELETE TEMP_RHS WHERE FD_NUMBER = :Fd_Number ;
)
quit69:
EXEC SQL CLOSE C69 ;
EXEC SQL OPEN C72 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit72 ;
for ( ;;)
(
    EXEC SQL FETCH C72 INTO :Role_Number ;
    EXEC SQL
    DELETE .TEMP_LHS
    WHERE  LHS_NUMBER IN
    (
        SELECT  RHS_NUMBER
    )

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        FROM    TEMP_RHS
        WHERE   FD_NUMBER = :Fd_Number
    ) AND FD_NUMBER = :Role_Number;
EXEC SQL
DELETE    TEMP_RHS
WHERE    RHS_NUMBER IN
(
    SELECT    LHS_NUMBER
    FROM      TEMP_LHS
    WHERE     FD_NUMBER = :Fd_Number
    ) AND FD_NUMBER = :Role_Number;

EXEC SQL OPEN C73 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit73 ;
for (;;)
{
    EXEC SQL FETCH C73 INTO :LHS_Number ;
    EXEC SQL INSERT INTO TEMP_LHS (FD_NUMBER,LHS_NUMBER) VALUES (:Role_Number,:LHS_Number) ;
}
quit73:
EXEC SQL CLOSE C73 ;
}
quit72:
EXEC SQL CLOSE C72 ;
EXEC SQL DROP VIEW TEMP ;
EXEC SQL COMMIT WORK ;
}

void CreateMetaTableBCNF ()
{
    int i,count;
    char Buff[10];

    EXEC SQL DECLARE C74 CURSOR FOR
    SELECT    DISTINCT TABLE_NAME
    FROM      TEMP_FD
    GROUP BY TABLE_NAME
    HAVING    COUNT(*) = :Temp_Number ;

    EXEC SQL DECLARE C75 CURSOR FOR
    SELECT    TABLE_NAME
    FROM      TEMP_FD
    WHERE     TABLE_NAME = :Table_Name
    FOR UPDATE OF TABLE_NAME ;

    EXEC SQL DECLARE C76 CURSOR FOR
    SELECT    FD_NUMBER, TABLE_NAME
    FROM      TEMP_FD
    ORDER BY TABLE_NAME ;

    EXEC SQL DECLARE C77 CURSOR FOR
    SELECT    LHS_NUMBER
    FROM      TEMP_LHS
    WHERE     FD_NUMBER = :Fd_Number ;

    EXEC SQL DECLARE C78 CURSOR FOR
    SELECT    RHS_NUMBER
    FROM      TEMP_RHS
    WHERE     FD_NUMBER = :Fd_Number ;

    EXEC SQL DECLARE C79 CURSOR FOR
    SELECT    TEMP_LHS.FD_NUMBER, TEMP_LHS.LHS_NUMBER
    FROM      TEMP_RHS, TEMP_LHS
    WHERE     TEMP_RHS.FD_NUMBER = TEMP_LHS.FD_NUMBER AND TEMP_RHS.RHS_NUMBER = TEMP_LHS.LHS_NUMBER ;

    EXEC SQL WHENEVER NOT FOUND CONTINUE;
    EXEC SQL DELETE ATTR_OF_INDEX ;
    if (ProcessOracleErrorCode (" DELETE * FROM ATTR_OF_INDEX ") == FALSE) ;
    EXEC SQL DELETE TABLE_OF_INDEX ;
    if (ProcessOracleErrorCode (" DELETE * FROM TABLE_OF_INDEX ") == FALSE) ;
    EXEC SQL DELETE ATTR_OF_TABLE ;
    if (ProcessOracleErrorCode (" DELETE * FROM ATTR_OF_TABLE ") == FALSE) ;
    EXEC SQL OPEN C79 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit79 ;
    for (;;)
    {
        EXEC SQL FETCH C79 INTO :Fd_Number,:LHS_Number ;
        EXEC SQL
        DELETE    TEMP_RHS
        WHERE     FD_NUMBER = :Fd_Number AND RHS_NUMBER = :LHS_Number ;
        if (ProcessOracleErrorCode (" DELETE TEMP_RHS ") == FALSE) ;
    }
quit79:
EXEC SQL CLOSE C79 ;

EXEC SQL WHENEVER NOT FOUND CONTINUE ;
EXEC SQL
SELECT    MAX(COUNT(*)) INTO :OraCount
FROM      TEMP_FD
GROUP BY TABLE_NAME ;
for (i=2;i<=OraCount;i++)
{
    Temp_Number = i ;
    EXEC SQL OPEN C74 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit74 ;
    for (;;)
    {
        EXEC SQL FETCH C74 INTO :Table_Name ;
        Table_Name.arr[Table_Name.len] = '\0' ;
        EXEC SQL OPEN C75 ;

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของลิขสิทธิ์

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EXEC SQL WHENEVER NOT FOUND GOTO quit75 ;
count = 1 ;
for (;;)
{
    EXEC SQL FETCH C75 INTO :Table_Name ;
    Table_Name.arr[Table_Name.len] = '\0' ;
    Istrcat (Table_Name.arr,"#") ;
    itoa (count, Buff, 10) ;
    Istrcat (Table_Name.arr, Buff) ;
    Table_Name.len = strlen (Table_Name.arr) ;
    EXEC SQL
    UPDATE TEMP_FD SET TABLE_NAME = :Table_Name WHERE CURRENT OF C75 ;
    if (ProcessOracleErrorCode ("UPDATE FD") == FALSE) ;
    count++ ;
}
quit75:
EXEC SQL CLOSE C75 ;
}
quit74:
EXEC SQL CLOSE C74 ;
}
Index_Number = 1 ;
EXEC SQL OPEN C76 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit76 ;
for (;;)
{
    EXEC SQL FETCH C76 INTO :Fd_Number, :Table_Name ;
    Table_Name.arr[Table_Name.len] = '\0' ;
    EXEC SQL INSERT INTO TABLE_OF_INDEX (INDEX_NUMBER, TABLE_NAME) VALUES (:Index_Number, :Table_Name) ;
    if (ProcessOracleErrorCode ("INSERT INTO TABLE_OF_INDEX") == FALSE) ;
    EXEC SQL OPEN C77 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit77 ;
    for (;;)
    {
        EXEC SQL FETCH C77 INTO :LHS_Number ;
        EXEC SQL INSERT INTO ATTR_OF_INDEX (INDEX_NUMBER, ATTR_NUMBER) VALUES (Index_Number, :LHS_Number) ;
        if (ProcessOracleErrorCode ("INSERT INTO ATTR_OF_INDEX") == FALSE) ;
        Istrcpy (Not_Null.arr, "Y") ;
        Not_Null.len = strlen (Not_Null.arr) ;
        EXEC SQL INSERT INTO ATTR_OF_TABLE (TABLE_NAME, ATTR_NUMBER, NOT_NULL)
        VALUES (:Table_Name, :LHS_Number, :Not_Null) ;
        if (ProcessOracleErrorCode ("INSERT INTO ATTR_OF_TABLE (LHS)") == FALSE) ;
    }
quit77:
EXEC SQL CLOSE C77 ;
EXEC SQL OPEN C78 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit78 ;
for (;;)
{
    EXEC SQL FETCH C78 INTO :RHS_Number ;
    Istrcpy (Not_Null.arr, "N") ;
    Not_Null.len = strlen (Not_Null.arr) ;
    EXEC SQL INSERT INTO ATTR_OF_TABLE (TABLE_NAME, ATTR_NUMBER, NOT_NULL)
    VALUES (:Table_Name, :RHS_Number, :Not_Null) ;
    if (ProcessOracleErrorCode ("INSERT INTO ATTR_OF_TABLE (RHS)") == FALSE) ;
}
quit78:
EXEC SQL CLOSE C78 ;
Index_Number++ ;
}
quit76:
EXEC SQL CLOSE C76 ;
CreateSqlTable ("Bcnf.sql") ;
}

```

Filename: PROCESS4.PC

```

#include "erextem.inc"
EXEC SQL BEGIN DECLARE SECTION ;
    VARCHAR Attr_Name[30] ;
    VARCHAR Temp_Name[30] ;
    VARCHAR Table_Name[30] ;
    int Attr_Number ;
    int Index_Number ;
    int Jd_Number ;
    int Temp_Number ;
    int OraCount ;
EXEC SQL END DECLARE SECTION ;
EXEC SQL INCLUDE SQLCA ;

void ProcessFifth ()
{
    FARPROC IpProcJd ;

    SetCapture (hDrawWnd) ;
    hOldCursor = SetCursor (hWaitCursor) ;
    EXEC SQL WHENEVER NOT FOUND CONTINUE ;
    EXEC SQL DELETE JD ;
    if (ProcessOracleErrorCode ("DELETE FROM JD") == FALSE) ;
    EXEC SQL DELETE ATTR_OF_JD ;
    if (ProcessOracleErrorCode ("DELETE FROM ATTR_OF_JD") == FALSE) ;
    SetCursor (hOldCursor) ;
    ReleaseCapture () ;
    EXEC SQL CREATE VIEW TEMP_JD AS
    SELECT TABLE_NAME, ATTR_NUMBER
    FROM ATTR_OF_TABLE
    WHERE TABLE_NAME IN
    (

```

เอกสารนี้เป็นเอกสารที่เลือกมาเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FROM TABLE_OF_INDEX,ATTR_OF_INDEX,ATTR_OF_TABLE
WHERE ATTR_OF_TABLE.TABLE_NAME = TABLE_OF_INDEX.TABLE_NAME AND
TABLE_OF_INDEX.INDEX_NUMBER = ATTR_OF_INDEX.INDEX_NUMBER
GROUP BY TABLE_OF_INDEX.TABLE_NAME
HAVING COUNT (DISTINCT ATTR_OF_TABLE.ATTR_NUMBER) =
COUNT (DISTINCT ATTR_OF_INDEX.ATTR_NUMBER) AND
COUNT (DISTINCT ATTR_OF_INDEX.ATTR_NUMBER) > 2
);
EXEC SQL SELECT COUNT(*) INTO :OraCount FROM TEMP_JD ;
If (OraCount != 0)
{
    IpProcJd = MakeProcInstance (JdProc, hInst) ;
    DialogBox (hInst, "JdBox", hMainWnd, IpProcJd) ;
    FreeProcInstance (IpProcJd) ;
}
else
{
    MessageBox (GetFocus(), "All table in Boyce/Codd NF. are also in Fifth NF. ", "Complete ", MB_OK) ;
    CreateSqlTable ("Fifth.sql") ;
}
EXEC SQL DROP VIEW TEMP_JD ;
}

BOOL FAR PASCAL __export JdProc (HWND hDlg, unsigned message, WORD wParam, LONG lParam)
{
    EXEC SQL DECLARE C80 CURSOR FOR
    SELECT DISTINCT TABLE_NAME
    FROM TEMP_JD ;

    EXEC SQL DECLARE C81 CURSOR FOR
    SELECT JD_NUMBER
    FROM JD
    WHERE TABLE_NAME = :Table_Name ;

    EXEC SQL DECLARE C85 CURSOR FOR
    SELECT JD_NUMBER
    FROM ATTR_OF_JD
    WHERE ATTR_NUMBER IN
    (
        SELECT ATTR_NUMBER
        FROM ATTR_OF_JD
        WHERE JD_NUMBER = :Jd_Number
    )
    GROUP BY JD_NUMBER
    HAVING COUNT (*) = :Temp_Number
    MINUS
    SELECT JD_NUMBER
    FROM ATTR_OF_JD
    WHERE JD_NUMBER = :Jd_Number ;

    switch (message)
    {
        case WM_INITDIALOG:
        {
            CursorOpen = FALSE ;
            EXEC SQL OPEN C80 ;
            EXEC SQL WHENEVER NOT FOUND GOTO quit80 ;
            for (;;)
            {
                EXEC SQL FETCH C80 INTO :Table_Name ;
                Table_Name.arr[Table_Name.len] = '\0' ;
                SendDlgItemMessage (hDlg, IDC_TNAME, CB_ADDSTRING, 0, (LPARAM) ((LPSTR) Table_Name.arr)) ;
            }
            quit80:
            EXEC SQL CLOSE C80 ;
            SendDlgItemMessage (hDlg, IDC_TNAME, CB_SELECTSTRING, 0, (LPARAM) ((LPSTR) Table_Name.arr)) ;
            Table_Name.len = strlen (Table_Name.arr) ;
            ClearJdListBox (hDlg) ;
            return (TRUE) ;
        }
        case WM_COMMAND:
        {
            switch (wParam)
            {
                case IDC_EXIT:
                {
                    if (CursorOpen == TRUE)
                    {
                        CursorOpen = FALSE ;
                        EXEC SQL CLOSE C81 ;
                    }
                    EndDialog (hDlg, TRUE) ;
                    EnableMenuItem (GetMenu (hMainWnd), IDM_FIFTH, MF_ENABLED) ;
                    return (TRUE) ;
                }
            }
            break ;
            case IDC_RUN:
            {
                if (CursorOpen == TRUE)
                {
                    CursorOpen = FALSE ;
                    EXEC SQL CLOSE C81 ;
                }
                SetCapture (hDrawWnd) ;
                hOldCursor = SetCursor (hWaitCursor) ;
                if (GenFifth () == NOERROR)
                {
                    SetCursor (hOldCursor) ;
                    ReleaseCapture () ;
                    EndDialog (hDlg, TRUE) ;
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        EnableMenuItem (GetMenu(hMainWnd),IDM_FIFTH, MF_DISABLED|MF_GRAYED);
        EnableMenuItem (GetMenu(hMainWnd),IDM_BCNF, MF_DISABLED|MF_GRAYED);
        return (TRUE);
    }
    else
    {
        SetCursor (hOldCursor);
        ReleaseCapture ();
        Istrcpy (Table_Name.ar,Temp_Name.ar);
        Table_Name.len = Istrlen (Table_Name.ar);
        ClearJdListBox (hDlg);
    }
}
break;
case IDC_SELECT:
{
    CursorOpen = TRUE;
    EXEC SQL OPEN C81;
    EnableWindow (GetDlgItem (hDlg,IDC_FETCH),TRUE);
    EnableWindow (GetDlgItem (hDlg,IDC_SELECT),FALSE);
}
break;
case IDC_DELETE:
{
    EXEC SQL DELETE FROM JD WHERE JD_NUMBER = :Jd_Number;
    EXEC SQL DELETE FROM ATTR_OF_JD WHERE JD_NUMBER = :Jd_Number;
}
case IDC_FETCH:
{
    EXEC SQL WHENEVER NOT FOUND CONTINUE;
    EXEC SQL FETCH C81 INTO :Jd_Number;
    If (sqlca.sqlcode == 1403)
    {
        CursorOpen = FALSE;
        EXEC SQL CLOSE C81;
        EnableWindow (GetDlgItem (hDlg,IDC_FETCH),FALSE);
        EnableWindow (GetDlgItem (hDlg,IDC_DELETE),FALSE);
        EnableWindow (GetDlgItem (hDlg,IDC_SELECT),TRUE);
        MessageBox (GetFocus (), "No More Data", "Fetch Error", MB_OK);
        ClearJdListBox (hDlg);
    }
    else
    {
        EnableWindow (GetDlgItem (hDlg,IDC_DELETE),TRUE);
        SelectIndexJdListBox (hDlg);
    }
}
break;
case IDC_INSERT:
{
    if (CursorOpen == TRUE)
    {
        CursorOpen = FALSE;
        EXEC SQL CLOSE C81;
    }
    ClearJdListBox (hDlg);
    EnableWindow (GetDlgItem (hDlg,IDC_CONFIRM),TRUE);
    EnableWindow (GetDlgItem (hDlg,IDC_CANCEL),TRUE);
    EnableWindow (GetDlgItem (hDlg,IDC_SELECT),FALSE);
    EnableWindow (GetDlgItem (hDlg,IDC_FETCH),FALSE);
    EnableWindow (GetDlgItem (hDlg,IDC_DELETE),FALSE);
    EnableWindow (GetDlgItem (hDlg,IDC_INSERT),FALSE);
    EnableWindow (GetDlgItem (hDlg,IDC_RUN),FALSE);
    EnableWindow (GetDlgItem (hDlg,IDC_EXIT),FALSE);
}
break;
case IDC_CANCEL:
{
    EnableWindow (GetDlgItem (hDlg,IDC_CONFIRM),FALSE);
    EnableWindow (GetDlgItem (hDlg,IDC_CANCEL),FALSE);
    EnableWindow (GetDlgItem (hDlg,IDC_SELECT),TRUE);
    EnableWindow (GetDlgItem (hDlg,IDC_FETCH),FALSE);
    EnableWindow (GetDlgItem (hDlg,IDC_DELETE),FALSE);
    EnableWindow (GetDlgItem (hDlg,IDC_INSERT),TRUE);
    EnableWindow (GetDlgItem (hDlg,IDC_RUN),TRUE);
    EnableWindow (GetDlgItem (hDlg,IDC_EXIT),TRUE);
    ClearJdListBox (hDlg);
}
break;
case IDC_CONFIRM:
{
    HLOCAL hMem;
    int *lpItems;
    DWORD index,dwIndex,cItems;

    EnableWindow (GetDlgItem (hDlg,IDC_CONFIRM),FALSE);
    EnableWindow (GetDlgItem (hDlg,IDC_CANCEL),FALSE);
    EnableWindow (GetDlgItem (hDlg,IDC_SELECT),TRUE);
    EnableWindow (GetDlgItem (hDlg,IDC_FETCH),FALSE);
    EnableWindow (GetDlgItem (hDlg,IDC_DELETE),FALSE);
    EnableWindow (GetDlgItem (hDlg,IDC_INSERT),TRUE);
    EnableWindow (GetDlgItem (hDlg,IDC_RUN),TRUE);
    EnableWindow (GetDlgItem (hDlg,IDC_EXIT),TRUE);
    cItems = SendDlgItemMessage (hDlg, IDC_ATTR, LB_GETSELCOUNT, 0, 0);
    if (cItems < 2)
    {
        ErrorMessage ("Not enough attribute to join");
        break;
    }
}
EXEC SQL SELECT MAX(JD_NUMBER) INTO :Jd_Number FROM JD;

```

เอกสารนี้เป็นเอกสารที่ส EXEC SQL SELECT MAX(JD\_NUMBER) INTO :Jd\_Number FROM JD; อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

quit82:
    EXEC SQL CLOSE C82;
}

void SelectIndexJdListBox (HWND hDlg)
{
    EXEC SQL DECLARE C83 CURSOR FOR
    SELECT ATTR_NAME
    FROM ATTR_DETAIL
    WHERE ATTR_NUMBER IN
        (
            SELECT ATTR_NUMBER
            FROM ATTR_OF_JD
            WHERE JD_NUMBER = :Jd_Number
        );

    ClearJdListBox (hDlg);
    EXEC SQL OPEN C83;
    EXEC SQL WHENEVER NOT FOUND GOTO quit83;
    for (;;)
    {
        DWORD dwIndex;

        EXEC SQL FETCH C83 INTO :Attr_Name;
        Attr_Name.arr[Attr_Name.len] = '\0';
        dwIndex = SendDlgItemMessage (hDlg, IDC_ATTR, LB_FINDSTRING, 0, (LPARAM) ((LPSTR) Attr_Name.arr));
        SendDlgItemMessage (hDlg, IDC_ATTR, LB_SETSEL, TRUE, (LPARAM) MAKELPARAM(dwIndex, 0));
    }
quit83:
    EXEC SQL CLOSE C83;
}

BOOL GenFifth ()
{
    ErrorFlag = NOERROR;
    if (!IsValidGroupJd () == NOERROR)
        MapFifth ();
    else
        ErrorFlag = ERROR;
    return (ErrorFlag);
}

BOOL IsValidGroupJd ()
{
    char Buff[100];

    EXEC SQL DECLARE C84 CURSOR FOR
    SELECT DISTINCT TABLE_NAME
    FROM JD;

    EXEC SQL DECLARE C89 CURSOR FOR
    SELECT JD_NUMBER
    FROM ATTR_OF_JD
    WHERE JD_NUMBER IN
        (
            SELECT JD_NUMBER
            FROM JD
            WHERE TABLE_NAME = :Table_Name
        );

    lstrcpy (Temp_Name.arr, Table_Name.arr);
    Temp_Name.len = lstrlen (Temp_Name.arr);
    ErrorFlag = NOERROR;
    EXEC SQL OPEN C84;
    EXEC SQL WHENEVER NOT FOUND GOTO quit84;
    for (;;)
    {
        EXEC SQL FETCH C84 INTO :Table_Name;
        Table_Name.arr[Table_Name.len] = '\0';

        EXEC SQL
        SELECT COUNT(DISTINCT ATTR_NUMBER) INTO :OraCount
        FROM ATTR_OF_JD
        WHERE JD_NUMBER IN
            (
                SELECT JD_NUMBER
                FROM JD
                WHERE TABLE_NAME = :Table_Name
            );

        EXEC SQL
        SELECT COUNT(DISTINCT ATTR_NUMBER) INTO :Temp_Number
        FROM TEMP_JD
        WHERE TABLE_NAME = :Table_Name;
        if (OraCount != Temp_Number)
        {
            ErrorFlag = ERROR;
            lstrcpy (Buff, "Some attribute in Table ");
            lstrcat (Buff, Table_Name.arr);
            lstrcat (Buff, " no selected");
            ErrorMessage (Buff);
        }
        EXEC SQL OPEN C89;
        EXEC SQL WHENEVER NOT FOUND GOTO quit89;
        for (;;)
        {
            EXEC SQL FETCH C89 INTO :Jd_Number;
            EXEC SQL
            SELECT COUNT(DISTINCT ATTR_NUMBER) INTO :OraCount

```

เอกสารนี้เป็นเอกสารของบริษัทฯ ขอสงวนสิทธิ์ในเนื้อหา ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FROM ATTR_OF_JD
WHERE ATTR_NUMBER IN
(
    SELECT ATTR_NUMBER
    FROM ATTR_OF_JD
    WHERE JD_NUMBER = :Jd_Number
) AND
JD_NUMBER IN
(
    SELECT JD_NUMBER
    FROM JD
    WHERE TABLE_NAME = :Table_Name AND JD_NUMBER != :Jd_Number
);
if (OraCount == 0)
{
    ErrorFlag = ERROR;
    Istrcpy (Buff,"Consequence of join not cover all attribute in ");
    Istrcat (Buff,Table_Name.arr);
    ErrorMessage (Buff);
}
quit89:
EXEC SQL CLOSE C89;
}
quit84:
EXEC SQL CLOSE C84;
return (ErrorFlag);
}

void MapFifth ()
{
    int count;
    char Buff[10];

    EXEC SQL DECLARE C86 CURSOR FOR
    SELECT DISTINCT TABLE_NAME
    FROM JD;

    EXEC SQL DECLARE C87 CURSOR FOR
    SELECT JD_NUMBER
    FROM JD
    WHERE TABLE_NAME = :Table_Name;

    EXEC SQL DECLARE C88 CURSOR FOR
    SELECT ATTR_NUMBER
    FROM ATTR_OF_JD
    WHERE JD_NUMBER = :Jd_Number;

    EXEC SQL OPEN C86;
    EXEC SQL WHENEVER NOT FOUND GOTO quit86;
    for (;;)
    {
        EXEC SQL FETCH C86 INTO :Table_Name;
        Table_Name.arr[Table_Name.len] = '\0';
        EXEC SQL OPEN C87;
        EXEC SQL WHENEVER NOT FOUND GOTO quit87;
        count = 1;
        for (;;)
        {
            EXEC SQL FETCH C87 INTO :Jd_Number;
            EXEC SQL SELECT MAX(INDEX_NUMBER) INTO :Index_Number FROM TABLE_OF_INDEX;
            Index_Number++;
            Istrcpy (Temp_Name.arr,Table_Name.arr);
            Istrcat (Temp_Name.arr,"*");
            Itoa (count,Buff,10);
            Istrcat (Temp_Name.arr,Buff);
            Temp_Name.len = Istrlen (Temp_Name.arr);
            EXEC SQL INSERT INTO TABLE_OF_INDEX (INDEX_NUMBER,TABLE_NAME) VALUES (:Index_Number,:Temp_Name);
            count++;
            EXEC SQL OPEN C88;
            EXEC SQL WHENEVER NOT FOUND GOTO quit88;
            for (;;)
            {
                EXEC SQL FETCH C88 INTO :Attr_Number;
                EXEC SQL INSERT INTO ATTR_OF_INDEX (INDEX_NUMBER,ATTR_NUMBER)
                VALUES (:Index_Number,:Attr_Number);
                EXEC SQL INSERT INTO ATTR_OF_TABLE (TABLE_NAME,ATTR_NUMBER,NOT_NULL)
                VALUES (:Temp_Name,:Attr_Number,'Y');
            }
            quit88:
            EXEC SQL CLOSE C88;
        }
        quit87:
        EXEC SQL CLOSE C87;
        EXEC SQL WHENEVER NOT FOUND CONTINUE;
        EXEC SQL DELETE FROM TABLE_OF_INDEX WHERE TABLE_NAME = :Table_Name;
        EXEC SQL DELETE FROM ATTR_OF_TABLE WHERE TABLE_NAME = :Table_Name;
        EXEC SQL DELETE FROM ATTR_OF_INDEX
        WHERE INDEX_NUMBER IN
        (
            SELECT INDEX_NUMBER
            FROM TABLE_OF_INDEX
            WHERE TABLE_NAME = :Table_Name
        );
    }
    quit86:
    EXEC SQL CLOSE C86;
    CreateSqlTable ("Fifth.sql");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "erxitem.inc"
EXEC SQL BEGIN DECLARE SECTION ;
  VARCHAR UserID[20];
  VARCHAR Password[20];
  VARCHAR OraImmedate[132];
  VARCHAR Entity_Name[30];
  VARCHAR Relation_Name[30];
  VARCHAR Attr_Name[30];
  VARCHAR DataType[20];
  VARCHAR Table_Name[30];
  VARCHAR Not_Null[2];
  VARCHAR Domain_Name[30];
  VARCHAR Invert[2];
  VARCHAR Lower_Limit[2];
  VARCHAR Min[30];
  VARCHAR Max[30];
  VARCHAR Upper_Limit[2];
  int Type;
  int Attr_Number;
  int Composit_Number;
  int Role_Number;
  int Index_Number;
  int Fd_Number;
  int Temp_Number;
  int Partial;
  int Cardinal;
  int OraCount;
EXEC SQL END DECLARE SECTION ;
EXEC SQL INCLUDE SQLCA ;
```

```
void ErrorMessage (LPSTR message)
```

```
{
  MessageBox (GetFocus(),message,"Error !",MB_OK);
}
```

```
void IntegerMessage (int x)
```

```
{
  char Buff[10];
  itoa (x,Buff,10);
  MessageBox (GetFocus(),Buff,"Error !",MB_OK);
}
```

```
BOOL IsBlankString (BYTE FAR* String)
```

```
{
  int i;
  for (i=0;String[i] != '\0';i++)
  {
    if (String[i] != ' ')
      return (FALSE);
  }
  return (TRUE);
}
```

```
void ClearBuffer (LPSTR Buffer,int Length)
```

```
{
  int i;
  for (i=0;i<Length;i++)
    Buffer[i] = '\0';
}
```

```
void SetCursorCenterPos (HWND hWnd)
```

```
{
  RECT Rect;
  POINT ptCursor;

  GetClientRect (hWnd,&Rect);
  GetCursorPos (&ptCursor);
  ScreenToClient (hWnd,&ptCursor);
  ptCursor.x = (int) (Rect.right-Rect.left)/2;
  ptCursor.y = (int) (Rect.bottom-Rect.top)/2;
  ClientToScreen (hWnd,&ptCursor);
  SetCursorPos (ptCursor.x,ptCursor.y);
}
```

```
void CreateErMetaFile (BYTE FAR* FileName)
```

```
{
  ErMetaFile.hMetaDC = CreateMetaFile (NULL);
}
```

```
void PlayErMetaFile (HDC hDC)
```

```
{
  HDC hTempMetaDC;
  HMF hMF;

  hMF = CloseMetaFile (ErMetaFile.hMetaDC);
  hTempMetaDC = CreateMetaFile (NULL);
  PlayMetaFile (hTempMetaDC,hMF);
  SetWindowOrg (hDC,HorzThumbPos,VertThumbPos);
  PlayMetaFile (hDC,hMF);
  DeleteMetaFile (hMF);
  ErMetaFile.hMetaDC = hTempMetaDC;
}
```

void SaveErMetafile (void) รหัสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    HDC      hTempMetaDC;
    HMETAFILE hMF;

    hMF = CloseMetaFile (ErMetaFile.hMetaDC);
    hTempMetaDC = CreateMetaFile (NULL);
    PlayMetaFile (hTempMetaDC,hMF);
    CopyMetaFile (hMF,szFileName);
    DeleteMetaFile (hMF);
    ErMetaFile.hMetaDC = hTempMetaDC;
}

void DeleteErMetaFile (void)
{
    HMETAFILE hMF;

    hMF = CloseMetaFile (ErMetaFile.hMetaDC);
    DeleteMetaFile (hMF);
}

BOOL IsValidPosition (HRGN hRgn)
{
    int      Output;
    NODE     *Temp;
    HRGN     hTotalRgn;
    HGLOBAL  Next,hMem;

    if (FIRST != NULL)
    {
        hTotalRgn = CreateRectRgn (0,0,0,0);
        hMem = FIRST;
        do
        {
            Temp = (NODE*) GlobalLock (hMem);
            Next = Temp->Next;
            Output = CombineRgn (hTotalRgn,hRgn,Temp->hRgn,RGN_AND);
            GlobalUnlock (hMem);
            if (Output != NULLREGION)
            {
                DeleteObject (hTotalRgn);
                return (FALSE);
            }
            hMem = Next;
        } while (Next != NULL);
        DeleteObject (hTotalRgn);
    }
    return (TRUE);
}

int WhatObjectSelected (int x,int y)
{
    int      Object;
    NODE     *Temp;
    BOOL     fPtn;
    HGLOBAL  Next,hMem;

    if (FIRST != NULL)
    {
        hMem = FIRST;
        do
        {
            Temp = (NODE*) GlobalLock (hMem);
            Next = Temp->Next;
            fPtn = PtnInRegion (Temp->hRgn,x,y);
            Object = Temp->ObjectType;
            GlobalUnlock (hMem);
            if (!fPtn)
                hMem = Next;
            else
                return (Object);
        } while (Next != NULL);
    }
    return (NO_OBJECT);
}

```

```

HGLOBAL IsRoleSelected (int Number)
{
    NODE     *Temp;
    HGLOBAL  Next,hMem;

    if (FIRST != NULL)
    {
        hMem = FIRST;
        do
        {
            Temp = (NODE*) GlobalLock (hMem);
            Next = Temp->Next;
            if (Temp->Number == Number)
            {
                switch (Temp->ObjectType)
                {
                    case TOTAL_1:
                    case TOTAL_N:
                    case TOTAL_M:
                    case PARTIAL_1:
                    case PARTIAL_N:
                    case PARTIAL_M:
                {
                    return (hMem);
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่เว้นวงไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break ;
    }
    }
    GlobalUnlock (hMem) ;
    hMem = Next ;
} while (Next != NULL) ;
}
return (NULL) ;
}
}
HGLOBAL IsAttributesSelected (int Number)
{
    NODE *Temp ;
    HGLOBAL Next,hMem ;
    # (FIRST != NULL)
    {
        hMem = FIRST ;
        do
        {
            Temp = (NODE*) GlobalLock (hMem) ;
            Next = Temp->Next ;
            if (Temp->Number == Number)
            {
                switch (Temp->ObjectType)
                {
                    case N_ATTRIBUTE:
                    case K_ATTRIBUTE:
                    case P_ATTRIBUTE:
                    case MV_ATTRIBUTE:
                    {
                        return (hMem) ;
                    }
                }
                break ;
            }
        }
        GlobalUnlock (hMem) ;
        hMem = Next ;
    } while (Next != NULL) ;
}
return (NULL) ;
}
}

```

```

HGLOBAL IsObjectSelected (int x,int y)
{
    NODE *Temp ;
    BOOL fPtn ;
    HGLOBAL Next,hMem ;
    # (FIRST != NULL)
    {
        hMem = FIRST ;
        do
        {
            Temp = (NODE*) GlobalLock (hMem) ;
            Next = Temp->Next ;
            fPtn = PtnRegion (Temp->hRgn,x,y) ;
            GlobalUnlock (hMem) ;
            if (!fPtn)
                hMem = Next ;
            else
                return (hMem) ;
        } while (Next != NULL) ;
    }
    return (NULL) ;
}
}

```

```

void InvertRole (int Number)
{
    HDC hDC ;
    NODE *Temp ;
    HGLOBAL Next,hMem ;
    hDC = GetDC (hDrawWnd) ;
    # (FIRST != NULL)
    {
        hMem = FIRST ;
        do
        {
            Temp = (NODE*) GlobalLock (hMem) ;
            Next = Temp->Next ;
            if (Temp->Number == Number)
            {
                switch (Temp->ObjectType)
                {
                    case TOTAL_1:
                    case TOTAL_N:
                    case TOTAL_M:
                    case PARTIAL_1:
                    case PARTIAL_N:
                    case PARTIAL_M:
                    {
                        CombineRgn (hDisplayRgn,hDisplayRgn,Temp->hRgn,RGN_OR) ;
                    }
                }
                break ;
            }
        }
        GlobalUnlock (hMem) ;
        hMem = Next ;
    } while (Next != NULL) ;
}
}

```

เอกสารนี้เป็นเอกสารที่เขียนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    ReleaseDC (hDrawWnd,hDC);
}

void InvertAttribute (int Number)
{
    HDC      hDC;
    NODE     *Temp;
    HGLOBAL  Next,hMem;

    hDC = GetDC (hDrawWnd);
    # (FIRST != NULL)
    {
        hMem = FIRST;
        do
        {
            Temp = (NODE*) GlobalLock (hMem);
            Next = Temp->Next;
            # (Temp->Number == Number)
            {
                switch (Temp->ObjectType)
                {
                    case N_ATTRIBUTE:
                    case K_ATTRIBUTE:
                    case P_ATTRIBUTE:
                    case MV_ATTRIBUTE:
                    {
                        CombineRgn (hDisplayRgn,hDisplayRgn,Temp->hRgn,RGN_OR);
                    }
                    break;
                }
            }
        }
        GlobalUnlock (hMem);
        hMem = Next;
    } while (Next != NULL);
    ReleaseDC (hDrawWnd,hDC);
}

void AdjustLineCenter (POINT *From,POINT *To)
{
    RECT      Rect;
    NODE     *Temp;
    HGLOBAL  hMem;
    HRGN     hFromRgn,hToRgn;
    LINEDDAPROC lpLDDAProc;

    hMem = IsObjectSelected (From->x,From->y);
    Temp = (NODE*) GlobalLock (hMem);
    hFromRgn = Temp->hRgn;
    GetRgnBox (hFromRgn,&Rect);
    From->x = (Rect.right+Rect.left)/2;
    From->y = (Rect.bottom+Rect.top)/2;
    GlobalUnlock (hMem);
    hMem = IsObjectSelected (To->x,To->y);
    Temp = (NODE*) GlobalLock (hMem);
    hToRgn = Temp->hRgn;
    GetRgnBox (hToRgn,&Rect);
    To->x = (Rect.right+Rect.left)/2;
    To->y = (Rect.bottom+Rect.top)/2;
    GlobalUnlock (hMem);
    fPInRgn = TRUE;
    lpLDDAProc = (LINEDDAPROC) MakeProcInstance ((FARPROC) LineDDAProc ,hInst);
    LineDDA (From->x,From->y,To->x,To->y,lpLDDAProc,hFromRgn);
    FreeProcInstance ((FARPROC) lpLDDAProc);
    From->x = xPosition;
    From->y = yPosition;
    fPInRgn = TRUE;
    lpLDDAProc = (LINEDDAPROC) MakeProcInstance ((FARPROC) LineDDAProc ,hInst);
    LineDDA (To->x,To->y,From->x,From->y,lpLDDAProc,hToRgn);
    FreeProcInstance ((FARPROC) lpLDDAProc);
    To->x = xPosition;
    To->y = yPosition;
}

void CreateDoubleLine (POINT From,POINT To,POINT *BeginPoint1,POINT *EndPoint1,POINT *BeginPoint2,POINT *EndPoint2)
{
    #if (abs(From.x - To.x) >= abs(From.y-To.y))
    {
        BeginPoint1->x = From.x;
        BeginPoint1->y = From.y-2;
        EndPoint1->x = To.x;
        EndPoint1->y = To.y-2;
        BeginPoint2->x = From.x;
        BeginPoint2->y = From.y+2;
        EndPoint2->x = To.x;
        EndPoint2->y = To.y+2;
    }
    #if (abs(From.x - To.x) < abs(From.y-To.y))
    {
        BeginPoint1->x = From.x-2;
        BeginPoint1->y = From.y;
        EndPoint1->x = To.x-2;
        EndPoint1->y = To.y;
        BeginPoint2->x = From.x+2;
        BeginPoint2->y = From.y;
        EndPoint2->x = To.x+2;
        EndPoint2->y = To.y;
    }
}

```

เอกสาร AdjustLineNormal (BeginPoint1,EndPoint1); เป็นการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AdjustLineNormal (BeginPoint2,EndPoint2);
}

void AdjustLineNormal (POINT *From,POINT *To)
{
    NODE          *Temp;
    HGLOBAL       hMem;
    HRGN          hFromRgn,hToRgn;
    LINEDDAPROC   lpLDDAProc;

    if ((hMem = IsObjectSelected (From->x,From->y)) != NULL)
    {
        Temp = (NODE*) GlobalLock (hMem);
        hFromRgn = Temp->hRgn;
        GlobalUnlock (hMem);
        fPtnRgn = TRUE;
        lpLDDAProc = (LINEDDAPROC) MakeProcInstance ((FARPROC) LineDDAProc ,hInst);
        LineDDA (From->x,From->y,To->x,To->y,lpLDDAProc,hFromRgn);
        FreeProcInstance ((FARPROC) lpLDDAProc);
        From->x = xPosition;
        From->y = yPosition;
    }
    if ((hMem = IsObjectSelected (To->x,To->y)) != NULL)
    {
        Temp = (NODE*) GlobalLock (hMem);
        hToRgn = Temp->hRgn;
        GlobalUnlock (hMem);
        fPtnRgn = TRUE;
        lpLDDAProc = (LINEDDAPROC) MakeProcInstance ((FARPROC) LineDDAProc ,hInst);
        LineDDA (To->x,To->y,From->x,From->y,lpLDDAProc,hToRgn);
        FreeProcInstance ((FARPROC) lpLDDAProc);
        To->x = xPosition;
        To->y = yPosition;
    }
}

void CALLBACK LineDDAProc (int xPos,int yPos,LPARAM lpData)
{
    if (fPtnRgn)
    {
        if (!(fPtnRgn = PtnRegion ((HRGN) lpData,xPos,yPos)))
        {
            xPosition = xPos;
            yPosition = yPos;
        }
    }
}

BOOL DeleteLink (void)
{
    NODE          *Temp;
    HGLOBAL       Next;

    if (FIRST != NULL)
    {
        do
        {
            Temp = (NODE*) GlobalLock (FIRST);
            Next = Temp->Next;
            DeleteObject (Temp->hRgn);
            GlobalUnlock (FIRST);
            if (GlobalFree (FIRST) != NULL)
                return (FALSE);
            FIRST = Next;
        } while (FIRST != NULL);
    }
    return (TRUE);
}

BOOL AppendNode (HRGN hRgn,int Length,int Type,LPSTR Name,int Number,LPSTR Domain)
{
    HGLOBAL       hMem;
    NODE          *ptr,*Temp;

    if ((hMem = GlobalAlloc (GMEM_MOVEABLE,sizeof (NODE))) != NULL)
    {
        if ((ptr = (NODE*) GlobalLock (hMem)) != NULL)
        {
            if (LAST == NULL)
                FIRST = hMem;
            else
            {
                Temp = (NODE*) GlobalLock(LAST);
                Temp->Next = hMem;
                GlobalUnlock(LAST);
            }
            ptr->hRgn = hRgn;
            ptr->MetaStartRec = CurrentMetaRec;
            ptr->MetaRecLength = Length;
            ptr->ObjectType = Type;
            lstrcpy (ptr->ObjectName,Name);
            lstrcpy (ptr->Constraint,Domain);
            ptr->Number = Number;
            ptr->Prev = LAST;
            ptr->Next = NULL;
            LAST = hMem;
            GlobalUnlock(hMem);
            CurrentMetaRec += Length;
            return (TRUE);
        }
    }
    else
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาก และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        ErrorMessage ("Can't Global Lock Memory");
        return (FALSE);
    }
}
else
{
    ErrorMessage ("Can't Alloc Memory");
    return (FALSE);
}
}

BOOL DeleteNode (HGLOBAL hMem)
{
    NODE *Temp;
    HGLOBAL Next,Prev;

    UpdateLink (hMem);
    Temp = (NODE*) GlobalLock(hMem);
    Prev = Temp->Prev;
    Next = Temp->Next;
    DeleteObject (Temp->hRgn);
    GlobalUnlock(hMem);
    if (Prev != NULL)
    {
        Temp = (NODE*) GlobalLock(Prev);
        Temp->Next = Next;
        GlobalUnlock(Prev);
    }
    else
    {
        FIRST = Next;
        if (Next != NULL)
        {
            Temp = (NODE*) GlobalLock(Next);
            Temp->Prev = Prev;
            GlobalUnlock(Next);
        }
        else
        {
            LAST = Prev;
            GlobalFree(hMem);
            return (TRUE);
        }
    }
}

void UpdateLink (HGLOBAL hMem)
{
    NODE *Temp;
    HGLOBAL Next;
    int RecLength;

    Temp = (NODE*) GlobalLock (hMem);
    RecLength = Temp -> MetaRecLength;
    Next = Temp->Next;
    CurrentMetaRec -= RecLength;
    GlobalUnlock (hMem);
    while (Next != NULL)
    {
        hMem = Next;
        Temp = (NODE*) GlobalLock (hMem);
        Temp->MetaStartRec -= RecLength;
        Next = Temp->Next;
        GlobalUnlock (hMem);
    }
}

void SaveLink (void)
{
    int hFile,count;
    NODE *Temp;
    HGLOBAL Next,hMem;
    OFSTRUCT OfStruct;
    struct {
        int ObjectType;
        char Name[30];
        char Domain[30];
        int MetaStartRec;
        int MetaRecLength;
        int Number;
    } data;

    if (FIRST != NULL)
    {
        hFile = OpenFile (LinkListFileName,&OfStruct,OF_CREATE);
        if (hFile >=0)
        {
            hMem = FIRST;
            do
            {
                Temp = (NODE*) GlobalLock (hMem);
                Next = Temp->Next;
                data.ObjectType = Temp->ObjectType;
                strcpy (data.Name,Temp->ObjectName);
                strcpy (data.Domain,Temp->Constraint);
                data.MetaStartRec = Temp->MetaStartRec;
                data.MetaRecLength = Temp->MetaRecLength;
                data.Number = Temp->Number;
                count = write (hFile,&data,sizeof (data));
                GlobalUnlock (hMem);
                hMem = Next;
            } while (Next != NULL);
            close (hFile);
        }
    }
}

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
    ErrorMessage ("Can't Save file Node Link List");
}
}
}

BOOL LoadLink (void)
{
    int         hFile,count;
    NODE        *Temp,*ptr;
    HGLOBAL     hMem;
    OFSTRUCT    OfStruct;
    struct {
        int     ObjectType;
        char    Name[30];
        char    Domain[30];
        int     MetaStartRec;
        int     MetaRecLength;
        int     Number;
    } data;

    hFile = OpenFile (LinkListFileName,&OfStruct,OF_READ);
    if (hFile >=0)
    {
        FIRST = NULL;
        LAST = NULL;
        do
        {
            count = read (hFile,&data,sizeof (data));
            if ((hMem = GlobalAlloc (GMEM_MOVEABLE,(DWORD) sizeof (NODE))) != NULL)
            {
                if ((ptr = (NODE*) GlobalLock (hMem)) != NULL)
                {
                    if (LAST == NULL)
                        FIRST = hMem;
                    else
                    {
                        Temp = (NODE*) GlobalLock(LAST);
                        Temp->Next = hMem;
                        GlobalUnlock(LAST);
                    }
                    ptr->hRgn = NULL;
                    ptr->MetaStartRec = data.MetaStartRec;
                    ptr->MetaRecLength = data.MetaRecLength;
                    ptr->ObjectType = data.ObjectType;
                    ptr->Number = data.Number;
                    lstrcpy (ptr->ObjectName,data.Name);
                    lstrcpy (ptr->Constraint,data.Domain);
                    ptr->Prev = LAST;
                    ptr->Next = NULL;
                    LAST = hMem;
                    GlobalUnlock(hMem);
                }
                else
                {
                    ErrorMessage ("Can't Global Lock Memory");
                    return (FALSE);
                }
            }
            else
            {
                ErrorMessage ("Can't Alloc Memory");
                return (FALSE);
            }
        } while (!eof(hFile));
        close (hFile);
        CreateMetaFileRegion ();
    }
    else
        ErrorMessage ("Can't Load file Node Link List");
}

void CreateMetaFileRegion (void)
{
    NODE        *Temp;
    HGLOBAL     Next,hMem;
    MFENUMPROC  lpRgnMetaProc;
    HMETAFILE   hMF;

    if (FIRST != NULL)
    {
        hMem = FIRST;
        do
        {
            Temp = (NODE*) GlobalLock (hMem);
            Next = Temp->Next;
            CurrentMetaRec = 0;
            hMF = CloseMetaFile (ErMetaFile.hMetaDC);
            ErMetaFile.hMetaDC = CreateMetaFile (NULL);
            lpRgnMetaProc = (MFENUMPROC) MakeProcInstance ((FARPROC) RegionMetaFileProc,hInst);
            EnumMetaFile (ErMetaFile.hMetaDC,hMF,lpRgnMetaProc,(LPARAM) Temp);
            FreeProcInstance ((FARPROC) lpRgnMetaProc);
            GlobalUnlock (hMem);
            hMem = Next;
        } while (Next != NULL);
        CurrentMetaRec++;
    }
}

int FAR PASCAL RegionMetaFileProc

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
HDC hMetaDC, HANDLETABLE FAR* lpHTable, METARECORD FAR* lpMFR, int cObj, BYTE FAR* lpClientData
```

```
NODE *Temp;
```

```
CurrentMetaRec++;
```

```
Temp = (NODE FAR*) lpClientData;
```

```
switch (Temp->ObjectType)
```

```
{
```

```
case R_ENTITY:
```

```
case W_ENTITY:
```

```
{
```

```
if (CurrentMetaRec == Temp->MetaStartRec)
```

```
Temp->hRgn = CreateRectRgn (lpMFR->rdParm[3], lpMFR->rdParm[2], lpMFR->rdParm[1], lpMFR->rdParm[0]);
```

```
}
```

```
break;
```

```
case N_RELATION:
```

```
case L_RELATION:
```

```
{
```

```
if (CurrentMetaRec == Temp->MetaStartRec)
```

```
{
```

```
POINT Point[4];
```

```
Point[0].x = lpMFR->rdParm[1];
```

```
Point[0].y = lpMFR->rdParm[2];
```

```
Point[1].x = lpMFR->rdParm[3];
```

```
Point[1].y = lpMFR->rdParm[4];
```

```
Point[2].x = lpMFR->rdParm[5];
```

```
Point[2].y = lpMFR->rdParm[6];
```

```
Point[3].x = lpMFR->rdParm[7];
```

```
Point[3].y = lpMFR->rdParm[8];
```

```
Temp->hRgn = CreatePolygonRgn (Point, sizeof (Point)/sizeof (POINT), WINDING);
```

```
}
```

```
}
```

```
break;
```

```
case N_ATTRIBUTE:
```

```
case K_ATTRIBUTE:
```

```
case P_ATTRIBUTE:
```

```
case MV_ATTRIBUTE:
```

```
{
```

```
if (CurrentMetaRec == Temp->MetaStartRec+2)
```

```
Temp->hRgn = CreateEllipticRgn (lpMFR->rdParm[3], lpMFR->rdParm[2], lpMFR->rdParm[1], lpMFR->rdParm[0]);
```

```
}
```

```
break;
```

```
case PARTIAL_1:
```

```
case PARTIAL_N:
```

```
case PARTIAL_M:
```

```
{
```

```
if (CurrentMetaRec == Temp->MetaStartRec+2)
```

```
{
```

```
HDC hDC;
```

```
RECT Rect;
```

```
DWORD dwExtent;
```

```
char *Ptr;
```

```
hDC = GetDC (hDrawWnd);
```

```
dwExtent = GetTextExtent (hDC, Temp->ObjectName, strlen (Temp->ObjectName));
```

```
ReleaseDC (hDrawWnd, hDC);
```

```
Ptr = (char *) lpMFR;
```

```
Rect.left = *((int *) (Ptr+(lpMFR->rdParm[0]+11)));
```

```
Rect.right = *((int *) (Ptr+(lpMFR->rdParm[0]+11))) + (LOWORD (dwExtent));
```

```
Rect.top = *((int *) (Ptr+(lpMFR->rdParm[0]+9)));
```

```
Rect.bottom = *((int *) (Ptr+(lpMFR->rdParm[0]+9))) + (HIWORD (dwExtent));
```

```
Temp->hRgn = CreateRectRgn (Rect.left, Rect.top, Rect.right, Rect.bottom);
```

```
}
```

```
}
```

```
break;
```

```
case TOTAL_1:
```

```
case TOTAL_N:
```

```
case TOTAL_M:
```

```
{
```

```
if (CurrentMetaRec == Temp->MetaStartRec+4)
```

```
{
```

```
HDC hDC;
```

```
RECT Rect;
```

```
DWORD dwExtent;
```

```
char *Ptr;
```

```
hDC = GetDC (hDrawWnd);
```

```
dwExtent = GetTextExtent (hDC, Temp->ObjectName, strlen (Temp->ObjectName));
```

```
ReleaseDC (hDrawWnd, hDC);
```

```
Ptr = (char *) lpMFR;
```

```
Rect.left = *((int *) (Ptr+(lpMFR->rdParm[0]+11)));
```

```
Rect.right = *((int *) (Ptr+(lpMFR->rdParm[0]+11))) + (LOWORD (dwExtent));
```

```
Rect.top = *((int *) (Ptr+(lpMFR->rdParm[0]+9)));
```

```
Rect.bottom = *((int *) (Ptr+(lpMFR->rdParm[0]+9))) + (HIWORD (dwExtent));
```

```
Temp->hRgn = CreateRectRgn (Rect.left, Rect.top, Rect.right, Rect.bottom);
```

```
}
```

```
}
```

```
break;
```

```
case TEXT:
```

```
{
```

```
if (CurrentMetaRec == Temp->MetaStartRec)
```

```
{
```

```
HDC hDC;
```

```
RECT Rect;
```

```
DWORD dwExtent;
```

```
char *Ptr;
```

```
hDC = GetDC (hDrawWnd);
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dwExtent = GetTextExtent (hDC,Temp->ObjectName,1strlen (Temp->ObjectName)) ;
ReleaseDC (hDrawWnd,hDC) ;
Ptr = (char *) lpMFR ;
if (1strlen (Temp->ObjectName) > 1)
{
    Rect.left = *((int *) (Ptr+(lpMFR->rdParm[0])+10)) ;
    Rect.right = *((int *) (Ptr+(lpMFR->rdParm[0])+10)) + (LOWORD (dwExtent)) ;
    Rect.top = *((int *) (Ptr+(lpMFR->rdParm[0])+8)) ;
    Rect.bottom = *((int *) (Ptr+(lpMFR->rdParm[0])+8)) + (HIWORD (dwExtent)) ;
}
else
{
    Rect.left = *((int *) (Ptr+(lpMFR->rdParm[0])+11)) ;
    Rect.right = *((int *) (Ptr+(lpMFR->rdParm[0])+11)) + (LOWORD (dwExtent)) ;
    Rect.top = *((int *) (Ptr+(lpMFR->rdParm[0])+9)) ;
    Rect.bottom = *((int *) (Ptr+(lpMFR->rdParm[0])+9)) + (HIWORD (dwExtent)) ;
}
Temp->hRgn = CreateRectRgn (Rect.left,Rect.top,Rect.right,Rect.bottom) ;
}
break ;
}
PlayMetaFileRecord (hMetaDC,lpHTable,lpMFR,cObj) ;
return (TRUE) ;
}

```

```

void CreateFileName (void)
{
    int i ;
    for (i=0;szFileName[i] != '.' && i<1strlen(szFileName);i++)
    {
        LinkListFileName [i] = szFileName[i] ;
        OracleFileName[i] = szFileName[i] ;
    }
    szFileName[i] = '.' ;
    szFileName[i+1] = 'E' ;
    szFileName[i+2] = 'R' ;
    szFileName[i+3] = 'O' ;
    LinkListFileName[i] = '.' ;
    LinkListFileName[i+1] = 'L' ;
    LinkListFileName[i+2] = 'N' ;
    LinkListFileName[i+3] = 'K' ;
    LinkListFileName[i+4] = 'O' ;
    OracleFileName[i] = '.' ;
    OracleFileName[i+1] = 'O' ;
    OracleFileName[i+2] = 'R' ;
    OracleFileName[i+3] = 'A' ;
    OracleFileName[i+4] = 'O' ;
}

```

```

void ChangeFileTitle (void)
{
    int i ;
    for (i=0;szFileTitle[i] != '.' && i<1strlen(szFileTitle);i++)
    {
        szFileTitle[i] = '.' ;
        szFileTitle[i+1] = 'E' ;
        szFileTitle[i+2] = 'R' ;
        szFileTitle[i+3] = 'O' ;
    }
}

```

```

BOOL CreateProcessTable (void)
{
    EXEC SQL WHENEVER NOT FOUND CONTINUE ;
    EXEC SQL DELETE ATTR_OF_INDEX ;
    if (ProcessOracleErrorCode (" DELETE * FROM ATTR_OF_TABLE ") == FALSE) ;
    EXEC SQL DELETE TABLE_OF_INDEX ;
    if (ProcessOracleErrorCode (" DELETE * FROM TABLE_OF_INDEX ") == FALSE) ;
    EXEC SQL DELETE ATTR_OF_TABLE ;
    if (ProcessOracleErrorCode (" DELETE * FROM ATTR_OF_TABLE ") == FALSE) ;
    EXEC SQL DELETE ATTR_DETAIL ;
    if (ProcessOracleErrorCode (" DELETE * FROM ATTR_DETAIL ") == FALSE) ;
    EXEC SQL DELETE TEMP_ATTR_OF_INDEX ;
    if (ProcessOracleErrorCode (" DELETE * FROM TEMP_ATTR_OF_TABLE ") == FALSE) ;
    EXEC SQL DELETE TEMP_TABLE_OF_INDEX ;
    if (ProcessOracleErrorCode (" DELETE * FROM TEMP_TABLE_OF_INDEX ") == FALSE) ;
    EXEC SQL DELETE TEMP_ATTR_OF_TABLE ;
    if (ProcessOracleErrorCode (" DELETE * FROM TEMP_ATTR_OF_TABLE ") == FALSE) ;
    EXEC SQL INSERT INTO TEMP_ENTITY SELECT * FROM RELATION ;
    if (ProcessOracleErrorCode ("INSERT TEMP_ENTITY") == FALSE)
        return (FALSE) ;
    EXEC SQL INSERT INTO TEMP_RELATION SELECT * FROM RELATION ;
    if (ProcessOracleErrorCode ("INSERT TEMP_RELATION") == FALSE)
        return (FALSE) ;
    EXEC SQL INSERT INTO TEMP_ATTR SELECT * FROM ATTR ;
    if (ProcessOracleErrorCode ("INSERT TEMP_ATTR") == FALSE)
        return (FALSE) ;
    EXEC SQL INSERT INTO TEMP_ROLE SELECT * FROM ROLE ;
    if (ProcessOracleErrorCode ("INSERT TEMP_ROLE") == FALSE)
        return (FALSE) ;
    EXEC SQL COMMIT WORK ;
    if (ProcessOracleErrorCode ("INSERT PROCESS TABLE COMMIT") == FALSE)
        return (FALSE) ;
    return (TRUE) ;
}

```

```

BOOL DropProcessTable (void)
{

```

เอ็กซ์... EXEC SQL DELETE TEMP\_ENTITY; สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (ProcessOracleErrorCode ("DELETE TEMP_ENTITY") == FALSE)
    return (FALSE);
EXEC SQL DELETE TEMP_RELATION;
if (ProcessOracleErrorCode ("DELETE TEMP_RELATION") == FALSE)
    return (FALSE);
EXEC SQL DELETE TEMP_ATTR;
if (ProcessOracleErrorCode ("DELETE TEMP_ATTR") == FALSE)
    return (FALSE);
EXEC SQL DELETE TEMP_ROLE;
if (ProcessOracleErrorCode ("DELETE TEMP_ROLE") == FALSE)
    return (FALSE);
EXEC SQL COMMIT WORK;
if (ProcessOracleErrorCode ("DELETE PROCESS TABLE COMMIT") == FALSE)
    return (FALSE);
return (TRUE);
}

void DeleteOracleAttribute (int Number)
{
    Attr_Number = Number;
    EXEC SQL WHENEVER NOT FOUND CONTINUE;
    EXEC SQL DELETE FROM ATTR
    WHERE ATTR_NUMBER = :Attr_Number;
    if (ProcessOracleErrorCode ("Delete Attribute Tuple") == FALSE);
    EXEC SQL DELETE FROM INVERT
    WHERE ATTR_NUMBER = :Attr_Number;
    if (ProcessOracleErrorCode ("Delete Invert Tuple") == FALSE);
    EXEC SQL DELETE FROM N_CONSTRAINT
    WHERE ATTR_NUMBER = :Attr_Number;
    if (ProcessOracleErrorCode ("Delete N_Constraint Tuple") == FALSE);
    EXEC SQL DELETE FROM C_CONSTRAINT
    WHERE ATTR_NUMBER = :Attr_Number;
    if (ProcessOracleErrorCode ("Delete C_Constraint Tuple") == FALSE);
}

void DisconnectOracle ()
{
    EXEC SQL COMMIT WORK RELEASE;
    if (ProcessOracleErrorCode ("Disconnect Oracle") == FALSE);
}

void ConnectOracle ()
{
    LoginOracle ();
}

BOOL LoginOracle (void)
{
    Istrcpy (UserD.arr,"ERRE");
    UserD.len = Istrlen (UserD.arr);
    Istrcpy (Password.arr,"KMRE");
    Password.len = Istrlen (Password.arr);
    EXEC SQL CONNECT :UserD IDENTIFIED BY :Password;
    if (ProcessOracleErrorCode ("Connect") == FALSE)
    {
        return (FALSE);
    }
    return (TRUE);
}

BOOL LogoutOracle (void)
{
    EXEC SQL WHENEVER NOT FOUND CONTINUE;
    EXEC SQL DELETE ENTITY;
    if (ProcessOracleErrorCode ("DELETE ENTITY") == FALSE);
    EXEC SQL DELETE RELATION;
    if (ProcessOracleErrorCode ("DELETE RELATION") == FALSE);
    EXEC SQL DELETE ROLE;
    if (ProcessOracleErrorCode ("DELETE Role") == FALSE);
    EXEC SQL DELETE ATTR;
    if (ProcessOracleErrorCode ("DELETE ATTR") == FALSE);
    EXEC SQL DELETE ATTR_OF_INDEX;
    if (ProcessOracleErrorCode ("DELETE ATTR_OF_INDEX") == FALSE);
    EXEC SQL DELETE TABLE_OF_INDEX;
    if (ProcessOracleErrorCode ("DELETE TABLE_OF_INDEX") == FALSE);
    EXEC SQL DELETE ATTR_OF_TABLE;
    if (ProcessOracleErrorCode ("DELETE ATTR_OF_TABLE") == FALSE);
    EXEC SQL DELETE ATTR_DETAIL;
    if (ProcessOracleErrorCode ("DELETE ATTR_DETAIL") == FALSE);
    EXEC SQL DELETE TEMP_ATTR_OF_INDEX;
    if (ProcessOracleErrorCode ("DELETE TEMP_ATTR_OF_INDEX") == FALSE);
    EXEC SQL DELETE TEMP_TABLE_OF_INDEX;
    if (ProcessOracleErrorCode ("DELETE TEMP_TABLE_OF_INDEX") == FALSE);
    EXEC SQL DELETE TEMP_ATTR_OF_TABLE;
    if (ProcessOracleErrorCode ("DELETE TEMP_ATTR_OF_TABLE") == FALSE);
    EXEC SQL DELETE FD;
    if (ProcessOracleErrorCode ("DELETE FD") == FALSE);
    EXEC SQL DELETE LHS;
    if (ProcessOracleErrorCode ("DELETE LHS") == FALSE);
    EXEC SQL DELETE RHS;
    if (ProcessOracleErrorCode ("DELETE RHS") == FALSE);
    EXEC SQL DELETE TEMP_FD;
    if (ProcessOracleErrorCode ("DELETE TEMP_FD") == FALSE);
    EXEC SQL DELETE TEMP_LHS;
    if (ProcessOracleErrorCode ("DELETE TEMP_LHS") == FALSE);
    EXEC SQL DELETE TEMP_RHS;
    if (ProcessOracleErrorCode ("DELETE TEMP_RHS") == FALSE);
    EXEC SQL DELETE JD;
    if (ProcessOracleErrorCode ("DELETE JD") == FALSE);
    EXEC SQL DELETE ATTR_OF_JD;
}

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (ProcessOracleErrorCode ("DELETE ATTR_OF_JD") == FALSE) ;
EXEC SQL DELETE N_CONSTRAINT ;
if (ProcessOracleErrorCode ("DELETE N_CONSTRAINT") == FALSE) ;
EXEC SQL DELETE C_CONSTRAINT ;
if (ProcessOracleErrorCode ("DELETE C_CONSTRAINT") == FALSE) ;
EXEC SQL DELETE INVERT ;
if (ProcessOracleErrorCode ("DELETE INVERT") == FALSE) ;
EXEC SQL DELETE KILL ;
if (ProcessOracleErrorCode ("DELETE KILL") == FALSE) ;
EXEC SQL COMMIT WORK RELEASE ;
if (ProcessOracleErrorCode ("Exit Oracle") == FALSE)
return (FALSE) ;
return (TRUE) ;
}

BOOL ProcessOracleErrorCode (LPSTR Header)
{
if (sqlca.sqlcode < 0)
{
MessageBox (GetFocus(),sqlca.sqlerrm.sqlerrmc,Header,MB_OK) ;
return (FALSE) ;
}
return (TRUE) ;
}

BOOL IsValidOracle (LPSTR Name,LPSTR Type,LPSTR Constraint)
{
ConstraintFlag = NO_CONSTRAINT ;
SetCapture (hDrawWnd) ;
hOldCursor = SetCursor (hWaitCursor) ;
Istrcpy (Oralmmmediate.arr,"CREATE TABLE TEST ( ?) ;
Istrcat (Oralmmmediate.arr,Name) ;
Istrcat (Oralmmmediate.arr,"") ;
Istrcat (Oralmmmediate.arr,Type) ;
Istrcat (Oralmmmediate.arr,"") ;
Oralmmmediate.len = Istrlen (Oralmmmediate.arr) ;
EXEC SQL EXECUTE IMMEDIATE :Oralmmmediate ;
if (ProcessOracleErrorCode ("Create Test Table") == FALSE)
{
SetCursor (hOldCursor) ;
ReleaseCapture () ;
return (FALSE) ;
}
if (IsBlankString (Constraint) == FALSE)
{
EXEC SQL WHENEVER NOT FOUND CONTINUE ;
EXEC SQL SELECT DATA_Type INTO :Data_Type FROM COLS WHERE TABLE_NAME = 'TEST' ;
Data_Type.arr[Data_Type.len] = '\0' ;
if ((Istrcmp (Data_Type.arr,"NUMBER") == 0) || (Istrcmp (Data_Type.arr,"LONG") == 0))
{
if (CheckDomainNumberBuffer () == FALSE)
{
EXEC SQL DROP TABLE TEST ;
SetCursor (hOldCursor) ;
ReleaseCapture () ;
return (FALSE) ;
}
else
ConstraintFlag = CONSTRAINT_NUMBER ;
}
if ((Istrcmp (Data_Type.arr,"CHAR") == 0))
{
if (IsValidDomainCharBuffer () == FALSE)
{
EXEC SQL DROP TABLE TEST ;
SetCursor (hOldCursor) ;
ReleaseCapture () ;
return (FALSE) ;
}
else
ConstraintFlag = CONSTRAINT_CHAR ;
}
}
EXEC SQL DROP TABLE TEST ;
if (ProcessOracleErrorCode ("Drop Test Table") == FALSE)
{
SetCursor (hOldCursor) ;
ReleaseCapture () ;
return (FALSE) ;
}
EXEC SQL COMMIT WORK ;
if (ProcessOracleErrorCode ("Commit Test Table") == FALSE)
{
SetCursor (hOldCursor) ;
ReleaseCapture () ;
return (FALSE) ;
}
SetCursor (hOldCursor) ;
ReleaseCapture () ;
return (TRUE) ;
}

void IdentifyMapping ()
{
EXEC SQL DECLARE C45 CURSOR FOR
SELECT RELATION_NAME
FROM TEMP_RELATION
WHERE RELATION_TYPE = 3 ;
EXEC SQL DECLARE C46 CURSOR FOR

```

เอกสารนี้เป็นเอกสารของบริษัทฯ เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SELECT  ATTR_NUMBER,ATTR_NAME,ATTR_DATATYPE
FROM    TEMP_ATTR
WHERE   RELATION_NAME = :Relation_Name ;

EXEC SQL OPEN C45 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit45 ;
for (:)
{
    EXEC SQL FETCH C45 INTO :Relation_Name ;
    Relation_Name.arr[Relation_Name.len] = '\0' ;
    EXEC SQL OPEN C46 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit46 ;
    for (:)
    {
        EXEC SQL FETCH C46 INTO :Attr_Number,:Attr_Name,:DataType ;
        Attr_Name.arr[Attr_Name.len] = '\0' ;
        DataType.arr[DataType.len] = '\0' ;
        Istrcpy (Not_Null.arr,"N") ;
        Not_Null.len = Istrlen (Not_Null.arr) ;
        EXEC SQL
        SELECT  ENTITY_NAME INTO :Entity_Name
        FROM    TEMP_ROLE
        WHERE   RELATION_NAME = :Relation_Name AND CADINALITY = 1 ;
        EXEC SQL INSERT INTO ATTR_OF_TABLE (TABLE_NAME,ATTR_NUMBER,NOT_NULL)
        VALUES (:Entity_Name,:Attr_Number,:Not_Null) ;
        EXEC SQL
        SELECT  ATTR_NAME,ATTR_DATATYPE INTO :Attr_Name,:DataType
        FROM    TEMP_ATTR
        WHERE   ATTR_NUMBER = :Attr_Number ;
        Attr_Name.arr[Attr_Name.len] = '\0' ;
        DataType.arr[DataType.len] = '\0' ;
        EXEC SQL INSERT INTO ATTR_DETAIL (ATTR_NUMBER,ATTR_NAME,DATATYPE)
        VALUES (:Attr_Number,:Attr_Name,:DataType) ;
    }
quit46:
    EXEC SQL CLOSE C46 ;
}
quit45:
    EXEC SQL CLOSE C45 ;
}

BOOL ValidStringChar (void)
{
    int  i,Flag ;

    Flag = TRUE ;
    for (i=0;i<Istrlen (DomainBuffer) ;i++)
    {
        if (!((DomainBuffer[i] < (char) 0x7f && DomainBuffer[i] > (char) 0x1f) || DomainBuffer[i] == '\0'))
            Flag = FALSE ;
    }
    return (Flag) ;
}

BOOL IsValidDomainCharBuffer (void)
{
    int  i ;

    if (ValidStringChar() == TRUE)
    {
        i=0 ;
        Invert.arr[0] = 'N' ;
        Invert.arr[1] = '\0' ;
        for (i=0;DomainBuffer[i] == '\0' || DomainBuffer[i] == '\n' ;i++) ;
        if (DomainBuffer[i] == '\r')
        {
            Invert.arr[0] = 'Y' ;
            i++ ;
            for (i=0;DomainBuffer[i] == '\0' || DomainBuffer[i] == '\n' ;i++) ;
        }
        Invert.len = Istrlen (Invert.arr) ;
        if (DomainBuffer[i] == '\t')
        {
            step1:
                i++ ;
                for (i=0;DomainBuffer[i] == '\0' || DomainBuffer[i] == '\n' ;i++) ;
                if (DomainBuffer[i] == (char) 0x27)
                {
                    step2:
                        i++ ;
                        for (i=0;DomainBuffer[i] != (char) 0x27 && i<Istrlen (DomainBuffer) ;i++) ;
                        if (DomainBuffer[i] == (char) 0x27)
                        {
                            i++ ;
                            if (DomainBuffer[i] == (char) 0x27)
                                goto step2 ;
                            for (i=0;DomainBuffer[i] == '\0' || DomainBuffer[i] == '\n' ;i++) ;
                            if (DomainBuffer[i] == '\r')
                                return (TRUE) ;
                            else
                            {
                                if (DomainBuffer[i] == ',')
                                    goto step1 ;
                                else
                                    return (FALSE) ;
                            }
                        }
                    }
                }
            }
        }
    }
    return (FALSE) ;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        else
            return (FALSE);
    }
    else
        return (FALSE);
}
else
    return (FALSE);
}
}

BOOL CheckDomainNumberBuffer (void)
{
    int i,j;

    for (i=0,j=0;DomainBuffer[i] != '\0';i++)
    {
        if (ValidNumberChar (DomainBuffer[i]) == TRUE)
        {
            if ((DomainBuffer[i] != ' ') && (DomainBuffer[i] != '\t'))
                TempBuffer[j++] = DomainBuffer[i];
        }
        else
            return (FALSE);
    }
    TempBuffer[j] = '\0';
    if (ValidDomainNumberBuffer () == TRUE)
        return (TRUE);
    else
        return (FALSE);
}

BOOL ValidNumberChar (char ch)
{
    int i;

    for (i=0;i<sizeof(NumberTable);i++)
    {
        if (NumberTable[i] == ch)
            return (TRUE);
    }
    return (FALSE);
}

BOOL ValidDomainNumberBuffer (void)
{
    int i,dotpoint,position,count;

    if (TempBuffer[0] != '(') return (FALSE);
    if (TempBuffer[1] != '[' && TempBuffer [1] != '(') return (FALSE);
    i=1;
    position = MIN;

state1:
    count = 0;
    dotpoint = 0;
state2:
    i++;
    if (isDigit (TempBuffer[i]) == TRUE)
    {
        if (count != 0)
        {
            switch (TempBuffer[i])
            {
                case '.':
                case '+':
                case '>':
                case '<':
                    return (FALSE);
            }
        }
        if (TempBuffer[i] == '>' && position == MIN) return (FALSE);
        if (TempBuffer[i] == '<' && position == MAX) return (FALSE);
        count++;
        goto state2;
    }
    else
    {
        if (TempBuffer[i] == '.')
        {
            dotpoint++;
            goto state2;
        }
        if (dotpoint > 1 || count == 0) return (FALSE);
    }
    if (position == MIN)
    {
        if (TempBuffer[i] == ',')
        {
            position = MAX;
            goto state1;
        }
        else
            return (FALSE);
    }
    else
    {
        if (TempBuffer[i] != ']' && TempBuffer[i] != ')')
            return (FALSE);
        else
            position = MIN;
    }
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
 อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        goto state3;
    }
}
state3:
i++;
if (TempBuffer[i] != '}')
{
    if (TempBuffer[i] != ',') return (FALSE);
    i++;
    if (TempBuffer[i] != '[' && TempBuffer[i] != '(') return (FALSE);
    goto state1;
}
return (TRUE);
}

BOOL IsDigit (char ch)
{
    int i;
    static char Digit[] = { '0','1','2','3','4','5','6','7','8','9','+','*','>','<' };

    for (i=0;i<sizeof(Digit);i++)
    {
        if (Digit[i] == ch)
            return (TRUE);
    }
    return (FALSE);
}

BOOL LoadOracleTable (void)
{
    int hFile,count;
    OFSTRUCT OfStruct;

    hFile = OpenFile ("First.sql",&OfStruct,OF_CREATE);
    if (hFile >=0) close (hFile);
    hFile = OpenFile ("BCNF.sql" ,&OfStruct,OF_CREATE);
    if (hFile >=0) close (hFile);
    hFile = OpenFile ("Fifth.sql",&OfStruct,OF_CREATE);
    if (hFile >=0) close (hFile);
    EXEC SQL WHENEVER NOT FOUND CONTINUE;
    hFile = OpenFile (OracleFileName,&OfStruct,OF_READ);
    if (hFile >=0)
    {
        /* Load entity file to table */
        EXEC SQL DELETE ENTITY;
        count = read (hFile,&Entity,sizeof (Entity));
        while (strcmp (Entity.Name,"END") != 0)
        {
            Istrcpy (Entity_Name.arr,Entity.Name);
            Entity_Name.len = strlen (Entity_Name.arr);
            Type = Entity.Type;
            EXEC SQL INSERT INTO ENTITY (ENTITY_NAME,ENTITY_TYPE) VALUES (Entity_Name,Type);
            if (ProcessOracleErrorCode ("Load Oracle Entity File") == FALSE)
            {
                EXEC SQL ROLLBACK WORK;
                close (hFile);
                return (FALSE);
            }
            count = read (hFile,&Entity,sizeof (Entity));
        };
        /* Load Relation file to table */
        EXEC SQL DELETE RELATION;
        count = read (hFile,&Relation,sizeof (Relation));
        while (strcmp (Relation.Name,"END") != 0)
        {
            Istrcpy (Relation_Name.arr,Relation.Name);
            Relation_Name.len = strlen (Relation_Name.arr);
            Type = Relation.Type;
            EXEC SQL INSERT INTO RELATION (RELATION_NAME,RELATION_TYPE) VALUES (:Relation_Name,Type);
            if (ProcessOracleErrorCode ("Load Oracle Relation File") == FALSE)
            {
                EXEC SQL ROLLBACK WORK;
                close (hFile);
                return (FALSE);
            }
            count = read (hFile,&Relation,sizeof (Relation));
        };
        /* Load Attribute file to table */
        EXEC SQL DELETE ATTR;
        count = read (hFile,&Attr,sizeof (Attr));
        while (Attr.Attr_Number != 0xffff)
        {
            Attr_Number = Attr.Attr_Number;
            Istrcpy (Attr_Name.arr,Attr.Name);
            Attr_Name.len = strlen (Attr_Name.arr);
            Type = Attr.Type;
            Istrcpy (Entity_Name.arr,Attr.Entity_Name);
            Entity_Name.len = strlen (Entity_Name.arr);
            Istrcpy (Relation_Name.arr,Attr.Relation_Name);
            Relation_Name.len = strlen (Relation_Name.arr);
            Istrcpy (DataType.arr,Attr.DataType);
            DataType.len = strlen (DataType.arr);
            Composit_Number = Attr.Composit_Number;
            EXEC SQL INSERT INTO ATTR
            (ATTR_NUMBER,ATTR_NAME,ATTR_TYPE,ATTR_DATATYPE,ENTITY_NAME,RELATION_NAME,COMPOSIT_NUMBER)
            VALUES (:Attr_Number,:Attr_Name,Type,:DataType,:Entity_Name,:Relation_Name,:Composit_Number);
            if (ProcessOracleErrorCode ("Load Oracle Attribute File") == FALSE)
            {
                EXEC SQL ROLLBACK WORK;
                close (hFile);
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามทำซ้ำโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return (FALSE);
    }
    count = read (hFile,&Attr,sizeof (Attr));
};
/* Load role file to table */
EXEC SQL DELETE ROLE;
count = read (hFile,&Role,sizeof (Role));
while (Role.Role_Number != 0xffff)
{
    Role_Number = Role.Role_Number;
    Partial = Role.Paticipation;
    Cadinal = Role.Cadinality;
    Istrcpy (Entity_Name.arr,Role.Entity_Name);
    Entity_Name.len = Istrlen (Entity_Name.arr);
    Istrcpy (Relation_Name.arr,Role.Relation_Name);
    Relation_Name.len = Istrlen (Relation_Name.arr);
    EXEC SQL INSERT INTO ROLE (ROLE_NUMBER,PATICIPATION,CADINALITY,ENTITY_NAME,RELATION_NAME)
    VALUES (:Role_Number,:Partial,:Cadinal,:Entity_Name,:Relation_Name);
    If (ProcessOracleErrorCode ("Load Oracle Role File") == FALSE)
    {
        EXEC SQL ROLLBACK WORK;
        close (hFile);
        return (FALSE);
    }
    count = read (hFile,&Role,sizeof (Role));
};
/* Load n_constraint file to table */
EXEC SQL DELETE N_CONSTRAINT;
count = read (hFile,&nDomain,sizeof (nDomain));
while (nDomain.Attr_Number != 0xffff)

    ttr_Number = nDomain.Attr_Number;
    Istrcpy (Min.arr,nDomain.Min);
    Min.len = Istrlen (Min.arr);
    Istrcpy (Lower_Limit.arr,nDomain.Lower_Limit);
    Lower_Limit.len = Istrlen (Lower_Limit.arr);
    Istrcpy (Max.arr,nDomain.Max);
    Max.len = Istrlen (Max.arr);
    Istrcpy (Upper_Limit.arr,nDomain.Upper_Limit);
    Upper_Limit.len = Istrlen (Upper_Limit.arr);
    EXEC SQL INSERT INTO N_CONSTRAINT (ATTR_NUMBER,LOWER_LIMIT,MIN,MAX,UPPER_LIMIT)
    VALUES (:Attr_Number,:Lower_Limit,:Min,:Max,:Upper_Limit);
    if (ProcessOracleErrorCode ("Load Oracle nDomain File") == FALSE)
    {
        EXEC SQL ROLLBACK WORK;
        close (hFile);
        return (FALSE);
    }
    count = read (hFile,&nDomain,sizeof (nDomain));
};
/* Load c_constraint file to table */
EXEC SQL DELETE C_CONSTRAINT;
count = read (hFile,&cDomain,sizeof (cDomain));
while (cDomain.Attr_Number != 0xffff)
{
    Attr_Number = cDomain.Attr_Number;
    Istrcpy (Domain_Name.arr,cDomain.Domain_Name);
    Domain_Name.len = Istrlen (Domain_Name.arr);
    EXEC SQL INSERT INTO C_CONSTRAINT (ATTR_NUMBER,DOMAIN_NAME) VALUES (:Attr_Number,:Domain_Name);
    if (ProcessOracleErrorCode ("Load Oracle cDomain File") == FALSE)
    {
        EXEC SQL ROLLBACK WORK;
        close (hFile);
        return (FALSE);
    }
    count = read (hFile,&cDomain,sizeof (cDomain));
};
/* Load c_invert file to table */
EXEC SQL DELETE INVERT;
count = read (hFile,&cInvert,sizeof (cInvert));
while (cInvert.Attr_Number != 0xffff)
{
    Attr_Number = cInvert.Attr_Number;
    Istrcpy (Invert.arr,cInvert.Invert);
    Invert.len = Istrlen (Invert.arr);
    EXEC SQL INSERT INTO INVERT (ATTR_NUMBER,INVERT) VALUES (:Attr_Number,:Invert);
    if (ProcessOracleErrorCode ("Load Oracle cInvert File") == FALSE)
    {
        EXEC SQL ROLLBACK WORK;
        close (hFile);
        return (FALSE);
    }
    count = read (hFile,&cInvert,sizeof (cInvert));
};
/* Load Fd file to table */
EXEC SQL DELETE FD;
count = read (hFile,&Fd,sizeof (Fd));
while (Fd.Fd_Number != 0xffff)
{
    Fd_Number = Fd.Fd_Number;
    Istrcpy (Table_Name.arr,Fd.Table_Name);
    Table_Name.len = Istrlen (Table_Name.arr);
    EXEC SQL INSERT INTO FD (FD_NUMBER,TABLE_NAME) VALUES (:Fd_Number,:Table_Name);
    If (ProcessOracleErrorCode ("Load Oracle Fd File") == FALSE)
    {
        EXEC SQL ROLLBACK WORK;
        close (hFile);
        return (FALSE);
    }
    count = read (hFile,&Fd,sizeof (Fd));
};

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ห้ามมิให้คัดลอกหรือใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

);
/* Load LHS file to table */
EXEC SQL DELETE LHS ;
count = read (hFile,&LHS,sizeof (LHS)) ;
while (LHS.Fd_Number != 0xffff)
{
    Fd_Number = LHS.Fd_Number ;
    Attr_Number = LHS.Attr_Number ;
    EXEC SQL INSERT INTO LHS (FD_NUMBER,LHS_NUMBER) VALUES (:Fd_Number,:Attr_Number) ;
    if (ProcessOracleErrorCode ("Load Oracle LHS File") == FALSE)
    {
        EXEC SQL ROLLBACK WORK ;
        close (hFile) ;
        return (FALSE) ;
    }
    count = read (hFile,&LHS,sizeof (LHS)) ;
};
/* Load RHS file to table */
EXEC SQL DELETE RHS ;
count = read (hFile,&RHS,sizeof (RHS)) ;
while (RHS.Fd_Number != 0xffff)
{
    Fd_Number = RHS.Fd_Number ;
    Attr_Number = RHS.Attr_Number ;
    EXEC SQL INSERT INTO RHS (FD_NUMBER,RHS_NUMBER) VALUES (:Fd_Number,:Attr_Number) ;
    if (ProcessOracleErrorCode ("Load Oracle RHS File") == FALSE)
    {
        EXEC SQL ROLLBACK WORK ;
        close (hFile) ;
        return (FALSE) ;
    }
    count = read (hFile,&RHS,sizeof (RHS)) ;
};
/* Load Temp_Attr_of_Index to table */
EXEC SQL DELETE TEMP_ATTR_OF_INDEX ;
count = read (hFile,&Temp_Attr_of_Index,sizeof (Temp_Attr_of_Index)) ;
while (Temp_Attr_of_Index.Index_Number != 0xffff)
{
    Index_Number = Temp_Attr_of_Index.Index_Number ;
    Attr_Number = Temp_Attr_of_Index.Attr_Number ;
    EXEC SQL INSERT INTO TEMP_ATTR_OF_INDEX (INDEX_NUMBER,ATTR_NUMBER)
    VALUES (:Index_Number,:Attr_Number) ;
    if (ProcessOracleErrorCode ("Load Oracle TEMP_ATTR_OF_INDEX File") == FALSE)
    {
        EXEC SQL ROLLBACK WORK ;
        close (hFile) ;
        return (FALSE) ;
    }
    count = read (hFile,&Temp_Attr_of_Index,sizeof (Temp_Attr_of_Index)) ;
};
/* Load Temp_Table_of_Index to table */
EXEC SQL DELETE TEMP_TABLE_OF_INDEX ;
count = read (hFile,&Temp_Table_of_Index,sizeof (Temp_Table_of_Index)) ;
while (Temp_Table_of_Index.Index_Number != 0xffff)
{
    Index_Number = Temp_Table_of_Index.Index_Number ;
    strcpy (Table_Name,arr,Temp_Table_of_Index.Table_Name) ;
    Table_Name.len = strlen (Table_Name.arr) ;
    EXEC SQL INSERT INTO TEMP_TABLE_OF_INDEX (INDEX_NUMBER,TABLE_NAME)
    VALUES (:Index_Number,:Table_Name) ;
    if (ProcessOracleErrorCode ("Load Oracle TEMP_Table_OF_INDEX File") == FALSE)
    {
        EXEC SQL ROLLBACK WORK ;
        close (hFile) ;
        return (FALSE) ;
    }
    count = read (hFile,&Temp_Table_of_Index,sizeof (Temp_Table_of_Index)) ;
};
/* Load Temp_Attr_of_Table to table */
EXEC SQL DELETE TEMP_ATTR_OF_TABLE ;
count = read (hFile,&Temp_Attr_of_Table,sizeof (Temp_Attr_of_Table)) ;
while (strcmp (Temp_Attr_of_Table.Table_Name,"END") != 0)
{
    Attr_Number = Temp_Attr_of_Table.Attr_Number ;
    strcpy (Table_Name,arr,Temp_Attr_of_Table.Table_Name) ;
    Table_Name.len = strlen (Table_Name.arr) ;
    strcpy (Not_Null,arr,Temp_Attr_of_Table.Not_Null) ;
    Not_Null.len = strlen (Not_Null.arr) ;
    EXEC SQL INSERT INTO TEMP_ATTR_OF_TABLE (TABLE_NAME,ATTR_NUMBER,NOT_NULL)
    VALUES (:Table_Name,:Attr_Number,:Not_Null) ;
    if (ProcessOracleErrorCode ("Load Oracle TEMP_Attr_of_Table File") == FALSE)
    {
        EXEC SQL ROLLBACK WORK ;
        close (hFile) ;
        return (FALSE) ;
    }
    count = read (hFile,&Temp_Attr_of_Table,sizeof (Temp_Attr_of_Table)) ;
};
/* Load Attr_Detail to table */
EXEC SQL DELETE ATTR_DETAIL ;
count = read (hFile,&Attr_Detail,sizeof (Attr_Detail)) ;
while (Attr_Detail.Attr_Number != 0xffff)
{
    Attr_Number = Attr_Detail.Attr_Number ;
    strcpy (Attr_Name,arr,Attr_Detail.Attr_Name) ;
    Attr_Name.len = strlen (Attr_Name.arr) ;
    strcpy (DataType,arr,Attr_Detail.DataType) ;
    DataType.len = strlen (DataType.arr) ;
    EXEC SQL INSERT INTO ATTR_DETAIL (ATTR_NUMBER,ATTR_NAME,DATATYPE)
    VALUES (:Attr_Number,:Attr_Name,:DataType) ;
};

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เพื่อการวิจัยเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (ProcessOracleErrorCode ("Load Oracle Attr_Detail File") == FALSE)
{
    EXEC SQL ROLLBACK WORK ;
    close (hFile) ;
    return (FALSE) ;
}
count = read (hFile,&Attr_Detail,sizeof (Attr_Detail)) ;
};
/* Load Attr_of_Index to table */
EXEC SQL DELETE ATTR_OF_INDEX ;
count = read (hFile,&Attr_of_Index,sizeof (Attr_of_Index)) ;
while (Attr_of_Index.Index_Number != 0xffff)
{
    Index_Number = Attr_of_Index.Index_Number ;
    Attr_Number = Attr_of_Index.Attr_Number ;
    EXEC SQL INSERT INTO ATTR_OF_INDEX (INDEX_NUMBER,ATTR_NUMBER) VALUES (:Index_Number,:Attr_Number) ;
    if (ProcessOracleErrorCode ("Load Oracle ATTR_OF_INDEX File") == FALSE)
    {
        EXEC SQL ROLLBACK WORK ;
        close (hFile) ;
        return (FALSE) ;
    }
    count = read (hFile,&Attr_of_Index,sizeof (Attr_of_Index)) ;
};
/* Load Table_of_Index to table */
EXEC SQL DELETE TABLE_OF_INDEX ;
count = read (hFile,&Table_of_Index,sizeof (Table_of_Index)) ;
while (Table_of_Index.Index_Number != 0xffff)
{
    Index_Number = Table_of_Index.Index_Number ;
    Istrcpy (Table_Name,arr,Table_of_Index.Table_Name) ;
    Table_Name.len = Istrlen (Table_Name.arr) ;
    EXEC SQL INSERT INTO TABLE_OF_INDEX (INDEX_NUMBER,TABLE_NAME) VALUES (:Index_Number,:Table_Name) ;
    if (ProcessOracleErrorCode ("Load Oracle Table_OF_INDEX File") == FALSE)
    {
        EXEC SQL ROLLBACK WORK ;
        close (hFile) ;
        return (FALSE) ;
    }
    count = read (hFile,&Table_of_Index,sizeof (Table_of_Index)) ;
};
/* Load Attr_of_Table to table */
EXEC SQL DELETE ATTR_OF_TABLE ;
count = read (hFile,&Attr_of_Table,sizeof (Attr_of_Table)) ;
while (Istrcmp(Attr_of_Table.Table_Name,"END") != 0)
{
    Attr_Number = Attr_of_Table.Attr_Number ;
    Istrcpy (Table_Name,arr,Attr_of_Table.Table_Name) ;
    Table_Name.len = Istrlen (Table_Name.arr) ;
    Istrcpy (Not_Null,arr,Attr_of_Table.Not_Null) ;
    Not_Null.len = Istrlen (Not_Null.arr) ;
    EXEC SQL INSERT INTO ATTR_OF_TABLE (TABLE_NAME,ATTR_NUMBER,NOT_NULL)
    VALUES (:Table_Name,:Attr_Number,:Not_Null) ;
    if (ProcessOracleErrorCode ("Load Oracle Attr_of_Table File") == FALSE)
    {
        EXEC SQL ROLLBACK WORK ;
        close (hFile) ;
        return (FALSE) ;
    }
    count = read (hFile,&Attr_of_Table,sizeof (Attr_of_Table)) ;
};
EXEC SQL
SELECT COUNT (*) INTO :OraCount
FROM FD ;
FdFlag = FALSE ;
JdFlag = FALSE ;
if (OraCount > 0)
{
    FdFlag = TRUE ;
    EXEC SQL
    SELECT COUNT (*) INTO :OraCount
    FROM ATTR_OF_INDEX ;
    if (OraCount > 0)
        JdFlag = TRUE ;
}
EXEC SQL COMMIT WORK ;
if (ProcessOracleErrorCode ("Load Oracle File COMMIT") == FALSE)
{
    EXEC SQL ROLLBACK WORK ;
    close (hFile) ;
    return (FALSE) ;
}
close (hFile) ;
return (TRUE) ;
}
else
{
    ErrorMessage ("Can't Load Oracle File") ;
    return (FALSE) ;
}
}
}

```

```

BOOL SaveOracleTable (void)
{

```

```

    int hFile,count ;
    OFSTRUCT OfStruct ;

```

```

EXEC SQL DECLARE C1 CURSOR FOR
SELECT ENTITY_NAME,ENTITY_TYPE
FROM นศ ENTITY ;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EXEC SQL DECLARE C2 CURSOR FOR
SELECT RELATION_NAME,RELATION_TYPE
FROM RELATION;

EXEC SQL DECLARE C3 CURSOR FOR
SELECT ATTR_NUMBER,ATTR_NAME,ATTR_TYPE,ATTR_DATATYPE,ENTITY_NAME,RELATION_NAME,COMPOSIT_NUMBER
FROM ATTR;

EXEC SQL DECLARE C4 CURSOR FOR
SELECT ROLE_NUMBER,PATICIPATION,CADINALITY,ENTITY_NAME,RELATION_NAME
FROM ROLE;

EXEC SQL DECLARE C90 CURSOR FOR
SELECT ATTR_NUMBER,LOWER_LIMIT,MIN,MAX,UPPER_LIMIT
FROM N_CONSTRAINT;

EXEC SQL DECLARE C91 CURSOR FOR
SELECT ATTR_NUMBER,DOMAIN_NAME
FROM C_CONSTRAINT;

EXEC SQL DECLARE C92 CURSOR FOR
SELECT ATTR_NUMBER,INVERT
FROM INVERT;

EXEC SQL DECLARE C93 CURSOR FOR
SELECT FD_NUMBER,TABLE_NAME
FROM FD;

EXEC SQL DECLARE C94 CURSOR FOR
SELECT FD_NUMBER,LHS_NUMBER
FROM LHS;

EXEC SQL DECLARE C95 CURSOR FOR
SELECT FD_NUMBER,RHS_NUMBER
FROM RHS;

EXEC SQL DECLARE C96 CURSOR FOR
SELECT INDEX_NUMBER,ATTR_NUMBER
FROM TEMP_ATTR_OF_INDEX;

EXEC SQL DECLARE C97 CURSOR FOR
SELECT INDEX_NUMBER,TABLE_NAME
FROM TEMP_TABLE_OF_INDEX;

EXEC SQL DECLARE C98 CURSOR FOR
SELECT TABLE_NAME,ATTR_NUMBER,NOT_NULL
FROM TEMP_ATTR_OF_TABLE;

EXEC SQL DECLARE C99 CURSOR FOR
SELECT ATTR_NUMBER,ATTR_NAME,DATATYPE
FROM ATTR_DETAIL;

EXEC SQL DECLARE C100 CURSOR FOR
SELECT INDEX_NUMBER,ATTR_NUMBER
FROM ATTR_OF_INDEX;

EXEC SQL DECLARE C101 CURSOR FOR
SELECT INDEX_NUMBER,TABLE_NAME
FROM TABLE_OF_INDEX;

EXEC SQL DECLARE C102 CURSOR FOR
SELECT TABLE_NAME,ATTR_NUMBER,NOT_NULL
FROM ATTR_OF_TABLE;

hFile = OpenFile (OracleFileName,&OfStruct,OF_CREATE);
if (hFile >=0)
{
    /* save entity table to file */
    EXEC SQL OPEN C1 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit1 ;
    for (;;)
    {
        EXEC SQL FETCH C1 INTO :Entity_Name,:Type ;
        Entity_Name.arr[Entity_Name.len] = '\0' ;
        Istrcpy (Entity_Name,Entity_Name.arr) ;
        Entity.Type = Type ;
        count = write (hFile,&Entity,sizeof (Entity)) ;
    }
quit1:
    EXEC SQL CLOSE C1 ;
    Istrcpy (Entity_Name,"END") ;
    Entity.Type = 0xff ;
    count = write (hFile,&Entity,sizeof (Entity)) ;

    /* save relation table to file */
    EXEC SQL OPEN C2 ;
    EXEC SQL WHENEVER NOT FOUND GOTO quit2 ;
    for (;;)
    {
        EXEC SQL FETCH C2 INTO :Relation_Name,:Type ;
        Relation_Name.arr[Relation_Name.len] = '\0' ;
        Istrcpy (Relation_Name,Relation_Name.arr) ;
        Relation.Type = Type ;
        count = write (hFile,&Relation,sizeof (Relation)) ;
    }
quit2:
    EXEC SQL CLOSE C2 ;
    Istrcpy (Relation_Name,"END") ;
    Relation.Type = 0xff ;
}

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

count = write (hFile,&Relation,sizeof (Relation));

/* save attribute table to file */
EXEC SQL OPEN C3 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit3 ;
for (;;)
{
    EXEC SQL FETCH C3 INTO :Attr_Number,:Attr_Name,:Type,:DataType,:Entity_Name,:Relation_Name,:Composit_Number ;
    Attr.Attr_Number = Attr_Number ;
    Attr.Name.arr[Attr_Name.len] = '0' ;
    Istrcpy (Attr.Name,Attr_Name.arr) ;
    Attr.DataType.arr[Attr.DataType.len] = '0' ;
    Istrcpy (Attr.DataType,Attr.DataType.arr) ;
    Attr.Entity_Name.arr[Attr.Entity_Name.len] = '0' ;
    Istrcpy (Attr.Entity_Name,Attr.Entity_Name.arr) ;
    Attr.Relation_Name.arr[Attr.Relation_Name.len] = '0' ;
    Istrcpy (Attr.Relation_Name,Attr.Relation_Name.arr) ;
    Attr.Type = Type ;
    Attr.Composit_Number = Composit_Number ;
    count = write (hFile,&Attr,sizeof (Attr)) ;
}
quit3:
EXEC SQL CLOSE C3 ;
Attr.Attr_Number = 0xffff ;
count = write (hFile,&Attr,sizeof (Attr)) ;

/* save role table to file */
EXEC SQL OPEN C4 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit4 ;
for (;;)
{
    EXEC SQL FETCH C4 INTO :Role_Number,:Partial,:Cardinal,:Entity_Name,:Relation_Name ;
    Role.Role_Number = Role_Number ;
    Role.Participation = Partial ;
    Role.Cardinality = Cardinal ;
    Role.Entity_Name.arr[Role.Entity_Name.len] = '0' ;
    Istrcpy (Role.Entity_Name,Role.Entity_Name.arr) ;
    Role.Relation_Name.arr[Role.Relation_Name.len] = '0' ;
    Istrcpy (Role.Relation_Name,Role.Relation_Name.arr) ;
    count = write (hFile,&Role,sizeof (Role)) ;
}
quit4:
EXEC SQL CLOSE C4 ;
Role.Role_Number = 0xffff ;
count = write (hFile,&Role,sizeof (Role)) ;

/* save n_constraint to file */
EXEC SQL OPEN C90 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit90 ;
for (;;)
{
    EXEC SQL FETCH C90 INTO :Attr_Number,:Lower_Limit,:Min,:Max,:Upper_Limit ;
    nDomain.Attr_Number = Attr_Number ;
    Lower_Limit.arr[Lower_Limit.len] = '0' ;
    Istrcpy (nDomain.Lower_Limit,Lower_Limit.arr) ;
    Min.arr[Min.len] = '0' ;
    Istrcpy (nDomain.Min,Min.arr) ;
    Max.arr[Max.len] = '0' ;
    Istrcpy (nDomain.Max,Max.arr) ;
    Upper_Limit.arr[Upper_Limit.len] = '0' ;
    Istrcpy (nDomain.Upper_Limit,Upper_Limit.arr) ;
    count = write (hFile,&nDomain,sizeof (nDomain)) ;
}
quit90:
EXEC SQL CLOSE C90 ;
nDomain.Attr_Number = 0xffff ;
count = write (hFile,&nDomain,sizeof (nDomain)) ;

/* save c_constraint to file */
EXEC SQL OPEN C91 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit91 ;
for (;;)
{
    EXEC SQL FETCH C91 INTO :Attr_Number,:Domain_Name ;
    cDomain.Attr_Number = Attr_Number ;
    Domain_Name.arr[Domain_Name.len] = '0' ;
    Istrcpy (cDomain.Domain_Name,Domain_Name.arr) ;
    count = write (hFile,&cDomain,sizeof (cDomain)) ;
}
quit91:
EXEC SQL CLOSE C91 ;
cDomain.Attr_Number = 0xffff ;
count = write (hFile,&cDomain,sizeof (cDomain)) ;

/* save c_invert to file */
EXEC SQL OPEN C92 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit92 ;
for (;;)
{
    EXEC SQL FETCH C92 INTO :Attr_Number,:Invert ;
    cInvert.Attr_Number = Attr_Number ;
    Invert.arr[Invert.len] = '0' ;
    Istrcpy (cInvert.Invert,Invert.arr) ;
    count = write (hFile,&cInvert,sizeof (cInvert)) ;
}
quit92:
EXEC SQL CLOSE C92 ;
cInvert.Attr_Number = 0xffff ;
count = write (hFile,&cInvert,sizeof (cInvert)) ;

```

เอกสารนี้เป็นเอกสารที่ลงวันเวสสำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* save Fd to file */
EXEC SQL OPEN C93 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit93 ;
for (;;)
{
    EXEC SQL FETCH C93 INTO :Fd_Number,:Table_Name ;
    Fd.Fd_Number = Fd_Number ;
    Table_Name.arr[Table_Name.len] = '\0' ;
    Istrcpy (Fd.Table_Name,Table_Name.arr) ;
    count = write (hFile,&Fd,sizeof (Fd)) ;
}
quit93:
EXEC SQL CLOSE C93 ;
Fd.Fd_Number = 0xfffe ;
count = write (hFile,&Fd,sizeof (Fd)) ;

/* save LHS to file */
EXEC SQL OPEN C94 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit94 ;
for (;;)
{
    EXEC SQL FETCH C94 INTO :Fd_Number,:Attr_Number ;
    LHS.Fd_Number = Fd_Number ;
    LHS.Attr_Number = Attr_Number ;
    count = write (hFile,&LHS,sizeof (LHS)) ;
}
quit94:
EXEC SQL CLOSE C94 ;
LHS.Fd_Number = 0xfffe ;
count = write (hFile,&LHS,sizeof (LHS)) ;

/* save RHS to file */
EXEC SQL OPEN C95 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit95 ;
for (;;)
{
    EXEC SQL FETCH C95 INTO :Fd_Number,:Attr_Number ;
    RHS.Fd_Number = Fd_Number ;
    RHS.Attr_Number = Attr_Number ;
    count = write (hFile,&RHS,sizeof (RHS)) ;
}
quit95:
EXEC SQL CLOSE C95 ;
RHS.Fd_Number = 0xfffe ;
count = write (hFile,&RHS,sizeof (RHS)) ;

/* save temp_attr_of_index to file */
EXEC SQL OPEN C96 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit96 ;
for (;;)
{
    EXEC SQL FETCH C96 INTO :Index_Number,:Attr_Number ;
    Temp_Attr_of_Index.Index_Number = Index_Number ;
    Temp_Attr_of_Index.Attr_Number = Attr_Number ;
    count = write (hFile,&Temp_Attr_of_Index,sizeof (Temp_Attr_of_Index)) ;
}
quit96:
EXEC SQL CLOSE C96 ;
Temp_Attr_of_Index.Index_Number = 0xfffe ;
count = write (hFile,&Temp_Attr_of_Index,sizeof (Temp_Attr_of_Index)) ;

/* save temp_table_of_index to file */
EXEC SQL OPEN C97 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit97 ;
for (;;)
{
    EXEC SQL FETCH C97 INTO :Index_Number,:Table_Name ;
    Temp_Table_of_Index.Index_Number = Index_Number ;
    Table_Name.arr[Table_Name.len] = '\0' ;
    Istrcpy (Temp_Table_of_Index.Table_Name,Table_Name.arr) ;
    count = write (hFile,&Temp_Table_of_Index,sizeof (Temp_Table_of_Index)) ;
}
quit97:
EXEC SQL CLOSE C97 ;
Temp_Table_of_Index.Index_Number = 0xfffe ;
count = write (hFile,&Temp_Table_of_Index,sizeof (Temp_Table_of_Index)) ;

/* save temp_attr_of_Table to file */
EXEC SQL OPEN C98 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit98 ;
for (;;)
{
    EXEC SQL FETCH C98 INTO :Table_Name,:Attr_Number,:Not_Null ;
    Table_Name.arr[Table_Name.len] = '\0' ;
    Istrcpy (Temp_Attr_of_Table.Table_Name,Table_Name.arr) ;
    Temp_Attr_of_Table.Attr_Number = Attr_Number ;
    Not_Null.arr[Not_Null.len] = '\0' ;
    Istrcpy (Temp_Attr_of_Table.Not_Null,Not_Null.arr) ;
    count = write (hFile,&Temp_Attr_of_Table,sizeof (Temp_Attr_of_Table)) ;
}
quit98:
EXEC SQL CLOSE C98 ;
Istrcpy (Temp_Attr_of_Table.Table_Name,"END") ;
count = write (hFile,&Temp_Attr_of_Table,sizeof (Temp_Attr_of_Table)) ;

/* save Attr_Detail to file */
EXEC SQL OPEN C99 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit99 ;
for (;;)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EXEC SQL FETCH C99 INTO :Attr_Number,:Attr_Name,:DataType ;
Attr_Detail.Attr_Number = Attr_Number ;
Attr_Name.arr[Attr_Name.len] = '\0' ;
Istrcpy (Attr_Detail.Attr_Name,Attr_Name.arr) ;
DataType.arr[DataType.len] = '\0' ;
Istrcpy (Attr_Detail.DataType,DataType.arr) ;
count = write (hFile,&Attr_Detail,sizeof (Attr_Detail)) ;
}
quit99:
EXEC SQL CLOSE C99 ;
Attr_Detail.Attr_Number = 0xffff ;
count = write (hFile,&Attr_Detail,sizeof (Attr_Detail)) ;

/* save attr_of_index to file */
EXEC SQL OPEN C100 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit100 ;
for (;;)
{
EXEC SQL FETCH C100 INTO :Index_Number,:Attr_Number ;
Attr_of_Index.Index_Number = Index_Number ;
Attr_of_Index.Attr_Number = Attr_Number ;
count = write (hFile,&Attr_of_Index,sizeof (Attr_of_Index)) ;
}
quit100:
EXEC SQL CLOSE C100 ;
Attr_of_Index.Index_Number = 0xffff ;
count = write (hFile,&Attr_of_Index,sizeof (Attr_of_Index)) ;

/* save table_of_index to file */
EXEC SQL OPEN C101 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit101 ;
for (;;)
{
EXEC SQL FETCH C101 INTO :Index_Number,:Table_Name ;
Table_of_Index.Index_Number = Index_Number ;
Table_Name.arr[Table_Name.len] = '\0' ;
Istrcpy (Table_of_Index.Table_Name,Table_Name.arr) ;
count = write (hFile,&Table_of_Index,sizeof (Table_of_Index)) ;
}
quit101:
EXEC SQL CLOSE C101 ;
Table_of_Index.Index_Number = 0xffff ;
count = write (hFile,&Table_of_Index,sizeof (Table_of_Index)) ;

/* save attr_of_Table to file */
EXEC SQL OPEN C102 ;
EXEC SQL WHENEVER NOT FOUND GOTO quit102 ;
for (;;)
{
EXEC SQL FETCH C102 INTO :Table_Name,:Attr_Number,:Not_Null ;
Table_Name.arr[Table_Name.len] = '\0' ;
Istrcpy (Attr_of_Table.Table_Name,Table_Name.arr) ;
Attr_of_Table.Attr_Number = Attr_Number ;
Not_Null.arr[Not_Null.len] = '\0' ;
Istrcpy (Attr_of_Table.Not_Null,Not_Null.arr) ;
count = write (hFile,&Attr_of_Table,sizeof (Attr_of_Table)) ;
}
quit102:
EXEC SQL CLOSE C102 ;
Istrcpy (Attr_of_Table.Table_Name,"END") ;
count = write (hFile,&Attr_of_Table,sizeof (Attr_of_Table)) ;

EXEC SQL COMMIT WORK ;
close (hFile) ;
return (TRUE) ;
}
else
{
ErrorMessage ("Can't Save Oracle File") ;
return (FALSE) ;
}
}
}

```

```

void SaveStringConstriant (int Number)

```

```

{
int i,j;
static char TempDomain[100];

i=0;
for (i=0;DomainBuffer[i] == '' || DomainBuffer[i] == '\t';i++);
if (DomainBuffer[i] == '\0')
{
i++;
for (i=0;DomainBuffer[i] == '' || DomainBuffer[i] == '\t';i++);
}
step1:
i++;
j=0;
for (j=i;DomainBuffer[j] == '' || DomainBuffer[j] == '\t';j++);
step2:
i++;
for (i=j;DomainBuffer[i] != (char) 0x27;i++,j++)
TempDomain[j] = DomainBuffer[i];
i++;
if (DomainBuffer[i] == (char) 0x27)
{
TempDomain[j++] = (char) 0x27;
TempDomain[j++] = (char) 0x27;
goto step2;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TempDomain[] = '\0';
Attr_Number = Number;
Istrcpy (Domain_Name.arr,TempDomain);
Domain_Name.len = Istrlen (Domain_Name.arr);
if (Domain_Name.len > 30)
{
    Istrcat (TempDomain," String more than 30 character ");
    ErrorMessage (TempDomain);
}
EXEC SQL INSERT INTO C_CONSTRAINT (ATTR_NUMBER,DOMAIN_NAME) VALUES (:Attr_Number,:Domain_Name);
if (ProcessOracleErrorCode ("INSERT C_CONSTRAINT") == FALSE);
for (i=i;DomainBuffer[i] == ' ' || DomainBuffer[i] == '\t';i++);
if (DomainBuffer[i] != '?')
    goto step1;
EXEC SQL INSERT INTO INVERT (ATTR_NUMBER,INVERT) VALUES (:Attr_Number,:Invert);
if (ProcessOracleErrorCode ("INSERT INVERT") == FALSE);
}

```

void SaveNumberConstriant (int Number)

```

{
    int i,j;
    char Lower[2],min[30],max[30],Upper[2];

    i=0;
step1:
    i++;
    Lower[0] = TempBuffer[i];
    Lower[1] = '\0';
    i++;
    for (i=i,j=0;TempBuffer[i] != ',';i++,j++)
        min[j] = TempBuffer[i];
    min[j] = '\0';
    i++;
    for (i=i,j=0;TempBuffer[i] != ']' && TempBuffer[i] != '?';i++,j++)
        max[j] = TempBuffer[i];
    max[j] = '\0';
    Upper[0] = TempBuffer[i];
    Upper[1] = '\0';
    Attr_Number = Number;
    Istrcpy (Lower_Limit.arr,Lower);
    Lower_Limit.len = Istrlen (Lower_Limit.arr);
    Istrcpy (Min.arr,min);
    Min.len = Istrlen (Min.arr);
    Istrcpy (Max.arr,max);
    Max.len = Istrlen (Max.arr);
    Istrcpy (Upper_Limit.arr,Upper);
    Upper_Limit.len = Istrlen (Upper_Limit.arr);
    EXEC SQL INSERT INTO N_CONSTRAINT (ATTR_NUMBER,LOWER_LIMIT,MIN,MAX,UPPER_LIMIT)
    VALUES (:Attr_Number,:Lower_Limit,:Min,:Max,:Upper_Limit);
    if (ProcessOracleErrorCode ("INSERT N_CONSTRAINT") == FALSE);
    i++;
    if (TempBuffer[i] != '?')
        goto step1;
}

```

Filename: ERFIVEPC.RC

//Microsoft App Studio generated resource script  
//  
#include "resource.h"

#define APSTUDIO\_READONLY\_SYMBOLS  
//  
// Generated from the TEXTINCLUDE 2 resource.  
//

#define APSTUDIO\_HIDDEN\_SYMBOLS  
#include "windows.h"  
#undef APSTUDIO\_HIDDEN\_SYMBOLS

////////////////////////////////////  
#undef APSTUDIO\_READONLY\_SYMBOLS  
////////////////////////////////////

// Icon  
//  
ERFIVEPC ICON DISCARDABLE "ERFIVEPC.ICO"  
////////////////////////////////////

// Menu  
//

ERFIVEMENU MENU DISCARDABLE  
BEGIN

```

    POPUP "&File"
    BEGIN
        MENUITEM "&Open...",          IDM_OPEN
        MENUITEM "&Save",              IDM_SAVE
        MENUITEM "Save &As...",        IDM_SAVEAS
        MENUITEM "&Print",             IDM_PRINT
        MENUITEM SEPARATOR
        MENUITEM "E&xit",              IDM_EXIT
        MENUITEM SEPARATOR
        MENUITEM "About &ERFive...",    IDM_ABOUT
    END
    POPUP "&Process"
    BEGIN
        MENUITEM "Fi&r&rst Normal Form", IDM_FIRST
        MENUITEM "&B&oyce Codd Normal Form", IDM_BCNF

```

```

MENUITEM "Fi&th Normal Form",          IDM_FIFTH
END
POPUP "&Oracle"
BEGIN
MENUITEM "Log&in",                      IDM_LOGIN
MENUITEM "Log&out",                     IDM_LOGOUT
END
END

```

```

////////////////////////////////////

```

```

//
// Dialog
//
RENTITYBOX DIALOG DISCARDABLE 50, 60, 187, 54
STYLE DS_MODALFRAME | WS_POPUP
FONT 8, "System"
BEGIN
LTEXT          "Regular Entity Name:",-1,8,11,72,8,NOT WS_GROUP
EDITTEXT      IDC_EDIT,85,10,105,13,ES_OEMCONVERT
DEFPUSHBUTTON "Ok",IDOK,85,35,40,14
PUSHBUTTON    "Cancel",IDCANCEL,150,35,40,14
END

```

```

WENTITYBOX DIALOG DISCARDABLE 50, 60, 187, 54
STYLE DS_MODALFRAME | WS_POPUP
FONT 8, "System"
BEGIN
LTEXT          " Weak Entity Name:",-1,0,12,72,8,NOT WS_GROUP
EDITTEXT      IDC_EDIT,75,10,105,13,ES_OEMCONVERT
DEFPUSHBUTTON "Ok",IDOK,75,35,40,14
PUSHBUTTON    "Cancel",IDCANCEL,140,35,40,14
END

```

```

NRELATIONBOX DIALOG DISCARDABLE 50, 60, 197, 54
STYLE DS_MODALFRAME | WS_POPUP
FONT 8, "System"
BEGIN
LTEXT          "Normal Relation Name:",-1,3,11,77,8,NOT WS_GROUP
EDITTEXT      IDC_EDIT,85,10,105,13,ES_OEMCONVERT
DEFPUSHBUTTON "Ok",IDOK,85,35,40,14
PUSHBUTTON    "Cancel",IDCANCEL,150,35,40,14
END

```

```

IRELATIONBOX DIALOG DISCARDABLE 50, 60, 197, 54
STYLE DS_MODALFRAME | WS_POPUP
FONT 8, "System"
BEGIN
LTEXT          "Identify Relation Name:",-1,3,11,77,8,NOT WS_GROUP
EDITTEXT      IDC_EDIT,85,10,105,13,ES_OEMCONVERT
DEFPUSHBUTTON "Ok",IDOK,85,35,40,14
PUSHBUTTON    "Cancel",IDCANCEL,150,35,40,14
END

```

```

NATTRIBUTEBOX DIALOG DISCARDABLE 60, 60, 222, 67
STYLE DS_MODALFRAME | WS_POPUP
FONT 8, "System"
BEGIN
LTEXT          "Normal Attribute Name :",IDC_STATIC,5,5,115,8,NOT WS_GROUP
EDITTEXT      IDC_NAME,5,15,138,13,ES_AUTOHSCROLL | ES_OEMCONVERT
LTEXT          "Datatype :",IDC_STATIC,151,5,39,8
EDITTEXT      IDC_DATATYPE,150,15,65,13,ES_AUTOHSCROLL | ES_OEMCONVERT
LTEXT          "Domain Constraint :",IDC_STATIC,5,35,115,8,NOT WS_GROUP
EDITTEXT      IDC_DOMAIN,5,45,138,13,ES_AUTOHSCROLL | ES_OEMCONVERT
DEFPUSHBUTTON "Ok",IDOK,150,45,30,14
PUSHBUTTON    "Cancel",IDCANCEL,185,45,30,14
END

```

```

KATTRIBUTEBOX DIALOG DISCARDABLE 60, 60, 222, 67
STYLE DS_MODALFRAME | WS_POPUP
FONT 8, "System"
BEGIN
LTEXT          "Candidate Key Attribute Name :",IDC_STATIC,5,5,115,8,NOT WS_GROUP
EDITTEXT      IDC_NAME,5,15,138,13,ES_AUTOHSCROLL | ES_OEMCONVERT
LTEXT          "Datatype :",IDC_STATIC,151,5,39,8
EDITTEXT      IDC_DATATYPE,150,15,65,13,ES_AUTOHSCROLL | ES_OEMCONVERT
LTEXT          "Domain Constraint :",IDC_STATIC,5,35,115,8,NOT WS_GROUP
EDITTEXT      IDC_DOMAIN,5,45,138,13,ES_AUTOHSCROLL | ES_OEMCONVERT
DEFPUSHBUTTON "Ok",IDOK,150,45,30,14
PUSHBUTTON    "Cancel",IDCANCEL,185,45,30,14
END

```

```

PATATTRIBUTEBOX DIALOG DISCARDABLE 60, 60, 222, 67
STYLE DS_MODALFRAME | WS_POPUP
FONT 8, "System"
BEGIN
LTEXT          "Primary Key Attribute Name :",IDC_STATIC,5,5,115,8,NOT WS_GROUP
EDITTEXT      IDC_NAME,5,15,138,13,ES_AUTOHSCROLL | ES_OEMCONVERT
LTEXT          "Datatype :",IDC_STATIC,151,5,39,8
EDITTEXT      IDC_DATATYPE,150,15,65,13,ES_AUTOHSCROLL | ES_OEMCONVERT
LTEXT          "Domain Constraint :",IDC_STATIC,5,35,115,8,NOT WS_GROUP
EDITTEXT      IDC_DOMAIN,5,45,138,13,ES_AUTOHSCROLL | ES_OEMCONVERT
DEFPUSHBUTTON "Ok",IDOK,150,45,30,14
PUSHBUTTON    "Cancel",IDCANCEL,185,45,30,14
END

```

```

MVATTRIBUTEBOX DIALOG DISCARDABLE 60, 60, 222, 67
STYLE DS_MODALFRAME | WS_POPUP
FONT 8, "System"
BEGIN
LTEXT          "Multivalue Key Attribute Name :",IDC_STATIC,5,5,115,8,NOT WS_GROUP

```

เอกสารนี้เป็นเอกสารที่... อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EDITTEXT          IDC_NAME,5,15,138,13,ES_AUTOHSCROLL | ES_OEMCONVERT
LTEXT             "Datatype :",IDC_STATIC,151,5,39,8
EDITTEXT          IDC_DATATYPE,150,15,85,13,ES_AUTOHSCROLL | ES_OEMCONVERT
LTEXT             "Domain Constraint :",IDC_STATIC,5,35,115,8,NOT WS_GROUP
EDITTEXT          IDC_DOMAIN,5,45,138,13,ES_AUTOHSCROLL | ES_OEMCONVERT
DEFPUSHBUTTON    "Ok",IDOK,150,45,30,14
PUSHBUTTON       "Cancel",IDCANCEL,185,45,30,14

```

END

TEXTBOX DIALOG DISCARDABLE 50, 60, 157, 47

STYLE DS\_MODALFRAME | WS\_POPUP

FONT 8, "System"

BEGIN

```

LTEXT             "Text Name:",-1,3,10,42,10,NOT WS_GROUP
EDITTEXT          IDC_EDIT,45,10,105,12,ES_OEMCONVERT
DEFPUSHBUTTON    "Ok",IDOK,45,30,45,14
PUSHBUTTON       "Cancel",IDCANCEL,105,30,45,14

```

END

ABOUTBOX DIALOG DISCARDABLE 22, 17, 147, 83

STYLE DS\_MODALFRAME | WS\_CAPTION | WS\_SYSMENU

CAPTION "About ERFive"

FONT 8, "System"

BEGIN

```

CTEXT             "(B.Eng.,M.Eng. KMITL)",-1,0,40,144,8
CTEXT             "For Microsoft Windows in Standard Mode",-1,0,21,144,8
CTEXT             "Entity Relationship Model",-1,0,10,144,8
CTEXT             "Developed By Twatchai Suttosatham",-1,0,31,144,8
DEFPUSHBUTTON    "OK",IDOK,50,70,45,14,WS_GROUP
ICON              "ERFIVEPC",-1,15,61,18,20
CTEXT             "Version 1.0",-1,50,52,45,8

```

END

ABORTDLG DIALOG DISCARDABLE 20, 20, 90, 64

STYLE DS\_MODALFRAME | WS\_CAPTION | WS\_SYSMENU

CAPTION "PmtFile"

BEGIN

```

DEFPUSHBUTTON    "Cancel",IDCANCEL,29,44,32,14,WS_GROUP
CTEXT             "Sending",-1,0,8,90,8
CTEXT             "Text",IDC_FILENAME,0,18,90,8
CTEXT             "to print spooler",-1,0,28,90,8

```

END

FDBOX DIALOG DISCARDABLE 50, 50, 301, 175

STYLE DS\_MODALFRAME | WS\_POPUP | WS\_VISIBLE | WS\_CAPTION

CAPTION "Functional Dependency Input"

FONT 8, "MS Sans Serif"

BEGIN

```

PUSHBUTTON       "RUN",IDC_RUN,240,135,50,14
PUSHBUTTON       "Cancel",IDC_CANCEL,240,115,50,14,WS_DISABLED
PUSHBUTTON       "INSERT",IDC_INSERT,240,75,50,14
PUSHBUTTON       "DELETE",IDC_DELETE,240,55,50,14,WS_DISABLED
PUSHBUTTON       "SELECT",IDC_SELECT,240,15,50,14
PUSHBUTTON       "FETCH",IDC_FETCH,240,35,50,14,WS_DISABLED
LISTBOX          IDC_RHS,125,15,105,135,LBS_SORT | LBS_MULTIPLESEL | LBS_NOINTEGRALHEIGHT |
                 WS_VSCROLL | WS_TABSTOP
LTEXT             "Right Hand Side :",IDC_STATIC,125,5,80,8
PUSHBUTTON       "Confirm",IDC_CONFIRM,240,95,50,14,WS_DISABLED
LISTBOX          IDC_LHS,10,15,105,135,LBS_SORT | LBS_MULTIPLESEL |
                 LBS_NOINTEGRALHEIGHT | WS_VSCROLL | WS_TABSTOP
LTEXT             "Left Hand Side :",IDC_STATIC,10,5,80,8
COMBOBOX         IDC_TNAME,125,155,105,85,CBS_DROPDOWNLIST | CBS_SORT | WS_VSCROLL | WS_TABSTOP
RTEXT            "TABLE NAME :",IDC_STATIC,85,155,50,10
DEFPUSHBUTTON    "EXIT",IDC_EXIT,240,155,50,14

```

END

JDBOX DIALOG DISCARDABLE 50, 50, 186, 175

STYLE DS\_MODALFRAME | WS\_POPUP | WS\_VISIBLE | WS\_CAPTION

CAPTION "Join Dependency Input"

FONT 8, "MS Sans Serif"

BEGIN

```

PUSHBUTTON       "RUN",IDC_RUN,125,135,50,14
PUSHBUTTON       "Cancel",IDC_CANCEL,125,115,50,14,WS_DISABLED
PUSHBUTTON       "INSERT",IDC_INSERT,125,75,50,14
PUSHBUTTON       "DELETE",IDC_DELETE,125,55,50,14,WS_DISABLED
PUSHBUTTON       "SELECT",IDC_SELECT,125,15,50,14
PUSHBUTTON       "FETCH",IDC_FETCH,125,35,50,14,WS_DISABLED
LISTBOX          IDC_ATTR,10,15,105,125,LBS_SORT | LBS_MULTIPLESEL |
                 LBS_NOINTEGRALHEIGHT | WS_VSCROLL | WS_TABSTOP
LTEXT             "ATTRIBUTE:",IDC_STATIC,10,5,80,8
PUSHBUTTON       "Confirm",IDC_CONFIRM,125,95,50,14,WS_DISABLED
COMBOBOX         IDC_TNAME,10,157,105,85,CBS_DROPDOWNLIST | CBS_SORT | WS_VSCROLL | WS_TABSTOP
RTEXT            "TABLE NAME :",IDC_STATIC,10,145,50,10
DEFPUSHBUTTON    "EXIT",IDC_EXIT,125,155,50,14

```

END

#ifdef APSTUDIO\_INVOKED

////////////////////////////////////

//

// TEXTINCLUDE

//

1 TEXTINCLUDE DISCARDABLE

BEGIN

"resource.h"

END

2 TEXTINCLUDE DISCARDABLE

BEGIN

#define APSTUDIO\_HIDDEN\_SYMBOLS\n"

#include "windows.h"\n"

เอกสารนี้เป็นทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
"#undef APSTUDIO_HIDDEN_SYMBOLS\n"
"\"
END
```

```
3 TEXTINCLUDE DISCARDABLE
BEGIN
"\\n"
"\"
END
```

```
////////////////////////////////////
#endif // APSTUDIO_INVOKED
////////////////////////////////////
//
// Cursor
//
TOOLCURS    CURSOR DISCARDABLE    "TOOLCURS.CUR"
PENCIL      CURSOR DISCARDABLE    "PENCIL.CUR"
ASCII       CURSOR DISCARDABLE    "ASCII.CUR"
INDICATE    CURSOR DISCARDABLE    "INDICATE.CUR"
```

```
#ifndef APSTUDIO_INVOKED
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//
////////////////////////////////////
#endif // not APSTUDIO_INVOKED
```

Filename: SQLPROTO.H

Copyright (c) 1988 by Oracle Corporation \*

NAME

sqlproto.h
FUNCTION
Contains full function prototypes for sql\* routines generated by Pro\*C 1.3
NOTES

To use these prototypes, just #include it before the EXEC SQL DECLARE SECTION in a Pro\*C program to be precompiled with 1.3.17 or later of Pro\*C. This include file defines SQLPROTO so that code generated with 1.3.17 or later of Pro\*C will not redefine these functions. Code generated with 1.3.16 or earlier of Pro\*C should not have this file included unless the external definitions in the C file are commented out.

MODIFIED
Criswell 03/17/91 - Add sqlgri
Cotello 04/18/90 - MSDOS: creation
\*/

```
#ifndef SQLPROTO
#define SQLPROTO
```

```
extern void sqlab2 (int *,char [],unsigned long[],unsigned short[],short[]);
extern void sqlad2 (int*,char[],unsigned long[],unsigned short[],short[]);
extern void sqlbcc (int*,int*);
extern void sqlbs2 (int*,unsigned char[],unsigned long[],unsigned short[],unsigned short*,int*,int*);
extern void sqlcls (int*);
extern void sqlcom (int*);
extern void sqlcxe (int*);
extern void sqlfcc (void);
extern void sqlfch (int*,int*);
extern void sqlgb2 (unsigned int*,unsigned int*,char *,unsigned short[],unsigned short[],char *,unsigned short[],unsigned short[],unsigned short[]);
extern void sqlgd2 (unsigned int*,int*,char *,unsigned short[],unsigned short[],unsigned long[],unsigned short[]);
extern void sqlg2 (int*,unsigned char**,unsigned long*,unsigned short*,int*,unsigned char**,unsigned long*,unsigned short*);
extern void sqlgri (int*,char *[]);
extern void sqlgem (char *,int*);
extern void sqllo2 (int*,unsigned char[],unsigned long[],unsigned short[],unsigned long*,int*,int*,unsigned long*);
extern void sqloca (struct oraca *,char *,int*,int*);
extern void sqlopn (int*,int*,unsigned long*);
extern void sqlos2 (int*,unsigned char[],unsigned long[],unsigned short[],unsigned short[]);
extern void sqlosq (char *,int*);
extern void sqlpf2 (int*,unsigned char**,unsigned long*,unsigned short*,int*,char **,unsigned long*,unsigned short*);
extern void sqlrl (int*);
extern void sqlsc (struct sqlca *);
extern void sqlsc2 (struct sqlca *);
extern void sqlsc (int*);
extern void sqlsch (int*);
extern void sqlsqs (char *,unsigned int*);
extern void sqlti (int*,int*);
extern void sqltoc (int*,int*,unsigned long*);
extern void sqlwmr (void);
#endif
```

Filename: ERFIVEPC.ICO \* binary file \*

00 00 01 00 01 00 20 20 10 00 00 00 00 00 E8 0200 00 18 00 00 00 28
00 00 00 20 00 00 00 40 00 00 00 01 00 04 00 00 00 00 00 00 00 00
00 00
00 BF 00 00 BF 00 00 00 BF BF 00 00 00 BF 00 00 BF 00 BF BF BF
00 C0 C0 C0 00 80 80 80 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 FF
00 00 00 FF 00 FF 00 FF FF 00 00 FF FF FF 00 00 00 00 00 00 00 00
00 00 00 00 00 00 0A A0 09 00 00 0E 00 00 0A A0 99 90 00 0E 00 00 AA A0
0A A0 09 00 00 EE E0 00 A0 00 00 00 00 00 00 0A A0 09 00 00 0E 00
E0 00 A0 00 00 00 00 00 00 00 0A A0 09 00 00 E0 E0 00 AA A0 00 00
00 00 00 0A A0 09 00 0E E0 EE 00 A0 00 00 00 00 0A A0 09 00 0E 00
09 00 0E 00 0E 00 A0 00 00 00 00 00 00 0A A0 09 00 00 00 00 0A A0
A0 00 00 00 00 00 00 0A A0 99 90 0E 00 0E 00 AA A0 00 00 00 00
00 00 0A A0 00 00 00 00 00 00 00 00 00 00 0A A0 AA AA AA AA AA AA
00 00 00 00 0A AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ที่สำนักงานกฤษฎีกาทำขึ้นโดยไม่ขึ้นกับด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0A A0 00 00 00 00 00 00 00 00 00 00 00 00 0A A0 00 00 00 00  
00 00 00 00 00 00 00 00 00 00 0A AA AA AA AA AA AA AA AA AA AA  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 0E EE EE EE EE E0 09 90 00  
09 90 00 00 00 00 00 0E EE EE EE E0 09 90 00 09 90 00 0E E0 00 00  
00 00 0E E0 00 00 00 09 90 00 99 00 00 00 00 00 00 00 00 00 0E E0  
00 09 90 00 99 00 00 00 00 00 00 00 00 0E E0 00 00 00 09 90 00 99 00  
00 00 00 00 00 00 0E EE EE EE E0 09 99 99 00 00 00 00 00 00 00 0E E0  
0E EE EE EE E0 09 99 99 99 00 00 0E E0 00 00 00 00 09 90 00 09 90 00  
00 00 09 90 00 0E E0 00 00 00 09 90 00 09 90 00 00 00 00 00 0E EE  
EE EE E0 09 99 99 99 00 00 00 00 00 00 00 00 0E EE EE EE E0 09 99 99  
90 00  
00  
00  
00  
00  
00  
00 00

Filename: TOOLCURS.CUR / binary file %

00 00 02 00 01 00 20 20 00 00 00 00 00 00 30 01 00 00 16 00 00 00  
28 00 00 00 00 20 00 00 00 40 00 00 00 01 00 01 00 00 00 00 00 00  
00  
FF FF FF 00  
00  
00 00 07 FF 00 00 0F FF 80 00 1F FF 80 00 1F FF 80 00 1B FF 80 00  
1B FF 80 00 1B 8D 80 00 03 8D 80 00 03 8D 80 00 03 8D 80 00 03 8D  
00 00 03 00 00 03 00 00 03 00 00 03 00 00 03 00 00 00 00 00 00 00  
FF  
FF  
FF FF FF FF FF FF FF FF FF F8 00 7F FF F8 00 7F FF F8 00 7F FF F8  
7F FF E0 00 3F FF C0 00 3F FF C0 00 3F FF C0 00 3F FF C0 00 3F FF  
C0 00 3F FF C0 00 3F FF F8 00 3F FF F8 00 3F FF F8 00 3F FF F8 7F  
FF FF F8 7F FF FF F8 7F FF FF F8 7F FF FF FC FF FF FF FF

Filename: ASCII.CUR binary file %

00 00 02 00 01 00 20 20 00 00 07 00 07 00 30 01 00 00 16 00 00 00  
28 00 00 00 00 20 00 00 00 40 00 00 00 01 00 01 00 00 00 00 00 00  
00  
FF FF FF 00  
00 60 1C 00 00 20 38 00 00 30 38 00 00 1F F0 00 00 18 70 00 00 08  
E0 00 00 0C E0 00 00 05 C0 00 00 07 C0 00 03 83 80 00 02 83 80 00  
02 81 00 00 02 80 00 00 3E F8 00 00 21 08 00 00 3E F8 00 00 02 80  
00 00 02 80 00 00 02 80 00 00 03 80 00 00 00 00 00 00 00 00 00 00  
FF  
FF  
FF  
FC 7F FF FF C0 07 FF FF C1 07 FF FF C0 07 FF FF FC 7F FF FF FC 7F  
FF FF FC 7F FF FF FC 7F FF FF FF FF FF FF FF FF FF FF FF FF FF FF

Filename: INDICATE.CUR / binary file %

00 00 02 00 01 00 20 20 00 00 08 00 08 00 30 01 00 00 16 00 00 00  
28 00 00 00 00 20 00 00 00 40 00 00 00 01 00 01 00 00 00 00 00 00  
00  
FF FF FF 00  
00 00 00 00 00 18 00 00 00 24 00 00 00 24 00 00 08 48 00 00 05 48  
00 00 04 90 00 00 04 1E 00 00 04 02 00 00 04 04 00 00 04 08 00 00  
04 10 00 00 04 20 00 00 04 40 00 00 04 80 00 00 05 00 00 00 08 00  
00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
FF  
FF  
FF E7 FF FF FF C3 FF FF FF C3 FF FF F9 87 FF FF F8 87 FF FF F8 0F  
FF FF F8 01 FF FF F8 01 FF FF F8 03 FF FF F8 07 FF FF F8 0F FF FF  
F8 1F FF FF F8 3F FF FF F8 7F FF FF F8 FF FF F9 FF FF F9 FF FF FB  
FF FF

Filename: PENCIL.CUR / binary file %

00 00 02 00 01 00 20 20 00 00 08 00 08 00 30 01 00 00 16 00 00 00  
28 00 00 00 00 20 00 00 00 40 00 00 00 01 00 01 00 00 00 00 00 00  
00  
FF FF FF 00  
01 14 00 00 01 3A 00 00 01 75 00 00 00 BA 80 00 00 5D 40 00 00 2E  
A0 00 00 17 50 00 00 0B A8 00 00 05 D4 00 00 02 EA 00 00 01 75 00  
00 00 AA 80 00 00 55 40 00 00 2B A0 00 00 17 A0 00 00 0B 40 00 00  
04 80 00 00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
FF  
FF  
FE 01 FF FF FE 00 FF FF FF 00 7F FF FF 80 3F FF FF CO 1F FF FF E0  
0F FF FF F0 07 FF FF F8 03 FF FF CO 01 FF FF FE 00 FF FF FF 00 7F  
FF FF 80 3F FF FF CO 1F FF FF E0 1F FF FF F0 3F FF FF F8 7F FF FF  
FC FF

Filename: ERINIT.PC

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ภายใต้การใช้งาน ไม่อนุญาตให้แก้ไขหรือดัดแปลง/ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
```

```
EXEC SQL BEGIN DECLARE SECTION ;
  VARCHAR uid[20];
  VARCHAR pwd[20];
EXEC SQL END DECLARE SECTION ;
EXEC SQL INCLUDE SQLCA ;
main ()
```

```
{
  strcpy (uid.arr,"SYSTEM");
  uid.len = strlen (uid.arr);
  strcpy (pwd.arr,"MANAGER");
  pwd.len = strlen (pwd.arr);
  EXEC SQL WHENEVER SQLERROR GOTO errout;
  EXEC SQL CONNECT :uid IDENTIFIED BY :pwd ;
  EXEC SQL GRANT CONNECT, RESOURCE TO ERRE IDENTIFIED BY KMRE ;
  EXEC SQL COMMIT WORK RELEASE;
  strcpy (uid.arr,"ERRE");
  uid.len = strlen (uid.arr);
  strcpy (pwd.arr,"KMRE");
  pwd.len = strlen (pwd.arr);
  EXEC SQL CONNECT :uid IDENTIFIED BY :pwd ;
  EXEC SQL CREATE TABLE ENTITY
  (
    ENTITY_NAME CHAR(30) NOT NULL,
    ENTITY_TYPE NUMBER
  );
  EXEC SQL CREATE UNIQUE INDEX ENTITY_INDEX ON ENTITY (ENTITY_NAME) ;

  EXEC SQL CREATE TABLE RELATION
  (
    RELATION_NAME CHAR(30) NOT NULL,
    RELATION_TYPE NUMBER
  );
  EXEC SQL CREATE UNIQUE INDEX RELATION_INDEX ON RELATION (RELATION_NAME) ;

  EXEC SQL CREATE TABLE ROLE
  (
    ROLE_NUMBER NUMBER NOT NULL,
    PARTICIPATION NUMBER,
    CADINALITY NUMBER,
    ENTITY_NAME CHAR(30),
    RELATION_NAME CHAR(30)
  );
  EXEC SQL CREATE UNIQUE INDEX ROLE_INDEX ON ROLE (ROLE_NUMBER) ;

  EXEC SQL CREATE TABLE ATTR
  (
    ATTR_NUMBER NUMBER NOT NULL,
    ATTR_NAME CHAR(30),
    ATTR_TYPE NUMBER,
    ATTR_DATATYPE CHAR(20),
    ENTITY_NAME CHAR(30),
    RELATION_NAME CHAR(30),
    COMPOSIT_NUMBER NUMBER
  );
  EXEC SQL CREATE UNIQUE INDEX ATTR_INDEX ON ATTR (ATTR_NUMBER) ;

  EXEC SQL CREATE TABLE ATTR_OF_INDEX
  (
    INDEX_NUMBER NUMBER NOT NULL,
    ATTR_NUMBER NUMBER NOT NULL
  );
  EXEC SQL CREATE UNIQUE INDEX ATTR_OF_INDEX_INDEX ON ATTR_OF_INDEX (INDEX_NUMBER,ATTR_NUMBER) ;

  EXEC SQL CREATE TABLE TABLE_OF_INDEX
  (
    INDEX_NUMBER NUMBER NOT NULL,
    TABLE_NAME CHAR(30)
  );
  EXEC SQL CREATE UNIQUE INDEX TABLE_OF_INDEX_INDEX ON TABLE_OF_INDEX (INDEX_NUMBER) ;

  EXEC SQL CREATE TABLE ATTR_OF_TABLE
  (
    TABLE_NAME CHAR(30) NOT NULL,
    ATTR_NUMBER NUMBER NOT NULL,
    NOT_NULL CHAR(1)
  );
  EXEC SQL CREATE UNIQUE INDEX ATTR_OF_TABLE_INDEX ON ATTR_OF_TABLE (TABLE_NAME,ATTR_NUMBER) ;

  EXEC SQL CREATE TABLE ATTR_DETAIL
  (
    ATTR_NUMBER NUMBER NOT NULL,
    ATTR_NAME CHAR(30) NOT NULL,
    DATATYPE CHAR(20)
  );
  EXEC SQL CREATE UNIQUE INDEX ATTR_DETAIL_INDEX ON ATTR_DETAIL (ATTR_NUMBER) ;

  EXEC SQL CREATE TABLE FD
  (
    FD_NUMBER NUMBER NOT NULL,
    TABLE_NAME CHAR(30)
  );
  EXEC SQL CREATE UNIQUE INDEX FD_INDEX ON FD (FD_NUMBER) ;

  EXEC SQL CREATE TABLE LHS
  (
    FD_NUMBER NUMBER NOT NULL,
```

```

        LHS_NUMBER    NUMBER    NOT NULL
    );
    EXEC SQL CREATE UNIQUE INDEX LHS_INDEX ON LHS (FD_NUMBER,LHS_NUMBER) ;

EXEC SQL CREATE TABLE RHS
(
    FD_NUMBER    NUMBER    NOT NULL,
    RHS_NUMBER   NUMBER    NOT NULL
);
EXEC SQL CREATE UNIQUE INDEX RHS_INDEX ON RHS (FD_NUMBER,RHS_NUMBER) ;

EXEC SQL CREATE TABLE JD
(
    JD_NUMBER    NUMBER    NOT NULL,
    TABLE_NAME  CHAR(30)  NOT NULL
);
EXEC SQL CREATE UNIQUE INDEX JD_INDEX ON JD (JD_NUMBER,TABLE_NAME) ;

EXEC SQL CREATE TABLE ATTR_OF_JD
(
    JD_NUMBER    NUMBER    NOT NULL,
    ATTR_NUMBER   CHAR(30)  NOT NULL
);
EXEC SQL CREATE UNIQUE INDEX ATTR_OF_JD_INDEX ON ATTR_OF_JD (JD_NUMBER,ATTR_NUMBER) ;

EXEC SQL CREATE TABLE N_CONSTRAINT
(
    ATTR_NUMBER   NUMBER    NOT NULL,
    LOWER_LIMIT  CHAR(1)   NOT NULL,
    MIN           CHAR(30)  NOT NULL,
    MAX           CHAR(30)  NOT NULL,
    UPPER_LIMIT  CHAR(1)   NOT NULL
);
EXEC SQL CREATE UNIQUE INDEX N_CONSTRAINT_INDEX
ON N_CONSTRAINT (ATTR_NUMBER,LOWER_LIMIT,MIN,MAX,UPPER_LIMIT) ;

EXEC SQL CREATE TABLE C_CONSTRAINT
(
    ATTR_NUMBER   NUMBER    NOT NULL,
    DOMAIN_NAME   CHAR(30)  NOT NULL
);
EXEC SQL CREATE UNIQUE INDEX C_CONSTRAINT_INDEX ON C_CONSTRAINT (ATTR_NUMBER,DOMAIN_NAME) ;

EXEC SQL CREATE TABLE INVERT
(
    ATTR_NUMBER   NUMBER    NOT NULL,
    INVERT        CHAR(1)   NOT NULL
);
EXEC SQL CREATE UNIQUE INDEX INVERT_INDEX
ON INVERT (ATTR_NUMBER,INVERT) ;

EXEC SQL CREATE TABLE KILL
(
    ATTR_NUMBER   NUMBER    NOT NULL
);

EXEC SQL CREATE TABLE TEMP_ENTITY           AS SELECT * FROM ENTITY ;
EXEC SQL CREATE TABLE TEMP_RELATION        AS SELECT * FROM RELATION ;
EXEC SQL CREATE TABLE TEMP_ATTR            AS SELECT * FROM ATTR ;
EXEC SQL CREATE TABLE TEMP_ROLE            AS SELECT * FROM ROLE ;
EXEC SQL CREATE TABLE TEMP_FD              AS SELECT * FROM FD ;
EXEC SQL CREATE TABLE TEMP_LHS             AS SELECT * FROM LHS ;
EXEC SQL CREATE TABLE TEMP_RHS             AS SELECT * FROM RHS ;
EXEC SQL CREATE TABLE TEMP_ATTR_OF_INDEX   AS SELECT * FROM ATTR_OF_INDEX ;
EXEC SQL CREATE TABLE TEMP_TABLE_OF_INDEX AS SELECT * FROM TABLE_OF_INDEX ;
EXEC SQL CREATE TABLE TEMP_ATTR_OF_TABLE   AS SELECT * FROM ATTR_OF_TABLE ;

EXEC SQL COMMIT WORK RELEASE ;
errexit:
errrpt():
EXEC SQL WHENEVER SQLERROR CONTINUE;
EXEC SQL ROLLBACK WORK RELEASE;
return(1);
}

void errrpt()
{
    printf("%7Ds (%d)\n", sqlca.sqlerrm.sqlerrmc, -sqlca.sqlcode);
    return(0);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ประวัติผู้เขียน

ชื่อผู้เขียน	นายธวัชชัย สุทธิทศธรรม
วันเดือนปีเกิด	วันที่ 27 ธันวาคม พ.ศ. 2509
สถานที่เกิด	จังหวัดกรุงเทพมหานคร
บิดาชื่อ	นายสุทธิ สุทธิทศธรรม
มารดาชื่อ	นางกัญญารัตน์ สุทธิทศธรรม
วุฒิการศึกษาระดับประถม	โรงเรียนวัดใหม่ช่องลม
วุฒิการศึกษาระดับมัธยม	โรงเรียนปทุมคงคา
วุฒิการศึกษาระดับปริญญาตรี	วิศวกรรมศาสตรบัณฑิต สาขาวิชา วิศวกรรมไฟฟ้า ภาควิชาวิศวกรรม อิเล็กทรอนิกส์ สถาบันเทคโนโลยีพระ จอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2530
ผลงานการศึกษาระดับปริญญาตรี	เคยถวายผลงานวิจัยทางคอมพิวเตอร์ กราฟฟิก แต่พระบาทสมเด็จพระเจ้าอยู่ หัว ในงานพระจอมเกล้าลาดกระบัง นิทรรศ 30 ขณะศึกษาอยู่ชั้นปีที่ 4
ผลงานทางวิชาการที่ได้รับการตีพิมพ์	เรื่องระบบอัตโนมัติช่วยในการออกแบบ ฐานข้อมูลแบบรีเลชันแนล ในการประชุม วิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 16 เรื่องระบบอัตโนมัติช่วยออกแบบฐานข้อมูล รีเลชันแนล ในวารสารคอมพิวเตอร์ของ สมาคมคอมพิวเตอร์แห่งประเทศไทย ใน พระบรมราชูปถัมภ์ ปีที่ 20 ฉบับที่ 103 กันยายน-ตุลาคม 2536
ประสบการณ์การทำงาน	วิศวกรระดับ 4 ฝ่ายฝึกอบรม กองฝึก อบรมอุปกรณ์ การไฟฟ้าฝ่ายผลิตแห่ง ประเทศไทย เมื่อปี 2531 ถึง 2533

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้