

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาโปรแกรมสำเร็จรูปสำหรับการออกแบบฐานข้อมูล

โดย



นางสาวนงค์นุช	ชยาติวณีย์	รหัส 33501013
นางสาวนรรัตน์	หอมวิเศษ	รหัส 33501014
นางสาวปรีนดา	ศิริชา	รหัส 33501017
นายอััจฉ์	ธนประสิทธิ์พัฒนา	รหัส 33501040

ปัญหาพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ปพ.  
๒๕๓๖

๒๕๓๖

เลขหมู่.....

เลขทะเบียน.....

วัน,เดือน,ปี.....

6/2533683

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2536

**SOFTWARE DEVELOPMENT FOR DATABASE DESIGN**

**BY**

<b>MISS NONGNUCH</b>	<b>KAYATIVANAI</b>	<b>CODE 33501013</b>
<b>MISS NAWARAT</b>	<b>HOMVISASE</b>	<b>CODE 33501014</b>
<b>MISS PARINDA</b>	<b>SIRICHA</b>	<b>CODE 33501017</b>
<b>MR. ATCH</b>	<b>THANAPRASITPATTANA</b>	<b>CODE 33501040</b>

**A Special Project Submitted in Partial Fulfillment  
of the Requirement for the Degree of Bachelor of Science**

**Department of Mathematics and Computer Science  
Faculty of Science  
King Mongkut's Institute of Technology Ladkrabang**

**1993**

ชื่อหัวข้อปัญหาพิเศษ การพัฒนาโปรแกรมสำเร็จรูปสำหรับการออกแบบฐานข้อมูล

โดย นางสาวนงคฺ์นุช ขยาติวน้อย รหัส 33501013  
นางสาวนวรรตน์ หอมวิเศษ รหัส 33501014  
นางสาวปรินดา ศิริชา รหัส 33501017  
นายอัจจ์ ธนประสิทธิ์พัฒนา รหัส 33501040

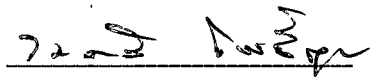
ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์วีระ บุญจริง  
อาจารย์สิริลักษณ์ เตียพิริยะกิจ

ภาควิชาคณิตศาสตร์ประยุกต์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

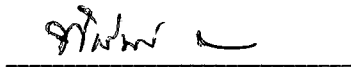
อนุมัติให้นำโครงการพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต



(รศ.ดร.ไมตรี โปธิ์สุข)

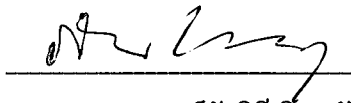
หัวหน้าภาควิชาฯ

คณะกรรมการโครงการพิเศษ



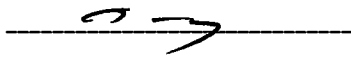
(ผู้ช่วยศาสตราจารย์พัชรินทร์ เหมชาติ)

ประธานกรรมการ



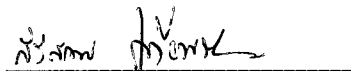
(รองศาสตราจารย์ภัคคินี ยิมเรวัตติ)

กรรมการ



(อาจารย์วีระ บุญจริง)

อาจารย์ที่ปรึกษาปัญหาพิเศษ



(อาจารย์สิริลักษณ์ เตียพิริยะกิจ)

อาจารย์ที่ปรึกษาปัญหาพิเศษ

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

### บทคัดย่อ

**ชื่อหัวข้อพิเศษ** การพัฒนาโปรแกรมสำเร็จรูปสำหรับการออกแบบฐานข้อมูล

<b>นักศึกษา</b>	นางสาวนงค์นุช ขยาติว নয়	รหัส 33501013
	นางสาวนวรรตน์ หอมวิเศษ	รหัส 33501014
	นางสาวปรินดา ศิริชา	รหัส 33501017
	นายอัฉน์ ธนประสิทธิ์พัฒนา	รหัส 33501040

<b>อาจารย์ที่ปรึกษา</b>	อาจารย์วิระ บุญจริง
	อาจารย์สิริลักษณ์ เตียพิริยะกิจ

<b>ภาควิชา</b>	คณิตศาสตร์และวิทยาการคอมพิวเตอร์
----------------	----------------------------------

<b>ปีการศึกษา</b>	2536
-------------------	------

### บทคัดย่อ

โครงการนี้มีวัตถุประสงค์เพื่อ พัฒนาโปรแกรมสำหรับการออกแบบฐานข้อมูล กระบวนการพัฒนาเริ่มจากการศึกษาทฤษฎีการออกแบบฐานข้อมูล การออกแบบซอฟต์แวร์และการพัฒนาโปรแกรม ผลจากโครงการนี้ จะได้โปรแกรมที่สามารถใช้ในการออกแบบฐานข้อมูลที่ข้อมูลอยู่ในรูปแบบนอร์มัลระดับบอยส์คอตต์

## ABSTRACT

The purpose of this project is aimed at development of database design software. The development process starts with studying database design theories, designing software and developing programs. The results program can be used to design database in Boyce-Codd normal form,

### **กิติกรรมประกาศ**

คณะผู้จัดทำปัญหาพิเศษฉบับนี้ขอขอบพระคุณอาจารย์ทุก ๆ ท่านที่ได้ช่วยให้คำแนะนำแนวทางต่าง ๆ ในการทำปัญหาพิเศษ โดยเฉพาะอย่างยิ่ง อาจารย์วีระ บุญจริง และ อาจารย์สิริลักษณ์ เตียพิริยะกิจ ซึ่งเป็นอาจารย์ที่ปรึกษาปัญหาพิเศษ ที่ได้กรุณาให้คำปรึกษาแนะนำในการทำปัญหาพิเศษฉบับนี้แก่คณะผู้จัดทำ ขอขอบคุณกรรมการทุกท่านที่ได้กรุณาใช้เวลาอันมีค่า เพื่อตรวจสอบปัญหาพิเศษฉบับนี้จนกระทั่งสำเร็จลงด้วยดี

**คณะผู้จัดทำ**

**สารบัญรูป**

	หน้า
รูปที่ 3.1 โครงสร้างโปรแกรมสำหรับการออกแบบฐานข้อมูล	9
รูปที่ 3.2 โครงสร้างโปรแกรมสำหรับการประมวลผล	9

## สารบัญ

	หน้า
บทคัดย่อปัญหาพิเศษภาษาไทย	ก
บทคัดย่อปัญหาพิเศษภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญรูป	ง
บทที่ 1 ความเป็นมาของปัญหา	
ความสำคัญและที่มาของปัญหา	1
วัตถุประสงค์ของปัญหา	1
ขอบเขตของปัญหา	1
ขั้นตอนการดำเนินงาน	1
ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีฐานข้อมูลรีเลชันนัล	
โมเดลข้อมูลแบบรีเลชันนัล(Relational Model)	3
กระบวนการนอร์มัลไลซ์(Normalization Process)	7
บทที่ 3 การออกแบบโปรแกรมสำเร็จรูปสำหรับการออกแบบฐานข้อมูล	
โครงสร้างระบบ	9
การออกแบบส่วนติดต่อกับผู้ใช้	13
การออกแบบแฟ้มข้อมูล	13
บทที่ 4 การประเมินผลระบบ	19
บทที่ 5 บทสรุปและข้อเสนอแนะ	20
ภาคผนวก	
บรรณานุกรม	

# บทที่ 1

## ความเป็นมาของปัญหา

### ความสำคัญและที่มาของปัญหา

ฐานข้อมูลจัดเป็นหัวใจของระบบงานคอมพิวเตอร์หากมีการจัดเก็บในฐานข้อมูลที่เหมาะสมจะทำให้ระบบงานคอมพิวเตอร์ประสบผลสำเร็จ การจัดเก็บข้อมูลที่เหมาะสมนั้นจะต้องมีการออกแบบการจัดเก็บที่ดีซึ่งจะต้องอาศัยหลักการและทฤษฎีสนับสนุน ในปัจจุบันมีโปรแกรมที่ช่วยการออกแบบฐานข้อมูลจำนวนมากแต่จะเป็นฐานข้อมูลขนาดใหญ่และมีราคาแพงอีกทั้งยังเป็นโปรแกรมที่สั่งมาจากต่างประเทศซึ่งส่งผลกระทบต่อดุลการค้าของประเทศ นอกจากนี้กลุ่มผู้พัฒนาระบบงานขนาดย่อมจัดเป็นกลุ่มพัฒนาระบบที่แพร่หลายที่สุด หากกลุ่มนี้มีการออกแบบฐานข้อมูลที่ดีโดยใช้เครื่องมือที่ดีจะส่งผลให้ระบบงานต่างๆประสบผลสำเร็จ ด้วยเหตุนี้จึงมีแนวคิดที่จะสร้างโปรแกรมสำเร็จรูปเพื่อช่วยในการออกแบบฐานข้อมูลขึ้นมา

### วัตถุประสงค์ของปัญหา

เพื่อจัดทำโปรแกรมสำเร็จรูปสำหรับออกแบบระบบฐานข้อมูล ที่สามารถใช้งานบนเครื่องไมโครคอมพิวเตอร์ได้

### ขอบเขตของปัญหา

การจัดทำโปรแกรมสำเร็จรูปสำหรับการออกแบบฐานข้อมูล โดยใช้ฟอกซ์โปรนี้ จำกัดการใช้เฉพาะฐานข้อมูลขนาดเล็กที่มีจำนวนแอตทริบิวต์(attribute) ไม่เกิน 20 ตัว และสามารถนอร์มัลไลซ์(normalize) ข้อมูลได้ถึงระดับบอยส์คอตต์(BoyceCodd) นอกจากนี้ยังสามารถแก้ไขชื่อตารางที่เป็นผลจากการนอร์มัลไลซ์เพื่อเตรียมพร้อมสำหรับการสร้างฐานข้อมูล

### ขั้นตอนการดำเนินงาน

- 1 ศึกษาทฤษฎีการออกแบบฐานข้อมูล
- 2 ศึกษาการเขียนโปรแกรมฟอกซ์โปร(Foxpro)
- 3 ออกแบบระบบ
- 4 พัฒนาระบบ
- 5 ทดสอบและปรับปรุงข้อผิดพลาดจากโปรแกรม
- 6 เขียนรายงานและคู่มือประกอบการใช้งาน

**ประโยชน์ที่คาดว่าจะได้รับ**

โปรแกรมสำเร็จรูปที่พัฒนาขึ้นนี้จะสามารถใช้เป็นเครื่องมือของนักวิเคราะห์ระบบในการออกแบบฐานข้อมูล ซึ่งสามารถใช้ร่วมกับเครื่องมืออื่นเป็นอย่างดี อีกทั้งสามารถนำผลการออกแบบไปสร้างฐานข้อมูลบนระบบจัดการฐานข้อมูลใด ๆ ได้

## บทที่ 2

# ทฤษฎีฐานข้อมูลรีเลชันนัล

### โมเดลข้อมูลแบบรีเลชันนัล (Relational Model)

โมเดลข้อมูลแบบรีเลชันนัล เป็นโมเดลฐานข้อมูลที่เป็นที่ใช้งานแพร่หลายมากที่สุดในบรรดาโมเดลฐานข้อมูลที่มีอยู่ในปัจจุบันวิธีการที่จะให้เห็นภาพของโมเดลนี้ที่ง่ายที่สุด คือการมองภาพว่าฐานข้อมูลนั้นถูกเก็บอยู่ในตารางสองมิติ

รีเลชัน(relation)หรือตาราง(table) คือ ตารางสองมิติที่คอลัมน์ต่างๆ ในรีเลชันมีความสัมพันธ์กัน และแต่ละแถวของตารางจะแสดงค่าข้อมูลในแต่ละคอลัมน์

เช่น ตารางการสั่งซื้อ

รหัสการสั่ง	วันที่สั่ง	รหัสสินค้า	ชื่อสินค้า	จำนวนที่สั่ง
12489	020931	AX12	เตารีด	11
12491	020931	BT04	เตา	1
		BZ66	จักรยายน	4
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

เขียนอยู่ในรูปแบบของรีเลชันได้เป็น การสั่งซื้อ(รหัสการสั่ง,วันที่สั่ง,รหัสสินค้า,ชื่อสินค้า,จำนวนที่สั่ง)

ทัพเพิล (tuple) คือ แถวของข้อมูลที่อยู่ในตาราง ข้อมูลในแต่ละแถวล้วนต้องมีความแตกต่างกัน ไม่มีข้อมูลในแถวใด ๆ ที่เหมือนกันทั้งหมดทีเดียว เพราะเท่ากับเป็นการซ้ำซ้อนโดยเปล่าประโยชน์ เช่น ทัพเพิล ในตารางสั่งซื้อ ทัพเพิลแรกคือ 12489,020931,AX12,เตารีด,11

แอตทริบิว (attribute)หรือคอลัมน์(column) คือ ข้อมูลที่อยู่ในแนวหลักของตาราง แต่ละหลักจะใช้แสดงลักษณะเฉพาะหัวข้อใดเพียงหัวข้อเดียวเท่านั้น และชื่อหัวข้อในแต่ละหลักภายใต้ตารางเดียวกันต้องแตกต่าง เช่น แอตทริบิวต์ต่าง ๆ ในตารางการสั่งซื้อ คือ รหัสการสั่งซื้อ,วันที่สั่ง,รหัสสินค้า,ชื่อสินค้า และจำนวนที่สั่ง

**ดีเทอร์มิแนนท์ (determinant)** คือ แอตทริบิวหรือกลุ่มของแอตทริบิวที่มีคุณสมบัติสามารถใช้เลือกค่าของแอตทริบิวอื่นได้เพียงหนึ่งค่าเสมอ เช่น รหัสสินค้าใช้บอกชื่อสินค้าได้ รหัสสินค้าจึงเป็นดีเทอร์มิแนนท์

**ฟังก์ชันการขึ้นต่อกัน (functional dependency, FD)** เรากล่าวว่า B มีฟังก์ชันการขึ้นต่อ A ถ้าค่าของ A เป็นดีเทอร์มิแนนท์(determinant) ใช้เลือกค่าของ B เราแทนด้วยสัญลักษณ์ " A -> B " เช่น รหัสสินค้า -> ชื่อสินค้า เป็นฟังก์ชันการขึ้นต่อกัน เพราะรหัสสินค้าใช้บอกชื่อสินค้าได้

**รีเลชันชิพ (relationship)** คือ ความสัมพันธ์ระหว่างรีเลชัน เนื่องจากข้อมูลที่ถูกเก็บอยู่ในฐานข้อมูลใดๆได้แก่ ข้อมูลของเหล่ารีเลชันที่มีความสัมพันธ์กัน และรีเลชันคู่ที่มีความสัมพันธ์กันก็จะมีบางแอตทริบิวที่เหมือนกันและใช้ค่าของแอตทริบิวในส่วนที่เหมือนกันเป็นตัวระบุข้อมูลในรีเลชันที่มีความสัมพันธ์กัน เช่น การสั่งซื้อสินค้า (รหัสการสั่ง,วันที่สั่ง,รหัสสินค้า) และสินค้า(รหัสสินค้า,ชื่อสินค้า) รหัสสินค้าจากรีเลชันการสั่งซื้อสินค้าจะเป็นตัวระบุ ชื่อสินค้าในรีเลชันสินค้า โดยรูปแบบของความสัมพันธ์อาจเป็นไปได้ดังนี้

- ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (one-to-one)
- ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (one-to-many)
- ความสัมพันธ์แบบกลุ่มต่อหนึ่ง (many-to-one)
- ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (many-to-many)

**คีย์หลัก (primary key)** ประกอบไปด้วยค่าของแอตทริบิวหนึ่งตัวหรือมากกว่านั้น ที่จะทำให้สามารถใช้เป็นตัวเจาะจงบอกได้ว่า เรากำลังอ้างอิงถึงทัพเพิลอันไหนของรีเลชัน เช่น สินค้า(รหัสสินค้า,ชื่อสินค้า,ราคาสินค้า) มีรหัสสินค้า เป็นคีย์หลักของรีเลชันเพราะใช้บอกชื่อและราคาสินค้าซึ่งเป็นข้อมูลทั้งหมดของทัพเพิลได้

**คีย์คู่แข่ง (candidate key)** คือ แอตทริบิวหรือกลุ่มของแอตทริบิวที่มีคุณสมบัติที่จะเป็นคีย์หลักได้ แต่ไม่ได้รับการคัดเลือกเป็นคีย์หลัก เช่น สินค้า(รหัสสินค้า,ชื่อสินค้า,ราคาสินค้า)เมื่อเราเลือกรหัสสินค้าเป็นคีย์หลัก ชื่อสินค้าซึ่งสามารถใช้บอก รหัสและราคาสินค้าซึ่งเป็นข้อมูลทั้งหมดของทัพเพิลได้เหมือนกัน ก็จะเป็นคีย์คู่แข่ง

**นันคีย์ (nonkey)** คือ แอตทริบิวที่ไม่เป็นส่วนใดส่วนหนึ่งของคีย์ เช่น สินค้า(รหัสสินค้า,ชื่อสินค้า,ราคาสินค้า) จะมีชื่อสินค้าและราคาสินค้าเป็นนันคีย์

**ฟังก์ชันการขึ้นต่อกันที่ซ้ำซ้อน (FD redundant)** คือ ฟังก์ชันการขึ้นต่อกันที่ดีเทอร์มิแนนท์และนันคีย์ของมันสามารถกล่าวถึงได้ โดยใช้ฟังก์ชันการขึ้นต่อกันอื่นที่มีอยู่ภายในฐานข้อมูล

**มินิมอลคอบเวอร์ (minimal cover)** คือ กลุ่มของฟังก์ชันการขึ้นต่อกัน ที่ไม่มีฟังก์ชันการขึ้นต่อกันที่ซ้ำซ้อนรวมอยู่

## รูปแบบนอร์มัล (Normal Form)

รูปแบบนอร์มัลระดับที่ 1 (First Normal Form, 1NF)

รีเลชันที่ไม่นอร์มัล(unnormalized relation)คือ รีเลชันที่มีข้อมูลในบางช่องมากกว่า 1 ค่า ดังนั้น การปรับปรุงให้รีเลชันมีรูปแบบนอร์มัลระดับที่ 1 ได้แก่การขจัดกลุ่มที่ซ้ำกัน (repeating group) โดยการแยกกลุ่มที่ซ้ำกันออกไปอยู่คนละแถว คีย์ตัวใหม่จะประกอบด้วยคีย์เดิมผนวกกับแอตทริบิวต์ถือเป็นคีย์หลักของกลุ่มที่ซ้ำ

เช่น การสั่งซื้อ

รหัสการสั่ง	รหัสสินค้า	ชื่อสินค้า	ราคา	จำนวนที่สั่ง
12489	AX12	เตารีด	280	11
12491	BT04	เตา	8000	1
	BZ66	เครื่องซักผ้า	6000	1
12494	CB03	จักรยาน	3500	4

จากตัวอย่างรีเลชันการสั่งซื้อ ในการสั่งแต่ละครั้งลูกค้าสามารถสั่งซื้อสินค้าได้มากกว่า 1 อย่าง จึงถือว่า"รหัสการสั่ง"เป็นกลุ่มที่ซ้ำกัน ถ้าแยกรหัสการสั่ง 12491 ออกเป็น 2 แถวก็จะได้รีเลชันที่เป็นรูปแบบนอร์มัลระดับที่ 1 ที่มีคีย์หลักประกอบด้วยรหัสการสั่ง และรหัสสินค้า ดังตารางต่อไปนี้

การสั่งซื้อ (1NF)

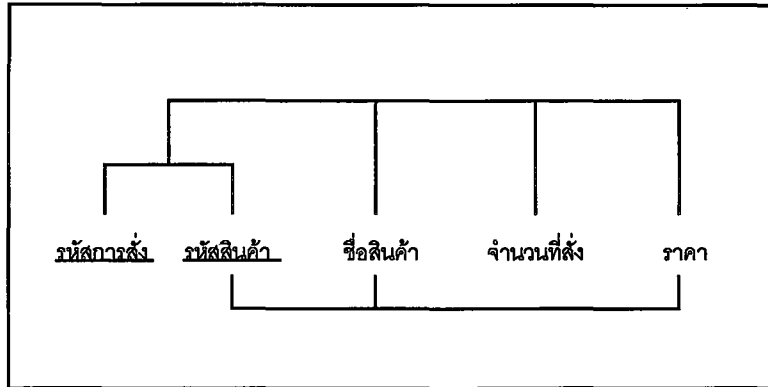
รหัสการสั่ง	รหัสสินค้า	ชื่อสินค้า	ราคา	จำนวนที่สั่ง
12489	AX12	เตารีด	280	11
12491	BT04	เตา	8000	1
12491	BZ66	เครื่องซักผ้า	6000	1
12494	CB03	จักรยาน	3500	4

การสั่งซื้อ(รหัสการสั่ง,รหัสสินค้า,ราคา,จำนวนที่สั่ง)

รูปแบบนอร์มัลระดับที่ 2 (Second Normal Form, 2NF)

แม้ว่าจะมีการปรับปรุงรีเลชันให้เป็นรูปแบบนอร์มัลระดับที่ 1 แล้วก็ตาม ถ้าในรีเลชันมี นันคีย์บางตัวขึ้นอยู่กับเพียงบางส่วนของคีย์ ก็จะทำให้เกิดความซ้ำซ้อนของแอตทริบิวต์นั้นในหลาย ทัพเพิล ซึ่งนอกจากจะเปลืองเนื้อที่โดยใช่เหตุแล้ว ยังทำให้เกิดปัญหาในการแก้ไขข้อมูล(update-anomaly) เพราะเราต้องตามไปแก้ไขข้อมูลในหลายแถว ซึ่งไม่สามารถรับประกันได้ว่าการแก้ไขเกิดขึ้น ทุกแถว เราจึงต้องขจัดการซ้ำกันบางส่วนนี้ ปรับให้เป็นรูปแบบนอร์มัลระดับที่ 2 โดยการแตกรีเลชันขึ้น

จึงต้องขจัดการซ้ำกันบางส่วนนี้ ปรับให้เป็นรูปแบบนอร์มัลระดับที่ 2 โดยการแตกรีเลชันขึ้นมาใหม่สำหรับฟังก์ชันการขึ้นกันที่เป็นปัญหา และกำหนดให้ตัวที่เป็นดีเทอมีแนนท์เป็นคีย์หลักของรีเลชันใหม่



#### ไดอะแกรมการขึ้นกัน

จากตัวอย่าง รีเลชันการสั่งซื้อ(รหัสการสั่ง,รหัสสินค้า,ราคา,จำนวนที่สั่ง) เมื่อพิจารณาร่วมกับไดอะแกรมการขึ้นกัน ชื่อและราคาสินค้า ขึ้นกับรหัสสินค้าซึ่งเป็นส่วนหนึ่งของคีย์หลัก จึงต้องแตกรหัสสินค้า,ชื่อสินค้า และราคาสินค้าออกสู่อรีเลชันใหม่ที่เป็นรูปแบบนอร์มัลระดับที่2 ดังนี้

การสั่งซื้อ(รหัสการสั่ง,รหัสสินค้า,จำนวนที่สั่ง)

สินค้า(รหัสสินค้า,ชื่อสินค้า,ราคา)

#### รูปแบบนอร์มัลระดับที่ 3 (Third Normal Form, 3NF)

เนื่องจากรูปแบบนอร์มัลระดับที่ 2 ไม่เพียงพอที่จะขจัดปัญหาการซ้ำซ้อนที่เนื่องมาจากการขึ้นต่อกันระหว่งนันคีย์ได้ จึงต้องปรับปรุงรีเลชันให้เป็นรูปแบบนอร์มัลระดับที่ 3 การทำรีเลชันให้เป็นรูปแบบนอร์มัลระดับที่ 3 ทำโดยการดึงนันคีย์คู่ที่เป็นปัญหาแตกออกสู่อรีเลชันใหม่ และกำหนดให้ตัวที่เป็นดีเทอมีแนนท์เป็นคีย์หลักของรีเลชันใหม่

จากตัวอย่างรีเลชันการสั่งซื้อที่เป็นนอร์มัลระดับที่ 2 และไดอะแกรมการขึ้นกัน ราคาสินค้าขึ้นกับชื่อสินค้า ซึ่งแอตทริบิวทั้งสองเป็นนันคีย์ทั้งคู่ จึงต้องแตกราคาสินค้าและรหัสสินค้าออกสู่อรีเลชันใหม่ที่เป็นรูปแบบนอร์มัลระดับที่ 3 ดังนี้

การสั่งซื้อ(รหัสการสั่ง,รหัสสินค้า,จำนวนที่สั่ง)

ชื่อ(รหัสสินค้า,ชื่อสินค้า), ราคา(ชื่อสินค้า,ราคาสินค้า)

รูปแบบนอร์มัลของบอยส์คอดด์(Boyce-Codd Normal Form, BCNF)

แม้ว่าปัญหาในการแก้ไขข้อมูลอันเนื่องมาจากนั้นคีย์จะถูกจัดให้หมดไป ในรูปแบบนอร์มัลระดับที่ 3 แล้วก็ตาม แต่ปัญหาในการแก้ไขข้อมูล (update anomaly) อันเนื่องมาจากคีย์หลักก็ยังคงอยู่ เพราะมี ดีเทอมีแนนท์บางตัวไม่เป็นคีย์คู่แข่ง การที่จะขจัดปัญหาเหล่านี้จะต้องปรับปรุงรีเลชันให้อยู่ในระดับสูงขึ้นไปเรียกว่า รูปแบบนอร์มัลของบอยส์คอดด์

### กระบวนการนอร์มัลไลซ์(Normalization Process)

การนอร์มัลไลซ์หรือวิธีการออกแบบฐานข้อมูล ในปัจจุบันทำได้ 2 ทาง คือ ดีคอมโพสิชัน (Decomposition) และซินทีซิส (Synthesis)

วิธีซินทีซิสตรงข้ามกับวิธีดีคอมโพสิชันเพราะการดีคอมโพสิชัน จะเริ่มโดยการรวมแอตทริบิวต์ทั้งหมดเป็นรีเลชันขนาดใหญ่ก่อน แล้วจึงแตกสู่เซตที่เล็กกว่า แต่สำหรับซินทีซิสจะนำฟังก์ชันการขึ้นต่อกันมา พิจารณาแบ่งแอตทริบิวต์ออกเป็นกลุ่มให้ภายในกลุ่มเดียวกันมีดีเทอมีแนนท์เหมือนกัน

### ดีคอมโพสิชัน

ขั้นตอนการทำงานเขียนเป็นอัลกอริทึม ได้ดังต่อไปนี้

1. สร้างรีเลชันที่ประกอบด้วยแอตทริบิวต์ทั้งหมด (Universal Relation, UR)
2. ขจัดความซ้ำซ้อนของฟังก์ชันการขึ้นต่อกัน(FD redundancy)เพื่อหามินิมอลคอบเวอร์ (Minimal Cover) โดยใช้วิธีเมมเบอร์ชิพ (Membership)

อัลกอริทึมของเมมเบอร์ชิพ เพื่อตรวจสอบฟังก์ชันการขึ้นต่อกัน :  $x \rightarrow Y$

1.  $T = \{x\}$
2. พิจารณาฟังก์ชันการขึ้นต่อกันที่เหลือว่า ดีเทอมีแนนท์ของมันอยู่ใน  $T$  หรือไม่ ?  
ใช่ ใส่มันคีย์ของฟังก์ชันการขึ้นต่อกันลงใน  $T$   
ไม่ใช่ ข้ามไปทำข้อ 4
3. พิจารณาว่า  $T$  เปลี่ยนแปลงหรือไม่ ?  
ใช่ กลับไปทำข้อ 2 ใหม่
4. พิจารณาว่า  $y$  เป็นสมาชิกของ  $T$  หรือไม่ ?  
ใช่  $x \rightarrow y$  เป็นฟังก์ชันการขึ้นต่อกันที่ซ้ำซ้อน

3. พิจารณาว่า รีเลชันที่ได้เป็นรูปแบบนอร์มัลของบอยส์คอตต์หรือไม่ ?

ไม่ , แต่กรีเลชันออกเป็นรีเลชันย่อย

4. นำรีเลชันใหม่ กลับไปวนทำ(3) จนกระทั่งทุกกรีเลชันเป็นรูปแบบนอร์มัลของบอยส์คอตต์ทั้งหมด

ขั้นที่สี่

ขั้นตอนการทำงานเขียนเป็นอัลกอริทึม ได้ดังต่อไปนี้

1. จัดแอดทริบิวพิเศษออกจากดีเทอร์มิแนนท์

2. จัดความซ้ำซ้อนของฟังก์ชันการขึ้นแก่กัน(FD redundancy)เพื่อหาขั้นต่ำมอลคอร์ดเวอร์ (Minimal Cover) โดยใช้วิธีเมมเบอร์ชิพ (Membership)

3. แบ่งฟังก์ชันการขึ้นต่อกันออกเป็นกลุ่ม ๆ โดยให้แต่ละกลุ่มมีดีเทอร์มิแนนท์เดียวกัน

4. นำแต่ละกลุ่มมาสร้างเป็นรีเลชัน

5. พิจารณาว่า รีเลชันที่ได้เป็นรูปแบบนอร์มัลของบอยส์คอตต์หรือไม่ ?

ไม่ , แต่กรีเลชันออกเป็นรีเลชันย่อย

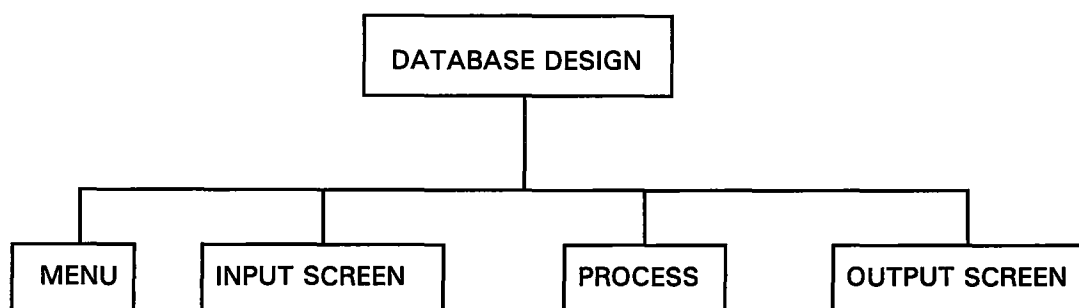
6. นำรีเลชันใหม่ กลับไปวนทำ(5) จนกระทั่งทุกกรีเลชันเป็นรูปแบบนอร์มัลของบอยส์และคอตต์ ทั้งหมด

# บทที่ 3

## การออกแบบโปรแกรม สำเร็จรูปสำหรับออกแบบฐานข้อมูล

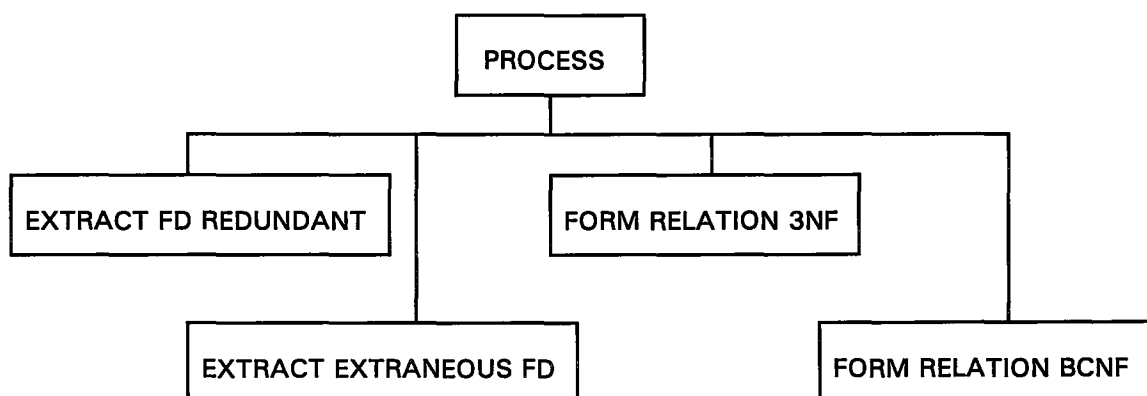
### โครงสร้างระบบ

การออกแบบระบบฐานข้อมูลมีโครงสร้าง ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 โครงสร้างโปรแกรมสำหรับการออกแบบฐานข้อมูล

การประมวลผล(process) มีโครงสร้างดังรูปที่ 3.2



รูปที่ 3.2 โครงสร้างโปรแกรมส่วนการประมวลผล

### อัลกอริทึมกำจัดฟังก์ชันการขึ้นต่อกันที่ซ้ำซ้อน (EXTRACT FD REDUNDANT)

1. อ่านข้อมูล 1 เรคอร์ดจากแฟ้มข้อมูลของฟังก์ชันการขึ้นต่อกันเก็บหมายเลขเรคอร์ดในตัวแปร REC เก็บแอตทริบิวของเทอร์มินันท์ในตัวแปร T และเก็บแอตทริบิวของนันคีย์ลงในตัวแปร N
2. อ่านข้อมูลเรคอร์ดต่อไปจากแฟ้มข้อมูลเก็บแอตทริบิวของดีเทอร์มิแนนท์ ลงในตัวแปร DET และเก็บแอตทริบิวของนันคีย์ลงในตัวแปร NONDET
3. พิจารณาว่าสมาชิกทุกตัวใน DET เป็นสมาชิกใน T หรือไม่
  - ถ้าอยู่ : ใส่สมาชิกทุกตัวใน DET ลงใน T
  - ไม่อยู่ : ไป 5
4. ทุกครั้งที่ T เปลี่ยนแปลง ให้กลับไป 3
5. ถ้า N เป็นสมาชิกใน T แล้ว ฟังก์ชันการขึ้นต่อกันซ้ำซ้อน (FD redundant)
6. อ่านข้อมูลหมายเลขเรคอร์ดที่มีค่าเท่ากับ REC + 1
7. พิจารณาว่าจบไฟล์หรือไม่
  - ถ้าไม่ใช่ : กลับไป 1
  - ถ้าใช่ : จะได้มินิมอลคอเวอร์

### อัลกอริทึมเพื่อกำจัดแอตทริบิวพิเศษ (Extraneous Attributes) ออกจากมินิมอลคอเวอร์

#### กรณีที่ 1 มีการขึ้นต่อกันภายในดีเทอร์มิแนนท์ของฟังก์ชันการขึ้นต่อกัน

1. เซตไฟล์พอยเตอร์สำหรับไฟล์มินิมอลคอเวอร์ไว้ 2 อัน คือไฟล์พอยเตอร์ pt และ pc ให้ชี้ไปที่เรคอร์ดแรกของแฟ้มข้อมูล
2. ใช้ไฟล์พอยเตอร์ pt อ่านข้อมูล 1 เรคอร์ด จากแฟ้มข้อมูลมินิมอลคอเวอร์ เก็บแอตทริบิวของดีเทอร์มิแนนท์ และแอตทริบิวของนันคีย์ลงในตัวแปร dt และ nt ตามลำดับเซตไฟล์พอยเตอร์ pc ไปที่เรคอร์ดแรกของแฟ้มข้อมูล
3. ใช้ไฟล์พอยเตอร์ pc อ่านข้อมูล 1 เรคอร์ด จากแฟ้มข้อมูลมินิมอลคอเวอร์ เก็บแอตทริบิวของดีเทอร์มิแนนท์ และแอตทริบิวของนันคีย์ลงใน dc และ nc ตามลำดับ
4. ถ้า ข้อมูลใน dt และ dc ไม่เหมือนกัน
  - ใช่ ถ้า ทุกแอตทริบิวใน dc อยู่ใน dt และทุกแอตทริบิวใน nc อยู่ใน dt
  - ใช่ กำจัดแอตทริบิวใน nc ออกจาก dt
5. กลับไปทำ (3) จนกระทั่งจบแฟ้มข้อมูล (มินิมอลคอเวอร์)
6. บันทึกแอตทริบิวของ dt และ nt ลงในแฟ้มข้อมูลมินิมอลคอเวอร์ในตำแหน่งที่ไฟล์พอยเตอร์ pt ชี้อยู่
7. กลับไปทำ (2) จนกระทั่งจบแฟ้มข้อมูล (มินิมอลคอเวอร์)

**กรณีที่ 2** มีการขึ้นต่อกันระหว่างบางแอตทริบิวของดีเทอร์มิแนนท์ กับ บางแอตทริบิวที่เป็นนัย

1. เซตไฟล์พอยเตอร์สำหรับไฟล์มินิมอลคอเวอร์ไว้ 2 อัน คือไฟล์พอยเตอร์ pt และ pc ให้ชี้ไปที่เรคอร์ดแรกของแฟ้มข้อมูล
2. ใช้ไฟล์พอยเตอร์ pt อ่านข้อมูล 1 เรคอร์ด จากแฟ้มข้อมูลมินิมอลคอเวอร์ เก็บแอตทริบิวของดีเทอร์มิแนนท์ และแอตทริบิวของนัยลงในตัวแปร dt และ nt ตามลำดับและเซตไฟล์พอยเตอร์ pc ไปที่เรคอร์ดแรกของแฟ้มข้อมูล
3. ใช้ไฟล์พอยเตอร์ pc อ่านข้อมูล 1 เรคอร์ด จากแฟ้มข้อมูลมินิมอลคอเวอร์ เก็บแอตทริบิวของดีเทอร์มิแนนท์ และแอตทริบิวของนัยลงใน dc และ nc ตามลำดับ
4. ถ้า ข้อมูลใน dt และ dc ไม่เหมือนกัน  
 ใ้ชี้ ถ้า ทุกแอตทริบิวใน dc อยู่ใน dt และทุกแอตทริบิวใน nc อยู่ใน nt  
 ใ้ชี้ ถ้า จัดแอตทริบิวใน nc ออกจาก nt
5. กลับไปทำ (3) จนกระทั่งจบแฟ้มข้อมูล (มินิมอลคอเวอร์)
6. บันทึกแอตทริบิวของ dt และ nt ลงในแฟ้มข้อมูลมินิมอลคอเวอร์ในตำแหน่งที่พอยเตอร์ pt ชี้อยู่
7. กลับไปทำ (2) จนกระทั่งจบแฟ้มข้อมูล (มินิมอลคอเวอร์)

**กรณีที่ 3** มีการขึ้นต่อกันภายในนัย ของฟังก์ชันการขึ้นต่อกัน

1. เซตไฟล์พอยเตอร์สำหรับแฟ้มข้อมูลมินิมอลคอเวอร์ไว้ 2 อัน คือไฟล์พอยเตอร์ pt และ pc ให้ชี้ไปที่เรคอร์ดแรกของแฟ้มข้อมูล
2. ใช้ไฟล์พอยเตอร์ pt อ่านข้อมูล 1 เรคอร์ด จากแฟ้มข้อมูลมินิมอลคอเวอร์ เก็บแอตทริบิวของดีเทอร์มิแนนท์ และแอตทริบิวของนัยลงในตัวแปร dt และ nt ตามลำดับ และเซตไฟล์พอยเตอร์ pc ไปที่เรคอร์ดแรกของแฟ้มข้อมูล
3. ใช้ไฟล์พอยเตอร์ pc อ่านข้อมูล 1 เรคอร์ดจากแฟ้มข้อมูลมินิมอลคอเวอร์ เก็บแอตทริบิว ของดีเทอร์มิแนนท์ และแอตทริบิวของนัยลงในตัวแปร dc และ nc ตามลำดับ
4. ถ้า ทุกแอตทริบิวใน dc อยู่ใน nt และทุกแอตทริบิวใน nc อยู่ใน nt  
 ใ้ชี้ ถ้า จัดแอตทริบิวใน nc ออกจาก nt
5. กลับไปทำ (3) จนกระทั่งจบแฟ้มข้อมูล (มินิมอลคอเวอร์)
6. บันทึกแอตทริบิวของ dt และ nt ลงในแฟ้มข้อมูลมินิมอลคอเวอร์ในตำแหน่งที่พอยเตอร์ pt ชี้อยู่
7. กลับไปทำ (2) จนกระทั่งจบแฟ้มข้อมูล (มินิมอลคอเวอร์)

### อัลกอริทึมของการสร้างรีเลย์ชั้นนอร์มัลฟอร์มระดับ 3 จากมินิมอลคอคเวอร์

1. อ่านข้อมูล 1 เรคอร์ด จากแฟ้มข้อมูลมินิมอลคอคเวอร์ เพื่อใช้เป็นรีเลย์หลักแล้วลบเรคอร์ดนั้นออก จากแฟ้มข้อมูล
2. อ่านข้อมูล 1 เรคอร์ด จากแฟ้มข้อมูลมินิมอลคอคเวอร์ เพื่อตรวจสอบค่าดีเทอร์มิแนนท์
3. ถ้า ดีเทอร์มิแนนท์ของฟังก์ชันการขึ้นต่อกัน เท่ากัน ทั้งสองเรคอร์ด  
ใช่ รวมแอดทริบิวชันคีย์ของฟังก์ชันการขึ้นต่อกันของทั้งสองเรคอร์ดเข้าด้วยกัน กำหนดให้เป็น กลุ่มแอดทริบิวชันคีย์ค่าใหม่ของเรคอร์ดที่อ่านมาเพื่อสร้างรีเลย์ชั้น, แล้วลบเรคอร์ดที่อ่านเพื่อ ตรวจสอบค่าดีเทอร์มิแนนท์ออกจากแฟ้มข้อมูล
4. กลับไปทำ (2) จนกระทั่งจบแฟ้มข้อมูล (มินิมอลคอคเวอร์)
5. บันทึกเรคอร์ดข้อมูลการเปลี่ยนแปลงครั้งสุดท้ายของฟังก์ชันการขึ้นต่อกันที่อ่านมา เพื่อสร้างรีเลย์ชั้น ลงในแฟ้มข้อมูลรีเลย์ชั้นนอร์มัลฟอร์มระดับที่ 3
6. เซตไฟล์พอยเตอร์ไปที่ต้นแฟ้มข้อมูลมินิมอลคอคเวอร์, แล้วกลับไปทำ (1) จนกระทั่งแฟ้มข้อมูลมินิมอลคอคเวอร์เปลี่ยนเป็นแฟ้มข้อมูลว่าง

### อัลกอริทึมปรับปรุงรีเลย์ชั้นนอร์มัลฟอร์มระดับที่ 3 ให้เป็นรีเลย์ชั้นนอร์มัลฟอร์มบอยส์คอดด์

1. อ่านข้อมูล 1 เรคอร์ด จากแฟ้มข้อมูลรีเลย์ชั้นนอร์มัลฟอร์มระดับที่ 3  
 ถ้า จำนวนนัยของรีเลย์ชั้น = 0  
ใช่ บันทึกรีเลย์ชั้นนอร์มัลฟอร์มระดับที่ 3 ลงในแฟ้มข้อมูลรีเลย์ชั้นนอร์มัลฟอร์มของบอยส์คอดด์ แล้วข้ามไปทำ (5)
2. อ่านข้อมูล 1 เรคอร์ด จากแฟ้มข้อมูลมินิมอลคอคเวอร์
3. ถ้า ทุกแอดทริบิวชันคีย์ของดีเทอร์มิแนนท์ อยู่ในคีย์หลักหรือนัยของรีเลย์ชั้น  
ใช่ ถ้า บางแอดทริบิวชันคีย์ของดีเทอร์มิแนนท์ อยู่ในนัยของรีเลย์ชั้น และ มีบางแอดทริบิวชันคีย์ทางด้านซ้ายของฟังก์ชันการขึ้นต่อกันอยู่ในคีย์หลักของรีเลย์ชั้น และ มีบางแอดทริบิวชันคีย์หลัก ที่ไม่พบอยู่ในการอยู่ในฟังก์ชันการขึ้นต่อกัน  
ใช่ สร้างรีเลย์ชั้นใหม่ 2 รีเลย์ชั้น , แล้วบันทึกรีเลย์ชั้นใหม่เหล่านั้นลงท้าย แฟ้มข้อมูลรีเลย์ชั้นนอร์มัลฟอร์มระดับที่ 3  
ไม่ กลับไปทำ (2) จนกระทั่งจบแฟ้มข้อมูล (มินิมอลคอคเวอร์)  
ไม่ กลับไปทำ(2) จนกระทั่งจบแฟ้มข้อมูล (มินิมอลคอคเวอร์)

## โครงสร้างของรีเลย์ขั้นใหม่

**รีเลย์ขั้นที่ 1** คีย์หลัก คือ ทุกแอตทริบิวต์ในดีเทอร์มิแนนท์

**นันทิย์** คือ บางแอตทริบิวต์ที่อยู่ทางด้านซ้ายของฟังก์ชันการขึ้นต่อกัน ซึ่ง พบอยู่ในรีเลย์  
    ชั้นนอร์มัลฟอร์มระดับที่ 3

**รีเลย์ขั้นที่ 2** คีย์หลัก คือ ทุกแอตทริบิวต์ในดีเทอร์มิแนนท์ บางแอตทริบิวต์ใน คีย์ซึ่งไม่พบอยู่ในฟังก์ชัน  
    การขึ้นต่อกัน

**นันทิย์** คือ บางแอตทริบิวต์ในนันทิย์ ซึ่งไม่พบอยู่ในฟังก์ชันการขึ้นต่อกัน

4. ถ้า จบเพิ่มข้อมูล (มินิมอลคอบเวอร์)

**ใช่** บันทึกรีเลย์ชั้นนอร์มัลฟอร์มที่อ่านจากข้อ (1) ลงท้ายไฟล์รีเลย์ชั้นนอร์มัลฟอร์มของบอยส์คอดด์

5. กลับไปทำ (1) จนกระทั่งจบเพิ่มข้อมูล (รีเลย์ชั้นนอร์มัลฟอร์มระดับที่ 3)

### การออกแบบส่วนติดต่อกับผู้ใช้

    การออกแบบส่วนติดต่อกับผู้ใช้แบ่งเป็น 2 ส่วน คือ เมนูกับจอภาพ รายละเอียดแสดงในภาค  
    ผนวก คู่มือการใช้

### การออกแบบเพิ่มข้อมูล

    เพิ่มข้อมูลมีทั้งหมด 12 แฟ้ม ถูกสร้างขึ้นโดยใช้โปรแกรมช่วยงานเกี่ยวกับการจัดการเพิ่ม  
    ข้อมูลที่มีอยู่ในฟอกซ์โปร

    เพิ่มข้อมูลทั้ง 12 แฟ้มนี้ ถูกแบ่งอยู่ในส่วนของการทำงานที่ต่างกัน 3 ส่วน คือ

- ส่วนของเพิ่มข้อมูลนำเข้า
- ส่วนของเพิ่มข้อมูลที่ใช้ระหว่างขบวนการวิเคราะห์ข้อมูล
- ส่วนของเพิ่มข้อมูลแสดงผล

    ในที่นี้ใช้ x แทนตัวอักษร

    9 แทนตัวเลข

    N แทน NUMERIC

    C แทน CHARACTER

    และความยาวมีหน่วยเป็นไบต์

## 1. เพิ่มข้อมูลนำเข้า

### เพิ่มฐานข้อมูลรวม

เป็นแฟ้มที่ใช้รวบรวมรายชื่อของฐานข้อมูลทั้งหมดที่ถูกสร้างขึ้นและยังคงใช้งานอยู่ และเก็บรวมรายชื่อของแฟ้มข้อมูลทั้งหมดที่มีความเกี่ยวข้องกับฐานข้อมูลแต่ละชุด

ชื่อแฟ้มข้อมูล : DBNMREC.DBF

ความยาวเรคคอร์ด : 41 ไบต์

ตัวแปร (Field)	รายละเอียด (Description)	ชนิด (Type)	ความยาว (Length)	รูปแบบ (Form)
REC_NO	หมายเลขประจำฐานข้อมูล	N	7	99.9
DBNM	ชื่อฐานข้อมูล	C	8	xx..x
ATTFILE	ชื่อแฟ้มข้อมูลแอดทริบิว	C	8	xx..x
FDATT	ชื่อแฟ้มข้อมูลแอดทริบิวของฟังก์ชันการขึ้นต่อกัน	C	8	xx..x
FDSIZE	ชื่อแฟ้มข้อมูลขนาดของฟังก์ชันการขึ้นต่อกัน	C	8	xx..x
RATT	ชื่อแฟ้มข้อมูลแอดทริบิวของรีเลชัน	C	8	xx..x
RSIZE	ชื่อไฟล์ขนาดของรีเลชัน	C	8	xx..x

- แฟ้มข้อมูลแอดทริบิว เป็นแฟ้มที่ใช้เก็บแอดทริบิวทั้งหมดของระบบฐานข้อมูลอันหนึ่ง ข้อมูลถูกเรียงตามลำดับการนำเข้า

ชื่อแฟ้มข้อมูล : ATTFILE.DBF

ความยาวเรคคอร์ด : 10 ไบต์

ตัวแปร (Field)	รายละเอียด (Description)	ชนิด (Type)	ความยาว (Length)	รูปแบบ (Form)
ATT	ชื่อแอดทริบิว	C	10	xx..x

- เพิ่มข้อมูลฟังก์ชันการขึ้นต่อกัน เป็นเพิ่มใช้เก็บแอตทริบิวที่เป็นดีเทอร์มิแนนท์และนัยสลับกันไป เรียงตามลำดับการนำเข้าข้อมูลฟังก์ชันการขึ้นต่อกัน

ชื่อเพิ่มข้อมูล : FDATT.DBF

ความยาวเรคอร์ด: 10 ไบต์

ตัวแปร (Field)	รายละเอียด (Description)	ชนิด (Type)	ความยาว (Length)	รูปแบบ (Form)
ATT	ชื่อแอตทริบิว	C	10	xx..x

- เพิ่มข้อมูลขนาดของดีเทอร์มิแนนท์และนัยของฟังก์ชันการขึ้นต่อกัน เป็นเพิ่มที่ใช้เก็บจำนวนของแอตทริบิวที่ประกอบขึ้นเป็นดีเทอร์มิแนนท์ และนัยของฟังก์ชันการขึ้นต่อกันสลับกันไป เรียงตามลำดับการนำเข้าของข้อมูลฟังก์ชันการขึ้นต่อกัน

ชื่อเพิ่มข้อมูล : FDSIZE.DBF

ความยาวเรคอร์ด: 7/4 ไบต์

ตัวแปร (Field)	รายละเอียด (Description)	ชนิด (Type)	ความยาว (Length)	รูปแบบ (Form)
SIZEL	ขนาดของดีเทอร์มิแนนท์	N	7/8	99
SIZER	ขนาดขนาดของนัย	N	7/8	99



## 2. เพิ่มข้อมูลที่ใช้ระหว่างขบวนการวิเคราะห์ข้อมูล

- เพิ่มข้อมูลแอตทริบิวต์ของมินิมอลคอเวออร์ เป็นเพิ่มที่ใช้เก็บแอตทริบิวต์ที่เป็นดีเทอร์มิแนนท์และนัยของมินิมอลคอเวออร์

ชื่อเพิ่มข้อมูล : FDATT1.DBF

ความยาวเรคอร์ด: 10 ไบต์

ตัวแปร (Field)	รายละเอียด (Description)	ชนิด (Type)	ความยาว (Length)	รูปแบบ (Form)
ATT	ชื่อแอตทริบิวต์	C	10	xx..x

- เพิ่มข้อมูลขนาดของมินิมอลคอเวออร์ เป็นเพิ่มที่ใช้เก็บจำนวนของแอตทริบิวต์ที่ประกอบขึ้นเป็นดีเทอร์มิแนนท์ และนัยของมินิมอลคอเวออร์

ชื่อเพิ่มข้อมูล : FDTSIZE1.DBF

ความยาวเรคอร์ด: 7/4 ไบต์

ตัวแปร (Field)	รายละเอียด (Description)	ชนิด (Type)	ความยาว (Length)	รูปแบบ (Form)
SIZEL	ขนาดของดีเทอร์มิแนนท์	N	7/8	99
SIZER	ขนาดของนัย	N	7/8	99

- เพิ่มแอตทริบิวต์ของมินิมอลคอเวออร์ที่ไม่เป็นแอตทริบิวต์พิเศษ เป็นเพิ่มที่ใช้เก็บแอตทริบิวต์ของมินิมอลคอเวออร์ ซึ่งได้กำจัดแอตทริบิวต์พิเศษออกไปแล้ว

ชื่อเพิ่มข้อมูล : FDTATT.DBF

ความยาวเรคอร์ด: 10 ไบต์

ตัวแปร (Field)	รายละเอียด (Description)	ชนิด (Type)	ความยาว (Length)	รูปแบบ (Form)
ATT	ชื่อแอตทริบิวต์	C	10	xx..x

- เพิ่มข้อมูลขนาดของมินิมอลคอเวอร์ที่ไม่เป็นแอตทริบิวพิเศษเป็นแฟ้มที่ใช้เก็บจำนวนของแอตทริบิวที่ประกอบขึ้นเป็นดีเทอร์มิแนนท์ และนันคีย์ของมินิมอลคอเวอร์ซึ่งได้กำจัดแอตทริบิวพิเศษออกไปแล้ว

ชื่อแฟ้มข้อมูล : FDTSIZE.DBF

ความยาวเรคอร์ด: 7/4 ไบต์

ตัวแปร (Field)	รายละเอียด (Description)	ชนิด (Type)	ความยาว (Length)	รูปแบบ (Form)
SIZEL	ขนาดของดีเทอร์มิแนนท์	N	7/8	99
SIZER	ขนาดของนันคีย์	N	7/8	99

- เพิ่มข้อมูลแอตทริบิวของรีเลชันนอร์มัลฟอร์มระดับที่ 3 เป็นแฟ้มที่ใช้เก็บแอตทริบิวที่เป็นคีย์หลักและนันคีย์ ของรีเลชันนอร์มัลฟอร์มระดับที่ 3

ชื่อแฟ้มข้อมูล : BCNFATT.DBF

ความยาวเรคอร์ด: 10 ไบต์

ตัวแปร (Field)	รายละเอียด (Description)	ชนิด (Type)	ความยาว (Length)	รูปแบบ (Form)
ATT	ชื่อแอตทริบิว	C	10	xx..x

- เพิ่มข้อมูลขนาดของรีเลชันนอร์มัลฟอร์มระดับที่3  
เป็นแฟ้มที่ใช้เก็บจำนวนของแอตทริบิวที่ประกอบขึ้นเป็นคีย์หลักและนันคีย์ ของรีเลชันนอร์มัลฟอร์มระดับที่3

ชื่อแฟ้มข้อมูล : BCNF SIZE.DBF

ความยาวเรคอร์ด: 7/4 ไบต์

ตัวแปร (Field)	รายละเอียด (Description)	ชนิด (Type)	ความยาว (Length)	รูปแบบ (Form)
SIZEL	ขนาดของดีเทอร์มิแนนท์	N	7/8	99
SIZER	ขนาดของนันคีย์	N	7/8	99

### 3. เพิ่มข้อมูลแสดงผล

- เพิ่มข้อมูลแอดทริบิวของรีเลชันนอร์มัลฟอร์มระดับบอยส์คอตต์ เป็นเพิ่มที่เก็บแอดทริบิวที่เป็นคีย์หลัก และนันทคีย์ของแต่ละรีเลชันซึ่งอยู่ในระดับบอยส์คอตต์นอร์มัลฟอร์ม

ชื่อเพิ่มข้อมูล : ATTRE.DBF

ความยาวเรคอร์ด: 10 ไบต์

ตัวแปร (Field)	รายละเอียด (Description)	ชนิด (Type)	ความยาว (Length)	รูปแบบ (Form)
ATT	ชื่อแอดทริบิว	C	10	xx..x

- เพิ่มข้อมูลขนาดของคีย์หลักและนันทคีย์ของรีเลชันนอร์มัลฟอร์มระดับบอยส์คอตต์ ใช้เก็บจำนวนของแอดทริบิวที่ประกอบกันเป็นคีย์หลักและนันทคีย์สลับกันไปเรียงตามลำดับของรีเลชัน ซึ่งอยู่ในรูปแบบนอร์มัลฟอร์มของบอยส์คอตต์

ชื่อเพิ่มข้อมูล : RTABLE.DBF

ความยาวเรคอร์ด: 39/4 ไบต์

ตัวแปร (Field)	รายละเอียด (Description)	ชนิด (Type)	ความยาว (Length)	รูปแบบ (Form)
RELANM	ชื่อของรีเลชัน	C	8	xx..x
SIZEL	ขนาดของคีย์	N	7/8	99
SIZER	ขนาดของนันทคีย์	N	7/8	99

## บทที่ 4

### การประเมินผลระบบ

ระบบที่พัฒนาขึ้นนี้ ประเมินผลจากการทดลองใช้งานดังนี้

- 1.สามารถออกแบบฐานข้อมูลขนาดเล็ก ซึ่งระบบฐานข้อมูลไม่ใหญ่มากทำให้เกิดความคล่องตัวและรวดเร็วในการออกแบบฐานข้อมูลมากกว่าการคิดด้วยตนเอง และยังทำให้ผลลัพธ์ที่ได้มีความถูกต้องมากขึ้น
- 2.การใช้งานของโปรแกรมอาจมีความยุ่งยากซับซ้อนหลายขั้นตอน ความไม่คล่องตัวอาจเกิดขึ้นในระยะแรกของการใช้งาน แต่เมื่อเข้าใจลำดับและวิธีการใช้งานของโปรแกรมอย่างถูกวิธีแล้วความผิดพลาดก็จะไม่เกิดขึ้น
- 3.สามารถนำฐานข้อมูลเดิมที่ได้สร้างไว้แล้วมาแก้ไขปรับปรุงได้
4. เมื่อสามารถสร้างฐานข้อมูลได้แล้ว สามารถนำเพิ่มข้อมูลของฐานข้อมูลนี้ไปป้อนข้อมูลนำเข้าได้ทันทีจากโปรแกรมออกแบบฐานข้อมูลโดยตรงและจะเก็บข้อมูลเหล่านี้ไว้ใช้งานได้ทันที

อย่างไรก็ตามข้อจำกัดของระบบ มีดังนี้

1. เป็นฐานข้อมูลที่สามารถรับข้อมูลแอดทริบิวต์ได้ไม่เกิน 20 แอดทริบิวต์ ซึ่งเป็นระบบฐานข้อมูลขนาดเล็ก
- 2.การประมวลผลของโปรแกรมสามารถออกแบบฐานข้อมูลได้เพียงระดับบอยส์คอดด์เท่านั้นทำให้การประมวลผลของงานที่ต้องใช้การออกแบบฐานข้อมูลที่สูงกว่านี้ไม่สามารถทำได้
- 3.ความเร็วในการทำงานของโปรแกรมยังอยู่ในขั้นต่ำเนื่องมาจากโปรแกรมมีการติดต่อกับหน่วยความจำสำรองบ่อยเกินไป

## บทที่ 5

### สรุปและข้อเสนอแนะ

การพัฒนาโปรแกรมออกแบบฐานข้อมูลในการทำปัญหาพิเศษในครั้งนี้ เหมาะสมมากสำหรับการใช้งานในระดับฐานข้อมูลที่มีขนาดเล็ก ซึ่งไม่มีความจำเป็นที่จะต้องให้โปรแกรมออกแบบฐานข้อมูลในระดับสูงกว่านี้ โดยผลจากการพัฒนาโปรแกรม ทำให้ได้ทราบลำดับขั้นของการออกแบบฐานข้อมูลซึ่งมีความซับซ้อนแต่สามารถนำมาประยุกต์ให้มาประมวลผลได้บนเครื่องไมโครคอมพิวเตอร์ ทำให้เกิดความสะดวกและรวดเร็วในการออกแบบฐานข้อมูลเป็นอย่างมาก แต่ปัญหาที่เกิดขึ้นบางประการที่พบได้ในการใช้งานโปรแกรมชนิดนี้ คือ อาจจะมีความยุ่งยากในการป้อนข้อมูล การผิดพลาดในการเตรียมข้อมูลของผู้ใช้โปรแกรมเป็นพิเศษ เนื่องจากเมื่อเกิดความผิดพลาดในการใส่ข้อมูลแล้ว ขั้นตอนในการแก้ไขข้อมูลของโปรแกรมออกแบบฐานข้อมูลนี้ จะมีความยุ่งยากในการแก้ไข เช่นการแก้ไขฟังก์ชันที่ขึ้นต่อกัน(functional dependency) เป็นต้น แต่โดยรวมแล้ว ถ้าผู้ใช้งานโปรแกรมรู้จักวิธีการใช้งานอย่างถูกต้อง ก็จะเกิดประโยชน์อย่างมากแก่งานด้านการออกแบบฐานข้อมูล

อย่างไรก็ตามโปรแกรมการออกแบบฐานข้อมูลนี้มีข้อจำกัดในการใช้งาน ซึ่งผู้ที่จะนำโปรแกรมนี้ไปใช้ควรจะทราบไว้ล่วงหน้าว่า สามารถนำไปใช้งานด้านการออกแบบฐานข้อมูลซึ่งมีขนาดเล็ก นั่นคือ มีข้อมูลของจำนวนแอตทริบิวต์ไม่เกิน 20 แอตทริบิวต์ อีกทั้งขั้นตอนของการออกแบบฐานข้อมูลให้อยู่ในรูปแบบนอร์มัลของบอยส์คอตต์เท่านั้น แต่ก็เหมาะสมกับระดับฐานข้อมูลในระดับไม่ใหญ่มาก

เนื่องมาจากโปรแกรมคอมพิวเตอร์ช่วยงานการออกแบบฐานข้อมูลขั้นนี้ ได้พัฒนามาจากทฤษฎีพื้นฐานของการออกแบบฐานข้อมูล ซึ่งการออกแบบฐานข้อมูลที่ต้องตรงตามทฤษฎีอาจไม่ใช่ฐานข้อมูลที่เหมาะสมในการใช้งานที่สุดก็ได้จึงทำให้ฐานข้อมูลบางอันที่เป็นผลจากการทำงานของโปรแกรมสำเร็จรูปนี้มีประสิทธิภาพการทำงานที่ไม่สมบูรณ์เท่าที่ควร จึงสมควรที่จะได้มีการศึกษาค้นคว้าเพิ่มเติมเพื่อพัฒนาโปรแกรมให้สมบูรณ์ยิ่งขึ้น สามารถออกแบบฐานข้อมูลที่มีประสิทธิภาพการทำงานสูงสุดได้

**ภาคผนวก**

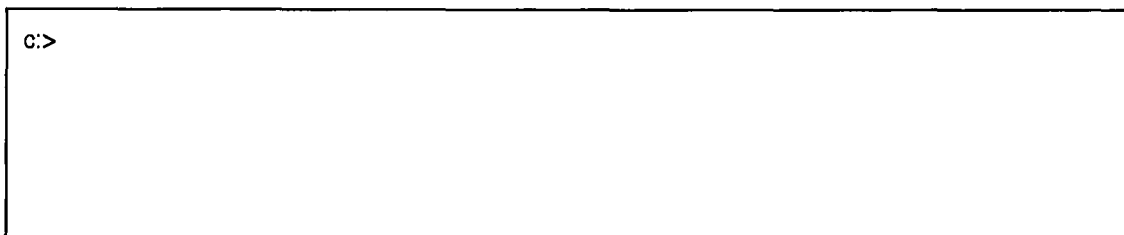
## คู่มือการใช้งาน

ในการใช้งานโปรแกรมสำเร็จรูปสำหรับการออกแบบฐานข้อมูลจะต้องประกอบด้วยส่วนต่างๆ ดังนี้

- ส่วนประกอบทางด้านฮาร์ดแวร์ (Hardware)
  - เครื่องไมโครคอมพิวเตอร์ 1 เครื่อง
  - จอสี 1 จอ
- ส่วนประกอบทางด้านซอฟต์แวร์ (Software)
  - แฟ้มข้อมูลสกุล EXE 1 ไฟล์ คือ DBMS.EXE

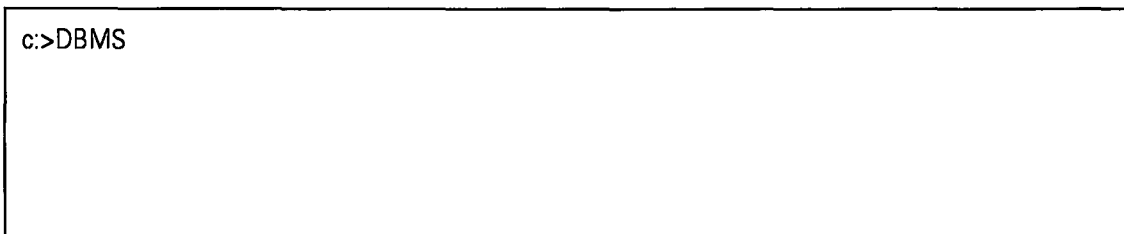
### การเข้าสู่โปรแกรมสำเร็จรูปสำหรับการออกแบบฐานข้อมูล

เมื่อมีอุปกรณ์และแฟ้มข้อมูลครบดังที่กล่าวแล้ว จะเริ่มเข้าสู่โปรแกรมสำเร็จรูปนี้ได้โดยขั้นตอนที่ 1 เปิดเครื่องคอมพิวเตอร์ หรือทำให้เครื่องคอมพิวเตอร์แสดงสถานะพร้อม นั่นคือจะปรากฏดังรูปที่ ก-1



รูปที่ ก-1 แสดงสถานะพร้อมของเครื่องคอมพิวเตอร์

ขั้นตอนที่ 2 พิมพ์ DBMS แล้วกด <ENTER> ดังรูปที่ ก-2



รูปที่ ก-2 แสดงการเข้าสู่โปรแกรมสำเร็จรูปสำหรับการออกแบบระบบฐานข้อมูล

หลังจากนั้นจะเข้าสู่การทำงานของโปรแกรมสำเร็จรูปสำหรับการออกแบบฐานข้อมูลซึ่งจะปรากฏเมนูให้เลือกในส่วนของการทำงานก่อน ดังรูปที่ ก-3

### วิธีการใช้แถบสว่างเลือกการทำงาน

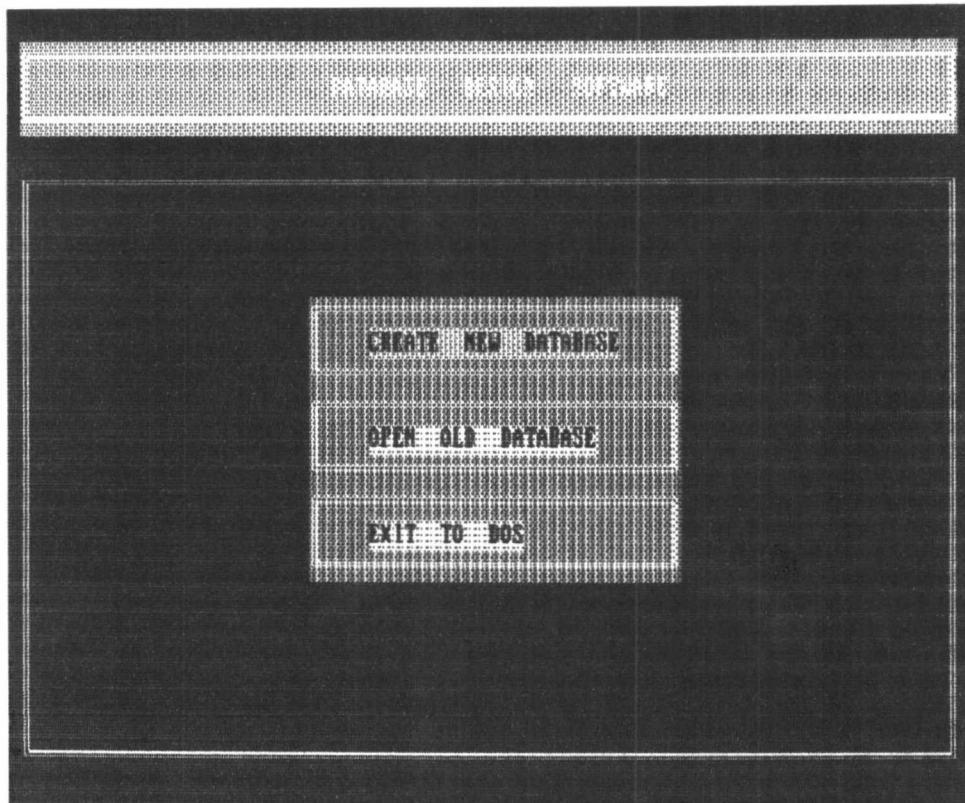
ลูกศรขึ้นเลื่อนแถบสว่างไปยังเรคอร์ดก่อนหน้า 1 เรคอร์ด หรือไปเรคอร์ดล่างสุดเพื่อแถบสว่างปัจจุบันอยู่ที่เรคอร์ดบนสุด

ลูกศรลงเลื่อนแถบสว่างไปยังเรคอร์ดต่อไป 1 เรคอร์ด หรือไปเรคอร์ดล่างสุดเพื่อแถบสว่างปัจจุบันอยู่ที่เรคอร์ดล่างสุด

ลูกศรไปซ้าย เลื่อนแถบสว่างไปยังเรคอร์ดที่อยู่ด้านซ้าย หรือกลับไปเรคอร์ดขวาสุดเมื่อแถบสว่างปัจจุบันอยู่ที่เรคอร์ดซ้ายสุด

ลูกศรไปขวาเลื่อนแถบสว่างไปยังเรคอร์ดที่อยู่ด้านขวา หรือกลับไปเรคอร์ดซ้ายสุดเมื่อแถบสว่างปัจจุบันอยู่ที่เรคอร์ดขวาสุด

กด "ENTER" เพื่อเข้าไปทำงานในส่วนหัวข้อที่แถบสว่างปรากฏอยู่

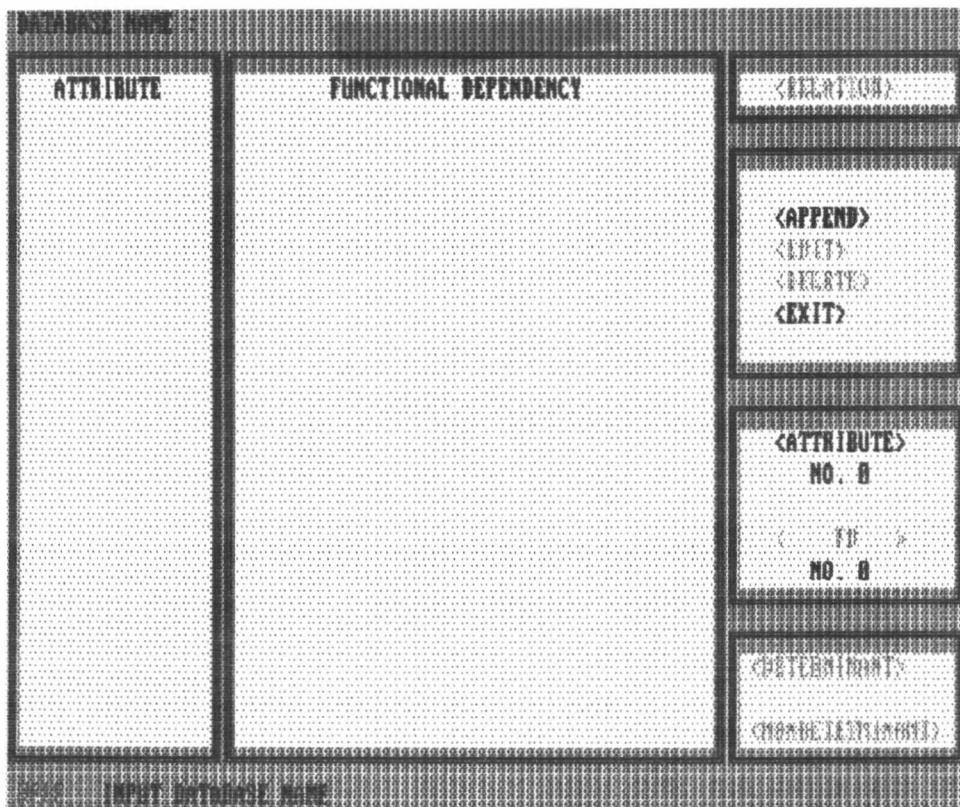


รูปที่ ก-3 แสดงเมนูให้เลือกส่วนของการทำงาน

จากรูปที่ ก-3 ถ้าเลือกหัวข้อ "CREATE NEW DATABASE" จะปรากฏดังรูป ก-4 เพื่อเข้าสู่การ  
สร้างฐานข้อมูลใหม่

จากรูปที่ ก-3 ถ้าเลือกหัวข้อ "OPEN OLD DATABASE" จะปรากฏดังรูป ก-11 เพื่อเข้าสู่การ  
ทำงานกับฐานข้อมูลเก่า

จากรูปที่ ก-3 ถ้าเลือกหัวข้อ "EXIT TO DOS " จะออกจากการทำงานของโปรแกรมสำเร็จรูป  
สำหรับการออกแบบฐานข้อมูล

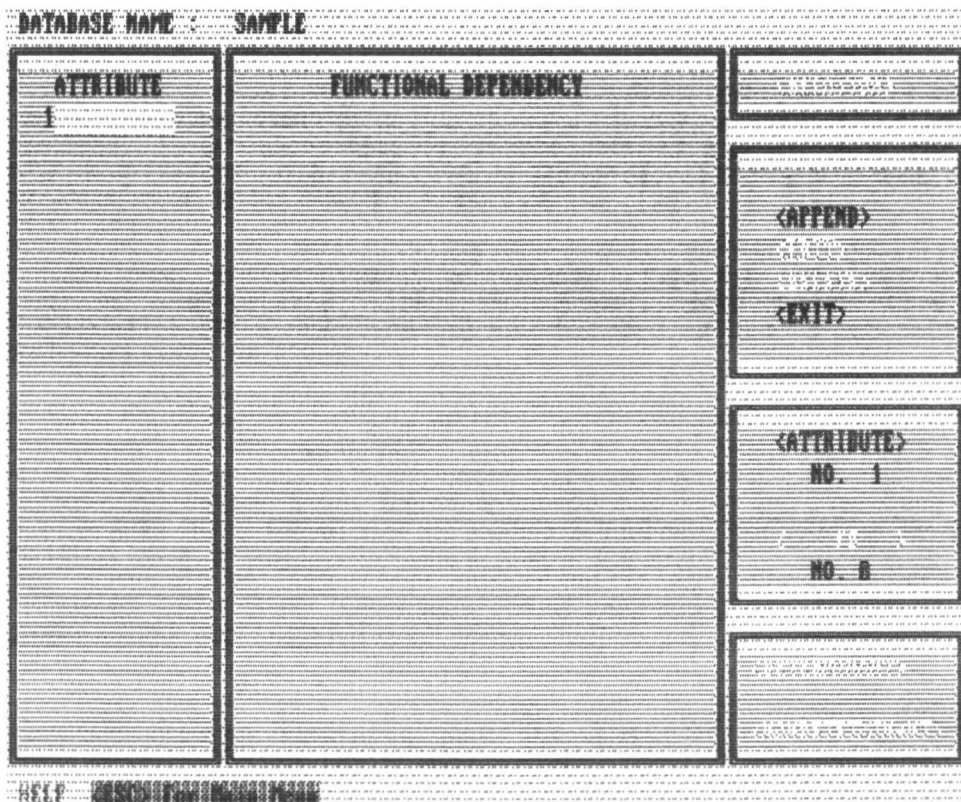


รูปที่ ก-4 แสดงการเข้าสู่การทำงานสร้างฐานข้อมูลใหม่

## การสร้างฐานข้อมูลใหม่

มีขั้นตอนการดำเนินงานดังนี้

1. ตั้งชื่อของฐานข้อมูล โดยที่ชื่อของฐานข้อมูลจะประกอบไปด้วย A-Z, 0-9 หรือ "\_" รวมทั้งหมดไม่เกิน 10 ตัว แล้วกด "ENTER" ผลของการทำงานเป็นดังรูปที่ ก-5



รูปที่ ก-5 แสดงหน้าจอการรอรับข้อมูลแอตทริบิวต์

2. ใส่ข้อมูลแอตทริบิวต์ซึ่งประกอบด้วย A-Z, 0-9 หรือ " " รวมทั้งหมดไม่เกิน 8 ตัว กด "ENTER" แล้วจึงใส่แอตทริบิวต์ตัวถัดไป

นอกจากการทำงานเพื่อเติมข้อมูลแอตทริบิวต์ด้วยการ กด "ESC" หรือ กด"ENTER"ซ้ำผลของการทำงานเป็น ดังรูปที่ ก-6

DATABASE NAME : SAMPLE	
ATTRIBUTE	FUNCTIONAL DEPENDENCY
1MARKET	
2CAPACITY	
3TIME	
4SPEED	
5KILOGRAM	
6TYPE	
7AMOUNT	
8TOTAL	
9	
	<RELATION>
	<APPEND>
	<EDIT>
	<DELETE>
	<EXIT>
	<ATTRIBUTE>
	NO. 9
	< FD >
	NO. 8
	<DETERMINANT>
	<NONDETERMINANT>
HELP	

รูปที่ ก-6 แสดงหน้าจอขณะใส่ข้อมูลแอตทริบิวต์

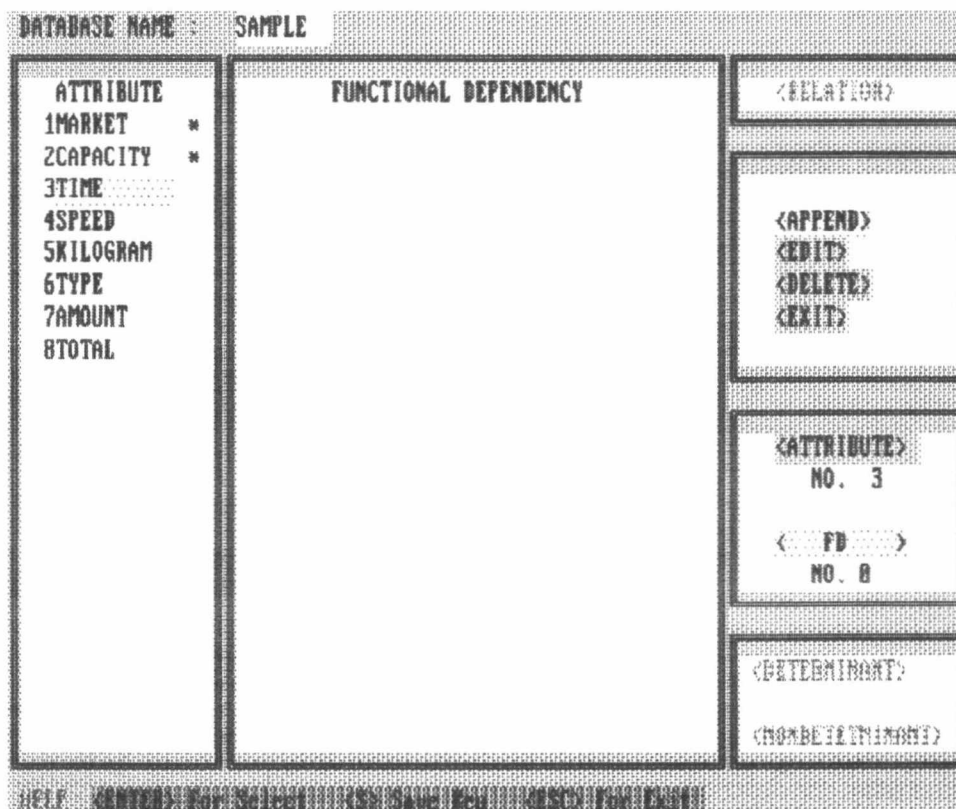
3. ใช้แถบส่วางเลือกการทำงานอย่างใดอย่างหนึ่ง อันได้แก่การเพิ่มข้อมูล (Append),การแก้แอดทริบิว (Edit), การลบแอดทริบิว(Delete) และออกจากการทำงานของโปรแกรม(Exit)

เลือกการทำงานเพื่อ **เพิ่มข้อมูล (Append)**

การเพิ่มข้อมูลทำได้ทั้ง 2 อย่าง คือ เพิ่มข้อมูลแอดทริบิว และเพิ่มข้อมูลฟังก์ชันการขึ้นต่อกัน ทำได้โดยการเลือกแถบส่วางแอดทริบิว หรือเลือกแถบส่วางฟังก์ชันการขึ้นต่อกัน

- การเพิ่มแอดทริบิว (Append Attribute) โดยการใส่แอดทริบิวที่ต้องการเพิ่มเติมสลับกับการกด "ENTER" และยุติการเพิ่มแอดทริบิวด้วยการกด "ESC" หรือกด "ENTER" ซ้ำ

- การเพิ่มข้อมูลฟังก์ชันการขึ้นต่อกัน (Append Function Dependency) ใช้แถบส่วางเลือกแอดทริบิวที่เป็นดีเทอร์มิแนนท์ ดังรูปที่ ก-7กด "S" เพื่อเปลี่ยนกลับไปกลับมา ระหว่างการเลือกแอดทริบิวเพื่อใช้เป็นดีเทอร์มิแนนท์ กับการเลือกแอดทริบิวเพื่อใช้เป็นนันคีย์ ดังรูปที่ ก-8 ออกจากการทำงานเพื่อเพิ่มข้อมูลฟังก์ชันการขึ้นต่อกันด้วยการกด "ESC"



รูปที่ ก-7 แสดงจอขณะเริ่มใส่ข้อมูลดีเทอร์มิแนนท์ของฟังก์ชันการขึ้นแก่กัน

DATABASE NAME : SAMPLE	
<b>ATTRIBUTE</b> 1MARKET 2CAPACITY 3TIME 4SPEED 5KILOGRAM 6TYPE 7AMOUNT 8TOTAL	<b>FUNCTIONAL DEPENDENCY</b> MARKET,CAPACITY—>
	<RELATION>  <APPEND> <EDIT> <DELETE> <EXIT>
	<ATTRIBUTE> NO. 1  < FD > NO. 1
	<DETERMINANT>  <NONDETERMINANT>
HELP <ENTER> For Select (S) Save Now Neg (ESC) For Exit	

รูปที่ ก-8 แสดงจอขณะเตรียมใส่ข้อมูลนัคัยของฟังก์ชันการขึ้นแกกัน

### เลือกการทำงานลบแอตทริบิว (Delete)

ใช้แถบสว่างเลือกแอตทริบิวที่จะลบออก ดังรูปที่ ก-9 ซึ่งสามารถยกเลิกคำสั่งลบแอตทริบิวแต่ละตัวได้ด้วยการกด "ENTER" ซ้ำอีกครั้ง

กด "D" เพื่อยืนยัน และออกจากการทำงานลบแอตทริบิว หรือกด "ESC" เมื่อต้องการยกเลิกคำสั่งลบที่ได้ทำไปทั้งหมด

DATABASE NAME : SAMPLE	
ATTRIBUTE	FUNCTIONAL DEPENDENCY
1MARKET	
2CAPACITY	
3TIME	
4SPEED	
5KILOGRAM	
6TYPE	
7AMOUNT	
8TOTAL1	
9TOTAL2 *	
10TOTAL3 *	

<RELATION>

<APPEND>  
<EDIT>  
<DELETE>  
<EXIT>

<ATTRIBUTE>  
NO. 1

< FD >  
NO. 8

<DETERMINANT>  
<NONDETERMINANT>

HELP <ENTER> For Select (D) For Delete (ESC) For Exit

รูปที่ ก-9 แสดงหน้าจอขณะทำการลบแอตทริบิว

### เลือกการทำงานแก้ไขแอตทริบิว (Edit)

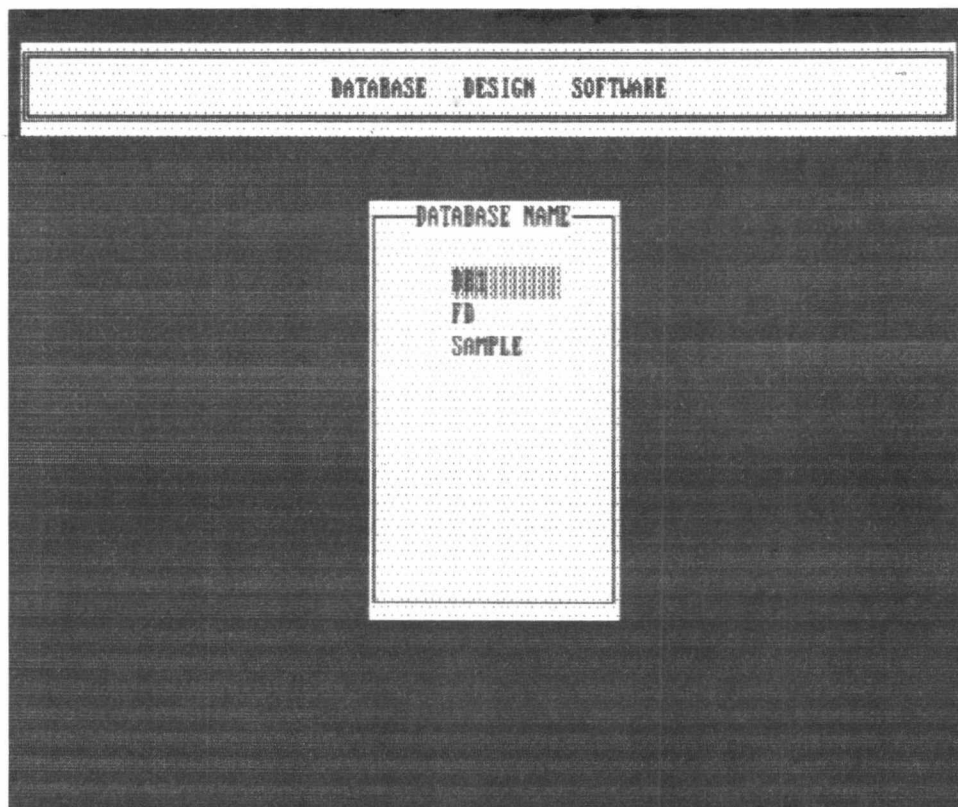
เลื่อนแถบสว่างไปยังแอตทริบิวที่ต้องการจะแก้ไข ทำการแก้ไขข้อมูลให้ถูกต้อง แล้วกด "ENTER" ออกจากการทำงานแก้ไขแอตทริบิวด้วยการกด "ESC" ผลของการทำงานเป็นดังรูป

DATABASE NAME : SAMPLE		
ATTRIBUTE	FUNCTIONAL DEPENDENCY	<RELATION>
1MARKET		<APPEND>
2CAPACITY		<EDIT>
3TIME		<DELETE>
4SPEED		<EXIT>
5KILOGRAM		
6TYPE		<ATTRIBUTE>
7AMOUNT		NO. 8
8TOTAL		< FB >
		NO. 8
		<DETERMINANT>
		<NONDETERMINANT>

HELP <ESC> FOR HELP P<PRESS <TAB> FOR NEXT SCREEN

รูปที่ ก-10 แสดงหน้าจอภายหลังแก้ไขข้อมูล

## การทำงานฐานข้อมูลเก่า



รูปที่ ก-11 แสดงหน้าจอเพื่อการเรียกฐานข้อมูลเก่ามาใช้งาน

เลือกรฐานข้อมูลที่ต้องการโดยใช้แถบส่วาง หลังจากนั้นข้อมูลแอตทริบิวและฟังก์ชันการขึ้นต่อกันทั้งหมดของฐานข้อมูลทีเลือก จะถูกไหลดออกมาแสดงที่หน้าจอดังที่ปรากฏในรูปที่ ก-12

DATABASE NAME : SAMPLE	
<b>ATTRIBUTE</b> 1 MARKET 2 CAPACITY 3 TIME 4 SPEED 5 KILOGRAM 6 TYPE 7 AMOUNT 8 TOTAL	<b>FUNCTIONAL DEPENDENCY</b> 1 MARKET,CAPACITY → TIME,SPEED 2 TIME → SPEED,AMOUNT 3 CAPACITY → KILOGRAM 4 TYPE → MARKET 5 KILOGRAM,TOTAL → AMOUNT
	<b>&lt;RELATION&gt;</b> <APPEND> <EDIT> <DELETE> <EXIT>
	<b>&lt;ATTRIBUTE&gt;</b> NO. 1 < FD > NO. 5
	<b>&lt;DETERMINANT&gt;</b> <NONDETERMINANT>
HELP	

รูปที่ ก-12 แสดงข้อมูลแอตทริบิวและฟังก์ชันการขึ้นต่อกันของฐานข้อมูลเก่า

**การทำงานกับฐานข้อมูลเดิม หรือ ฐานข้อมูลใหม่ที่มีข้อมูลแอตทริบิวและข้อมูลฟังก์ชันการขึ้นต่อกันพร้อมแล้ว**

ใช้แถบสว่างเพื่อเลือกการทำงานอย่างใดอย่างหนึ่งได้แก่ สร้างรีเลชัน (Relation), เพิ่มข้อมูล(Append), แก้ไขข้อมูล(Edit) และการออกจากการทำงานของโปรแกรม (Exit)

#### **เลือกการทำงานเพิ่มข้อมูล (Append)**

การเพิ่มข้อมูลทำได้ทั้ง 2 อย่าง คือ เพิ่มข้อมูลแอตทริบิว และเพิ่มข้อมูลฟังก์ชันการขึ้นต่อกันทำได้โดยการเลือกแถบสว่างแอตทริบิว หรือแถบสว่างฟังก์ชันการขึ้นต่อกัน

- การเพิ่มแอตทริบิว (Append Attribute) ทำโดยการใส่แอตทริบิวที่ต้องการเพิ่มเติมสลับกับการกด"ENTER" ออกจากการทำงานเพิ่มแอตทริบิวด้วยการกด "ESC" หรือกด "ENTER" ซ้ำ

- การเพิ่มข้อมูลฟังก์ชันการขึ้นต่อกัน (Append Function Dependency) ใช้แถบสว่างเลือกแอตทริบิวที่เป็นดีเทอร์มิแนนท์ ดังรูปที่ ก-7 กด "S" เพื่อเปลี่ยนกลับไปกลับมา ระหว่างการเลือกแอตทริบิวเพื่อใช้เป็นดีเทอร์มิแนนท์ กับการเลือกแอตทริบิวเพื่อใช้เป็นนันคีย์ ดังรูปที่ ก-8 ออกจากการทำงานเพิ่มฟังก์ชันการขึ้นต่อกันด้วยการกด "ESC"

#### **เลือกการทำงานแก้ไขข้อมูล (Edit)**

การแก้ไขข้อมูลแอตทริบิว ทำโดยใช้แถบสว่างเลือกแอตทริบิวที่ต้องการจะแก้ไข ออกจากการทำงานแก้ไขฟังก์ชันการขึ้นต่อกันด้วยการกด "ESC" ผลของการทำงานเป็น ดังรูปที่ ก-10

#### **เลือกการทำงานลบข้อมูล (Delete)**

การลบข้อมูลมีอยู่ 2 ชนิด คือ ลบข้อมูลแอตทริบิว และลบข้อมูลฟังก์ชันการขึ้นต่อกันทำได้โดยการเลือกแถบสว่างแอตทริบิว หรือแถบสว่างฟังก์ชันการขึ้นต่อกัน

- การลบแอตทริบิว (Delete Attribute) ใช้แถบสว่างเลือกแอตทริบิวที่จะลบออกผลการทำงานเป็นดังรูปที่ ก-9 สามารถยกเลิกคำสั่งลบแอตทริบิวแต่ละตัวได้ด้วย การกด "ENTER" ซ้ำอีกครั้ง กด "D" เพื่อออกจากการทำงานลบแอตทริบิว กด "ESC" เมื่อต้องการยกเลิกคำสั่งลบที่ได้ทำไปทั้งหมด

- การลบฟังก์ชันการขึ้นต่อกัน (Delete Function Dependency) ใช้แถบสว่าง เลือกฟังก์ชันการขึ้นต่อกันที่ต้องการจะลบออกดังรูปที่ ก-13 สามารถยกเลิกคำสั่งลบ ฟังก์ชันการขึ้นต่อกันแต่ละตัว ได้ด้วยการกด "ENTER" ซ้ำอีกครั้ง กด "D" เพื่อออกจากการทำงานลบแอตทริบิว กด "ESC" เมื่อต้องการยกเลิกคำสั่งลบที่ได้ทำไปทั้งหมด ผลการทำงานจะเป็น ดังรูปที่ ก-14

DATABASE NAME: SAMPLE	
<b>ATTRIBUTE</b> 1 MARKET 2 CAPACITY 3 TIME 4 SPEED 5 KILOGRAM 6 TYPE 7 AMOUNT 8 TOTAL	<b>FUNCTIONAL DEPENDENCY</b> MARKET, CAPACITY → TIME, SPEED TIME → SPEED, AMOUNT CAPACITY → KILOGRAM TYPE → MARKET KILOGRAM, TOTAL → AMOUNT
	<b>&lt;RELATION&gt;</b>  <APPEND> <EDIT> <DELETE> <EXIT>
	<b>&lt;ATTRIBUTE&gt;</b> NO. 1  <b>&lt;FD&gt;</b> NO. 3
	<b>&lt;DETERMINANT&gt;</b>  <b>&lt;NONDETERMINANT&gt;</b>
HELP CONTROL FOR SELECT (D) FOR DELETE (ESC) FOR EXIT	

รูปที่ ก-13 แสดงหน้าจอขณะลบข้อมูลฟังก์ชันการขึ้นต่อกัน

DATABASE NAME		SAMPLE
<b>ATTRIBUTE</b> 1MARKET 2CAPACITY 3TIME 4SPEED 5KILOGRAM 6TYPE 7AMOUNT 8TOTAL	<b>FUNCTIONAL DEPENDENCY</b> 1 CAPACITY—> KILOGRAM 2 TYPE—> MARKET 3 KILOGRAM,TOTAL—> AMOUNT	<b>&lt;RELATION&gt;</b>  <b>&lt;APPEND&gt;</b> <b>&lt;EDIT&gt;</b> <b>&lt;DELETE&gt;</b> <b>&lt;EXIT&gt;</b>
		<b>&lt;ATTRIBUTE&gt;</b> NO. 1  <b>&lt; FD &gt;</b> NO. 1
		<b>&lt;DETERMINANT&gt;</b>  <b>&lt;NONDETERMINANT&gt;</b>
<b>HELP (ENTER) For Select (D) For Delete (ESC) For Exit</b>		

รูปที่ ก-14 แสดงผลของการลบข้อมูลฟังก์ชันการขึ้นต่อกัน

### เลือกการทำงานสร้างรีเลชัน (Relation)

เพื่อให้โปรแกรมทำงานออกแบบฐานข้อมูล ซึ่งผลของการทำงานจะแสดงรีเลชันทั้งหมด ที่อยู่ในระดับบอยส์คอตต์ ดังรูปที่ ก-15

RELATION	ATTRIBUTE		
R1	MARKET	CAPACITY	TIME
R2	TIME	SPEED	AMOUNT
R3	CAPACITY	KILOGRAM	
R4	TYPE	MARKET	
R5	KILOGRAM	TOTAL	AMOUNT

รูปที่ ก-15 แสดงผลของการออกแบบรีเลชัน

กด F10 ยุติการทำงานกับรีเลชัน กลับสู่หน้าจอในรูปที่ ก-12

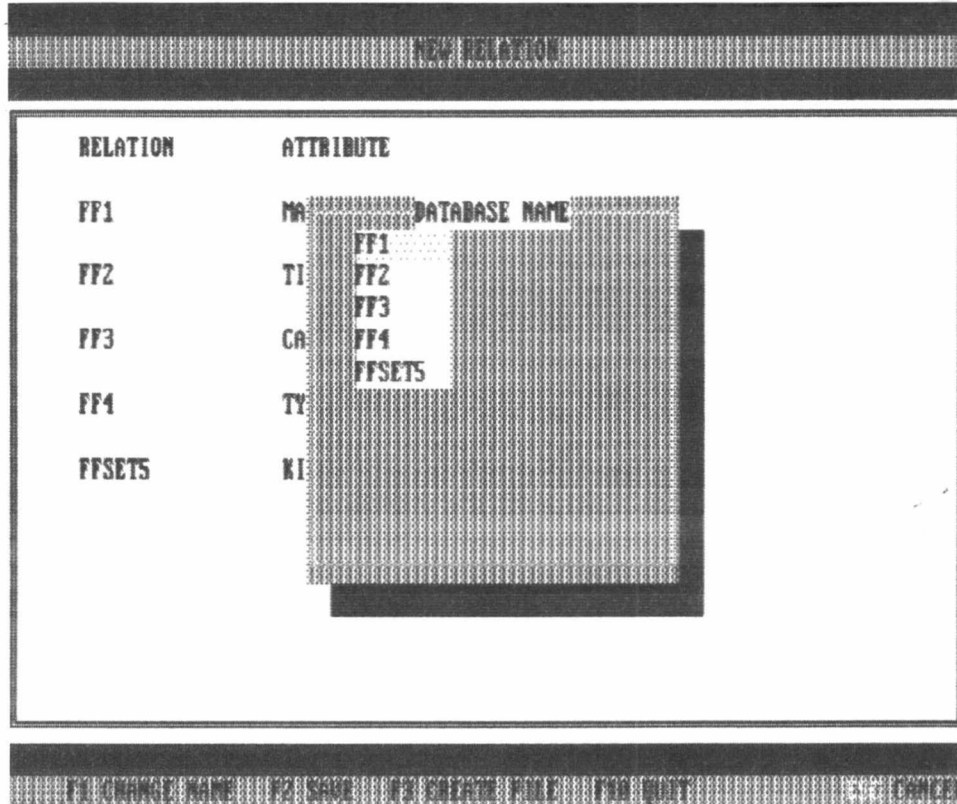
กด F1 เมื่อต้องการตั้งชื่อ หรือ เปลี่ยนชื่อของรีเลชัน โดยใช้แถบสว่างเลือกรีเลชันที่ต้องการ แล้วทำการเติมชื่อของรีเลชัน ซึ่งประกอบด้วยอักขระ A-Z, 0-9, "\_" ทั้งหมดไม่เกิน 8 ตัว

กต F2 เพื่อยืนยันการตั้งชื่อรีเลย์ที่ได้ทำลงไป ผลของการทำงานจะเป็น ดังรูปที่ ก-16

RELATION	ATTRIBUTE		
FF1	MARKET	CAPACITY	TIME
FF2	TIME	SPEED	AMOUNT
FF3	CAPACITY	KILOGRAM	
FF4	TYPE	MARKET	
FF5	KILOGRAM	TOTAL	AMOUNT

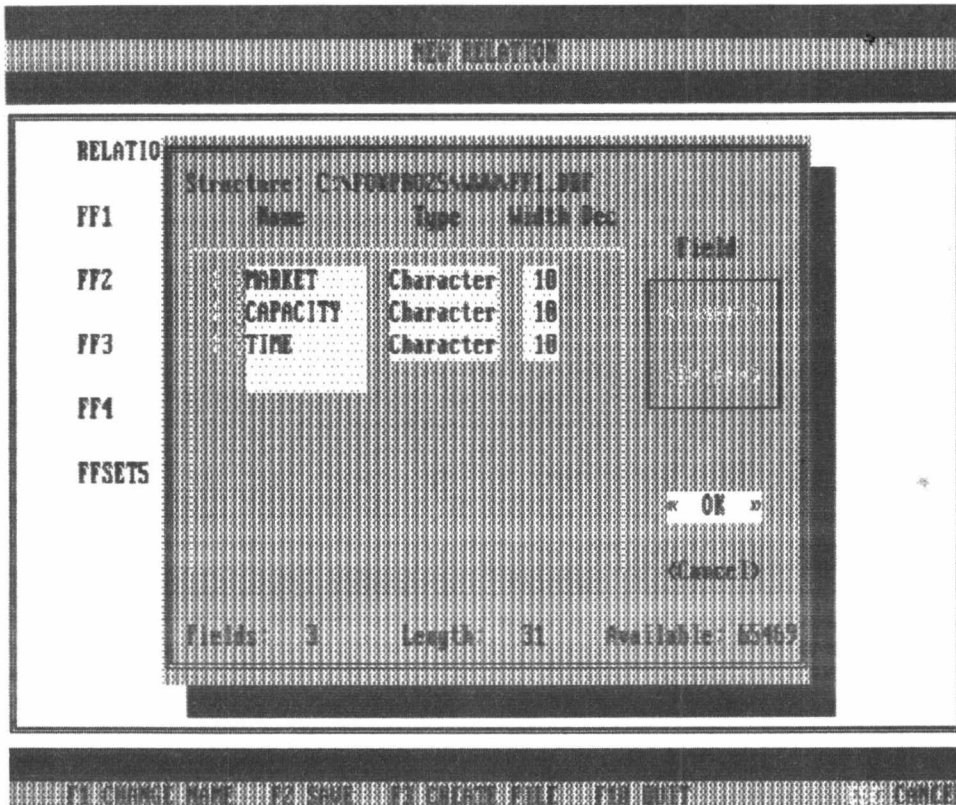
รูปที่ ก-16 แสดงผลของการเปลี่ยนชื่อรีเลย์

กด F3 เพื่อเข้าสู่การทำงานสร้างไฟล์ ซึ่งจะปรากฏหน้าต่างแสดงชื่อของรีเลชันทั้งหมดของฐานข้อมูลที่กำลังทำงานอยู่แล้วใช้แถบสว่างเลือกรีเลชันที่ต้องการ ดังรูปที่ ก-17



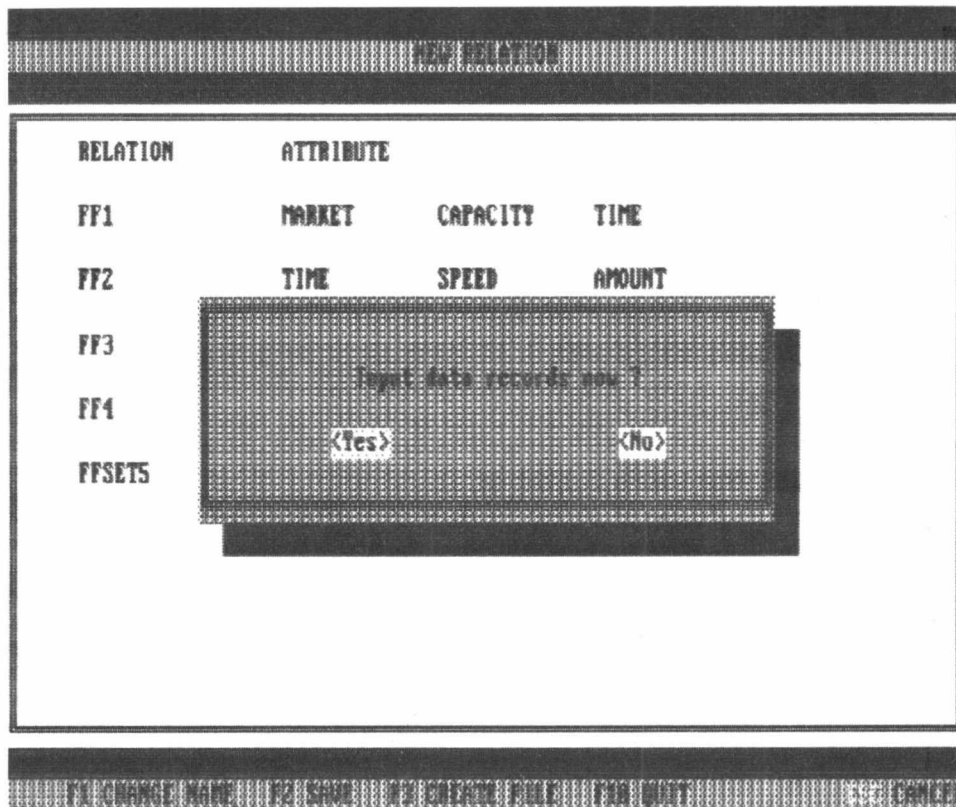
รูปที่ ก-17 แสดงหน้าจอของการสร้างไฟล์

ผลจากการเลือกกรีนั้นจะเป็นดังรูป ก-18 เป็นการรองรับการป้อนข้อมูลรายละเอียดเกี่ยวกับโครงสร้างของแอดทริบิวทั้งหมดภายในไฟล์อันได้แก่ ชนิดของข้อมูล(TYPE), ขนาดของข้อมูล(WIDTH) และจำนวนตัวเลขหลังทศนิยม(DEC)



รูปที่ ก-18 แสดงหน้าจอการป้อนข้อมูลรายละเอียดโครงสร้างของแอดทริบิว

ใช้แถบสว่างเลือก "OK" เพื่อยืนยันการใช้โครงสร้างของแอดทริบิวที่ใส่ลงไป หรือใช้แถบสว่างเลือก"CANCEL" เพื่อยกเลิกโครงสร้างของแอดทริบิวที่ใส่ลงไปผลการทำงานปรากฏ ดังรูปที่ ก-19



รูปที่ ก-19 แสดงหน้าจอภายหลังการสร้างไฟล์

ใช้แถบสว่างเลือก "YES" จะปรากฏหน้าจอดังรูปที่ ก-20 เพื่อรอการป้อนข้อมูลไฟล์

FF1	
Market	CUC
Capacity	BBB
Time	DDD
Market	FGFG
Capacity	FFF
Time	NNN
Market	DDD
Capacity	OO
Time	WWW
Market	.....
Capacity	.....
Time	.....

รูปที่ ก-20 แสดงหน้าจอของการรอรับข้อมูลลงไฟล์

กด "ESC" เพื่อยกเลิกข้อมูลที่ป้อนลงไปทั้งหมด หรือกด "Ctrl W" เพื่อยืนยันข้อมูลที่ป้อนลงไปทั้งหมด หลังจากนั้นหน้าจอจะกลับสู่รูปที่ ก-14

กด F10 ยุติการทำงานกับรีเลย์ชั้น กลับสู่หน้าจอในรูปที่ ก-10

## บรรณานุกรม

ดวง บงกชเกตุสกุล, คู่มือการใช้คำสั่งและฟังก์ชันฟอกร์โปร, หจก.เฮช-เอน การพิมพ์, 2534

รศ.ดร.ดวงแก้ว สวามิภักดิ์, ระบบฐานข้อมูล, หจก.เฮช-เอน การพิมพ์, 2534

Glenn A. Jackson., Relational Database Design with Microcomputer applications, Germany:  
Druckhaus Beltz, 1991

Igor T. Hawryskiewicz., Relational Database Design Introduction, New Jersey: Prentice  
Hall,Int., 1990

Toby J. Teorey., Database Modeling and Design, New Jersey: Prentice Hall,Intc., 1998