



ปีการศึกษา 2533

การจัดเก็บรายการสินค้า(stocks) โดยใช้คอมพิวเตอร์

โดย

นาย กิตติ กำลั้งแพทย

นาย ชัยวิทย์ วัฒนมงคลลาภ

อาจารย์ที่ปรึกษา

อาจารย์ ภาคกร หตะสังกาศ



ปฏิทินพจนานุกรมการศึกษา 2533

เรื่อง การจัดเก็บรายการสินค้า โดยใช้คอมพิวเตอร์



ผู้จัดทำ

1. นาย กิตติ กำลิ่งแพทย์

2. นาย ชัยวิทย์ วัฒนมงคลลาภ

..... อาจารย์ที่ปรึกษา

(.....)

เลขที่ T 33488 ก 3
เลขทะเบียน 027841
วัน, พจน. ปี 12 ก.ค. 34

027841

ยพ.
กชก
๒๓๓๗

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่ใช่ว่าจะเผยแพร่ไปใช้ประโยชน์ตามทางอื่น

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดเก็บรายการสินค้าโดยให้คอมพิวเตอร์

นาย กิตติ กำลังแพทย์

นาย ชัยวิทย์ วัฒนมงคลลาภ

อาจารย์ที่ปรึกษา

อาจารย์ ภากร หตะสังกาศ

ปีการศึกษา 2533

บทคัดย่อ

ปริญญาชั้นมัธยมต้นเป็นการศึกษาถึงการประยุกต์ใช้เครื่องไมโครคอมพิวเตอร์ (MICRO COMPUTER) ในงานจัดเก็บรายการสินค้า ซึ่งสามารถบอกรายละเอียด เช่น การคำนวณราคาต้นทุนของสินค้า ค่าแรงงานต่อชิ้น การรับสินค้าเข้า (receive List) การส่งสินค้าออก (issue List) รายงานรายการของวัสดุ (Material Report) รายงานสินค้าที่เหลืออยู่ในคลังเก็บสินค้า (Stock Report) ซึ่งสามารถดูข้อมูลต่างๆที่กล่าวมาแล้วได้จากจอแสดงผล (Display Monitor) หรือพิมพ์ข้อมูลที่ต้องการเป็นเอกสาร (Hard copy) ได้จากเครื่องพิมพ์ (Printer) ที่ต่อพ่วงอยู่ก็ได้ ซึ่งโปรแกรม (Program) นี้สามารถจะใช้งานได้กับทุกๆ หน่วยงานหรือห้างร้านหรือบริษัทไม่จำกัดประเภทของสินค้าเพราะจัดเก็บสินค้าแต่ละประเภทเป็นรหัส (Code) และสามารถใช้กับเครื่องคอมพิวเตอร์ทั่วไปที่คล้ายคลึงกับเครื่องไอบีเอ็ม รุ่น เอ็กซ์ที หรือ เอที (Compatible IBM model XT or AT) ผู้ใช้ก็ไม่จำเป็นต้องมีความรู้ทางด้านคอมพิวเตอร์มาก่อนเพราะโปรแกรมได้จัดทำ คำอธิบาย ภาษาไทยประกอบอยู่ด้วย ซึ่งได้มีคำอธิบายแต่ละคำสั่งหรือทั้งระบบของโปรแกรมทั้งหมดด้วย ซึ่งจะสะดวกในการนำไปใช้งานและการพัฒนาประยุกต์ใช้งานได้ต่อไป ซึ่งคณะผู้จัดทำนั้นจะจะให้โปรแกรมนี้สามารถทำงานเก็บอยู่ใน แผ่นฟลอปปีดิสก์ (Floppy disk) ขนาด 1.2 เมกะไบต์ (Megabyte) เพื่อสะดวกในการใช้งานกับเครื่องที่ไม่มีฮาร์ดดิสก์ (Hard disk) และสามารถเรียกใช้งานบนระบบปฏิบัติการ (Operating System) ได้โดยไม่ต้องเรียกใช้โปรแกรมสำเร็จรูป (Package Program)

Stock Inventory Control

Mr. Kitti Kamlangpat

Mr. Chaiwat Wattanamongkonlap

Adviser

Mr. Pakorn Hutasingkas

Graduate Year 1990

ABSTRACT

This purpose of project is to develop the application software running on Micro-computer. This application program is the stock inventory control system. It keeps the record of price calculation, labor per piece, receive list, issue list, material report, inventory level. The information can be viewed on display monitor or hard copy which is connected to the micro-computer system.

This application is widely implemented for any business without any limitation because it records the inventory part by digit code. It can be installed and running on any compatible XT/AT personal computer.

It does not need computer knowledge for user to run this program because there is Thai instruction explain step/procedure to run this program that make it easy for users. Also ease of future develop for more requirements.

The program is recorded in the 1.2MB floppy disk which can run on computer which has no hard disk installed and can run without other software support needed.

สารบัญ

	หน้า
บทที่ 1 บทนำ	4
บทที่ 2 ความเป็นมา วัตถุประสงค์ หน้าที่และความสำคัญ ของการจัดซื้อ	8
บทที่ 3 คอมพิวเตอร์กับการจัดซื้อ	15
บทที่ 4 ระบบจัดการฐานข้อมูล (Database Management System)	23
บทที่ 5 ดีเบสทีฟพลัส (dBASE III Plus)	27
บทที่ 6 การทำงานของโปรแกรม	77
บทที่ 7 แพลนผังการทำงานและโปรแกรม	89
บทสรุป	165
บทวิจารณ์	166
กิตติกรรมประกาศ	167
หนังสืออ้างอิง	168

บทที่ 1

บทนำ

เนื่องจากโครงการนี้จะเป็นการนำคอมพิวเตอร์ไปใช้ในระบบจัดเก็บสินค้า ซึ่งขนาดของร้านค้าเล็กจนถึงบริษัทใหญ่สามารถนำไปใช้ได้โดยไม่คำนึงถึงประเภทของสินค้าและผู้ใช้ (USER) ก็ไม่จำเป็นต้องมีความรู้ทางด้านการใช้เครื่องคอมพิวเตอร์มากนัก การเก็บรายละเอียดก็สะดวกเพราะสินค้าแต่ละชนิดก็ใช้รหัสสินค้าเมนูรายละเอียดมีแสดงก็เป็นภาษาไทยและโครงการนี้มุ่งให้ใช้ (support) กับเครื่องคอมพิวเตอร์ส่วนบุคคลไอบีเอ็ม หรือที่คล้ายคลึงกันกับรุ่นเล็กที่ หรือ เอที ซึ่งในการดำเนินงานของโครงการได้ศึกษาระบบของการจัดเก็บสินค้าคงคลังและความเป็นไปได้ ที่จะนำโมดูลคอมพิวเตอร์มาใช้งาน และการวางแผนของการจัดเก็บข้อมูลการศึกษาโปรแกรม (Software) ที่จะนำมาใช้ในระบบและการเขียนแผนผังการทำงานของโปรแกรม (Flow Chart) การวางโครงสร้าง (Structure) การเขียนโปรแกรมระบบงานต่างๆ

ได้ดำเนินงานทั้งหมดเป็น 2 ขั้นตอน คือ ขั้นตอนแรกนี้ภายใน ภาคเรียนที่ 1 ได้ทำการเขียนแผนผังการทำงานของโปรแกรมทั้งหมดรวมถึงตัวโปรแกรมด้วยซึ่งใช้โปรแกรมสำเร็จรูปดีเบสทีฟพลัส (Dbase III+) ส่วนขั้นตอนที่ 2 จัดทำในภาคเรียน 2 ได้ทำการประยุกต์การใช้งานของโปรแกรมให้สามารถเก็บบนแผ่นฟลอปปีดิสก์ 1.2 เมกะไบต์และสามารถทำงานบนระบบปฏิบัติการได้ ซึ่งได้นำโปรแกรมคลิปเปอร์ (Clipper) มาคอมไพล์ (compile) โปรแกรมที่เขียนด้วยดีเบสทีฟพลัสซึ่งได้ทำการแก้ไขเพิ่มเติมให้สามารถใช้ได้กับคลิปเปอร์ซึ่งเมื่อคอมไพล์แล้วสามารถทำงานได้เร็วและสะดวกขึ้นเพราะไม่ต้องเรียกใช้โปรแกรมดีเบส

1.1 แนวทางการวิจัย และทฤษฎีที่เกี่ยวข้อง

ขั้นตอนการวิเคราะห์และออกแบบระบบ (System Analysis & Design) ในการออกแบบและดำเนินการได้ปฏิบัติตามขั้นตอนของการวิเคราะห์และออกแบบระบบ (System Analysis & Design) ซึ่งมีขั้นตอนดังนี้

1.1.1 วิเคราะห์ระบบ (System Analysis)

ก. การศึกษาเบื้องต้น (Preliminary Study) เป็นขบวนการแรกในการพัฒนา เพื่อที่จะเข้าใจปัญหาและสาเหตุที่แท้จริงของปัญหา กำหนดวัตถุประสงค์ของระบบใหม่ตลอดจนขอบข่ายที่จะแก้ปัญหา ซึ่งจะคำนึงถึงความเหมาะสมทางด้านเศรษฐกิจ เทคนิค การปฏิบัติงานในองค์การและความเหมาะสมของงานนั้น ๆ ขั้นตอนได้แก่

- กำหนดปัญหาและความต้องการ (Determination of Problem and Requirement) โดยการทำซ้ำเท็จจริง
- กำหนดวัตถุประสงค์ (Specification of objective)
- การศึกษาความเหมาะสม (Feasibility Study)
- การวางแผนงาน (System planning)

ท. การวิเคราะห์ระบบงานปัจจุบัน (Present system analysis)

เมื่อกำหนดแผนงานเรียบร้อยแล้วก็จะพร้อมที่จะดำเนินการวิเคราะห์ระบบได้โดยจะศึกษาจากรายละเอียดของระบบ ซึ่งจะเป็นแนวทางการปรับปรุง

- การวิเคราะห์ข้อมูลที่ได้รวบรวมมา (Analysis of Collected Data) เป็นการวิเคราะห์ข้อมูลที่ใช้ทั้ง input output ว่ามีความสัมพันธ์กันอย่างไรมีที่มาอย่างไร
- การวิเคราะห์กระบวนการของระบบ (Analysis of System Process) จากการทำงานของระบบปัจจุบันและปัญหาที่เกิดขึ้นเพื่อจะนำมาวิเคราะห์หาความต้องการของระบบ และแนวทางปรับปรุง

1.1.2 การออกแบบ (System Design)

การออกแบบระบบ อาจจะเป็นการออกแบบใหม่ทั้งระบบหรือปรับปรุงระบบเก่าเพียงบางส่วน ซึ่งผู้วิเคราะห์งานจะนำผลจากการศึกษาและวิเคราะห์ ความสามารถขององค์การมาพิจารณา เพื่อออกแบบระบบงานใหม่ให้เหมาะสม ขั้นตอนการออกแบบได้แก่

ก. การออกแบบผลลัพธ์ (Output Design)

การออกแบบจะมีวัตถุประสงค์เพื่อเป็นสื่อความหมายในการดำเนินการให้เข้าใจง่ายและเหมาะสม การออกแบบจะดูจาก ผลลัพธ์ของระบบเดิม หาอุปกรณ์การเก็บข้อมูลที่เหมาะสม ตลอดจนวางรูปแบบของรายงานที่คอมพิวเตอร์จะแสดงออกมา

ข. การออกแบบข้อมูลนำเข้า (Input Design)

ส่วน Input ของระบบใดระบบหนึ่งประกอบด้วย การรวบรวมข้อมูลตรวจสอบความถูกต้อง การลงทะเบียนและการบันทึกข้อมูลบนอุปกรณ์ของคอมพิวเตอร์ การออกแบบต้องพิจารณาถึงสภาพทั่วไปขององค์การ เพื่อให้ข้อมูลที่ได้นั้นสอดคล้องกับความต้องการของ เสา่ก่พท กล่าวคือรูปแบบ และเนื้อหาของข้อมูลหรือสิ่งที่นำเข้าตลอดจนประเภทข้อมูล ปริมาณและความถี่ในการใช้งานสัมพันธ์อยู่กับ เสา่ก่พท ที่ต้องการ

ค. การออกแบบแฟ้มข้อมูล (File Design)

การออกแบบแฟ้มข้อมูลจะพิจารณาการออกแบบที่เกี่ยวข้องกับโครงสร้าง การจัดองค์การ (Organization) และอุปกรณ์ที่ใช้เก็บข้อมูล (Storage service)

- File Structure กำหนดโครงสร้างของแฟ้มข้อมูลว่าประกอบด้วยเรคคอร์ด (Record) อะไรบ้างและ file นั้นใช้เป็น Master file หรือ transaction file
- File organization จะต้องพิจารณาควบคู่กับการประมวลผลการใช้ข้อมูล ซึ่งอาจเป็นแบบ Sequential Organization หรือ Direct Access Organization Storage Device ควรจะคำนึงถึงขนาด ความเร็วและวิธีการ access ซึ่งใช้ในการออกแบบจะดูความเหมาะสมของเครื่องไมโครคอมพิวเตอร์ที่ใช้ด้วย

ง. การออกแบบประมวลผลด้วยคอมพิวเตอร์ (Computer System Design) จะมีการวางระบบงานอย่างกว้างๆ เพื่อเป็นแนวทางในการกำหนดรายละเอียดของระบบย่อยๆ ภายในระบบงานนั้นๆ เช่น การกำหนดรายละเอียดเกี่ยวกับ output Input และ Process ซึ่งรายละเอียดเหล่านี้จะอาศัยจากรายละเอียดของเครื่องคอมพิวเตอร์และข้อกำหนดรายละเอียดของโปรแกรม

1.1.3 การพัฒนาและการออกแบบติดตั้ง (Development and Implementation)

เป็นการวางแผนและพัฒนาสิ่งที่ได้ออกแบบไว้แล้ว และนำไปใช้งานกับระบบจริง ตั้งแต่การออกแบบโปรแกรม แก้ไขข้อผิดพลาดและทดสอบกับข้อมูลจริง ๆ เป็นต้น

1.1.4 การเปลี่ยนแปลงแก้ไขในขนาด (Maintenance)

จะกระทำเมื่อติดตั้งระบบใหม่เรียบร้อยแล้ว แล้วมีการเพิ่มเติมเพื่อขยายขนาดของระบบและแก้ไขข้อผิดพลาดซึ่งตรวจสอบได้ ภายหลัง เมื่อติดตั้งระบบแล้ว

1.2 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย

1. ช่วยลดภาระงานประจำที่ซ้ำซากและเชิงซ้ำให้เป็นไปอย่างรวดเร็ว
2. ช่วยให้การประมวลผลข้อมูลเป็นไปอย่างถูกต้องและแม่นยำมากขึ้น
3. ช่วยให้ได้ข้อมูลข่าวสาร เพื่อการบริหารแก่ผู้บริหาร ให้เป็นไปอย่างสะดวก ถูกต้อง และรวดเร็ว

4. ช่วยพัฒนาความคิดให้เป็นการยอมรับในเทคโนโลยีใหม่ๆ ในการพัฒนาระบบให้ทันสมัย และดีขึ้นกว่าระบบเดิม
5. เพื่อช่วยวางแผนในการขยายตัวและปรับปรุงระบบในอนาคตได้สะดวกและรวดเร็วยิ่งขึ้น

1.3 เครื่องมือและอุปกรณ์ที่ใช้ในงานวิจัย

1.3.1 เครื่องคอมพิวเตอร์ IBM PC XT หรือ AT หรือ Compatible

- 360 Kb Floppy Disk 1 Ea
- 1.2 Mb Floppy Disk 1 Ea
- Monochrome monitor
- Hercules graphics/printer port
- Thai Card

1.3.2 เครื่องพิมพ์ ขนาด 132 Column (แคร์ขนาดยาว)

1.3.3 โปรแกรม Dbase III+

1.3.4 โปรแกรม Clipper V5.0

บทที่ 2

ความเป็นมา วัตถุประสงค์ หน้าที่และความสำคัญของการจัดซื้อ

(Introductions)

การจัดซื้อโดยทั่วไปมีอยู่ 2 ประเภทใหญ่ๆ ด้วยกันคือ การซื้อมาเพื่อขายต่อ และ การซื้อมาเพื่อใช้ หรือเพื่อเปลี่ยนสภาพให้เป็นของอย่างอื่นการซื้อมาเพื่อขายต่อ (Purchasing for Resale) เป็นการซื้อโดยพ่อค้าหรือผู้เก็งกำไรเดิมทีนั้นเป็นความรับผิดชอบของพ่อค้าในการที่จะเสาะแสวงหาสินค้ามาขาย ซึ่งก็เป็นผลทำให้มีการค้นพบโลกใหม่ เช่น การค้นพบมหาสมุทรอินเดีย เป็นต้น ในอดีตพ่อค้าใช้เวลาเพียงบางส่วนเกี่ยวข้องกับปัญหาการขาย แต่เวลาส่วนใหญ่ของเข่าใช้ในการเสาะแสวงหาสินค้ามาเพื่อขาย โดยการจัดซื้อของแปลกๆใหม่ๆ ที่เหมาะสมและเป็นที่ต้องการของลูกค้าเขาจำเป็นต้องศึกษาปัญหาต่างๆ ต้องรู้เทคนิคและวิธีที่จะคาดคะเนว่าผู้บริโภคต้องการอะไร และจะต้องซื้อหามาในราคาที่ไม่บวกกำไรเข้าไปแล้ว ยังเป็นที่พอใจของผู้บริโภค รวมทั้งคุณภาพและบริการก็เป็นที่ถูกใจด้วยเช่น ในหน้าผนพ่อค้าอาจจะเห็นซื้อสินค้ามาผิด เขาก็จะล้มเหลวในธุรกิจของเขาได้ การซื้อมาเพื่อใช้หรือเปลี่ยนสภาพ (Purchasing for Consumption or Conversion) กิจการอุตสาหกรรมทุกกิจการย่อมต้องการวัสดุและของใช้ต่างๆ มาใช้ในการผลิต ก่อนเริ่มทำการผลิตวัสดุที่จะใช้จะต้องมีอยู่พร้อม และจะต้องมั่นใจว่ามีเพียงพอตามระยะเวลาของการผลิตที่กำหนดไว้

ปัญหาของผู้ซื้อในธุรกิจอุตสาหกรรม (Industrial buyer) จึงอยู่ที่ว่า เขาควร จะผลิตอะไรจะใช้อะไรเป็นส่วนประกอบ จะซื้อวัสดุส่วนประกอบจากที่ไหน และจะต้องประสานงานอย่างใกล้ชิดระหว่างฝ่ายซื้อและฝ่ายขายในเรื่องการพยากรณ์ยอดขาย และการกำหนดตารางการผลิตเพราะถ้าหากคิดผิด จะทำให้เกิดความเสียหาย คือต้องเสียค่าใช้จ่ายอันเกิดจากความล่าช้า การผลิตขาดประสิทธิภาพ ได้สินค้าที่น้อยกว่ากำหนด ผิดข้อตกลงในการนำส่งสินค้าซึ่งจะทำให้ลูกค้าไม่พอใจ

2.1 ความหมาย

การจัดซื้อ (Purchasing)

เป็นกิจกรรมที่รับผิดชอบในการประกันว่า บรรดาวัสดุของใช้และอุปกรณ์ที่ต้องใช้ในกิจการนั้นมีพร้อมอยู่เสมอ แต่อาจมีหลายท่านที่เข้าใจสับสนกันระหว่างการจัดซื้อ (Purchasing)

และการจัดหา (Procurement) ตามความหมายที่ใช้กันทั่วไป คำว่าการจัดหานั้นมีความหมายกว้างคลุมไปถึงการเช่า การซื้อ การผลิตขึ้นมาเอง การเปลี่ยน โอนและอื่นๆ ที่ทำให้ได้มาซึ่งพัสดุและอุปกรณ์ที่ต้องการใช้ ไม่ได้หมายถึงการซื้ออย่างเดียว คำว่าจัดหานี้ใช้แพร่หลายในกิจการทหาร

2.2 วัตถุประสงค์การจัดซื้อ

ในการศึกษาเกี่ยวกับการบริหารการจัดซื้อ จำเป็นต้องเข้าใจวัตถุประสงค์ให้ชัดเจนเพราะเป็นจุดหมายหรือเป้าหมายที่ต้องทำให้สำเร็จกล่าวคือกำหนดให้แคบและจำเพาะเจาะจงลงไป เรื่องนี้ทางด้านการบริหารธุรกิจที่คิดค้าหากำไร และโดยเฉพาะในกิจการอุตสาหกรรมต่างๆได้กำหนดวัตถุประสงค์ของตนไว้ชัดเจน ซึ่งพอสรุปเป็นตัวอย่างมาศึกษาได้บ้าง ดังนี้

2.2.1 ตัวอย่างที่ 1 วัตถุประสงค์ของงานการจัดซื้อสำหรับกิจการด้านอุตสาหกรรมก็เพื่อเสริมสร้างกำไรให้แก่กิจการในด้านที่การในด้านที่การจัดซื้อเกี่ยวข้องกับ ส่วนวัตถุประสงค์เฉพาะที่ช่วยให้บรรลุวัตถุประสงค์หลักนั้น สรุปสั้น ๆ ได้ดังนี้

- เพื่อให้ได้มาซึ่งพัสดุ (ที่ใช้ในการผลิต) ที่มีคุณสมบัติเหมาะสมที่สุดกับที่ตั้งใจจะใช้
- เพื่อให้ได้มาซึ่งจำนวนพัสดุที่สามารถสนับสนุนการผลิตได้อย่างต่อเนื่องไม่ขาดตอน และในเวลาเดียวกันจะต้องให้การลงทุนในสต็อกต่ำที่สุดเท่าที่จะทำได้ โดยสอดคล้องกับลักษณะจำนวนสั่งซื้อที่ประหยัด และสอดคล้องกับสภาวะของตลาด
- เพื่อให้ได้มาซึ่งพัสดุของใช้และการบริการในต้นทุน ที่ต่ำที่สุดเมื่อเปรียบเทียบกับคุณสมบัติและการบริการที่ต้องการ
- เพื่อเป็นแหล่งข่าวเสรีที่จะให้หน่วยงานอื่น ๆ ที่สนใจและเกี่ยวข้องได้ทราบข่าวสารเกี่ยวกับผู้ขาย (Suppliers) และคาดว่าเป็นผู้ขาย (Potential Supplier) ในเรื่องพัสดุใหม่ๆกรรมวิธีในการผลิตใหม่ ๆ วิทยาการใหม่ๆสภาวะเศรษฐกิจ และตลอดจนการพยากรณ์ราคา

2.2.2 ตัวอย่างที่ 2 วัตถุประสงค์หลักของการจัดซื้อสำหรับอุตสาหกรรมการผลิตพอรวบรวมได้ดังนี้

- เพื่อสนับสนุนการผลิตด้วยพัสดุวัตถุดิบ เพื่อให้การผลิตเป็นไปตามกำหนดการผลิต (Manufacturing schedule) ตลอดไปโดยไม่ขาดตอน

- เพื่อให้การลงทุนในสต็อกต่ำที่สุด โดยไม่เสียผลทางด้านความปลอดภัยและการประหยัด
- เพื่อหลีกเลี่ยงปัญหาพัสดุซ้ำกัน พัสดุเสีย และพัสดุล้าสมัย
- เพื่อจัดหาพัสดุในต้นทุนที่ต่ำที่สุด โดยไม่เสียผลทางด้านคุณสมบัติของพัสดุและบริการที่ต้องการ
- เพื่อรักษาคุณสมบัติมาตรฐานของพัสดุโดยคำนึงถึงความเหมาะสมในการใช้เป็นหลัก
- เพื่อให้กิจการมีกำไร ตั้งอยู่ในภาวะที่มีการแข่งขันได้ ในงานของส่วนพัสดุก่อซื้อที่ส่วนเกี่ยวข้องด้วย

2.3 หน้าที่ของการจัดซื้อ

การดำเนินการจัดซื้อภายใต้แผนงานจัดซื้อของแต่ละกิจการย่อมมีมากน้อยแตกต่างกันออกไป และงานนั้นๆ ก็มีลักษณะทั้งที่เป็นงานบริหาร และงานประจำวัน หน้าที่ของแผนกจัดซื้อโดยทั่วไป อาจแยกออกได้เป็นหัวข้อใหญ่ ๆ ได้ดังนี้

2.3.1 หน้าที่หลัก

- งานเก็บและรวบรวมข่าวสารเกี่ยวกับการจัดซื้อ ข่าวสารที่สำคัญ ได้แก่ ข้อมูลเกี่ยวกับการซื้อ ราคา พัสดุคงคลัง อัตราการใช้พัสดุ แหล่งขาย คุณลักษณะ เฉพาะ (Specification) และแคตตาล็อก (Catalog)
- งานการค้นคว้าวิจัย ซึ่งได้แก่ การศึกษาตลาด การศึกษาพัสดุ การวิเคราะห์ราคาพัสดุ (Value analysis) การสอบสวนแหล่งขาย การตรวจโรงงานผลิตของผู้ขายการพัฒนาแหล่งขายใหม่ การพัฒนาพัสดุและแหล่งขายสำรอง เป็นต้น
- การปฏิบัติงานซื้อ ซึ่งมีการตรวจสอบคำขอหรือใบเบิก การขอราคา การวิเคราะห์ราคา การพิจารณาเลือกว่า จะซื้อด้วยการทำสัญญาหรือซื้อจากท้องตลาดโดยตรง กำหนดการซื้อและการนำส่ง การสัมภาษณ์พนักงานขาย การเจรจาในการทำ สัญญา การออกใบสั่งซื้อการตรวจสอบเงื่อนไขในสัญญาการติดตามเรื่องเกี่ยวกับ การนำส่ง การตรวจรับพัสดุ การตรวจสอบใบเสร็จ การมีหนังสือโต้ตอบกับผู้ขาย และการเจรจายึดหยุ่นกับผู้ขาย
- งานการจัดงานพัสดุ ซึ่งได้แก่การรักษาระดับพัสดุคงคลังให้ต่ำที่สุด รักษาคุณภาพ



ของพิสดุคงคลัง ปรับปรุงอัตราค่าธรรมเนียมของพิสดุคงคลัง การโยกย้ายพิสดุ การรวบรวมความต้องการ การหลักเลียงพิสดุคงคลังมากเกินไป และพิสดุค้าสมัย กำหนดมาตรฐานของหีบห่อและเครื่องบรรจุ ทำบัญชีวิสดุที่ส่งคืน และการรายงานสถานภาพของพิสดุตามระยะเวลา

- งานเบ็ดเตล็ดอื่น ๆ เช่น การประมาณราคา การจำหน่ายของเหลือใช้ของเสีย และของล้าสมัย ดำเนินการเกี่ยวกับนโยบายการช่วยเหลือซึ่งกันและกัน

2.3.2 หน้าที่รอง

หน้าที่เหล่านี้เป็นหน้าที่ที่แผนกจัดซื้อได้รับมอบหมายให้ปฏิบัติด้วย

- การพิจารณาว่าควรจะมีหรือซื้อหรือเช่า
- มาตรฐานพิสดุ (Standardization)
- คุณสมบัติเฉพาะ (Specification)
- การพิจารณาพิสดุที่ใช้ทดแทนกันได้
- การงบประมาณพิสดุ
- การควบคุมพิสดุ (Inventory Control)
- เลือกอุปกรณ์การลงทุน
- โครงการก่อสร้างต่าง ๆ
- โปรแกรมการผลิตในส่วนที่เกี่ยวข้องกับพิสดุ

2.3.3 หน้าทีอื่นที่เกี่ยวข้องกับการจัดซื้อ

หน้าที่ต่อไปนี้เป็นงานที่สัมพันธ์กับงานการจัดซื้อ บางกิจการก็ได้มอบหมายงานเหล่านี้

ให้อยู่ในความรับผิดชอบของแผนกจัดซื้อด้วย

- คลังพิสดุ กิจการต่างๆ มีจำนวนประมาณครั้งหนึ่งที่มีมอบหมายให้อยู่ในหน้าที่ของแผนก จัดซื้อด้วย
- การจัดงานการค้าเลียงขนส่ง การจัดแบ่งและการวางแผนค่าเลียงขนส่งพิสดุที่ จะนำเข้ามา ถือว่าเป็นหน้าที่ของการจัดซื้อ ความรับผิดชอบในการจัดงานการค้า เลียงขนส่งทั้งหมดที่กิจการต่างๆ ดำเนินการอยู่ มีจำนวนประมาณหนึ่งในสิ่งที่ มอบหมายให้อยู่ในหน้าที่ของแผนกจัดซื้อ

- การประกันภัย
- ที่ดินและอสังหาริมทรัพย์
- กิจเกี่ยวกับยานพาหนะและการซ่อมทำ
- การตรวจพัสดุ

2.4 การจัดซื้อกับการผลิตหรือการใช้

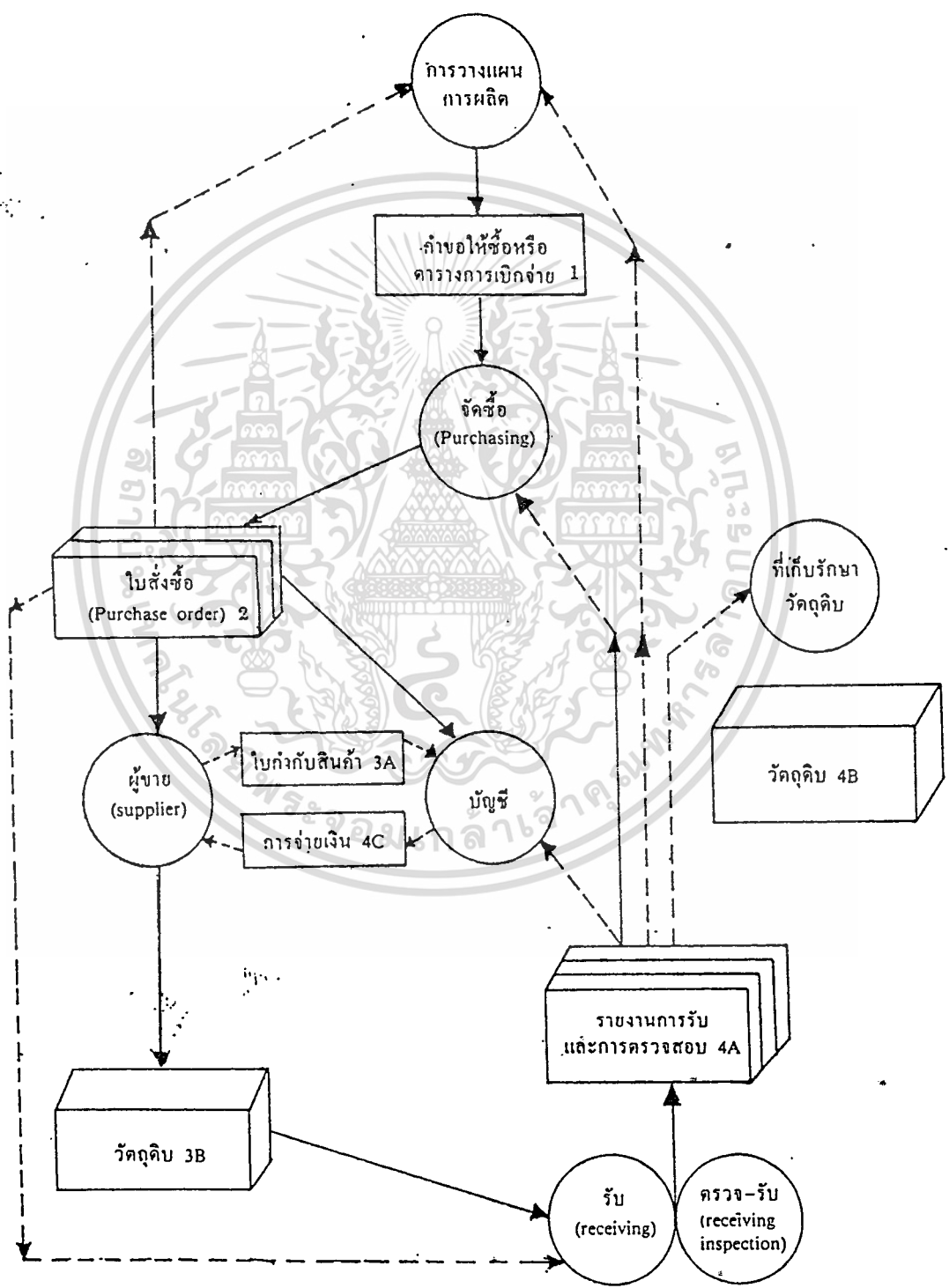
ปัญหาคุณสมบัติของพัสดุ การตกลงใจขั้นสุดท้ายในเรื่อง แบบและคุณสมบัติของพัสดุ ที่จัดซื้อซื้อแล้วแต่แผนกออกแบบหรือแผนกที่ใช้จะกำหนดอย่างไร แต่ปัญหาว่าจะเลือกซื้อที่ไหน เป็นเรื่องของแผนกจัดซื้อโดยตรง อย่างไรก็ตาม แผนกใช้ก็มีสิทธิที่จะเสนอแนะได้ในกรณีที่มีการกำหนดพัสดุเป็นอันตราสือชื่อ (brand) ซึ่งนับว่าเป็นการจำกัดขอบเขตของงานจัดซื้อ ดังนั้น การที่ได้ให้การซื้อเสรีมากขึ้น ก็มักจะเพิ่มเติมข้อความว่า "หรือพัสดุที่มีคุณสมบัติเหมือนหรือใกล้เคียง"(or equal) เข้าไปด้วย ทั้งนี้ เพื่อให้ฝ่ายจัดซื้อได้มีโอกาสเลือกพัสดุอย่างอื่นและ จากแหล่งอื่นที่คิดว่ามีความเหมาะสมในการใช้พอๆ กันมาพิจารณา เหตุผลก็คือว่า แผนกจัดซื้อจำเป็นต้องพัฒนาแหล่งซาวัวอย่างน้อย 2 หรือ 3 แห่งหรือมากกว่านั้นไว้เพื่อให้มีความมั่นใจว่าสามารถจัดซื้อได้ตลอดเวลาไม่ขาดแคลน แผนกผลิตหรือแผนกใช้พัสดุก็จะให้ความร่วมมือด้วยการทดสอบพัสดุที่แผนกจัดซื้อนำมานั้นว่าจะสามารถนำมาใช้ได้เพียงใดหรือไม่ ถ้าใช้ได้ต่อไป พัสดุนั้นจะอยู่ใน"รายการที่ยอมรับ" (Approved list) เมื่อเป็นเช่นนั้น ต่อไปแผนกจัดซื้อก็สามารถจัดซื้อพัสดุได้ตามความประสงค์ของผู้ใช้ได้ตลอดไป โดยมีสิทธิที่จะเลือกแหล่งซาวัวที่เห็นเหมาะสม ในการจัดทำคุณลักษณะเฉพาะของพัสดุนั้น แผนกจัดซื้อก็มีบทบาทที่จะให้ความเห็นได้เหมือนกัน คือ ถ้าเห็นว่าละเอียดและเป็นทางวิชาการมากเกินไปอาจเป็นอุปสรรคต่อการจัดซื้อได้ เช่น อาจทำให้จำกัดแหล่งซาวัว หรืออาจทำให้ราคาสูงขึ้นโดยไม่จำเป็น ก็เสนอความคิดเห็นเพื่อทำให้เข้าใจมาตรฐานพัสดุในท้องตลาด หรือยอมให้มีช่วงของความผิดพลาด (tolerances) บ้างพอสมควร

2.5 ตัวอย่างวิธีปฏิบัติเกี่ยวกับการดำเนินการจัดซื้อ

ตัวอย่างวิธีปฏิบัติที่แสดงข้างล่างนี้เป็นวิธีปฏิบัติของบริษัทอุตสาหกรรมการผลิตทั่วไป โดยจะแสดงให้เห็นว่าการผลิตเริ่มที่จุดใดและการเสนอความต้องการเข้ามาสู่แผนกจัดซื้ออย่างไรและสุดท้ายออกเป็นใบสั่งซื้ออย่างไร

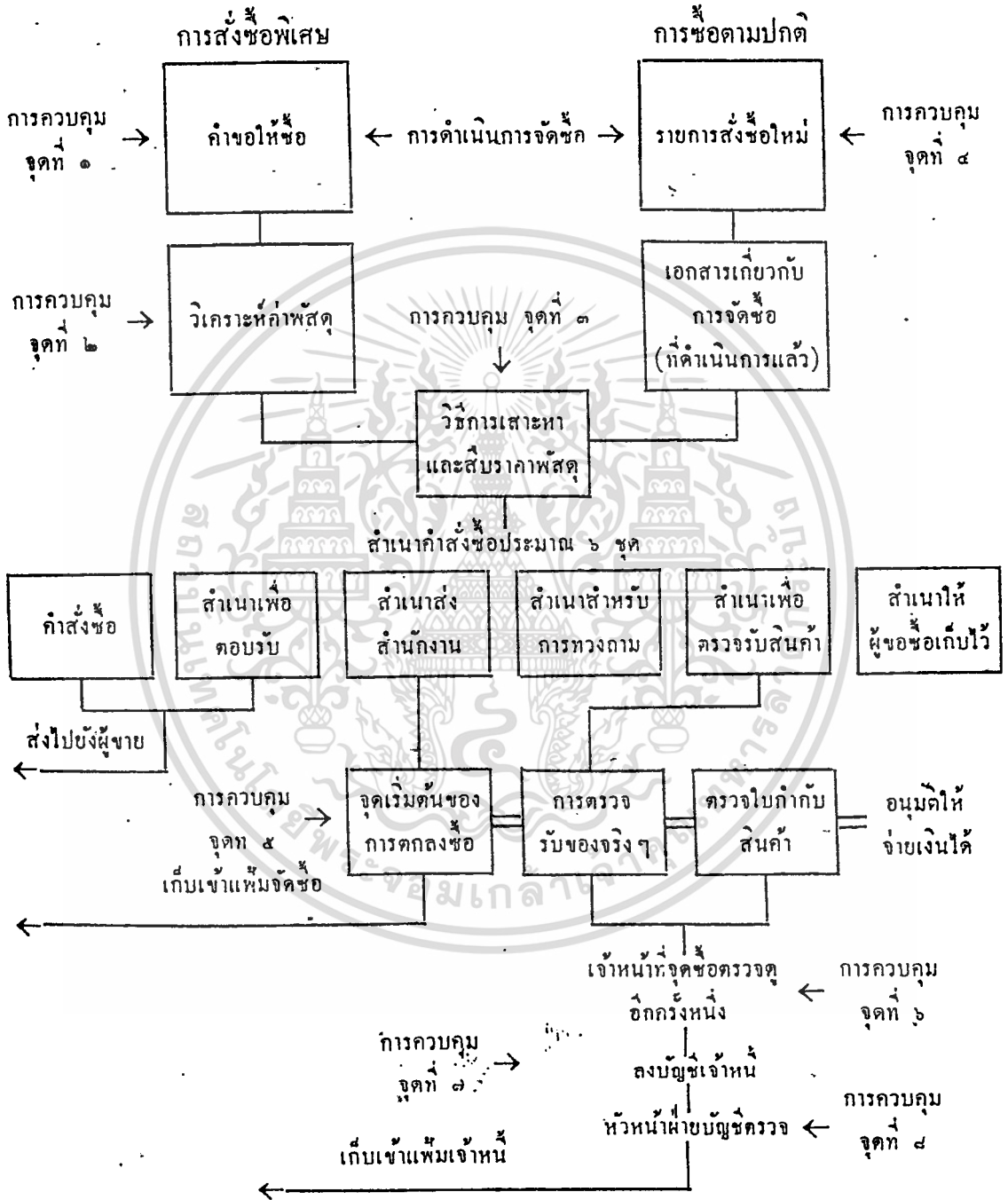
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป ๒-๑ แสดงขั้นตอนการจัดการพัสดุมาใช้ในการผลิต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป ๒-๒ ผังแสดงขั้นตอนวิธีการจัดซื้อ



- ตัวอย่างแบบฟอร์มต่าง ๆ
๑. ใบจัดหาหรือใบขอให้ซื้อวัสดุ
 ๒. ใบสั่งซื้อ
 ๓. ผลการตรวจรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

คอมพิวเตอร์กับการจัดซื้อ

(Computerized Purchasing)

การกำหนดการไหลของข่าวสาร

การไหลของข่าวสารเป็นเส้นเลือดตามแนวความคิดของการบริหารธุรกิจ การตัดสินใจที่ดีย่อมขึ้นอยู่กับข่าวสารที่สมบูรณ์ ถูกต้องแน่นอน และทันเวลา (timely, complete, and accurate information) แต่ที่จะพัฒนาและจัดให้มีข่าวสารที่สมบูรณ์ถูกต้องแน่นอนและทันเวลา เป็นปัญหาที่ยุ่งยากสับสนแก่นักบริหารเป็นประจำ

ในอดีต ปัจจัยสำคัญที่มีอิทธิพลต่อหลักการจัดหน่วยงาน คือ ความจำเป็นที่จะให้ผู้จัดการอยู่ใกล้ชิดกับแหล่งข่าว และปัจจัยนี้แหละที่เป็นอุปสรรคต่อการจัดแบบ "แยกการ" (decentralization) โดยทั่วไป เพราะผู้บริหารต้องการจะให้มีส่วนการบังคับบัญชาชั้นที่สุด เพื่อจะได้รับข่าวสารที่จะใช้บริหารธุรกิจอยู่ตลอดเวลา

ปัจจุบันนี้ การพัฒนาอุปกรณ์ Electronic Data Processing (EDP) และการนำเอามาใช้ในรูปของ Integrated Data Processing systems (IDP) ทำให้ผู้บริหารสามารถควบคุมการไหลของข่าวสารต่างๆ ได้รวดเร็ว สะดวกและถูกต้องแน่นอน อย่างไม่น่าเชื่อ

ระบบการจัดซื้อก็ขึ้นอยู่กับการไหลของข่าวสารเช่นเดียวกัน และข่าวสารที่ต้องการนอกจากข่าวสารจากหน่วยงานอื่น ๆ ในบริษัทแล้ว ยังมีข่าวสารจากภายนอกบริษัทด้วย

ทุกหน่วยงานในบริษัทย่อมจำเป็นต้องมีข่าวสาร และ หรือต้องการข่าวสารจากแผนกจัดซื้อ ข่าวสารสำคัญที่ส่งให้แผนกจัดซื้อแยกได้เป็นสองประเภท คือ

1. ข้อความแสดงความต้องการพิเศษ และการบริการที่ได้รับจากภายนอกบริษัท
2. ค่าขอข่าวสารที่แผนกจัดซื้อมีอยู่หรือที่แผนกจัดซื้อได้รับจากภายนอกบริษัท

เพื่อให้เข้าใจแหล่งข่าวและสาระสำคัญของข่าวที่ส่งมาซึ่งแผนกจัดซื้อ ให้ศึกษาจาก พังและรายละเอียดย่อย ๆ ดังนี้

3.1 การวางแผน

ให้ข่าวสารเกี่ยวกับความต้องการในระยะยาวที่เกี่ยวข้องกับสิ่งอำนวยความสะดวกในเรื่องพิเศษและการบริหารของบริษัท การวางแผนที่ดีมีประสิทธิภาพย่อมมีความสำคัญเป็นพิเศษในการเตรียมวัตถุดิบโดยเฉพาะในช่วงที่จัดหาอาจ

3.2 การพยากรณ์การขาย

การพัฒนาการพยากรณ์การขายที่รัดกุมเป็นข้อ่าสารที่เป็นประโยชน์ต่อการจัดซื้อในด้าน การวางแผนจัดหาในอีก 12 เดือนข้างหน้า ปกติการดำเนินงานธุรกิจจะบังเกิดผลดีมีประสิทธิภาพสูงสุดเมื่อมีแผนความต้องการที่มีกำหนดการและวิธีที่จะได้พัสดุอย่างมีแบบแผน คราบเท่าที่แผนการจัดซื้อได้ทราบรายการและจำนวนโดยประมาณล่วงหน้าพอสมควรมากเท่าใด ก็อยู่ในฐานะที่ได้เปรียบในการเกิดความสัมพันธ์ระหว่างสภาวะของตลาดและความต้องการของบริษัทมากเท่านั้น

3.3 การงบประมาณและการควบคุมการเงิน

ข้อ่าสารการงบประมาณช่วยในการพิจารณาข้อ่าสารด้านการวางแผนและการพยากรณ์การขายว่า จะมีอุปสรรคด้านการควบคุมการเงินอย่างไรบ้าง ข้อ่าจำกัดหรืออุปสรรคดังกล่าวอาจพิจารณาเป็นข้อ่าค่าใช้จ่ายของการจัดซื้อ หรือพิจารณาแก้ไขไปทางอื่น ๆ ได้แทนที่จะแก้ไขด้าน การควบคุมพัสดุคงคลังแบบ "ซื้อเมื่อต้องการ" (Buy-as-required)

3.4 การบัญชี

ให้ข้อ่าในเรืองการชำระเงินแก่ผู้ขาย ข้อมูลศึกษาเกี่ยวกับข้อ่าตกลงใจว่าจะผลิตหรือซื้อข้อ่าเปรียบเทียบกับข้อ่าค่าใช้จ่ายจริงกับงบประมาณ เป็นต้น

3.5 กฎหมาย

งานด้านกฎหมายจะให้ข้อ่าสารเกี่ยวกับสัญญาวิธีปฏิบัติและอื่น ๆ ทั้งนี้เพราะงานจัดซื้อเป็นงานสำคัญที่บริษัทกระจาย อำนาจให้ทำสัญญาซื้อพัสดุและการบริการภายนอกได้

3.6 การช่าง

ฝ่ายช่างให้ข้อ่าสารด้านชนิดของพัสดุที่ต้องการ คุณลักษณะเฉพาะของคุณสมบัติที่ต้องการซึ่งแผนการจัดซื้อสามารถพิจารณาคุณลักษณะเฉพาะได้เมื่อเห็นว่า คุณลักษณะเฉพาะอย่างอื่นให้ ประโยชน์ต่อบริษัทในด้านจัดซื้อได้ดีกว่า

3.7 การผลิตและการควบคุมการผลิต

ปกติการผลิตจะให้ข้อ่าสารด้านคุณสมบัติที่ต้องการ ส่วนการควบคุมการผลิตและการทำหมายกำหนดการผลิตจะให้ข้อ่าสารว่าต้องการพัสดุดะไร ในช่วงการผลิตใดต้องการเท่าใดข้อ่าสารเหล่านี้เมื่อรวบรวมได้สมบูรณ์ก็จะ เป็นข้อ่าสารที่ใช้ในการวางแผนการจัดซื้อได้

3.8 การควบคุมพัสดุคงคลัง

เป็นข้อ่าสารขึ้นต้นว่าอะไรบางอย่างที่จำเป็นต้องซื้อหรือจะสั่งซื้อเมื่อใด นโยบายพัสด

คงคลังซึ่งควบคุมการลงทุนด้านนี้ในระยะเวลาหนึ่ง ๆ จะเป็นข่าวสารช่วยในการตกลงใจซื้อในจำนวนที่ประหยัด (Economic order quantities) นโยบายนี้สอดคล้องกับพิจารณาที่กำหนดจากปัจจัยต่าง ๆ คือ ทรัพยากรด้านการเงินของบริษัท แผนงาน สภาวะทางด้านตลาดระยะเวลาในการจัดหาวัสดุและอื่น

3.9 การควบคุมคุณสมบัติและการรับพัสดุ

ข่าวสารจากสองแห่งนี้จะเป็นการแจ้งให้ทราบว่า ผู้ขายได้มอบวัสดุที่ตกลงกันไว้แล้วว่ามีจำนวนและคุณสมบัติถูกต้องหรือไม่ ข่าวสารดังกล่าวนี้เป็นงานส่วนหนึ่งที่ทำให้การดำเนินการจัดซื้อครบถ้วนสมบูรณ์

3.10 ผลิตภัณฑ์ใหม่

การพัฒนาผลิตภัณฑ์ใหม่ๆ ในระยะที่ผ่านมา มีความสำคัญต่อความสำเร็จของบริษัทมากขึ้นทุกวัน หากข่าวสารเกี่ยวกับโครงการพัฒนาผลิตภัณฑ์ใหม่ไม่ถึงแผนกจัดซื้อตั้งแต่ตอนเริ่มต้น การสนับสนุนของแผนกจัดซื้อก็มักไม่ได้ผลเต็มที่นัก

3.11 ข่าวสารจากภายนอกบริษัทที่เข้ามาในแผนกจัดซื้อ

แผนกจัดซื้อเป็นแผนกหนึ่งที่ติดต่อกับโลกภายนอกของบริษัท ข่าวสารส่วนใหญ่จะมีความสำคัญต่อการดำเนินการของบริษัท ดังจะอธิบายตามผังและรายละเอียดย่อ ๆ ดังนี้

3.11.1 ภาวะการตลาดทั่วไป

ผู้บริหารการจัดซื้อและผู้ซื้อที่ปรึกษาสามารถ จะกลายเป็นผู้เชี่ยวชาญด้านภาวะการตลาดและธุรกิจทั่วไป พนักงานขายของแหล่งขายนิคมสารเกี่ยวกับการค้า นิคมสารของสมาคมการค้าและการพบปะสังสรรค์ของนักธุรกิจ จะเป็นแหล่งข่าวเกี่ยวกับราคาอุปสงค์ อุปทาน และภาวะการแข่งขัน

3.11.2 แหล่งขาย

พนักงานของแหล่งขาย สื่อโฆษณาทุกแบบ การส่งเสริมการตลาด การแสดงสินค้า รายงานการเงินและเงินเชื่อ ซึ่งแหล่งขายจัดทำขึ้นเพื่อให้ข่าวสารแก่ลูกค้าหรือผู้ที่คาดว่าจะ เป็นลูกค้า เหล่านี้จะช่วยให้สามารถหาแหล่งขายแหล่งขายพัสดุที่ต้องการได้

3.11.3 ที่ความสามารถและอัตราการผลิตของผู้ขาย

ภาวะแรงงานในโรงงานและอุตสาหกรรมของผู้ขายข่าวสารเหล่านี้มีความสำคัญมากในการกำหนดนโยบายพัสดุคงคลัง และพิจารณาความแน่ใจในการผลิตที่ต่อเนื่องกันไปในส่วนที่พัสดุเกี่ยวข้องกันด้วย

3.11.4 ราคาและส่วนลดการขายและภาษี

3.11.5 การขนส่งและอัตราค่าขนส่ง

3.11.6 ข่าวมลพิษภัยและข่าวมลพิษภัยใหม่ ๆ

3.12 ข่าวดำเนินการในจากแผนกจัดซื้อไปสู่หน่วยงานต่าง ๆ ในบริษัท

เกือบไม่มีหน่วยงานใดในบริษัทที่ไม่เกี่ยวข้องกับกาจัดซื้อ ผังข้างล่างนี้เป็นการแสดงให้เห็นว่าข่าวดำเนินการสำคัญที่ส่งจากแผนกจัดซื้อให้แก่หน่วยงานต่าง ๆ

1. การบริหารทั่วไป
2. การช่าง
3. การพัฒนาผลิตภัณฑ์
4. การตลาด
5. การผลิต
6. กฎหมาย
7. การเงินและการบัญชี
8. คลังพัสดุ

การกำหนดนโยบายในแผนกคลังข้อมู้นั้นอยู่กับข่าวดำเนินการด้านพัสดุเกี่ยวกับระยะเวลาจัดหามา (Lead time) พิสัยที่มีอยู่ แนวโน้มของราคา และการพัฒนาของพัสดุซึ่งอาจมีพัสดุนอกคณได้ ข่าวดำเนินการดังกล่าวนี้ฝ่ายจัดซื้อเป็นแหล่งข่าวที่ดีที่สุด

3.13 คอมพิวเตอร์เป็นเครื่องมือช่วยในการไหลของข่าวดำเนินการ

ในการดำเนินงานเกี่ยวกับการจัดซื้อตั้งแต่ต้นจนเสร็จสิ้นแต่ละเรื่อง ข้อมู้นั้นต้องมีการรับส่งข่าวดำเนินการต่าง ๆ ตามขั้นตอนมากมาย ข่าวดำเนินการที่รับส่งกันนั้นส่วนใหญ่จัดทำเป็นแบบฟอร์ม เริ่มต้นที่ใบขอให้ซื้อ (Requisition) การขอทราบราคา (Requests for quotations) ใบสั่งซื้อ (Purchase orders) การเปลี่ยนแบบฟอร์มการสั่งซื้อ (Receiving reports) ใบเสร็จรับเงิน (Invoice) และอื่น ๆ เป็นต้น

งานเกี่ยวกับเอกสารเหล่านี้ต้องใช้ให้มีคนดำเนินการ ต้องปฏิบัติต่อเนื่องตลอดเวลา ถ้าดูอย่างผิวเผินว่าแผนกจัดซื้อทำอะไรบ้าง ก็จะมองเห็นแต่เจ้าหน้าที่ปฏิบัติเกี่ยวกับเอกสารเกือบทั้งสิ้นและนี่แหละที่เป็นความเข้าใจอย่างหนึ่งของคนที่เข้าไปที่เข้าใจว่า การจัดซื้อเป็นงานเสมียน (Clerical-function) เพราะการจัดการเกี่ยวกับเอกสารเหล่านี้ทำด้วยมือหรือด้วยคน (manually) ปัจจุบันการใช้เป็นสิ่งที่หลีกเลี่ยงไม่ได้ และเป็นผลต่อระยะเวลาจัดหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนพัสดุคงคลัง เงินลงทุน สัตวอก และตลอดจนตารางการผลิต นอกจากนี้ การจัดทำด้วยมือ ยังเป็นอุปสรรคต่อการจัดข้อมูล และการวิเคราะห์ข้อมูลเหล่านั้นด้วย คือ เข้าเกินไป จากผล การสำรวจระยะยาวดำเนินการโดยทั่วไป ตั้งแต่ออกไปร้องขอจนถึงออกไปสั่งซื้อของอุตสาหกรรมต่าง ๆ โดยเฉลี่ยประมาณ 11 วัน ส่วนสิ้นการรับใบสั่งซื้อ ส่งพัสดุ ตรวจสอบ และ จ่ายเงิน ต้องใช้เวลาอีกช่วงหนึ่งต่างหาก

การนำเอาเครื่องมือ Electronic Data Processing มาใช้ จะเป็นประโยชน์และสะดวกมากในการจัดการกับข่าวสารเหล่านั้นได้อย่างรวดเร็ว เช่น ตรวจสอบความถูกต้อง แยกแยะข้อมูลออกเพื่อใช้วิเคราะห์การปฏิบัติของผู้ขาย วิเคราะห์ราคาและอื่น ๆ แต่ดูเหมือนว่าหน่วยงานจัดซื้อจะเป็นหน่วยงานสุดท้ายที่นำเอาอุปกรณ์นี้มาใช้งาน เหตุที่เข้าไปเช่นนี้ เพราะมีเหตุผลหลายประการคือ

ประการแรก ตามผลการศึกษาการใช้คอมพิวเตอร์โดยทั่วไป ปรากฏว่างานด้านการบัญชีเป็นหน่วยแรกที่นำมาใช้ เมื่อคอมพิวเตอร์แพร่หลาย กิจกรรมต่าง ๆ ก็เริ่มที่การบัญชีก่อน ประการที่สอง หัวหน้าจัดซื้อปกติมีงานติดต่อกันตลอดวัน ไม่มีเวลาว่างพอที่จะพัฒนาแนวความคิดใหม่ ๆ มาปรับปรุงงานจัดซื้อ ส่วนใหญ่ยังไม่ค่อยมีความรู้ด้านนี้เพียงพอว่าคอมพิวเตอร์จะช่วยงานจัดซื้อได้อย่างไรจึงลังเลที่จะริเริ่มนำมาใช้

ประการที่สาม เมื่อนำเอาคอมพิวเตอร์มาใช้ในบริษัทครั้งแรก ยังไม่ได้กำหนดค่าใช้จ่ายด้านจัดซื้อด้วย ดังนั้นเมื่อต้องการให้ใช้ก็ต้องการเปลี่ยนแปลงมากพอสมควร เป็นเหตุให้เจ้าหน้าที่รับผิดชอบในการใช้คอมพิวเตอร์ไม่สู้เต็มใจจะเปลี่ยนหรือเพิ่มเติมให้

ประการที่สี่ บางบริษัทที่นำเอาคอมพิวเตอร์มาใช้ในรูปของ Integrated Data Processing โดยกำหนดเป็นระบบ (Systems group) ถ้ารวมงานด้านการจัดซื้อเป็นระบบหนึ่งด้วย จะมีปัญหาว่าเจ้าหน้าที่ด้านคอมพิวเตอร์มีความรู้ด้านการจัดซื้อไม่พอ ทำให้ประสิทธิภาพหรือผลจากการใช้คอมพิวเตอร์ไม่เป็นที่ประทับใจ

ความประสงค์ในการศึกษาเรื่องคอมพิวเตอร์ในที่นี้ไม่ได้อธิบายรายละเอียดเกี่ยวกับการติดตั้งและการใช้ EDP หรือ IDP แต่เพื่อต้องการอธิบายให้เข้าใจถึง แนวความคิดในการนำมาใช้เครื่องคอมพิวเตอร์ครบชุดจะมีอุปกรณ์เก็บข้อมูลเป็นล้านๆรายการมีเครื่องคำนวณที่คำนวณให้ได้คำตอบได้ในเวลาอันรวดเร็วมีเครื่องพิมพ์ที่พิมพ์ข่าวสารออกมาตามที่ต้องการได้อย่างรวดเร็ว ดังนั้น หากว่าได้ส่งข่าวสาร (input of information) เข้าไปอย่างถูกต้องและสมบูรณ์แล้ว เครื่องคอมพิวเตอร์นี้ จะช่วยอำนวยความสะดวกให้ในเรื่องต่อไปนี้

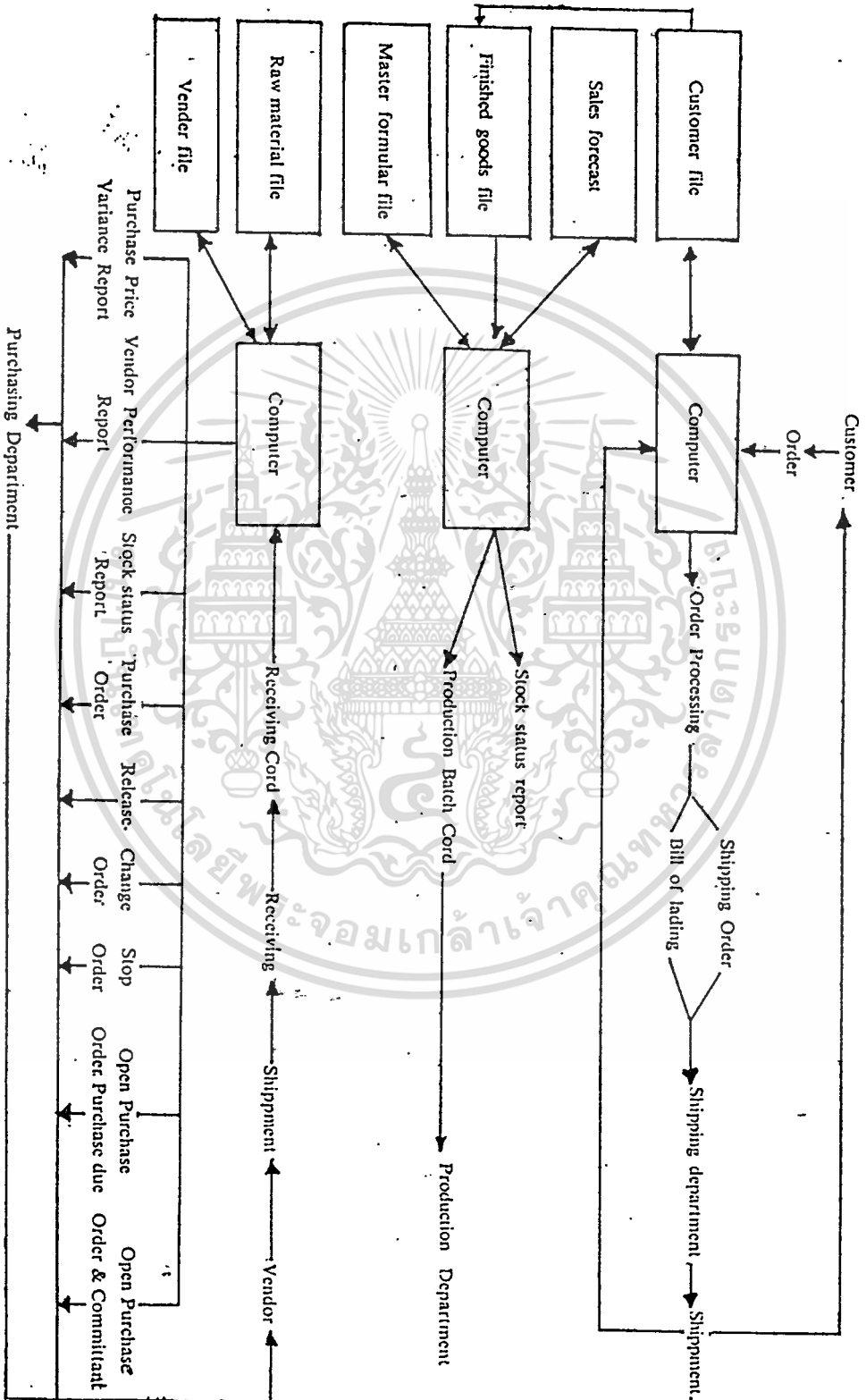
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ตรวจสอบรายการพัสดุ (Bill of information) และแยกออกไปเป็นรายการตามความต้องการให้เหมาะสำหรับตารางการผลิตที่กำหนด
2. กำหนดระดับพัสดุลงคลัง (Inventory level) และระดับสั่งเพิ่ม (Reorder points) ของแต่ละรายการพัสดุที่สต็อกไว้ในคลังได้อย่างมีประสิทธิภาพ
3. คำนวณจำนวนร้องขอที่ประหยัด (Economic order quantities) และจำนวนซื้อที่ประหยัด (Economic purchase quantities)
4. เตรียมและเขียนในสั่งซื้อ รวมทั้งเอกสารติดตามเรื่องที่เกี่ยวข้อง
5. ตรวจสอบใบเสร็จรับเงินกับเอกสารต่าง ๆ ที่เกี่ยวข้อง และเขียนเช็ค
6. รายงานข้อมูลต่างๆ (Statistical reports) เกี่ยวกับการร้องขอ การสั่งซื้อ การปฏิบัติของผู้ขาย การปฏิบัติของผู้ซื้อ (Buyer) การรวบรวมในสั่งซื้อตามกลุ่มพัสดุ ตามกลุ่มผู้ขายตามอาณาเขต หรือแล้วแต่จะต้องการอย่างไร

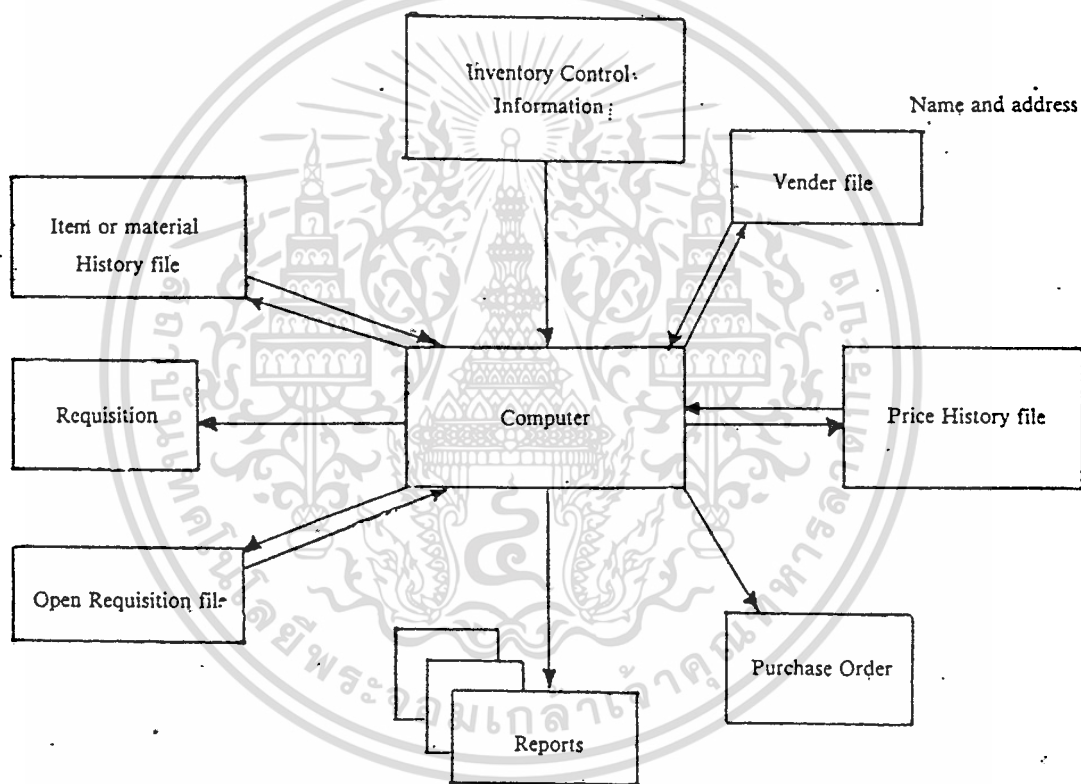
โปรแกรม automated purchasing systems มีขั้นตอนทำงานเองเกี่ยวกับการดำเนินการด้วยมือ คือ เริ่มตั้งแต่ การเสนอคำขอ ตรวจสอบคำขอกับพัสดุลงคลัง กำหนดจำนวนที่พึงต้องการเลือกแหล่งขาย เขียนใบสั่งซื้อ ติดตามผลเพื่อให้การนำส่งตามเวลาที่กำหนดลงบัญชีการรับ การตรวจสอบความถูกต้องของจำนวนและคุณสมบัติ ตรวจสอบใบเสร็จรับเงินและเขียนเช็คชำระเงิน

ประสิทธิภาพในการดำเนินการของระบบการจัดซื้อส่วนหนึ่งขึ้นอยู่กับการไหลของข่าวสารจากหน่วยงานต่าง ๆ ใน บริษัท และจากแหล่งภายนอกมาซึ่งแผนกจัดซื้อ และจากแผนกจัดซื้อไปยังแผนกต่าง ๆ เป็นต้นมาก การมองเห็นความสำคัญในเรื่องนี้เป็นที่ยอมรับกันทั่วไปแล้วในกลุ่มของนักธุรกิจ แม้จะเป็นการชอบด้วยเหตุผลแล้วที่จะคาดต่อไปไม่ได้ว่า ความเข้าใจในประโยชน์ของการนำเอาคอมพิวเตอร์มาใช้ จะช่วยในเรื่องการไหลของข่าวสารการจัดซื้อได้อย่างไรนั้นคงจะทำได้กว้างขวางมากขึ้นเรื่อย ๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



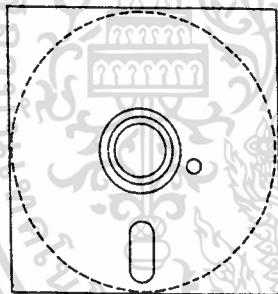
บทที่ 4

ระบบจัดการฐานข้อมูล

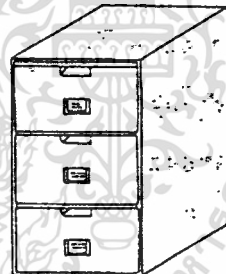
(Database Management System)

4.1 ฐานข้อมูลคืออะไร

ฐานข้อมูล หรือที่ภาษาอังกฤษเรียกว่า ดาตาเบส (Data Base) คือที่รวมของข้อมูลหรือข่าวสารต่างๆ ที่เราเก็บรวบรวมเอาไว้ ลองนึกถึงภาพของตู้เอกสารที่ใช้เก็บข้อมูลต่างๆ ถ้าเราใช้ตู้เอกสารเพื่อเก็บข้อมูลที่จำเป็นต้องใช้ในการทำงาน ตู้เก็บเอกสารนั้นก็คือ ฐานข้อมูลของเรา ในระบบข้อมูลที่อาศัยคอมพิวเตอร์นั้น การเก็บข้อมูลมิได้เก็บไว้ในลักษณะของเอกสารที่พิมพ์ไว้ในแฟ้ม แต่จะเก็บไว้ในหน่วยความจำที่เราสามารถจะเรียกมาใช้ในการเก็บข้อมูลหรือสร้างเป็นฐานข้อมูล คือ ฟิลอปปีดิสก์ หรือ ดิสเกตต์ และ วินเชสเตอร์ หรือ ฮาร์ดดิสค์



(ก) ดิสเกตต์: ฐานข้อมูลสำหรับระบบไมโครคอมพิวเตอร์

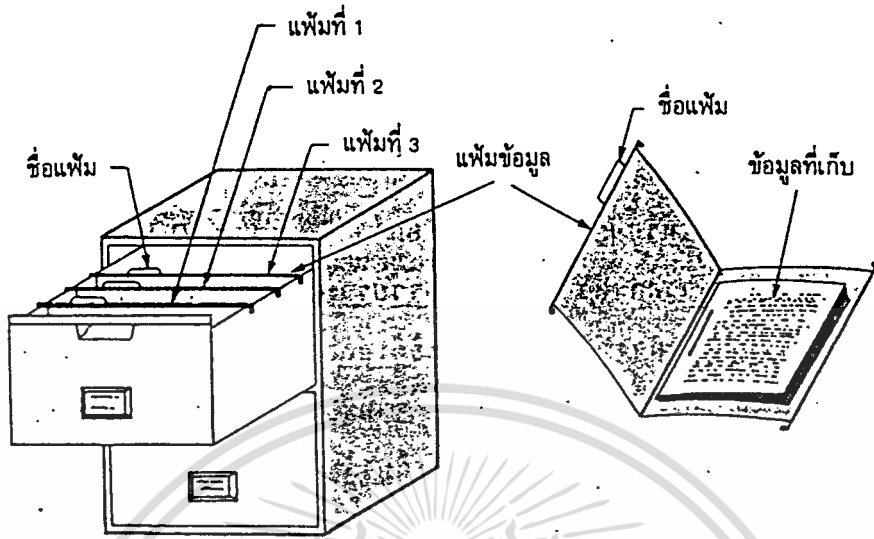


(ข) ตู้เอกสาร: ฐานข้อมูลในการจัดการข้อมูลด้วยคน

รูป 4.1 แสดงการเปรียบเทียบระหว่างการเก็บข้อมูลโดยใช้แผ่นดิสเกตต์กับตู้เอกสาร

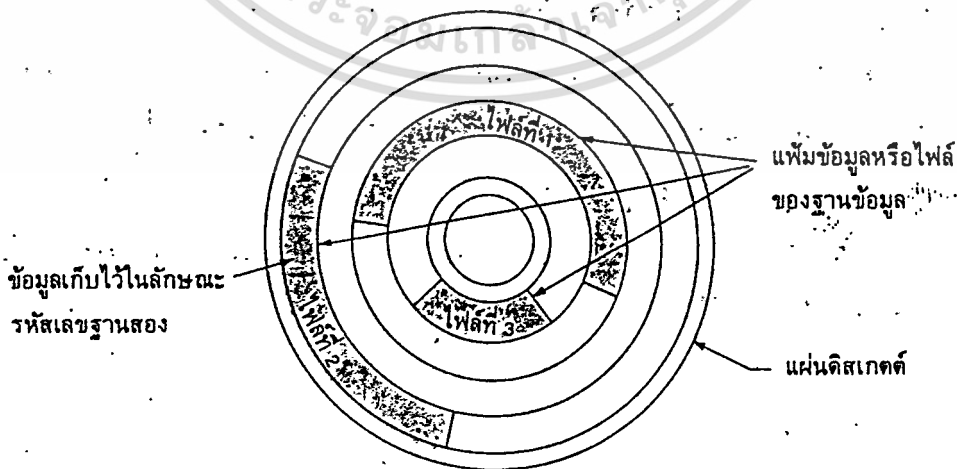
4.2 เพิ่มข้อมูล

ลองนึกถึงภาพการเก็บเอกสารหรือข้อมูลที่ทำกัน โดยการเก็บใส่ในแฟ้มแขวน เรามักจะเก็บข้อมูลเรื่องต่างๆมากมาย โดยแต่ละแฟ้มอาจมีชื่อเขียนกำกับไว้เพื่อสะดวกแก่การค้นหาและการเรียกใช้ดังแสดงในรูป



รูป 4.2 แสดงการเก็บแฟ้มข้อมูล

ในการเก็บข้อมูลด้วยคอมพิวเตอร์ก็เช่นเดียวกัน ในฐานข้อมูลจะต้องประกอบด้วยแฟ้มข้อมูลหรือ คาด้าไฟล์ (Data file) จำนวนหนึ่ง แต่ละไฟล์หรือแฟ้มข้อมูลจะต้องมี ชื่อไฟล์ (File Name) กำกับอยู่เพื่อสะดวกแก่การเรียกใช้ ข้อมูลของแต่ละไฟล์จะถูกเก็บไว้ในแผ่นดิสเกตต์ ซึ่งเป็นแผ่นพลาสติกเคลือบสารแม่เหล็ก และการเก็บข้อมูลจะเก็บในลักษณะของรหัสเลขฐานสองแผ่นดิสเกตต์แต่ละแผ่นจะมีความสามารถในการเก็บข้อมูลได้มากน้อยต่างกันบางแผ่นอาจเก็บข้อมูลได้เพียง 70,000 ตัวอักษรในขณะที่บางแผ่นอาจเก็บข้อมูลได้ถึง 10 ล้านตัวอักษร



รูป 4.3 แสดงลักษณะการเก็บข้อมูลโดยแผ่นดิสเกตต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3. ค่าฟิลด์และเรคอร์ด

ลองพิจารณาตัวอย่างของข้อมูลเรื่องต่าง ๆ ที่เก็บไว้ในแฟ้มดังต่อไปนี้
ไฟล์คะแนนสอบของนักเรียน

เลขประจำตัว	ชื่อ	สอบครั้งที่ 1	สอบครั้งที่ 2	สอบครั้งที่ 3
245462	สุรเดช กล้าหาญ	40	36	76
246248	สมชาย เดชขวัญ	32	50	82
259328	พิเชษฐ์ ดีสกุล	16	30	46
262735	กมล สมใจ	32	60	92
274285	วิรัช สกษณ์เดช	45	40	85

รูป 4.4 แสดงการเก็บไฟล์ข้อมูลคะแนนสอบของนักเรียน

แฟ้มข้อมูลนี้ใช้ในการเก็บข้อมูลของคะแนนสอบของนักเรียน. ซึ่งข้อมูลที่สำคัญที่ต้องเก็บไว้คือ เลขประจำตัว ชื่อของนักเรียนแต่ละคน ตลอดจนผลสอบแต่ละครั้ง ข้อมูลของนักเรียนแต่ละคนหรือแต่ละบรรทัดเรียกว่า เรคอร์ด(record) ถ้าพิจารณาต่อไปจะเห็นว่า ข้อมูลแต่ละเรคอร์ดหรือของแต่ละคนจะเก็บรายละเอียดในรูปแบบลักษณะเหมือนกัน ดังเช่นในคอลัมน์ที่ 1 จะมีเลขประจำตัว คอลัมน์ที่ 2 เป็นชื่อ คอลัมน์ที่ 3 เป็นผลของการสอบครั้งที่ 1 ฯลฯ ในแต่ละคอลัมน์ซึ่งรวบรวมข้อมูลประเภทเดียวกันไว้ เรียกว่า ค่าฟิลด์ (Data Field) หรือฟิลด์แต่ละฟิลด์มีชื่อกำกับ เรียกว่า ฟิลด์นาม (Field Name) หรือชื่อฟิลด์ สิ่งที่บรรจุอยู่ในแต่ละค่าฟิลด์ก็คือ ข้อมูลหรือค่า (Data) นั้นเอง

ลักษณะการจัดแบ่งฟิลด์ต่างๆ ของไฟล์นี้เราเรียกว่า โครงสร้างของไฟล์ (File Structure)

4.4 การจัดการฐานข้อมูล

การจัดการฐานข้อมูลหรือการจัดการค่าเบส (Database Management) เป็นเรื่องที่เกี่ยวข้องกับการจัดการกับการจัดกวางกับข้อมูลที่มีอยู่ โดยมีหัวใจสำคัญคือ ทำอย่างไรจึงจะทำให้เราสามารถ

- จัดเก็บข้อมูลง่าย
- เรียกใช้ข้อมูลสะดวก

ในแวดวงของไมโครคอมพิวเตอร์ ได้มีการพัฒนาซอฟต์แวร์ หรือโปรแกรมสำเร็จรูป เพื่อใช้ ช่วยในการจัดการฐานข้อมูลหลายโปรแกรมด้วยกัน โดยทุกโปรแกรมก็มุ่งที่จะให้ผู้ใช้ สามารถจัด เก็บข้อมูลได้ง่าย และเรียกใช้ข้อมูลได้สะดวก การทำงานพื้นฐานที่สำคัญของการจัดการฐาน ข้อมูลประกอบด้วย

1. การสร้างไฟล์ข้อมูล
2. การเพิ่มเติมข้อมูลลงในไฟล์
3. การจัดเรียงข้อมูลในไฟล์ตามที่ต้องการ
4. การค้นหาข้อมูลจากไฟล์
5. การจัดทำรายงานจากไฟล์ที่มีอยู่
6. การแก้ไขปรับปรุงข้อมูลในไฟล์
7. การลบข้อมูลออกจากไฟล์

นอกจากงานหลักๆ เหล่านี้แล้ว ยังมีงานลักษณะอื่นอีก เช่น การเปลี่ยนแปลงโครงสร้าง ของไฟล์ การโอนย้ายข้อมูลจากไฟล์หนึ่งไปยังอีกไฟล์หนึ่ง การเชื่อมต่อสองไฟล์เข้าด้วยกัน เป็นต้น

บทที่ 5

ดีเบสก์วิฬัส

(Dbase III+)

5.1 ซอฟต์แวร์จัดการฐานข้อมูล

บนเครื่องไมโครคอมพิวเตอร์นี้ ซอฟต์แวร์ที่ได้รับการพัฒนาขึ้นเพื่อการจัดระบบฐานข้อมูลมีอยู่หลายโปรแกรมด้วยกัน ที่คุ้นเคยและมีการใช้กันในบ้านเราดูเหมือนจะได้แก่ พีเอฟเอส (PFS)ดาต้าสตาร์ (Data Star) และซอฟต์แวร์ในตระกูลดีเบส นับตั้งแต่ดีเบสทู ดีเบสทรี และดีเบสทรีพลัส แต่ซอฟต์แวร์ที่ได้รับความนิยมแพร่หลายสูงสุด และมีใช้ในไมโครคอมพิวเตอร์เกือบทุกเครื่องไม่ว่าจะเป็นระดับ 8 บิต 16 บิต หรือ 32 บิต เห็นจะได้แก่ซอฟต์แวร์ตระกูลดีเบส

สำหรับดีเบสทรีพลัส เป็นซอฟต์แวร์ที่มีความสามารถสูง โดยเฉพาะอย่างยิ่งการเรียกใช้ข้อมูล การประมวลผลข้อมูลในไฟล์ นอกจากนี้ดีเบสทรีพลัสยังเป็นซอฟต์แวร์ที่ผู้ใช้สามารถเขียนชุดคำสั่ง หรือโปรแกรม เพื่อสั่งให้คอมพิวเตอร์ทำการประมวลผลข้อมูลตามต้องการได้อีกด้วย ลักษณะโปรแกรมในดีเบสทรีพลัสเป็นโปรแกรมโครงสร้าง ซึ่งทำให้การเขียนโปรแกรมทำได้ง่ายและตรวจสอบได้ดี ซอฟต์แวร์ดีเบสทรีพลัสจึงได้รับความนิยมอย่างมากในปัจจุบัน

5.2 วิวัฒนาการของซอฟต์แวร์ในตระกูลดีเบส

ผู้พัฒนาซอฟต์แวร์ในตระกูลดีเบสคือ เวย์ แรทลิฟ (Wayne Rathlieve) วิศวกรที่ทำงานในโครงการขององค์การนาซ่าหรือองค์การอวกาศแห่งสหรัฐอเมริกา เวย์ แรทลิฟได้เริ่มพัฒนาดีเบสทูตั้งแต่ปี พ.ศ. 2520 และนำออกจำหน่ายครั้งแรกเมื่อกลางปี พ.ศ. 2522 ภายหลังชื่อซอฟต์แวร์ว่า วัลแคน (Valcan) วัลแคนไม่ประสบความสำเร็จในด้านการตลาดเลย จนกระทั่งเวย์ แรทลิฟได้พบกับ จอร์จ เทตแห่งบริษัทแอสตันเทค ซึ่งเป็น บริษัทแอสตันเทค ซึ่งเป็นบริษัทซอฟต์แวร์ เสาส์และตัวแทนจำหน่ายซอฟต์แวร์ไมโครคอมพิวเตอร์ จอร์จ เทตได้ตกลงทำสัญญาเป็นตัวแทนจำหน่ายซอฟต์แวร์ของเวย์ แรทลิฟ และเปลี่ยนชื่อซอฟต์แวร์เสียใหม่เป็นดีเบสทู (dBASE II)

ดีเบสทูเริ่มวางจำหน่ายครั้งแรกในต้นปี พ.ศ. 2524 และประสบความสำเร็จอย่างงดงาม จนกลายเป็นซอฟต์แวร์ที่มีชื่อเสียงสูงสุดในบรรดาซอฟต์แวร์การจัดการข้อมูล นับแต่นั้นมาดีเบสทูได้รับการพัฒนาตามลำดับจนถึงดีเบสทรี (dBASE III) ที่ใช้ในเครื่องระดับ 16 บิตและ

เมื่อต้นปี พ.ศ. 2529 ก็ได้พัฒนาเป็น dBASE III+ ซึ่งมีความสามารถและความเร็วในการทำงานสูงมาก

5.3 โครงสร้างของดีเบสทรีพอลัส

ดีเบสทรีพอลัสเป็นซอฟต์แวร์ที่จัดเก็บข้อมูลในลักษณะเป็นตาราง คล้ายคลึงกับการเก็บข้อมูลในแฟ้มเอกสาร เราเรียกลักษณะของระบบการจัดเก็บข้อมูลนี้ว่า แบบรีเลชันนัลดาตาเบส (Relational Database) ลักษณะการจัดเก็บข้อมูลเก็บข้อมูลแบบรีเลชันนัลนี้เป็นวิธีการที่ง่ายแก่การเรียกใช้ และการทำความเข้าใจของผู้ใช้ข้อมูล

5.3.1 เริ่มต้นใช้ดีเบสทรีพอลัส

เมื่อจะเริ่มต้นใช้ ดีเบสทรีพอลัสจะต้องทำความเข้าใจกับเครื่องคอมพิวเตอร์ที่จะใช้เสียก่อน ซอฟต์แวร์ดีเบสทรีพอลัสจะถูกเก็บอยู่ในแผ่นดิสเกตต์ในลักษณะของไฟล์เช่นเดียวกัน แต่เป็นโปรแกรมไฟล์ ดังนั้นการที่จะใช้ดีเบสทรีพอลัสได้บนเครื่องคอมพิวเตอร์ที่ใช้จะต้องประกอบด้วย

1. เครื่องอ่านดิสค์อย่างน้อย 2 เครื่อง
2. แผ่นดิสเกตต์ที่มีโปรแกรมดีเบสทรีพอลัสอยู่
3. จอภาพหรือจอทีวี
4. เครื่องคอมพิวเตอร์ที่มี ซีพียู ซึ่งสามารถรันโปรแกรมจัดระบบงาน (operating system) MS-DOS หรือ PC-DOS ได้
5. มีหน่วยความจำภายในชนิด RAM ไม่น้อยกว่า 256 KB

5.3.2 โหลดโปรแกรมจัดระบบงานลงเครื่อง

เมื่อตรวจสอบว่ามีอุปกรณ์ครบตามที่กล่าวมาแล้ว ก็เริ่มต้นการโหลด (load) เอาโปรแกรมจัดระบบงานลงในหน่วยความจำของเครื่องคอมพิวเตอร์ดังนี้ขั้นตอนต่อไปนี้เป็นคือ

1. นำแผ่นดิสเกตต์ที่มีโปรแกรมจัดระบบงาน PC-DOS ไปใส่ในเครื่องอ่านดิสค์ในไดรฟ์ A
2. เปิดสวิตช์ของจอภาพ
3. เปิดสวิตช์ของเครื่อง (สวิตช์ของซีพียู) จะปรากฏข้อความบนจอภาพดังแสดง

Current date is Tue 1-01-1980

Enter new date:

4. คอมพิวเตอร์จะให้เรารู้ วัน เดือน ปี ที่ใช้งาน และเวลาทำงานให้แก่เครื่อง เพื่อที่จะได้เก็บบันทึกวันเวลาของการทำงานของเราไว้ในโปรแกรมหรือไฟล์ที่สร้างขึ้น

5. นำแผ่นดิสก์เต็มดิสก์ (System Disk) แผ่นที่ 1 ใส่ในไดรฟ์ A แทนแผ่นโปรแกรมดอส พิจารณารายการต่างๆ บนจอภาพ สิ่งที่แสดงอยู่เมื่อคีย์ DIR แล้วกดคีย์รีเทิร์นคือชื่อของไฟล์ต่างๆ ที่มีอยู่ในดิสก์เกตต์ แต่ละบรรทัดคือรายละเอียดของแต่ละไฟล์ประกอบด้วย ชื่อไฟล์ประเภทของไฟล์ จำนวนของหน่วยความจำที่ให้แก่และเวลาที่ทำการสร้างไฟล์ ไฟล์ต่างๆ บนแผ่นดิสก์เกตต์ จะมีความหมายและลักษณะการใช้งานแตกต่างกัน ตัวอย่างเช่น

COMMAND.COM เป็นโปรแกรมคำสั่งของระบบที่ PC-COS เรียกว่าใช้โดยทั่วไป ผู้ใช้งานจะไม่เรียกใช้โปรแกรมนี้เอง

DBASE.EXE เป็นโปรแกรมหลักของดีเบสทรีฟล์ส การเรียกใช้ดีเบสทรีฟล์สจะต้องเรียกผ่านโปรแกรมนี้

DBASE.MSG เป็นโอเวอร์เลย์ไฟล์ (Overlay File) ซึ่งเก็บคำสั่งของดีเบสทรีฟล์ส เมื่อใดที่ดีเบสต้องการทำงานโดยเรียกใช้คำสั่งเหล่านั้น โหลดไฟล์นี้ลงในหน่วยความจำ

HELP.DBS เป็นไฟล์ที่เก็บรายละเอียดของคำสั่งที่ผู้ใช้สามารถเรียกดูได้เมื่อต้องการขอความช่วยเหลือในระหว่างการใช้ดีเบสทรีฟล์ส

ASSIST.HLP เป็นไฟล์ที่เก็บเมนูในโหมดของ แอสซิสต์ (Assist)

5.3.3 โหลดดีเบสทรีฟล์สลงหน่วยความจำ

เมื่อพร้อมที่จะใช้งานดีเบสทรีฟล์ส จะต้องโหลดไฟล์ที่เก็บโปรแกรมดีเบสทรีฟล์สลงสู่หน่วยความจำเสียก่อนโดยการคีย์ชื่อไฟล์ DBASE แล้วคีย์รีเทิร์นดังแสดง

A>DBASE

คอมพิวเตอร์จะทำการโหลดดีเบสทรีฟล์สจากดิสก์เกตต์ลงสู่หน่วยความจำและจะปรากฏข้อความบนจอภาพคือ

dBASE III PLUS version 1.0 IBM/MSDOS
 Copyright (c) Ashton-Tate 1984, 1985, 1986. All Rights Reserved.
 dBASE, dBASE III, dBASE III PLUS, and Ashton-Tate
 are trademarks of Ashton-Tate

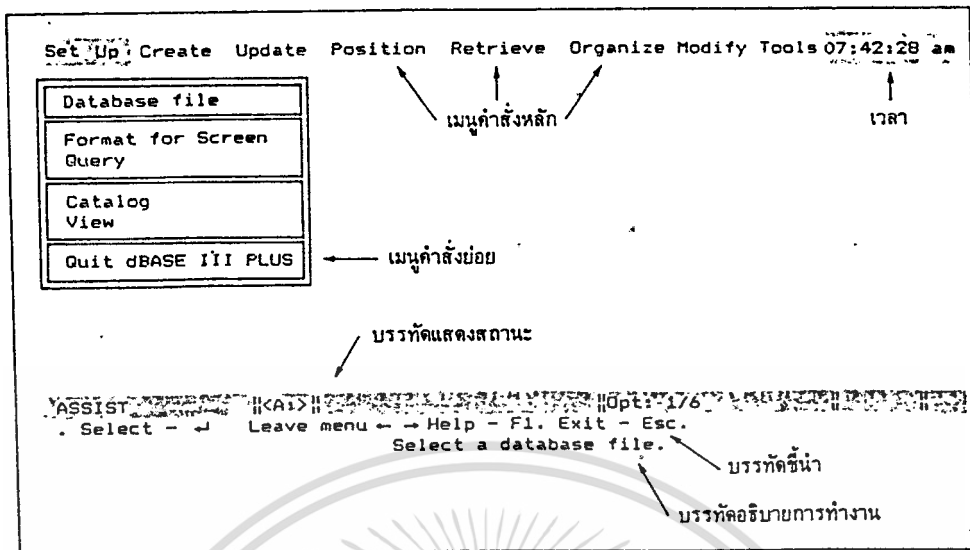
You may use the dBASE III PLUS software and printed materials in the dBASE III PLUS software package under the terms of the dBASE III PLUS Software License Agreement. In summary, Ashton-Tate grants you a paid-up, non-transferable, personal license to use dBASE III PLUS on one microcomputer or workstation. You do not become the owner of the package, nor do you have the right to copy or alter the software or printed materials. You are legally accountable for any violation of the License Agreement or of copyright, trademark, or trade secret laws.

Command Line: A:\A: >

Press **Y** to assent to the License Agreement and begin dBASE III PLUS

นำแผ่นดิสก์ 1 ออกจากไดรฟ์ A ใส่แผ่นดิสก์ 2 ลงในไดรฟ์ A แทน แล้ว
 กดรีเทิร์น จะได้ผลบนจอภาพคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



สิ่งที่เห็นบนจอภาพ

สิ่งที่ปรากฏบนจอภาพของดีเบสทีริวัลส์ ประกอบด้วย

เมนูคำสั่งหลัก เป็นคำสั่งที่ผู้ใช้สามารถเรียกใช้ได้ เมนูคำสั่งหลักนี้แสดงอยู่บรรทัด

บนสุด

เมนูคำสั่งย่อย สำหรับคำสั่งหลักแต่ละคำสั่งจะมีคำสั่งจะมีคำสั่งย่อยแสดงอยู่ใต้คำ

สั่งหลัก คำสั่งย่อยนี้จะแสดงบนจอภาพเมื่อตัวชี้คำสั่ง (Command Pointer) ชี้อยู่ตรงคำสั่งหลักนั้น ๆ

เวลา แสดงอยู่มุมขวาบนของจอภาพ

บรรทัดแสดงสถานะ หรือ Status Line ซึ่งจะแสดงสถานะการทำงานว่าอยู่ในสถานะใดทำงานอยู่กับดิสก์ไดรฟ์ใด มีไฟล์ที่อยู่ในระหว่างการใช้งานหรือไม่ เป็นต้น

บรรทัดชี้หน้า เป็นบรรทัดรองสุดท้าย ซึ่งจะบอกคีย์พิเศษ เพื่อเป็นการชี้หน้าให้ผู้ใช้ทำตาม เช่น ถ้าต้องการขอความช่วยเหลือให้กด F1 ถ้าต้องการออกจากเมนูให้ กด Esc

บรรทัดอธิบายการทำงาน เป็นบรรทัดสุดท้ายในจอภาพ ซึ่งจะอธิบายสถานะที่จะเกิดขึ้น ถ้าเลือกคำสั่งในเมนู คือสภาพการทำงานต่างๆ

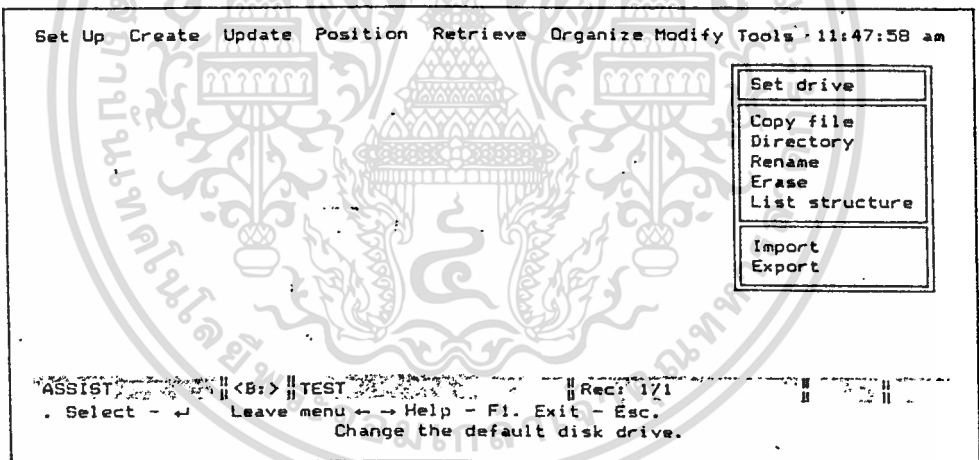
5.3.4 โหมดแอสซิสต์

เมื่อเข้าสู่ดีเบสทีริวัลส์ เราจะเห็นเมนูบนจอภาพ เมนูอยู่ในโหมดการทำงานของแอสซิสต์ เมนูของดีเบสทีริวัลส์ที่ลักษณะ เป็นเมนู 2 มิติ หรือพูลดาวน์เมนู (Pull Down Menu) คือการเรียกใช้จะทำได้ใน 2 ทิศทาง คำสั่งหลักในเมนูจะแสดงในบรรทัดสุดท้ายซึ่งมี

ข้อ 5 คำสั่งคือ

- Set up ใช้เพื่อโหลดไฟล์ลงสู่หน่วยความจำ
- Create ใช้เพื่อสร้างไฟล์ชนิดต่าง
- Update ใช้เพื่อปรับปรุง เปลี่ยนแปลง เพิ่มลด และแก้ไขข้อมูล
- Position ใช้เพื่อค้นหาข้อมูล
- Retrieve ใช้เพื่อเรียงลำดับข้อมูลและก๊อปปี้ไฟล์
- Tools ใช้เพื่อจัดการเกี่ยวกับดิสค์ไฟล์ เช่น เปลี่ยนชื่อไฟล์ ก๊อปปี้ไฟล์ ลบไฟล์ เป็นต้น

การเรียกใช้คำสั่งหลักทำได้โดยเลื่อนตัวชี้คำสั่งโดยกด -> หรือ <- คำชี้คำสั่งจะเลื่อนไปยังคำสั่งที่ต้องการเมื่อตัวชี้คำสั่งอยู่ตรงกับคำสั่งใดจะมีคำสั่งย่อยของคำสั่งหลักปรากฏอยู่ข้างล่างของคำสั่งนั้น ดังตัวอย่างเช่น



การเรียกใช้คำสั่งย่อยของแต่ละคำสั่งทำได้โดยกดคีย์ ; หรือ ; เพื่อเลื่อนตัวชี้คำสั่งย่อยไปยังคำสั่งที่ต้องการแล้วกดรีเทิร์นคีย์เบสทรีพลัสจะทำงานตามคำสั่งที่เลือกไว้

ใครควรจะใช้แอสซิสต์เมนู

ดีเบสทรีพลัสได้สร้างเมนูในโหมดของแอสซิสต์ เพื่อช่วยให้ผู้ใช้ไม่ต้องคีย์คำสั่งเอง เป็นการประหยัดเวลาและเพิ่มความรวดเร็วในการทำงาน แต่จากประสบการณ์ของผู้เขียนในการสอนดีเบสทรีพลัส พบว่าการใช้แอสซิสต์เมนูสำหรับผู้เริ่มต้นซึ่งไม่มีพื้นฐานในด้านการใช้คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่าง ๆ จะสร้างความยุ่งยากให้แก่ผู้ใช้เป็นอย่างมาก ทั้งนี้เพราะคำสั่งของดีเบสทรีพลัสมีความซับซ้อน มีเงื่อนไขมาก ผู้เริ่มต้นใช้ดีเบสทรีพลัสจึงมักประสบปัญหาในการใช้แอสซิสต์เมนู ผู้ที่ใช้แอสซิสต์เมนูได้ดีจึงควรเป็นผู้ที่คุ้นเคยกับคำสั่งของดีเบสทรีพลัสเสียก่อน

เมื่อต้องการทราบรายละเอียดคำสั่งใดก็คีย์หมายเลขของคำสั่งนั้นดีเบสทรีพลัสจะแสดงรายละเอียดของคำสั่งตามต้องการ

เมื่อต้องการออกจากการทำงานของความช่วยเหลือก็กด Esc

การขอความช่วยเหลือโดยการคีย์คำสั่ง Help แล้วตามด้วยคำสั่งที่ต้องการทราบรายละเอียด

การขอความช่วยเหลือเมื่อคีย์คำสั่งผิด ในระหว่างการทำงานโดย การคีย์คำสั่ง เมื่อเราคีย์คำสั่งผิด ดีเบสทรีพลัสจะบอกให้รู้และจะถามว่าต้องการความช่วยเหลือหรือไม่ด้วยความ

Do you want some help? (Y/N)

เมื่อต้องการความช่วยเหลือ ก็ให้กด Y ดีเบสทรีพลัสจะแสดงรายละเอียดของคำสั่งที่ควรจะเป็นออกมาทางจอภาพ

เมื่อต้องการออกจากการทำงานของความช่วยเหลือก็กด Esc

5.3.5 ฟังก์ชันคีย์

ดีเบสทรีพลัสได้กำหนดฟังก์ชันคีย์ ซึ่งเป็นคีย์ใช้งานพิเศษไว้ 10 คีย์ เพื่อช่วยให้ผู้ใช้ทำงานได้สะดวกขึ้น โดยเฉพาะคำสั่งที่มีการเรียกใช้บ่อย ๆ ผู้ใช้งานสามารถกดฟังก์ชันคีย์เพียงคีย์เดียวก็จะทำงานให้ตามที่กำหนดไว้ คำศัพท์ที่ดีเบสทรีพลัสกำหนดไว้ให้กับฟังก์ชันคีย์ทั้ง 10 คีย์ ประกอบด้วย

คีย์	คำสั่ง
F1	Help
F2	Assist
F3	List
F4	Dir
F5	Display Structure
F6	Display Status

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F7	Display Memory
F8	Display
F9	Append
F10	Edit

คำสั่งของฟังก์ชันคีย์สามารถเปลี่ยนแปลงได้ด้วยคำสั่ง SET FUNCTION ตัวอย่าง เช่น ถ้าต้องการกำหนดให้ฟังก์ชันคีย์ F10 เป็นคำสั่ง Quit ก็ทำได้ โดยคีย์คำสั่งคือ

SET FUNCTION 10 TO "Quit"

ฟังก์ชันคีย์ F10 จะเปลี่ยนเป็น Quit ตามต้องการ

5.4 การสร้างไฟล์ข้อมูล

การสร้างไฟล์ข้อมูลด้วย CREATE

เมื่อต้องการสร้างไฟล์ด้วยดีเบส คำสั่งที่ใช้คือ CREATE แล้วคีย์ร็ทเทอร์น

. CREATE

ดีเบสจะถามว่าต้องการตั้งชื่อไฟล์ที่จะสร้างใหม่คืออะไร ด้วยข้อความ

ENTER FILENAME :

ปัญหาต่อไปคือ เราจะใส่ชื่อไฟล์อะไรลงไป การตั้งชื่อไฟล์ในดีเบสมีหลักดังนี้

เกณฑ์ที่สำคัญ ซึ่งจะต้องปฏิบัติตาม

ข้อกำหนดการตั้งชื่อไฟล์

ชื่อไฟล์ของดีเบสที่รหัสจะต้องเป็นไปตามกฎต่อไปนี้คือ

1. อักขระตัวแรกต้องเป็นตัวอักษร
2. ชื่อไฟล์ยาวได้ไม่เกิน 8 ตัวอักษร
3. ถ้าต้องการให้สร้างไฟล์ไว้ในไดรฟ์ที่ต้องการ ให้ระบุชื่อไดรฟ์โดยกำกับหน้าชื่อไฟล์ เช่น A: B: เป็นต้น (ชื่อไดรฟ์ไม่นับรวมเป็นชื่อไฟล์ด้วย)

5.4.1 โครงสร้างข้อมูล

โครงสร้างข้อมูลที่จะต้องกำหนดให้กับดีเบสทรีพลัส ประกอบด้วย

Field Name	หมายถึง ชื่อของฟิลด์ที่จะตั้งให้แก่ฟิลด์
Type	หมายถึง ประเภทข้อมูลที่เก็บในฟิลด์นั้น
Width	หมายถึง ความกว้างหรือจำนวนตัวอักษร ที่จะเก็บไว้ในฟิลด์นั้น
Dec	หมายถึง จำนวนตำแหน่งหลังจุดทศนิยม (Decimal Places) สำหรับข้อมูลชนิดตัวเลขของฟิลด์นั้น

5.4.1.1 ข้อจำกัดของโครงสร้างข้อมูล

1. จำนวนฟิลด์สูงสุดต่อเรคอร์ดคือ 128 ฟิลด์
2. ชื่อของฟิลด์ (Field Name) ตั้งได้สูงสุด 10 ตัวอักษร จะเป็นอักษรตัวใหญ่หรือเล็ก ก็ได้ โดยตัวแรกต้องเป็นตัวอักษร (A-Z) ชื่อฟิลด์ห้ามไม่ให้มีช่องว่าง ประกอบด้วย ตัวเลขหรือตัวอักษร ห้ามใช้อักษรพิเศษอื่นโดยยกเว้น _ (ขีดเส้นใต้)
3. ประเภทของข้อมูลที่จะเก็บ สามารถแบ่งได้เป็น 5 ประเภท คือ
 - ตัวอักษร (Character) เป็นข้อมูลที่เก็บในลักษณะตัวอักษร ซึ่งประกอบด้วยตัวอักษร ตัวเลข สัญลักษณ์และเครื่องหมายต่าง ๆ มีความกว้างสูงสุดไม่เกิน 254 ตัวอักษร
 - ตัวเลข (Numeric) เป็นข้อมูลที่เก็บในลักษณะตัวเลข ซึ่งสามารถนำไปใช้ในการคำนวณทางคณิตศาสตร์ได้มีความกว้างสูงสุดได้ไม่เกิน 19 ตัวเลข
 - ตรรก (Logical) เป็นข้อมูลชนิดที่มีค่าทางตรรก คือ ค่าจริง (True) หรือเท็จ (False) เท่านั้น มีความกว้าง 1 ตัวอักษร ข้อมูลที่จะเก็บมีเพียง 4 อย่างคือ T (หรือ t) และ (Y หรือ y) แทนจริง (True) .T. F (หรือ f) และ (N หรือ n) แทนเท็จ (False).F.
 - วันที่ (Date) เป็นข้อมูลชนิดวันที่ซึ่งดีเบสทรีพลัสจะเก็บไว้ในลักษณะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จูเลียนเดก (ตัวเลขจำนวนวัน นับจากวันที่ 1 มกราคม ค.ศ. 1900) ความกว้างของฟิลด์เป็น 8 ตัวอักษร การป้อนข้อมูลต้องป้อนเป็น เดือน/วันที่ /ปี ค.ศ. แบบ MM/DD/YY

บันทึกย่อว่า (Memo) เป็นข้อมูลชนิดตัวอักษร ใช้บันทึกข้อความที่ต้องการสำหรับบาง เรคอร์ด มีความกว้างต่ำสุด 10 ตัวอักษร และเก็บข้อความได้สูงสุด 5000 ตัวอักษร

4. การกำหนดประเภทของฟิลด์ข้อมูล ทำได้ 2 วิธีคือ

4.1 กำหนดโดยคีย์อักษรตัวแรกของประเภทฟิลด์ คือ

- C ถ้าต้องการให้เป็นฟิลด์ชนิด Character
- N ถ้าต้องการให้เป็นฟิลด์ชนิด Numeric
- L ถ้าต้องการให้เป็นฟิลด์ชนิด Logical
- D ถ้าต้องการให้เป็นฟิลด์ชนิด Memo

4.2 กำหนดโดยการกดเป็นสเปซบาร์ (หรือป้อนช่องว่าง) เมื่อกด 1 ครั้ง ประเภทของฟิลด์จะเปลี่ยนไป ให้กดจนถึงประเภทของฟิลด์ที่ต้องการแล้วกดรีเทิร์น

5.4.2 เมื่อป้อนรายละเอียดผิด

ในการป้อนรายละเอียดของโครงสร้างข้อมูล อาจเกิดความผิดพลาดขึ้น เช่น ตั้งชื่อฟิลด์ผิด กำหนดความกว้างฟิลด์มากเกินไปจนญาติกำหนดประเภทฟิลด์ผิด กำหนดจำนวนทศนิยม มากกว่าความกว้างของฟิลด์ข้อมูล เมื่อเกิดกรณีเช่นนี้ขึ้น คือเบสทรีนั้จะแสดงข้อความให้รู้พร้อมทั้งทำเสียงดัง 1 ครั้ง เพื่อให้เราแก้ไขความผิดพลาด ตัวอย่างข้อความที่แสดงประกอบด้วย

Illegal Data length

Illegal Decimal length

การแก้ไขความผิดพลาดทำได้โดยการคีย์รายละเอียดที่ถูกต้องลงไปใหม่สำหรับกรณีที่เกิดรายละเอียดโครงสร้างข้อมูลผิดไปจากที่ต้องการ เช่นคีย์ชื่อฟิลด์ผิดไป ความกว้างผิดไปจากที่ต้องการ หรือคีย์ชื่อฟิลด์ตกไปบางฟิลด์ เราสามารถแก้ไขได้ (ก่อนที่จะ กดรีเทิร์นเพื่อเก็บโครงสร้างไว้) โดยใช้คอนโทรลคีย์ต่าง ๆ ดังต่อไปนี้คือ

SALARY สิ่งเกิดการทำเสียงร้องคอมพิวเตอร์ สำหรับฟิลด์ใดที่มีเสียงร้อง เมื่อจะคีย์ข้อมูลในฟิลด์ถัดไปไม่ต้องคีย์รีเทิร์น แต่ฟิลด์ใดที่เมื่อคีย์ข้อมูลหมดแล้วแต่ยังไม่มีเสียงร้องจะต้องคีย์รีเทิร์นจึงจะคีย์ข้อมูลในฟิลด์ถัดไปได้

นอกจากนี้ในการคีย์ข้อมูล จะต้องคีย์ให้ถูกต้องกับประเภทของฟิลด์ที่กำหนดไว้ เช่น ฟิลด์ประเภทตัวเลขจะไม่ยอมรับข้อมูลชนิดตัวอักษรหรืออักษรหรืออักขระพิเศษอื่นใด เมื่อคีย์ข้อมูลผิดประเภทเข้าไป คอมพิวเตอร์จะทำเสียงร้องเตือนให้รู้และไม่รับผลการคีย์ข้อมูลนั้น ในฟิลด์

ประเภทตรรกจะรับข้อมูลเพียง 1 ตัวอักษรซึ่งอาจจะเป็น Y หรือ y T หรือ t ซึ่งแทนจริง และ N หรือ n F หรือ f ซึ่งแทนเท็จ ส่วนฟิลด์ประเภทวันที่ จะต้องคีย์ตัวเลขของเดือน วันที่ และปี ค.ศ. (2 หลักสุดท้าย) ในกรณีที่คีย์ผิดจากความเป็นจริง เช่น เดือนที่ 13 หรือ วันที่ 35 คีย์เบสท์วัลส์จะไม่ยอมรับข้อมูล และจะให้เราคีย์ข้อมูลใหม่

5.5.2 คีย์ข้อมูลชนิดเมโม

สำหรับฟิลด์ข้อมูลชนิดเมโม การป้อนข้อมูลทำได้ดังต่อไปนี้คือ

1. เลื่อนเคอร์เซอร์ให้ตรงกับฟิลด์ที่ต้องการป้อนข้อมูล
2. คีย์ Ctrl-PgDn คีย์เบสท์วัลส์จะเตรียมที่สำหรับการรับข้อมูลชนิดเมโม โดยเตรียมที่ บนจอภาพในลักษณะฟูลสกรีนเอดิเตอร์ให้ทำการคีย์ข้อมูล
3. คีย์ข้อมูลที่ต้องการลงไป การจัดตัวอักษรในบรรทัดจะมีลักษณะคล้ายคลึงกับเอดิเตอร์ ของไชด์คิก และเวิร์ดสตาร์ คอนโทรลคีย์ที่สำคัญสำหรับการแก้ไขระหว่างการคีย์ข้อมูล ประกอบด้วย

คีย์	ความหมาย
->, <-	เลื่อนเคอร์เซอร์ไปทางขวาหรือซ้าย 1 ตัวอักษร
! , !	เลื่อนเคอร์เซอร์ขึ้นหรือลง 1 บรรทัด
Ins	ขอแทรกตัวอักษร
Del	ขอลบตัวอักษรตรงเคอร์เซอร์
Home	เลื่อนเคอร์เซอร์ไปข้างหน้า 1 คำ
End	เลื่อนเคอร์เซอร์ไปข้างหลัง 1 คำ

Ctrl - N (^ - N) ขอบทกรบทัด

Ctrl - T (^ - T) ลบคำที่อยู่ทางขวาของเคอร์เซอร์ 1 คำ

Ctrl - Y (^ - Y) ลบบรรทัด

PgUp เลื่อนขึ้นไป 1 หน้า

PgDn เลื่อนลงไป 1 หน้า

Ctrl - K F (^ - K F) เพื่อค้นหาค่าในเอกสาร

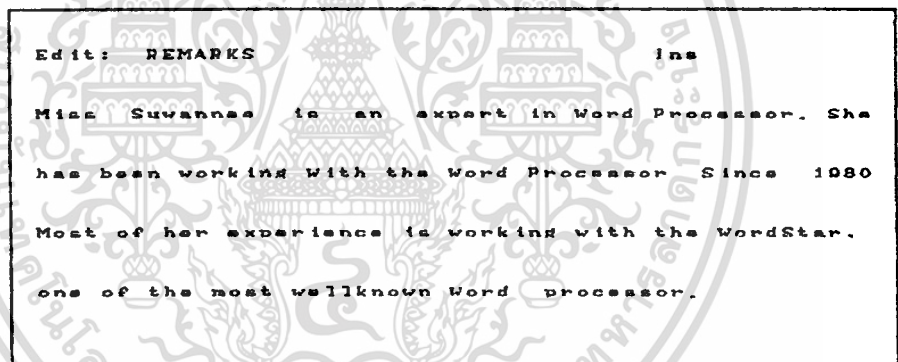
Ctrl - K L (^ - K L) เพื่อค้นหาค่าซ้ำอีกครั้ง

Ctrl - K B (^ - K B) เพื่อจัดพารากราฟใหม่หลังการแก้ไข

Ctrl - K R (^ - K R) เพื่ออ่านข้อความจากไฟล์อื่นมาใช้

Ctrl - K W (^ - K W) เพื่อเขียนข้อความเก็บในไฟล์

ในทันสมมติให้คีย์ข้อความของเมโมฟิลด์ ดังแสดงต่อไปนี้คือ



4. เมื่อคีย์ข้อความเสร็จก็ให้กด Ctrl-End เพื่อเก็บข้อมูลไว้ในไฟล์ข้อมูลชนิดเมโมจะถูกเก็บไว้ในไฟล์แยกต่างหากจากไฟล์ข้อมูล โดยจะเก็บไว้ในไฟล์แยกต่างหากจากไฟล์เหมือนกับตาเบสไฟล์ ไฟล์ชนิด .DBT นี้จะเก็บข้อมูลประเภทเมโม โดยชี้เนื้อที่อย่างต่ำ 512 ไบต์ และสูงสุดไม่เกิน 5,000 ไบต์ต่อเรคอร์ด

หลังจากป้อนข้อมูลสุดท้ายของเรคอร์ดแรกเสร็จแล้ว ดีเบสทรีวัลส์จะเตรียมพร้อมสำหรับการป้อนข้อมูลให้เรคอร์ดที่ 2 ต่อไปโดยแสดงผลบนจอภาพคือ

5.5.3 เมื่อต้องการแก้ไขข้อมูล

ในขณะที่ทำการคีย์ข้อมูลแต่ละเรคอร์ด อาจเกิดความผิดพลาดในการคีย์ข้อมูล การเลื่อนเคอร์เซอร์ไปมา เพื่อทำการแก้ไขตัวอักษร หรือตัวเลขที่พิมพ์ผิดไปแล้ว ทำได้โดยการใช้คอนโทรลคีย์ดังต่อไปนี้คือ

คีย์	ความหมาย
->, <-	เลื่อนเคอร์เซอร์ไปทางขวาหรือซ้าย 1 ตัวอักษร
! , !	เลื่อนเคอร์เซอร์ขึ้นหรือลง 1 ฟิลด์
Home	เลื่อนเคอร์เซอร์ไปต้นฟิลด์
End	เลื่อนเคอร์เซอร์ไปที่ท้ายฟิลด์
PgUp	เลื่อนขึ้น 1 เรคอร์ด
PgDn	เลื่อนลง 1 เรคอร์ด
Del	ลบตัวอักษรตรงเคอร์เซอร์
Ctrl - Y (^ - Y)	ลบข้อมูลในฟิลด์
Ctrl - U (^ - U)	ลบข้อมูลทั้งเรคอร์ด
Ins	ขอแทรกตัวอักษร
Ctrl-End (^ - End)	เก็บข้อมูลที่แก้ไขแล้วออกสู่โหมดการรับคำสั่ง
Esc	ยกเลิกข้อมูลที่แก้ไขที่ทำได้แล้ว ออกสู่โหมดการรับคำสั่ง
Ctrl-Home (^ - Home)	ขอแก้ไขข้อความในเมโมฟิลด์ เคอร์เซอร์ต้องอยู่ตรงเมโมฟิลด์จึงจะกด Ctrl-Home ได้

คอนโทรลคีย์ต่าง ๆ เหล่านี้สามารถเรียกมาใช้ปรากฏบนจอได้โดยกดฟังก์ชันคีย์ F1

5.5.4 การแทรกและลบตัวอักษร

วิธีแก้ไขความผิดพลาดของการคีย์ข้อมูลวิธีหนึ่งก็คือ การเลื่อนเคอร์เซอร์ไปในตำแหน่งที่พิมพ์ผิดแล้วคีย์ตัวอักษรหรือตัวเลขใหม่ที่ลงไปซ้ำที่เดิม แต่ในบางกรณีข้อมูลที่คีย์ลงไปแล้วอาจยาวมากและความผิดพลาดเกิดอยู่บริเวณต้น ๆ หรือกลางข้อความ การเลื่อนเคอร์เซอร์กลับไปเพื่อคีย์ทับของเก่าอาจเสียเวลาในกรณีเช่นนี้เราอาจใช้วิธีการลบตัวอักษรหรือแทรกตัวอักษร โดยการใช้คอนโทรลคีย์ คือ

Ins	การขอแทรกตัวอักษรตรงตำแหน่งของเคอร์เซอร์
Del	การขอลบตัวอักษรตรงตำแหน่งของเคอร์เซอร์
Ctrl-Y	การขอลบข้อมูลในฟิลด์ที่เคอร์เซอร์อยู่ออกทั้งหมด

ตัวอย่างเช่นในการคีย์ ชื่อ SUWANNEE ROONGPETCH เกิดคีย์ตัวอักษรตก

ไปเป็นดังนี้คือ

SWANNEE ROONGPETCH

เราต้องการแทรกตัว U หลังตัว S วิธีการแทรกตัวอักษรนี้ทำได้โดย

1. เลื่อนเคอร์เซอร์ไปให้ตรงกับตัวอักษรที่ต้องการแทรก คือ P (โดยใช้ คีย์ \rightarrow <-หรือ $!$, $!$)
2. คีย์ Ins จะเห็นคำว่า Insert บนบรรทัดบนสุดของจอภาพ
3. คีย์ตัวอักษรที่ต้องการแทรกลงไปคือ U จะได้ SUWANNEE

ROONGPETCH

4. คีย์ Ins อีกครั้งเพื่อยกเลิกการขอแทรกอักษร (ถ้าไม่มีคีย์ Ins ซ้ำดีเบสที่ผลลัพธ์จะถือว่าการคีย์อักษรตัวต่อ ๆ ไป เป็นการขอแทรกอักษรตัวต่อไป เป็นการขอแทรกอักษรทั้งสิ้น)

5.5.5 ลบตัวอักษร

ส่วนการลบตัวอักษรที่คีย์เกินไว้ ตัวอย่างเช่น

SUWANNEE ROONGPETCH

ซึ่งมีอักษร N เกินมา 1 ตัว การลบตัวอักษรทำได้โดย

1. เลื่อนเคอร์เซอร์ไปตรงกับตัวอักษรที่ต้องการจะลบออก ในที่นี้คือ N (โดยใช้ \rightarrow , <-หรือ $!$, $!$) SUWANNEE ROONGPETCH
2. คีย์ Del ตัวอักษรตรงตำแหน่งของเคอร์เซอร์จะถูกลบออกไปเป็น SUWANNEE ROONGPETCH

5.5.6 ลบข้อมูลในฟิลด์

ในกรณีที่ต้องการลบข้อมูลในฟิลด์ที่คีย์ไว้แล้วออกทั้งหมด ก็สามารถใช้คอน

โทรลคีย์ \wedge - Y ในการลบข้อมูลดังขั้นตอนคือ

1. เลื่อนเคอร์เซอร์ไปในฟิลด์ที่ต้องการลบออก (อยู่ตรงไหนก็ได้ในฟิลด์)
2. คีย์ \wedge - Y

ข้อมูลที่มีอยู่ก็จะถูกลบออกทั้งหมดพร้อม ๆ กัน

5.5.7 ลบข้อมูลทั้งเรคอร์ด

สำหรับการลบข้อมูลทั้งเรคอร์ด ทำได้โดยใช้คีย์ \wedge - U ดังนี้คือ

1. เลื่อนเคอร์เซอร์อยู่ในเรคอร์ดที่ต้องการลบ

2. คีย์ ^ - U

ข้อมูลที่มีในเรคอร์ดทั้งหมดจะถูกลบทิ้งไป

5.5.8 เมื่อจะหยุดคีย์ข้อมูล

เมื่อคีย์ข้อมูลต่าง ๆ จนหมดแล้ว ในกรณีตัวอย่างนี้ คือเมื่อคีย์ข้อมูลจนหมดเรคอร์ดที่ 15 แล้ว คีย์เบสทรีพลัสก็จะเตรียมพร้อมให้เราคีย์ข้อมูลเรคอร์ดต่อไปในจังหวะนี้ให้คีย์รีเทิร์นเลข โดยไม่ต้องป้อนข้อมูลใด ๆ คีย์เบสทรีพลัสจะถือว่าเป็นการเสร็จสิ้นการป้อนข้อมูล และจะขึ้น สัญญาณเตรียมพร้อม เพื่อรับคำสั่งต่าง ๆ ต่อไป

5.6 การเตรียมไฟล์และการแสดงข้อมูล

คำสั่งสำหรับการเตรียมไฟล์

เมื่อได้สร้างไฟล์เพื่อเก็บข้อมูลต่าง ๆ ด้วยคำสั่ง CREATE แล้ว คีย์เบสทรีพลัสจะเก็บข้อมูลไว้ในดิสเกตต์ เมื่อใดที่เราต้องการเรียกข้อมูลมาดูหรือใช้งาน จะต้องสั่งให้คีย์เบสทรีพลัสเตรียมไฟล์ให้ก่อนด้วยคำสั่ง USE รูปแบบพื้นฐานของคำสั่งคือ

use <ชื่อไฟล์ >

คือคีย์คำสั่ง use แล้วตามด้วยชื่อไฟล์ที่ต้องการให้เตรียมไว้ใช้งานศัพท์ทางคอมพิวเตอร์เรียกว่าการเปิดไฟล์(open file)

คำสั่ง use นั้น เปรียบเสมือนกับการสั่งให้เลขานุการของเราไปหยิบเพิ่มข้อมูลที่ต้องการที่อยู่ในตู้เอกสารวางไว้บนโต๊ะทำงาน เพื่อเตรียมที่จะให้เราดูข้อมูลต่อไปนั่นเอง

เมื่อต้องการจะดูรายละเอียดของข้อมูลในไฟล์ personel ก็คีย์คำสั่ง

. use personel

คีย์เบสทรีพลัสจะเตรียมไฟล์ให้ใช้งานต่อไป

5.6.1 ขอชื่อไฟล์ในดิสค์

เมื่อต้องการดูชื่อไฟล์ข้อมูลที่สร้างด้วยคีย์เบส คำสั่งที่ใช้คือ

LIST FILE

DISPLAY FILE

หรือ DIR

คีย์เบสจะแสดงชื่อทุกไฟล์ที่เป็นไฟล์ชนิด .dbf ออกมา

เมื่อต้องการดูไฟล์ที่มีอยู่ในดิสค์ คำสั่งที่ใช้คือ

LIST FILE LIKE *.*

DISPLAY FILE LIKE *.*

หรือ DIR *.*

นอกจากนี้การแสดงผลไฟล์ในลักษณะต่างๆ อาจใช้สัญลักษณ์ * และ ? ได้ ดังตัวอย่างเช่น

เมื่อต้องการแสดงชื่อไฟล์ทุกชื่อที่ขึ้นต้นด้วย C คำสั่งที่ใช้คือ

DIR C*.*

เมื่อต้องการแสดงชื่อไฟล์ที่ยาว 5 ตัวอักษร และมีอักษร C อยู่ตรงกลาง คำสั่งที่ใช้คือ

DIR ??C??.*

เมื่อต้องการแสดงชื่อไฟล์ทุกไฟล์ที่อยู่ในไดเรกทอรีพาท ชื่อ DBASE คำสั่งที่ใช้คือ

DIR DBASE*.*

เครื่องหมายดอกจัน (*) ในทันทมาสถึงทุกชื่อและทุกประเภท ตัวอย่างเช่น ถ้าต้องการดูชื่อไฟล์ทุกไฟล์ที่เป็นประเภท PRG คำสั่งที่ใช้คือ

DIR *.PRG

5.6.2 คำสั่งแสดงข้อมูล

คำสั่งที่ใช้ในการแสดงข้อมูลของไฟล์ที่สำคัญคือ LIST และ DISPLAY ทั้งสองคำสั่งนี้มีลักษณะการทำงานคล้ายคลึงกันมาก ซึ่งจะได้กล่าวถึงในรายละเอียดต่อไป เมื่อเราต้องการดูข้อมูลในไฟล์ทั้งหมด คำสั่งที่ใช้คือ

LIST

หรือ LSIT ALL

หรือ DISPLAY ALL

การแสดงผลจะแบ่งเป็นคอลัมน์ตามฟิลด์ข้อมูลที่กำหนดไว้ โดยคอลัมน์แรกจะเป็นหมายเลขของเรคอร์ด คอลัมน์นี้ดีเบสจะแสดงออกมาให้เอง แต่ถ้าเราไม่ต้องการให้แสดงส่วนนี้ออกมาก็สามารถสั่งได้ด้วยคำสั่ง

LIST OFF

หรือ DISPLAY OFF

5.6.3 คู่มือโครงสร้างของไฟล์

เมื่อต้องการตรวจสอบรายละเอียดของโครงสร้างไฟล์ที่กำลังใช้งานอยู่ ก็สามารถเรียกดูได้โดยใช้คำสั่ง

LIST STRUCTURE

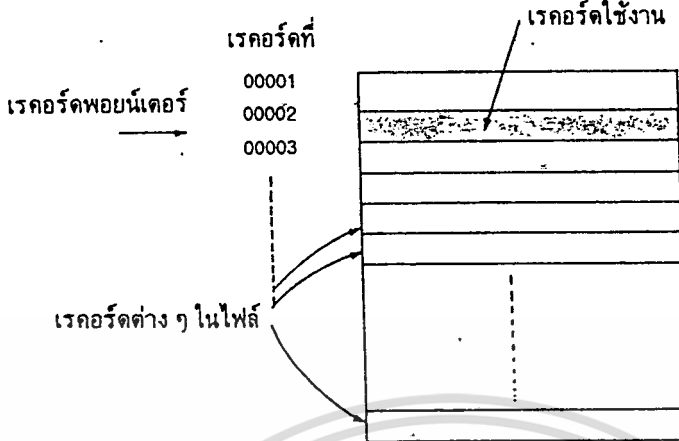
หรือ DISPLAY STRUCTURE

โครงสร้างไฟล์จะบอกรายละเอียดที่สำคัญคือ

1. ชื่อไฟล์ เช่น B:personel .dbf หมายถึง personel เป็นประเภท .dbf หรือ Database File คือไฟล์ที่สร้างโดยดีเบส ไฟล์นี้เก็บไว้ในดิสค์ไดรฟ์ B:
2. จำนวนเรคอร์ดทั้งหมดที่มีอยู่ในไฟล์
3. วันสุดท้ายที่มีการเปลี่ยนแปลง หรือแก้ไขข้อมูลในไฟล์
4. โครงสร้างข้อมูลที่สร้างไว้ว่ามีฟิลด์กี่ฟิลด์ ชื่ออะไรบ้าง เป็นประเภทอะไร ความกว้างเท่าใดและถ้าเป็นชนิดตัวเลขมีทศนิยมกี่ตำแหน่ง
5. จำนวนความกว้างทั้งหมดของเรคอร์ด ซึ่งความกว้างนี้มีค่าเท่ากับความกว้างทั้งหมดของฟิลด์รวมกันอีกหนคาง

5.6.4 เรคอร์ดพอยน์เตอร์

ในไฟล์ข้อมูลที่เตรียมไว้ในงานนั้น จะมีตัวชี้ตำแหน่งข้อมูลที่เรียกว่า เรคอร์ดพอยน์เตอร์ (Record Pointer) ซึ่งทำหน้าที่ชี้ตำแหน่งของข้อมูลในไฟล์ เรคอร์ดที่ถูกเรคอร์ดพอยน์เตอร์ชี้อยู่คือ แอกทีฟเรคอร์ด (Active Record) หรืออาจจะเรียกว่าเรคอร์ดใช้งาน ซึ่งเมื่อเริ่มเตรียมไฟล์ด้วยคำสั่ง USE นั้น เรคอร์ดพอยน์เตอร์ 1 เสมอ เรคอร์ดพอยน์เตอร์นี้จะเลื่อนไปมาได้ด้วยคำสั่งต่าง ๆ เช่น GOTO SKIP NEXT LIST หรือ DISPLAY และอื่น ๆ



เมื่อใช้คำสั่ง LIST หรือ DISPLAY ALL เรคอร์ดพอยน์เตอร์จะเลื่อนจากเรคอร์ดที่ 1 ไปเรื่อย ๆ พร้อมกับแสดงข้อมูลบนจอภาพ จนถึงเรคอร์ดสุดท้าย เรคอร์ดพอยน์เตอร์ก็จะหยุดอยู่ที่เรคอร์ดสุดท้ายนี้ จนกว่าจะมีการสั่งให้ย้ายเรคอร์ดพอยน์เตอร์ไปตำแหน่งอื่นต่อไป

5.6.5 DISPLAY ต่างกับ LIST อย่างไร

DISPLAY ต่างกับ LIST ในจุดสำคัญ ๆ 2 ประการคือ

1. เมื่อใช้คำสั่ง DISPLAY โดยไม่ได้ตามด้วย ALL คือเบสจะแสดงผลข้อมูลออกมาเพียง 1 เรคอร์ด โดยแสดงในตำแหน่งของเรคอร์ดพอยน์เตอร์ออกมาเท่านั้น ดังตัวอย่างเช่น

Record#	IDNO	NAME	SEX	MARRIED	AGE	DATE IN	SALARY	REMARKS
1	763	Suwannee	Rongpetch	F	.F.	23 03/12/84	5000.00	Memo

ในกรณีนี้แสดงว่า เรคอร์ดพอยน์เตอร์อยู่ที่เรคอร์ดที่ 1

2. เมื่อใช้คำสั่ง DISPLAY ALL ลักษณะการแสดงผลบนจอภาพจะคล้ายกับคำสั่ง LIST แต่แตกต่างกันตรงที่ ถ้าหากข้อมูลในไฟล์มีมากกว่า 20 เรคอร์ด และหยุดคอส เมื่อเราดูข้อมูลหมดแล้วกดคีย์ใดคีย์หนึ่ง แต่ถ้าใช้คำสั่ง LIST คือเบสก็จะแสดงข้อมูลไปเรื่อย ๆ ถ้าจำนวนเรคอร์ดมีมากเกินไปที่จะแสดงบนจอภาพเดียว เรคอร์ดต่าง ๆ ก็จะเลื่อนไปบนจอภาพจนดูไม่ทัน ซึ่งถ้าต้องการสั่งให้หยุดก็ได้คีย์ Ctrl - S จอภาพจะหยุดการเลื่อน เมื่อต้องการให้เลื่อนต่อไปก็กดคีย์ใด ๆ จอภาพจะแสดงเรคอร์ดต่อไปออกมา

นอกจากความแตกต่าง 2 จุดนี้แล้ว ลักษณะการใช้คำสั่ง DISPLAY และ LIST จะมีความคล้ายคลึงกัน ดังนั้นการอธิบายคำสั่งใด ๆ ต่อไปนี้ไม่ว่าจะเป็น LIST หรือ DISPLAY เราสามารถใช้คำสั่งทั้งสองทดแทนกันได้เลย

5.7 การแสดงข้อมูลในลักษณะต่าง ๆ

5.7.1 รูปแบบคำสั่ง LIST และ DISPLAY

จากตัวอย่างต่าง ๆ ที่แสดงมาแล้วจะเห็นว่าคำสั่ง LIST และ DISPLAY สามารถแสดงได้หลายลักษณะตามที่กำหนด รูปแบบคำสั่ง LIST และ DISPLAY

```
LIST [OFF] [<scope>] [<expression list>]
```

```
[FOR/WHILE <condition>][TO PRINT]
```

```
DISPLAY [OFF][<scope>][expression list]
```

```
[FOR/WHILE <condition>][TO PRINT]
```

ข้อความหรือคำสั่งที่อยู่ภายในเครื่องหมาย [] อาจจะไว้ได้ ความหมายของคำสั่งต่าง ๆ คือ

OFF หมายถึง การไว้ละเว้นไม่ต้องแสดงหมายเลขเรคอร์ด

scope หมายถึง ข้อความที่แสดงถึงขอบเขตของการแสดงข้อมูล เช่น

ALL หมายถึงการแสดงทุกเรคอร์ด

NEXT 5 หมายถึงการแสดง 5 เรคอร์ด ถัดไป เป็นต้น

expression หมายถึง ชื่อฟิลด์ที่ต้องการให้แสดงผล

FOR condition หมายถึง การกำหนดเงื่อนไขที่ต้องการให้แสดงเช่น

FOR SEX = "F" หมายถึงสำหรับฟิลด์ชื่อ SEX ที่มีข้อมูล

F บรรจุอยู่

TO PRINT หมายถึง การสั่งพิมพ์ผลออกทางเครื่องพิมพ์ (printer)

แทนการแสดง ผลบนจอภาพ

5.7.2 การกำหนดเงื่อนไข

ในการกำหนดเงื่อนไขหลังคำสั่ง FOR นั้น มีเครื่องหมายแสดงความสัมพันธ์ (relational operators) ที่สำคัญ 6 ตัวคือ

- = หมายถึง เท่ากับ
- > หมายถึง มากกว่า
- < หมายถึง น้อยกว่า
- >= หมายถึง มากกว่าหรือเท่ากับ
- <= หมายถึง น้อยกว่าหรือเท่ากับ
- <> หมายถึง ไม่เท่ากับ

นอกจากนี้ในกรณีที่ต้องการกำหนดเงื่อนไขแบบผสม ก็สามารถใช้เครื่องหมายตรรก (logical operators) ที่สำคัญอีก 3 ตัวคือ

- .AND. หมายถึง และ
- .OR. หมายถึง หรือ
- .NOT. หมายถึง ไม่ใช่

5.7.3 ตัวอย่างการ LIST และ DISPLAY ลักษณะต่าง ๆ

เพื่อให้เข้าใจถึงการแสดงผลด้วยคำสั่ง LIST และ DISPLAY ในลักษณะต่าง ๆ ลองพิจารณาตัวอย่างการแสดงผลดังต่อไปนี้

1. เมื่อต้องการให้แสดงข้อมูลเฉพาะฟิลด์ NAME SALARY และ DEATE IN ของข้อมูลทุกเรคอร์ด คำสั่งที่ใช้คือ

```
LIST ALL NAME, SALARU, DEATE IN
```

```
หรือ DISPLAY ALL NAME, SALARY, DATE IN
```

5.7.4 สรุปเงื่อนไขการแสดงผล

จากตัวอย่างที่แสดงมาเวลานั้น จะเห็นได้ว่าเงื่อนไขสำหรับการแสดงผลจะต้องมีการเปรียบเทียบข้อมูลในฟิลด์ สิ่งที่จะนำมาเปรียบเทียบจะต้องสอดคล้องกัน ซึ่งสามารถสรุปได้คือ

1. ฟิลด์ชนิดตัวอักษร จะต้องเปรียบเทียบกับตัวอักษร โดยตัวอักษรนี้จะต้องอยู่ภายในเครื่องหมายคำพูดด้วยเสมอ เช่น

LIST ALL FOR NAME = "Somsri"

2. พิลด์ชนิดตัวเลข จะต้องเปรียบเทียบกับตัวเลข โดยที่ตัวเลขไม่จำเป็น ต้องมีเครื่องหมายอะไรกำกับ เช่น

LIST ALL FOR SALARY >500

3. พิลด์ชนิดตรรก การเปรียบเทียบทำได้โดยการใช้อิลด์ ถ้าต้องการให้ ผลลัพธ์เป็นจริง แต่ถ้าต้องการให้ผลลัพธ์เป็นเท็จ ให้ใส่ค่า .NOT. นำหน้าชื่อฟิลด์นั้น ตัวอย่าง เช่น

LIST ALL FOR MARRIED

LIST ALL FOR .NOT. MARRIED

4. พิลด์ชนิดวันที่ การเปรียบเทียบต้องแปลงวันที่ในลักษณะของอักขระไปเป็น วันที่แบบจุดเลือนเสียก่อน ด้วยฟังก์ชัน Ctod หรือ อาจใช้ตัวเลขวันที่แบบจุดเลือนแทนก็ได้ ตัวอย่างเช่น

LIST ALL FOR DATE_IN >=Ctod ("10/01/85")

LIST ALL FOR DATE_IN >= 30000

5. พิลด์ชนิดเมโม ไม่สามารถนำมาใช้ได้ในการเปรียบเทียบเพื่อการ แสดงผล อย่างไรก็ตาม พิลด์เมโมนี้สามารถให้แสดงผลตามเงื่อนไขจากการเปรียบเทียบของฟิลด์ อื่นได้ ตัวอย่างเช่น

LIST ALL FOR REMARKS ='XYZ' ใช้ไม่ได้

LIST ALL FOR AGE >= 30 FIELDS REMARKS ใช้ได้

6. ถ้าการเปรียบเทียบภายในเงื่อนไข ทำได้โดยใช้ข้อมูลที่แตกต่าง จากชนิดของฟิลด์ดีเบสทรีพลัสจะไม่ยอมรับ เราจะต้องคีย์เงื่อนไขให้ถูกต้องเสียก่อนดีเบสทรีพลัส จึงจะทำการ

5.8 เรคอร์ดพอสเตอร์และการแก้ไขข้อมูล

5.8.1 การเลื่อนเรคอร์ดพอสเตอร์ด้วย GOTO

การเลื่อนเรคอร์ดพอสเตอร์ เป็นการสั่งให้เรคอร์ดพอสเตอร์เลื่อนไปตามเรคอร์ด ต่างๆในไฟล์ คำสั่ง GOTO เป็นการเลื่อนเรคอร์ดพอสเตอร์ไปยังหมายเลขเรคอร์ดพอสต์ เตอร์ไปยังหมายเลขเรคอร์ดที่ต้องการ รูปแบบคำสั่งคือ

GOTO n หรือ GO n'

เมื่อ n เป็นหมายเลขที่ต้องการ

5.8.2 GO TOP และ GO BOTTOM

คำสั่ง GO TOP เป็นการสั่งให้เรคอร์ดพอยน์เตอร์เลื่อนไปยังเรคอร์ดเตอร์เลื่อนไปยังเรคอร์ดที่ 1 ของ ไฟล์ ส่วน GO BOTTOM เป็นการสั่งให้เรคอร์ดพอยเตอร์เลื่อนไปที่เรคอร์ดสุดท้าย

5.8.3 SKIP คำสั่งสำหรับเลื่อนเรคอร์ดพอยน์เตอร์

การเลื่อนเรคอร์ดพอยเตอร์ไปยังข้างหน้าหรือข้างหลังจากตำแหน่งที่เรคอร์ดพอยน์เตอร์อยู่สามารถทำได้โดยใช้คำสั่งคือ

SKIP + n

เมื่อ n คือจำนวนเรคอร์ดที่ไป

5.8.4 การหาเรคอร์ดด้วย LOCATE และ CONTINUE

คำสั่ง locate ทำหน้าที่เลื่อนเรคอร์ดพอยน์เตอร์ไปยังเรคอร์ดที่ต้องการ รูปแบบคำสั่ง

LOCATE [<scope>][FOR <condition>] WHILE <condition>]

เมื่อ scope คือ ข้อความที่แสดงขอบเขตของการแสดงข้อมูล condition คือ เงื่อนไขที่ต้องการให้แสดง

5.8.5 การแก้ไขข้อมูลด้วยคำสั่ง EDIT

คำสั่ง edit เป็นการขอแก้ไขข้อมูลที่ละเรคอร์ด รูปแบบคำสั่ง

EDIT [<scope>][FIELDS<field list>][FOR<condition>]

[WHILE<condition>]

5.8.6 การแก้ไขหลายๆด้วยเรคอร์ดพร้อมกันด้วยคำสั่ง BROWSE รูปแบบของคำสั่ง

```
BROWSE [FIELDS<fields list>][WIDTH<expr>][LOCK<expr>]
[FREEZE<fields>][NOFOLLOW][NOAPPEND][NOMENU]
```

5.9 การเพิ่มและลบข้อมูลในไฟล์

5.9.1 คำสั่งที่ใช้ในการเพิ่มข้อมูลคือ APPEND ซึ่งมีรูปแบบคือ

```
APPEND [BLANK /FROM <file name>][FOR <condition>][TYPE]
<file type>][/DELIMITED [WITH BLANK /<delimiter>]]
```

BLANK หมายถึงเพิ่มเรคอร์ดว่าง

FROM <file name> หมายถึงการเพิ่มเรคอร์ดโดยการนำเรคอร์ดจากไฟล์อื่นมาเพิ่ม

FOR <condition> หมายถึง การเพิ่มข้อมูลเฉพาะที่มีเงื่อนไข ตามที่กำหนด

TYPE <file type> หมายถึง การระบุประเภทของไฟล์ที่จะนำมาเพิ่ม

DELIMITED หมายถึง การแยกฟิลด์ด้วยเครื่องหมาย ,

DELIMITED WITH BLANK หมายถึง การแยกฟิลด์ด้วยช่องว่าง

DELIMITED WITH <delimiter> หมายถึง การแยกฟิลด์ด้วยเครื่องหมายที่กำหนดด้วย <delimiter>

5.9.2 การแทรกข้อมูลด้วยคำสั่ง INSERT

คำสั่ง INSERT เป็นการขอแทรกข้อมูลลงในไฟล์ตามตำแหน่งของเรคอร์ดที่ต้องการ
การ รูปแบบคำสั่งคือ

```
INSERT [BLANK] [BEFORE]
```

เมื่อ BLANK เป็นการขอแทรกเรคอร์ดเปล่า

BEFORE เป็นการขอแทรกในตำแหน่งก่อนเรคอร์ดที่เรคอร์ดพอยน์เตอร์ชี้อยู่

INSERT เป็นการขอแทรกข้อมูลหลังเรคอร์ดที่เรคอร์ดพอยน์เตอร์ชี้อยู่

5.9.3 การลบข้อมูลด้วยคำสั่ง DELETE

คำสั่ง DELETE ทำหน้าที่ลบข้อมูลออกจากไฟล์ โดยมีรูปแบบการเขียนคำสั่ง คือ
DELETE [<SCOPE>] [FOR<condition>]
[WHILE <condition>]

เมื่อ Scope คือขอบเขตของการลบข้อมูลที่ต้องการ
condition คือเงื่อนไขของการลบข้อมูล
DELETE RECORD 5 เป็นการลบข้อมูล 1 เรคอร์ดที่ 5
DELETE ALL FOR DEALER = "BOON SINTHAI" เป็นการลบข้อมูลทุก
เรคอร์ดที่มีฟิลด์ DEALER= "BOON SINTHAI"
DELETE ALL WHILE PRICE <= 300 เป็นการลบข้อมูลตั้งแต่เรคอร์ด
แรกจนถึงเรคอร์ดที่มี ฟิลด์ PRICE > 300 จึงหยุดการลบข้อมูล

คำสั่ง DELETE นั้นยังสามารถทำการลบข้อมูลออกจากไฟล์อย่างแท้จริงเป็นแต่เพียงทำให้เรคอร์ดที่ถูกลบนั้นเหมือนไม่มีอยู่ในไฟล์ โดยการใส่เครื่องหมายไว้หน้าเรคอร์ด ซึ่งเรคอร์ดใดที่มีเครื่องหมายดอกจันอยู่ ดีเบสจะถือเสมือนว่าไม่มีเรคอร์ดนั้นอยู่ในไฟล์ แต่ข้อมูลจริงๆ ยังมีอยู่ไม่ถูกลบออกแต่อย่างใด และเราสามารถนำข้อมูลกลับมาใช้ได้ใหม่โดยใช้คำสั่ง RECALL

5.9.4 การเรียกข้อมูลคืนด้วยคำสั่ง RECALL

ข้อมูลเรคอร์ดที่มีดอกจันอยู่สามารถเรียกกลับมาใช้งานได้ใหม่โดยใช้คำสั่ง RECALL ซึ่งมีรูปคำสั่งคือ

RECALL [<SCOPE>] [FOR <condition>]
[WHILE <Condition>]

5.9.5 การลบข้อมูลอย่างถาวรด้วยคำสั่ง PACK

ในกรณีที่ต้องการลบข้อมูลออกจากไฟล์อย่างถาวรก็ทำได้โดยคำสั่ง PACK หลังจากลบข้อมูลด้วยคำสั่ง DELETE แล้ว คำสั่ง PACK จะลบข้อมูลที่มีดอกจันอยู่ออกจากไฟล์อย่างถาวร และไม่สามารถเรียกคืนมาด้วยคำสั่ง RECALL

5.10 การก๊อปปี้ไฟล์

5.10.1 การก๊อปปี้ไฟล์ข้อมูลของดีเบสด้วย COPY

คำสั่งที่ใช้ในการก๊อปปี้ข้อมูลของดีเบสคือ COPY ซึ่งมีรูปแบบคำสั่งคือ

```
COPY TO <file name>[[<scope>]][FIELD<field list>]
      [FOR<condition>][WHILE<condition>]
      [TYPE]<file type>]/DELIMITED [WITH
      BLANK /<delimiter>]
```

เมื่อ file name คือ ชื่อไฟล์ใหม่ที่ต้องการก๊อปปี้ไปไว้

scope คือ ข้อความที่แสดงถึงขอบเขตข้อมูลที่ต้องการก๊อปปี้ไป

condition คือ ข้อความที่กำหนดเงื่อนไขในการก๊อปปี้

field list คือ ฟิลด์ที่จะก๊อปปี้ไปไว้

TYPE<file type> คือ เป็นการกำหนดให้ก๊อปปี้ไปไว้ในไฟล์ประเภทใด

DELIMITED WITH BLANK/<delimiter> เป็นการกำหนดว่าจะใช้

อักขระอะไรเป็นตัวแยกฟิลด์

5.10.2 ก๊อปปี้ข้อมูลทั้งหมด

สมมติว่าต้องการก๊อปปี้ข้อมูลในไฟล์ทั้งหมด เพื่อเก็บไว้ในไฟล์อีกไฟล์หนึ่ง โดยตั้ง

ชื่อไฟล์ใหม่ว่า NEWM.DBF คำสั่งที่ใช้คือ

```
COPY TO NEWM
```

เมื่อดีเบสก๊อปปี้ไฟล์เสร็จจะได้ไฟล์ใหม่อีก 1 ไฟล์ ชื่อ NEWM.DBF ซึ่งมี

โครงสร้างและเก็บข้อมูลเหมือนกับ MACHINE.DBF ทุกอย่าง เมื่อเรียกไฟล์ NEWM แล้วทำ

การ LIST ออกมาก็จะได้ข้อมูล และโครงสร้างเหมือนไฟล์ MACHINE คำสั่งสำหรับการก๊อปปี้

การแสดงโครงสร้าง ตลอดจนข้อมูลของ NEWM

5.10.3 เมื่อต้องการก๊อปปี้เฉพาะบางฟิลด์

ในกรณีที่ต้องการก๊อปปี้ข้อมูลเฉพาะบางฟิลด์ไปเก็บไว้ในไฟล์ใหม่ ก็สามารถทำได้

โดยใส่รายชื่อฟิลด์ที่ต้องการก๊อปปี้ลงไปเป็นคำสั่ง COPY ในทำนองเดียวกันกับกรณีที่ต้องการก๊อปปี้

ข้อมูลเฉพาะบางลักษณะ ก็สามารถใช้อธิบายความกำหนดเงื่อนไขตามหลังคำสั่ง FOR ได้ ตัวอย่าง

เช่น

```
เมื่อต้องการก๊อปปี้ข้อมูลเฉพาะชื่อเครื่องจักร (MNAME) วันที่ติดตั้ง (INST DATE)
```

และบริษัทตัวแทนจำหน่ายเครื่องจักร (DEALER) โดยเก็บไว้ในไฟล์ชื่อ MEWM1 ชุดคำสั่งที่ใช้ และการแสดงผลต่าง ๆ แสดงดังต่อไปนี้

5.10.4 เมื่อต้องการก๊อปปี้เฉพาะโครงสร้างฟิลด์

ในกรณีที่ต้องการก๊อปปี้เฉพาะโครงสร้างของไฟล์ไว้ โดยไม่ต้องการก๊อปปี้ข้อมูลในไฟล์ไว้ด้วย คำสั่งที่ใช้คือ COPY STRUCTURE

สำหรับกรณีที่ต้องการก๊อปปี้เฉพาะบางฟิลด์ของโครงสร้างข้อมูลไปไว้ในไฟล์ใหม่ ก็ทำได้โดยการใส่ชื่อฟิลด์ข้อมูลที่ต้องการตามหลังคำสั่ง FIELD

5.10.5 การก๊อปปี้ของโครงสร้างไปเป็นข้อมูลในไฟล์

ในกรณีที่ต้องการนำเอาโครงสร้างของไฟล์ใดไฟล์หนึ่งไปเป็นข้อมูลของไฟล์ ใช้คำสั่งคือ

COPY TO<file name> STRUCTURE EXTENDED

5.10.6 การก๊อปปี้ไฟล์ไปเป็นไฟล์ประเภทอื่น

การก๊อปปี้ไฟล์ข้อมูลที่สร้างด้วยดีเบสทรีพลัสไปใช้ในโปรแกรมอื่น หรือนำไปเก็บไว้ในฟอร์มแมตที่อ่านได้ด้วยโปรแกรมอื่น หรือนำไปเก็บไว้ในฟอร์มแมตที่อ่านได้ด้วยโปรแกรมอื่นทำได้โดยรูปแบบคำสั่ง

COPY TO<file name>[[TYPE]<file type>/DELIMITED

[WITH BLANK/<delimiter>]]

ตัวอย่างเช่นเมื่อต้องการก๊อปปี้ข้อมูลจากไฟล์ MACHINE ไปเป็นไฟล์ประเภทเท็กซ์ไฟล์ ที่มีชื่อว่า TEMP1.TXT โดยแยกฟิลด์ด้วยเครื่องหมายและใส่เครื่องหมายคำพูดเปิดปิด " " กำกับชนิดตัวอักษร คำสั่งที่ใช้และการแสดงผล

5.10.7 การก๊อปปี้ข้อมูลจากไฟล์ประเภทอื่นที่ไม่ใช่ .DBF

ในกรณีที่ต้องการก๊อปปี้ข้อมูลที่สร้างจากโปรแกรมอื่น และไม่ได้เป็นประเภท .DBF เป็นไฟล์ข้อมูลที่สร้างจากภาษาเบสิกหรือเอดิเตอร์ต่าง ๆ เพื่อนำมาเพิ่มหรือใส่ไว้ในไฟล์ข้อมูลชนิด .DBF

เราสามารถใส่คำสั่ง APPEND รูปแบบคำสั่งเพื่อใช้ในการก๊อปปี้ข้อมูลจากไฟล์ประเภทอื่นมาใช้ดีเบสทรีพลัสคือ

APPEND FORM <file name>[FOR<condition>]

[WHILE <condition>][<type>

<file type>]/DELIMITED

[WITH BLANK / <delimiter>]

สังเกตว่าการแทรกฟิลด์ใช้เครื่องหมาย และฟิลด์ประเภทตัวอักษรกำกับด้วยเครื่องหมายค่าพูดเปิดปิด " "

การก๊อปปี้ไฟล์ดังกล่าวเพื่อนำมาเพิ่มให้กับไฟล์ MACHINE ทำได้โดยใช้คำสั่ง

.APPEND FROM TEST.TXT DELIMITED

5.11 การเรียงลำดับข้อมูลในไฟล์

5.11.1 การเรียงลำดับข้อมูลด้วยคำสั่ง SORT

จากไฟล์ข้อมูลที่สร้างด้วยดีเบส เมื่อต้องการเรียงลำดับข้อมูลตามฟิลด์ใดฟิลด์หนึ่ง คำสั่งที่ใช้การเรียงลำดับข้อมูล คือ SORT ซึ่งมีรูปแบบคำสั่งคือ

SORT [<scope>] TO <new file> ON <field1>
[/A] [/C] [/D] [, <field2>] [/A] [/C] [/D]
[FOR <condition>] [WHILE <condition>]

เมื่อ scope หมายถึง ขอบเขตของการเรียงลำดับข้อมูล เช่น ALL หรือ NEXT 10 เป็นต้น

new file หมายถึง ชื่อไฟล์ใหม่ที่จะเก็บข้อมูลหลังการเรียงลำดับ field 1 field 2 หมายถึง ชื่อฟิลด์ที่จะใช้เป็นหลักในการเรียงลำดับข้อมูล

- /A เป็นการกำหนดให้เรียงจากน้อยไปมาก (Ascending)
 - /D เป็นการกำหนดให้เรียงจากมากไปน้อย (Descending)
 - /C เป็นการกำหนดให้ถือว่าอักษรตัวใหญ่กับ ตัวเล็กมีลำดับเหมือนกัน
- FOR<condition> และ WHILE <condition> หมายถึงการกำหนดเงื่อนไขไป สำหรับเรียงข้อมูล

5.11.1 ลักษณะการสร้างไฟล์ของคำสั่ง SORT

โดยลักษณะการทำงานของคำสั่ง SORT นั้น คือเบสทรีนัลจะสร้างไฟล์ขึ้นอีก 1 ไฟล์ทุกครั้งที่มีการเรียกใช้คำสั่ง SORT ไฟล์ใหม่ที่สร้างนี้จะมีโครงสร้างข้อมูลต่าง ๆ เหมือนไฟล์เดิมทุกอย่าง แต่การเรียงลำดับข้อมูลจะแตกต่างกันไปตามแต่ที่กำหนดไว้ในคำสั่ง sort

5.11.2 การเรียงข้อมูลด้วยคำสั่ง INDEX

INDEX เป็นคำสั่งที่ใช้ในการเรียงข้อมูลได้เช่นเดียวกับคำสั่ง SORT โดยมีรูปแบบของคำสั่งคือ

INDEX ON <key expression> TO <index file name>[UNIQUE]

เมื่อ key expression คือ นิพจน์ของชื่อฟิลด์ที่ใช้เป็นหลักในการเรียงลำดับข้อมูล

index file name คือ ชื่อไฟล์ที่ใช้เก็บลักษณะของการเรียงลำดับไฟล์ ประ

เภทนี้มีชื่อเรียกว่า อินเด็กซ์ไฟล์ (index file).NDX

unique คือ เป็นการกำหนดว่า ถ้ามีข้อมูลซ้ำในลักษณะเดียวกันกับที่กา

หนดใน key dpression ให้ใช้เพียงเรคอร์ดแรกเรคอร์ดเดียว

การเรียงลำดับข้อมูลด้วยคำสั่ง INDEX จะทำการเรียงจากน้อยไปหามากเท่านั้น

กรณีที่ต้องการเรียงจากมากไปหาน้อยทำได้โดยการใส่เครื่องหมายลบไว้ หน้าชื่อฟิลด์ข้อมูล ตัวอย่างคำสั่ง INDEX คือ

INDEX ON HCODE TO CINDEX

INDEX ON HCODE TO CINDEX

เป็นการเรียงลำดับข้อมูลโดยเรียงตามลำดับข้อมูลโดยเรียงตามรหัสเครื่องจักร

(HCODE) จากน้อยไปหามาก เก็บผลการเรียงไว้ในอินเด็กซ์ไฟล์ ชื่อ CINDEX.NDX

INDEX ON DEALER TO DINDEX

เป็นการเรียงลำดับข้อมูล โดยเรียงตามรายชื่อบริษัทตัวแทนจำหน่ายเครื่องจักร

(DEALER) จากมากไปหาน้อย เก็บผลการเรียงไว้ในอินเด็กซ์ไฟล์ ชื่อ DINDEX.NDX

การเรียงลำดับข้อมูลด้วยคำสั่ง INDEX สามารถใช้เรียงลำดับโดยใช้หลายฟิลด์เป็นหลักได้ เช่นเดียวกับคำสั่ง SORT ตัวอย่างเช่น

INDEX ON MTYPE + MNAME TO MMINDEX

เป็นการเรียงลำดับข้อมูลโดยเรียงตาม MTYPE ถ้า MTYPE ซ้ำกันให้ เรียงตาม

MNAME เก็บผลลัพธ์การเรียงไว้ในอินเด็กซ์ไฟล์ชื่อ MMINDEX.NDX

5.11.3 ขั้นตอนการเรียงลำดับข้อมูลด้วยคำสั่ง INDEX

1. เตรียมไฟล์ ด้วยคำสั่ง USE
2. เรียงข้อมูลด้วยคำสั่ง INDEX เมื่อเรียงข้อมูลเสร็จจะได้ผลลัพธ์บนจอภาพ
3. เรียกดูข้อมูลที่เรียงแล้ว ด้วยคำสั่ง
.USE MACHINE INDEX MINDEX

ลักษณะการสร้างไฟล์ของคำสั่ง INDEX

ในการเรียงลำดับข้อมูลด้วยคำสั่ง INDEX นั้น ดีเบสมิได้สร้างไฟล์ข้อมูลใหม่ขึ้น เหมือนกับวิธีการของ SORT แต่ดีเบสจะสร้างไฟล์อีกชนิดหนึ่งที่เรียกว่าอินเด็กซ์ไฟล์ อินเด็กซ์ไฟล์มิได้มีลักษณะเป็นไฟล์ข้อมูล จึงไม่สามารถเรียกใช้ด้วยคำสั่ง USE และเมื่อเรียกดูด้วยคำสั่ง LIST FILE ก็จะไม่สามารรถเห็นชื่อไฟล์ชนิดนี้ได้ อินเด็กซ์ไฟล์ที่สร้างจากคำสั่ง INDEX นี้เป็นประเภท .NDX ซึ่งเรียกดูชื่อไฟล์ด้วยคำสั่ง

.LIST FILE LIKE *.NDX

5.11.4 การเรียงลำดับข้อมูลจากมากไปหาน้อย

การเรียงลำดับข้อมูลจากมากไปหาน้อยทำได้โดยการใส่เครื่องหมายลบ หน้าชื่อฟิลด์ที่ต้องการใช้เป็นหลักในการเรียงลำดับ

5.11.5 การเรียงลำดับข้อมูลหลายฟิลด์พร้อมๆกัน

การเรียงลำดับข้อมูลหลายฟิลด์พร้อมๆกัน อาจทำได้ด้วยคำสั่ง SORT หรือ คำสั่ง INDEX สำหรับคำสั่ง SORT ฟิลด์ที่จะใช้เป็นหลักในการเรียงลำดับต้องเขียนเรียงลำดับจากฟิลด์ที่มีความสำคัญมากไปหาฟิลด์ที่มีความสำคัญน้อย ตัวอย่างเช่น

```
SORT TO SORT1 ON MTYPE /A, DEALER, MNAME /D
```

เป็นการเรียงลำดับข้อมูลโดยใส่ฟิลด์ MTYPE เป็นหลักเรียงจากน้อยไปหามาก ฟิลด์ DEALER เป็นฟิลด์รองลงมา เรียงจากน้อยไปมากเช่นกัน (ไม่ใส่ /A หรือ /D ถ้ากับมีความหมายเหมือนกับ /A) และพิมพ์ MNAME เป็นฟิลด์ที่มีความสำคัญสุดท้ายในการจัดเรียงลำดับ

จำนวนฟิลด์ที่จะเป็นหลักในการเรียงลำดับนี้ ไม่จำกัดว่าจะมีได้กี่ฟิลด์ แต่ความยาวของคำสั่งจะต้องไม่เกิน 256 ตัวอักษร(รวมช่องว่างด้วย)

สำหรับคำสั่ง INDEX การเรียงลำดับโดยใช้หลายฟิลด์ จะต้องนำชื่อฟิลด์

มาเขียนเป็นนิพจน์ โดยให้ฟิลด์ที่มีความสำคัญมากที่สุดอยู่เป็นอันดับแรก และฟิลด์ที่มีความสำคัญรองลงไปอยู่เป็นลำดับถัดไปตามลำดับ ตัวอย่างเช่น

INDEX ON MTYPE +DEALET+ MNAME TO MSORT

เป็นการเรียงลำดับข้อมูลโดยใช้ฟิลด์ MTYPE เป็นฟิลด์หลัก ฟิลด์ DEALER เป็นฟิลด์รองลงมา และฟิลด์ MNAME เป็นความสำคัญลำดับสุดท้าย

จำนวนฟิลด์ที่ใช้ในการเรียงลำดับไม่จำกัดว่าจะต้องไม่เกินกี่ฟิลด์ แต่ความยาวสูงสุดของนิพจน์ชื่อฟิลด์จะต้องไม่เกิน 100 ตัวอักษร

5.11.6 ข้อสรุปในการเรียงลำดับข้อมูลหลายฟิลด์พร้อม ๆ กัน

1. การเรียงลำดับข้อมูลอาจใช้คำสั่ง SORT หรือ INDEX ก็ได้
2. ในกรณีคำสั่ง SORT เมื่อต้องการใช้หลายฟิลด์หลักจะต้องอยู่ลำดับแรก ฟิลด์รองเป็น ฟิลด์ถัด ๆ ไป ตามลำดับ ตัวอย่างเช่น
SORT TO MSORT ON MTYPE, DEALER, PRICE
3. จำนวนฟิลด์สำหรับคำสั่ง SORT มีได้ไม่จำกัด แต่ความยาวคำสั่งต้องไม่เกิน 256 ตัวอักษร (รวมช่องว่างด้วย)
4. การ INDEX หลายฟิลด์พร้อม ๆ กันทำได้โดยนำชื่อฟิลด์มาเชื่อมต่อกันด้วยเครื่องหมาย + เรียกว่า นิพจน์ชื่อฟิลด์ ตัวอย่างเช่น
INDEX ON DEALER + MNAME
INDEX ON DEALET + INST_DATE
5. เมื่อฟิลด์ที่จะทำการ INDEX เป็นฟิลด์ชนิดตัวเลข และฟิลด์ชนิดตัวอักษรจะต้อง แปลงฟิลด์ตัวเลขให้เป็นชนิดตัวอักษรเสียก่อน
ตัวอย่างเช่น

INDEX ON DEALER + STR(PRICE, 10, 2)

INDEX ON DEALER + INST_DATE + STR(PRICE, 10, 2)

ฟังก์ชันที่ใช้ในการแปลงฟิลด์ชนิดตัวอักษรคือ STR หรือสตรีฟังก์ชัน (String Function) ซึ่งมีรูปแบบคือ STR (ชื่อฟิลด์หรือค่าตัวเลข, ความกว้างฟิลด์, จำนวนทศนิยม) ฟังก์ชัน STR นี้จะแปลงข้อมูลที่เป็นตัวเลขให้เป็นชนิดตัวอักษรเสีย ก่อน จึงจะสามารถนำไปทำการ INDEX ร่วมกับฟิลด์ชนิดตัวอักษรอื่นได้

6. จำนวนฟิลด์สำหรับการ INDEX มีได้ไม่จำกัด แต่ความยาวของนิพจน์ ชื่อฟิลด์จะต้อง ยาวไม่เกิน 100 ตัวอักษร
7. ฟิลด์ที่จะใช้เป็นหลักในการเรียงลำดับข้อมูลด้วยคำสั่ง SORT หรือ INDEX นั้นต้องเป็นฟิลด์ประเภทตัวอักษร (Character) ตัวเลข (Numeric) และวันที่ (Date) เท่านั้นฟิลด์ชนิด ตรรก(Logical) และเมโม (Memo) ไม่สามารถใช้เป็นหลักในการเรียงลำดับข้อมูลได้

5.12 การ REINDEX และการค้นหาข้อมูลจากอินเด็กซ์ไฟล์

5.12.1 การทำ INDEX ซ้ำด้วยคำสั่ง REINDEX

เมื่อไฟล์ข้อมูลที่เคยได้ทำการ INDEX ไว้แล้วมีการแก้ไขเพิ่มเติม หรือลดข้อมูล การเรียงลำดับข้อมูลอาจเปลี่ยนแปลงไปจากที่เคยทำการ INDEX ไว้ในกรณีนี้ไม่จำเป็นต้องทำการ INDEX ใหม่ เพียงแต่ใช้คำสั่ง REINDEX ดีเบสก็จะจัดเรียงลำดับข้อมูลให้ใหม่ ดังนั้นในกรณีนี้จะต้องทำการ REINDEX ดังวิธีการคือ

1. เตรียมไฟล์ MACHINE และอินเด็กซ์ไฟล์ DPINDEX ด้วยคำสั่ง
.USE MACHINE INDEX MINDEX
2. ทำการ REINDEX ดีเบสจะทำการ INDEX ซ้ำใหม่และเก็บผลลัพธ์ไว้
3. ให้ดีเบสแสดงข้อมูลออกมาด้วยคำสั่ง LIST

5.12.2 การกำหนดให้อินเด็กซ์ไฟล์ปรับปรุงเมื่อมีการแก้ไขข้อมูลรูปแบบคำสั่งคือ

SET INDEX TO <index file name>

เมื่อเริ่มใช้คำสั่งนี้ ก็ไม่จำเป็นต้องใช้คำสั่ง REINDEX อีก พิจารณาตัวอย่าง การใช้คำสั่ง SET INDEX TO ดังต่อไปนี้

ต้องการให้การเพิ่มข้อมูลนี้มีผลต่อการอินเด็กซ์ของอินเด็กซ์ไฟล์ MINDEX และ PINDEX ขั้นตอนในการใช้คำสั่ง SET INDEX TO ทำได้ดังนี้คือ

1. เตรียมเพิ่มข้อมูลด้วยคำสั่ง
.USE MACHINE

2. ใช้คำสั่ง SET INDEX TO เพื่อให้การเพิ่มข้อมูลมีผลต่ออินเด็กซ์ไฟล์
SET INDEX TO MINDEX, PINDEX

3. เพิ่มข้อมูลให้แก่ไฟล์ด้วยคำสั่ง
.APPEND

5.12.3 การค้นหาข้อมูลด้วยคำสั่ง FIND

การเรียงลำดับข้อมูลด้วยคำสั่ง INDEX นอกจากทำให้ประหลัดเนื้อหาของดิสเกตต์ และสามารถเรียงลำดับข้อมูลได้หลายฟิลด์แล้ว ข้อดีที่สำคัญอีกประการหนึ่งของการเรียงลำดับข้อมูลด้วยคำสั่ง INDEX คือการค้นหาข้อมูลที่ได้รวดเร็วด้วยคำสั่ง FIND โดยมีรูปแบบของคำสั่งคือ

FIND<character string>

เมื่อ character string เป็นตัวอักษรหรือข้อความของเรคอร์ดที่ต้องการค้นหา

การค้นหาข้อมูลด้วยคำสั่ง FIND จะใช้ได้เฉพาะกับไฟล์ที่เรียงลำดับด้วยคำสั่ง INDEX เท่านั้น และข้อความที่ต้องการหาจะต้องเป็นของฟิลด์ที่ใช้เป็นหลักในการเรียงลำดับข้อมูล คำสั่ง FIND จะสามารถค้นหาข้อมูลที่ต้องการได้ในเวลาเพียงประมาณ 2 วินาที ไม่ว่าเรคอร์ดนั้นจะอยู่ในตำแหน่งใดในไฟล์ซึ่งค้นหาหน่อสมากเมื่อเทียบกับการค้นหาด้วยคำสั่ง LIST FOR โดยเฉพาะอย่างยิ่ง เมื่อไฟล์ข้อมูลมีเรคอร์ดจำนวนมาก

5.12.4 การค้นหาข้อมูลด้วยการใช้คำสั่ง SEEK

SEEK <expression>

เมื่อ expression เป็นชนิดตัวอักษรจะต้องมีเครื่องหมายคำพูด " " กำกับ แต่ถ้าเป็นนิพจน์ ชนิดตัวเลขหรือนิพจน์ที่ใช้ตัวแปรจากหน่วยความจำ ก็สามารถเขียนได้โดยตรงถ้าไม่มีข้อมูลเรคอร์ดใดสอดคล้องกับนิพจน์ที่กำหนด ดีเบสทีพลัสจะแสดงข้อความ NO FIND ให้รู้ คำสั่ง SEEK มีผลเช่นเดียวกับคำสั่ง FIND คือเรคอร์ดพอยน์เตอร์จะเปลี่ยนไปเรคอร์ดแรกที่ตรงกับนิพจน์ที่กำหนด

5.13 การแก้ไขโครงสร้างและการเปลี่ยนแปลงข้อมูล

5.13.1 การเปลี่ยนแปลงข้อมูลด้วยคำสั่ง CHANGE

CHANGE [<scope>][<FIELDS <field list>][<FOR/WHILE<condition>]

เมื่อ scope คือ ข้อความแสดงถึงขอบเขตการแก้ไขข้อมูล
fields list คือ ชื่อของฟิลด์ที่ต้องการเปลี่ยนแปลง

5.13.2 การแก้ไขข้อมูลด้วยคำสั่ง REPLACE

REPLACE [<scope>]<field> WITH <exp>[,<field>
WITH <exp>...] [<FOR/WHILE<condition>]

เมื่อ scope คือข้อความแสดงถึงขอบเขตการเปลี่ยนแปลงข้อมูล
field คือชื่อฟิลด์ข้อมูล
exp คือนิพจน์ที่แสดงถึงการเปลี่ยนแปลงหรือทดแทนข้อมูล
condition คือเงื่อนไขที่ต้องการเปลี่ยนแปลงข้อมูล

5.14 การคำนวณตัวเลขในไฟล์

5.14.1 นับจำนวนเรคอร์ดด้วยคำสั่ง COUNT

COUNT [<scope>][<FOR/WHILE<condition>][<TO<memvar>]

เมื่อ scope คือข้อความที่แสดงถึงขอบเขตการนับเรคอร์ด
condition คือเงื่อนไขของเรคอร์ดที่ต้องการนับ
memvar คือชื่อตัวแปรที่เก็บค่าจากผลการนับไว้

5.14.2 หาผลรวมของฟิลด์ด้วยคำสั่ง sum

SUM [<expression list>] [<scope>] [<FOR/WHILE<condition>]
[<TO<memvar>]

5.14.3 หาค่าเฉลี่ยด้วยคำสั่ง AVERAGE

AVERAGE [expression list] [scope] [<FOR/WHILE
<condition>] [<TO <memvar>]

5.14.4 การหาผลรวมของฟิลด์แล้วเก็บไว้ในไฟล์ใหม่ด้วยคำสั่ง TOTAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TOTAL ON <key field> TO <file name> [<scope>]

[FIELD <field list>] [FOR/WHILE <condition>]

เมื่อ key field	คือฟิลด์ที่ใช้เป็นหลัก
file name	คือชื่อไฟล์ที่เก็บผลรวม
scope	คือข้อความที่แสดงถึงขอบเขตการหาผลรวม
condition	คือเงื่อนไขที่กำหนดให้หาผลรวม
field list	คือฟิลด์ที่ต้องการหาผลรวม

ไฟล์ข้อมูลที่ใช้ในการหาผลรวมจะต้องเรียงลำดับข้อมูลตามคีย์ฟิลด์เสียก่อน จะด้วยวิธี index หรือ sort ก็ได้

5.15 การทำรายงานจากไฟล์ข้อมูล

5.15.1 การทำรายงาน

ในการจัดทำรายงาน เพื่อให้แสดงผลข้อมูลใส่ไฟล์ตามรูปแบบที่ต้องการนั้น สามารถทำได้ตามขั้นหลัก ๆ ดังนี้คือ

1. เตรียมไฟล์ข้อมูลที่ต้องการทำรายงานด้วยคำสั่ง USE
2. ออกแบบลักษณะของรายงานที่ต้องการด้วยการสร้างไฟล์ประเภท .FRM ด้วยคำสั่ง CREATE REPORT
3. กำหนดรูปแบบการพิมพ์รายงาน เช่น ข้อความของเฮดดิ้ง ความกว้างของหน้ากระดาษ จำนวนบรรทัดต่อหน้า หัวของคอลัมน์ ข้อมูลในแต่ละคอลัมน์และอื่น ๆ จากนั้นจึงเก็บรูปแบบไว้โดย ^- End
4. สั่งให้ตีเบสท์ปริ้นต์สั่งพิมพ์งานโดยใช้คำสั่ง REPORT FORM

5.15.2 รูปแบบคำสั่ง CREATE REPORT และ REPORT FORM

ไฟล์ประเภท .FRM ที่ใช้เพื่อเก็บรูปแบบของการพิมพ์รายงาน สามารถสร้างขึ้นได้ด้วยคำสั่ง CREATE REPORT ซึ่งมีรูปแบบคำสั่งคือ

CREATE REPORT <file name>

เมื่อ file name คือชื่อของไฟล์ที่เราตั้งขึ้นเพื่อให้เก็บรูปแบบของการพิมพ์รายงาน ในขั้นตอนการพิมพ์รายงาน จะต้องเรียกใช้คำสั่ง REPORT FORM รูปแบบคำสั่งคือ

REPORT FORM <file name > [<scope>] [FOR/WHILE<condition>]
 [TO PRINT][PLAIN] [NOJECT]
 [HEADING <character string>] [TO FILE <file name>]

เมื่อ file name คือชื่อของไฟล์

scope คือข้อความที่แสดงถึงขอบเขตการทำรายงาน

condition คือเงื่อนไขที่กำหนดให้ทำรายงาน

TO PRINT คือสั่งให้พิมพ์ออกทางเครื่องพิมพ์

PLAIN คือคำสั่งให้ยกเลิกการพิมพ์เลขหน้าและวันที่

HEADING คือคำสั่งให้พิมพ์ข้อความ character string บนหัวกระดาษ

ทุกแผ่น

NOEJECT คือคำสั่งให้พิมพ์ต่อเนื่องโดยไม่ต้องเว้นหัวกระดาษตอนขึ้นหน้าใหม่

TO FILE คือคำสั่งให้พิมพ์รายงานลงในไฟล์ซึ่งจะเก็บไว้ในไฟล์ประเภท

.TXT

5.15.3 พิมพ์รายงานผลการขาย

จากไฟล์ข้อมูล SALES ซึ่งเก็บข้อมูลด้านการขายของพนักงานแต่ละคนไฟล์ SALES มีโครงสร้างดังแสดง

การจัดทำรายงานเริ่มโดยสร้างแฟ้มเก็บรูปแบบของรายงานด้วยคำสั่ง CREATE REPORT ในที่นี้จะตั้งชื่อไฟล์ว่า SALEREP ดังนั้นคือคำสั่ง

- . USE SALES
- . CREATE REPORT SALEREP

ดีเบสทรีพลัส จะแสดงเมนูบนจอภาพดังแสดงคือ

Options	Groups	Columns	Locate	Exit 12:09:08 am
Page title				
Page width (positions)		80		
Left margin		8		
Right margin		0		
Lines per page		58		
Double space report		No		
Page eject before printing		Yes		
Page eject after printing		No		
Plain page		No		

CURSOR	<-- -->	Delete char:	Del	Insert column:	^N	Insert:	Ins
Char:		Delete word:	^T	Report format:	F1	Zoom in:	^PgDn
Word:	Home End	Delete column:	^U	Abandon:	Esc	Zoom out:	^PgUp

CREATE REPORT FROM: "C:\SALEREPR.FRM" Options: /2
 Select - - Leave menu - -
 Enter up to four lines of text to be displayed at the top of each report page.

5.15.4 เมนูแรกของการทำงาน

เมนูแรกของจอภาพจะแสดงเมนูคำสั่งบนบรรทัดแรก หรือคำสั่ง เพื่อกำหนดลักษณะรายงาน ซึ่งมีคำสั่งที่ต้องกำหนด 6 อย่างคือ

1. ข้อความที่จะพิมพ์บนหัวกระดาษ (Page Title) ส่วนการพิมพ์รายงานนี้จะพิมพ์ทุกแผ่นของรายงาน ในทันทีคือว่า
 The ABC Company SALES Report
 เสด็จขึ้นสามารถมีได้ถึง 4 บรรทัด เมื่อคีย์เสด็จเสร็จก็กดแป้น PgDn
2. คีย์ความยาวของตัวอักษรต่อบรรทัด (Page Width) (ค่าดีฟอลต์=80) แล้วกดรีเทิร์น
3. คีย์จำนวนการเว้นย่อหน้า (left margin) (ค่าดีฟอลต์ = 8) แล้วกดรีเทิร์น
4. คีย์จำนวนเว้นขอบกระดาษทางขวา (right margin) (ค่าดีฟอลต์ = 0) แล้วกดรีเทิร์น
5. คีย์จำนวนบรรทัดต่อหน้า (lines/page) (ค่าดีฟอลต์ = 58 บรรทัด)

แล้วกดรีเทิร์น

6. คีย์ y หรือ n เพื่อคอยคำถามที่ว่าต้องการให้พิมพ์บรรทัดหรือไม่
(y=yes,n=no)

5.15.5 แสดงรายงาน

เมื่อต้องการแสดงรายงานบนจอภาพ คำสั่งที่ใช้คือ

```
.use sales
.report form salerep
```

5.15.6 เมื่อต้องการให้พิมพ์รายงานออกทางเครื่องพิมพ์

เมื่อต้องการให้ตีเบสพิมพ์รายงานออกทางเครื่องพิมพ์ ก็สามารถทำได้โดย
เขียนคำสั่งแสดงคือ

```
.report form salerep to print
```

ตีเบสก็จะพิมพ์รายงานออกทางเครื่องพิมพ์ โดยมีรูปแบบเหมือนกับที่เห็นบนจอ

5.15.7 การชอแก้ไขรูปแบบรายงาน

สำหรับรูปแบบที่ถูกสร้างไว้ด้วยคำสั่ง reprot จะเก็บไว้ในไฟล์ประเภท

```
.modify report <file name>
```

5.16 การพิมพ์บัญชีรายชื่อเจ้าหน้าที่ของจดหมาย

ขั้นตอนในการจัดพิมพ์บัญชีรายชื่อสำหรับจัดส่งทางไปรษณีย์ทำได้ดังนี้คือ

1. เตรียมไฟล์ข้อมูลที่ต้องการพิมพ์บัญชีรายชื่อ ด้วยคำสั่ง use
2. สร้างไฟล์เพื่อเก็บรูปแบบของบัญชีรายชื่อที่ต้องการพิมพ์ ด้วยคำสั่ง
create label ไฟล์นี้จะเป็นไฟล์ประเภท .lbl
3. กำหนดลักษณะที่ต้องการของบัญชีรายชื่อที่จะพิมพ์ แล้วเก็บรูปแบบที่กำหนดไว้
4. สั่งให้ตีเบสที่พลัสจัดพิมพ์บัญชีรายชื่อ โดยใช้คำสั่ง label form

5.16.1 รูปแบบคำสั่ง create label และ label form

ไฟล์ประเภท .lbl ที่ใช้เพื่อเก็บรูปแบบของการพิมพ์รายชื่อสามารถสร้างขึ้น

ด้วยคำสั่ง create label ซึ่งมีรูปแบบคำสั่งคือ

```
create label <file name>
```

เมื่อ file name คือชื่อไฟล์ที่ตั้งขึ้นเพื่อเก็บรูปแบบการพิมพ์บัญชีรายชื่อในขั้นตอนของการพิมพ์บัญชีรายชื่อ จะต้องเรียกใช้คำสั่ง LABEL FORM ซึ่งมีรูปแบบคำสั่งคือ

```
LABEL FORM <file name> [scope][FOR/WHILE<condition>]
[TO PRINT][SAMPLE] [TO PRINT<file <file name>]
```

เมื่อ file name คือชื่อไฟล์

scope คือข้อความแสดงขอบเขตการพิมพ์บัญชีรายชื่อ

condition คือเงื่อนไขที่กำหนดให้พิมพ์บัญชีรายชื่อ

TO PRINT คือคำสั่งให้พิมพ์ออกทางเครื่องพิมพ์

SAMPLE คือคำสั่งให้พิมพ์เป็นตัวอักษรเฉพาะบางส่วน

TO FILE คือคำสั่งให้พิมพ์บัญชีรายชื่อลงในไฟล์ประเภท .TXT

5.17 สร้างไฟล์กำหนดเงื่อนไขแสดงข้อความ

5.17.1 ไฟล์ชนิด .QRY

ดีเบสที่พร้อมใส่ได้เพิ่มความสะดวกให้แก่ผู้ใช้งาน โดยเพิ่มลักษณะไฟล์ชนิด .QRY

สร้างได้ด้วยคำสั่ง

```
CREATE QUERY <ชื่อไฟล์>
```

เมื่อต้องการเรียกใช้ไฟล์ที่สร้างไว้ก็ใช้คำสั่ง SET FILTER TO ซึ่งมีรูปแบบคือ

```
SET FILTER TO FILE <ชื่อไฟล์ชนิด .QRY>
```

ขั้นตอนการสร้างและใช้ไฟล์ชนิด .QRY ทำได้ดังนี้คือ

1. เตรียมไฟล์ข้อมูลชนิด .DBF ที่ต้องการจะกำหนดเงื่อนไขการแสดงผลด้วยคำสั่ง USE
2. สร้างไฟล์ชนิด .QRY ด้วยคำสั่ง CREATE QUERY
3. เลือกฟิลด์ข้อมูลที่จะกำหนดเงื่อนไข
4. เลือกโอเปอเรเตอร์ที่จะใช้ในการกำหนดเงื่อนไข ได้แก่ Match (เท่ากับ), More Than (มากกว่า), Less Than (น้อยกว่า) และอื่นๆ
5. กำหนดเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. กำหนดโอเปอเรเตอร์ชนิดตรรกที่จะใช้เชื่อมเงื่อนไขซึ่งได้แก่ .AND.
หรือ .OR.
7. ถ้าจำเป็นต้องกำหนดเงื่อนไขการเชื่อม ก็ให้กำหนดเงื่อนไขการเชื่อม
โดยด้วยการ เลือก Nest จากเมนู
8. เซฟไฟล์ด้วยการเลือก Exit Save จากเมนู
9. เรียกใช้ไฟล์ .QRY ด้วยคำสั่ง SET FILTER TO

5.18 การปรับปรุงข้อมูลและการผสมไฟล์

ในดีเบสทรีพลัสมีคำสั่งที่ใช้เพื่อกระบวนการทั้งสองนี้คือ UPDATE และ JOIN

5.18.1 คำสั่ง UPDATE สำหรับปรับปรุงไฟล์ข้อมูล

ในการปรับปรุงไฟล์ข้อมูลนั้น ดีเบสทรีพลัสมีคำสั่งในการนี้คือ UPDATE ซึ่งมีรูป

แบบคือ

```
UPDATE ON <key field> FROM<alias> REPLACE
<field> WITH <exp>[,<field> WITH <exp>...]
```

เมื่อ key field คือฟิลด์ที่ใช้เป็นหลักในการปรับปรุงไฟล์

alias คือชื่อทรานแซคชันไฟล์ที่จะนำมาปรับปรุง

field คือชื่อฟิลด์ที่ต้องการปรับปรุง

exp คือนิพจน์ที่แสดงถึงการปรับปรุงข้อมูล

ในการปรับปรุงไฟล์ข้อมูลด้วยคำสั่ง UPDATE นั้นดีเบสจะเตรียมไฟล์ 2 ไฟล์คือ
ไฟล์ข้อมูลหรือที่เรียกว่า มาสเตอร์ไฟล์ (Master File) และทรานแซคชันไฟล์ แล้วใช้ข้อมูล
จากทรานแซคชันไฟล์ปรับปรุงข้อมูลในมาสเตอร์ไฟล์จนได้มาสเตอร์ไฟล์ใหม่

5.18.2 ตัวอย่างการปรับปรุงไฟล์ด้วย UPDATE

จากมาสเตอร์ไฟล์ BOOKSTOR และทรานแซคชันไฟล์ TRANBOOK การปรับปรุงข้อมูล
ในมาสเตอร์ไฟล์ สามารถทำได้ดังขั้นตอนต่อไปนี้คือ

1. ถ้าไฟล์ BOOKSTOR ยังไม่เรียงข้อมูลตามฟิลด์ CODE ให้เรียงข้อมูลตามฟิลด์

CODE เสียก่อนด้วยคำสั่ง SORT หรือ INDEX ทำนองเดียวกัน ทราบแน่ชัดขั้นไฟล์ให้เรียงข้อมูลตามฟิลด์ CODE เสียก่อน โดยเรียงจากน้อยไปหามาก การเรียงข้อมูลของทราบแน่ชัดขั้นไฟล์อาจใช้คำสั่ง SORT หรือ INDEX ในที่นี้ กำหนดให้ไฟล์ BOOKSTOR และ TRANBOOK ถูกอินเด็กซ์ไว้ด้วยไฟล์ CODE และ CODE - IN ด้วยคำสั่งดังแสดงคือ

```
.USE BOOKSTOR
.INDEX ON CODE TO CODE
.USE TRANBOOK
.INDEX ON CODE TO CODE-IN
```

2. เตรียมมาสเตอร์ไฟล์ไว้บริเวณ A ด้วยคำสั่ง

```
. SELECT 1
. USE TRANBOOK INDEX CODE
```

3. เตรียมทราบแน่ชัดขั้นไฟล์ไว้ในบริเวณ B ด้วยคำสั่ง

```
. SELECT 2
. USE TRANBOOK INDEX CODE-IN
```

4. ทำการปรับปรุงข้อมูล UPDATE ด้วยคำสั่ง

```
. SELECT 1
. UPDATE ON CODE FROM TRANBOOK REPLACE ONHAND WITH
ONHAND+B->JN OUT
```

5. คำนวณมูลค่าความของสกรีนใหม่โดยคำสั่ง

```
. REPLACE ALL AMOUNT WITH UNIT_P * ONHAND
```

6. เรียงดูข้อมูลหลังการปรับปรุงด้วยคำสั่ง LIST

5.18.3 การผสมไฟล์ด้วยคำสั่ง JOIN

ดีเบสที่รีพลัสมีคำสั่ง JOIN เพื่อใช้ในการสร้างไฟล์ใหม่โดยนำเอาไฟล์ที่มีอยู่แล้วผสมเข้าด้วยกันตามเงื่อนไขที่กำหนด รูปแบบของคำสั่ง JOIN คือ

```
JOIN WITH <alias> TO <file name> FOR <condition>
[FIELDS <field list>]
```

เมื่อ alias คือไฟล์ที่จะนำมาผสม

file name คือชื่อไฟล์ใหม่ที่เกิดจากการผสมไฟล์
 condition คือเงื่อนไขการนำข้อมูลมาผสมกัน
 field list คือชื่อฟิลด์ที่จะให้มีอยู่ในไฟล์ใหม่

วิธีใช้คำสั่ง JOIN

จากความต้องการข้างต้นเราสามารถสร้างไฟล์ STU-AD ดังวิธีต่อไปนี้ คือ

1. เตรียมไฟล์ ADVISOR.DBF ไว้ในบริเวณ A ด้วยคำสั่ง

```
.SELECT 1
```

```
.USE ADVISOR
```

2. เตรียมไฟล์ไว้ในบริเวณ B ด้วยคำสั่ง

```
.SELECT 2
```

```
.USE STUDENT
```

3. ผสมไฟล์ด้วยคำสั่ง

```
JOIN DWITH ADISOR TO STU-AD FOR AD_CODE = A->CODE
```

```
FIELD STU_NAME, AD-NAME
```

4. เรียกไฟล์ STU-AD ออกมาดูข้อมูลและโครงสร้าง

5.19 ตัวแปรในดีเบสทรีฟลีส

5.19.1 หน่วยความจำของคอมพิวเตอร์

หน่วยความจำหลักของเครื่องคอมพิวเตอร์มีชื่อเรียกว่าหน่วยความจำชนิดแรมคอมเอกเซส(Random Access Memory) หรือที่มักเรียกกันย่อ ๆ ว่า แรม (RAM) หน่วยความจำหลักนี้จะทำหน้าที่เก็บข้อมูล คำสั่ง และโปรแกรมต่าง ๆ ในระหว่างที่เครื่องทำงาน แต่สิ่งที่เก็บไว้ในแรมนี้ จะหายไปเมื่อดับเครื่อง ดีเบสอนุญาตให้ผู้ใช้งานใช้เนื้อที่ในหน่วยความจำเพื่อเก็บข้อมูลที่ต้อง

การได้เช่นเดียวกับการเก็บข้อมูลในไฟล์ ข้อมูลที่เราเก็บไว้ในไฟล์จะต้องมีการตั้งชื่อฟิลด์ การเก็บข้อมูล

ในหน่วยความจำของคอมพิวเตอร์ ก็ต้องมีการตั้งชื่อเช่นกัน ชื่อที่ตั้งนี้เรียกว่าตัวแปร ตัวแปรจะทำหน้าที่

เก็บข้อมูลชนิดตัวเลขหรือตัวอักษรที่ต้องการไว้ได้ จำนวนสูงสุดของตัวแปรสามารถตั้งได้ในดีเบสคือ 256 ตัวแปร

5.19.2 ? คำสั่งสำหรับการแสดงผล

ด้วยคำสั่งแสดงผล ? เราสามารถให้ดีเบสทำการคำนวณและแสดงผลเช่นเดียวกับเครื่องคิดเลขที่ใช้กันทั่วไป ลองโหลดดีเบสเข้าสู่เครื่องคอมพิวเตอร์แล้วคีย์คำสั่ง ? 1+1 แล้วคีย์รีเทิร์นดีเบสจะให้ผลการคำนวณ คือ 1 บวก 1 ได้ผลลัพธ์เป็น 2

ลองคีย์คำสั่ง ? เพื่อการคำนวณอีกคำสั่ง คือ ? 326.5(0.1*105) แล้วคีย์รีเทิร์นผลลัพธ์ที่ได้คือ 337 ทั้งนี้เพราะคำสั่งข้างต้นคือการหาผลรวมของ 326.5 กับผลคูณของ 0.1*105

จากตัวอย่างข้างต้นจะเห็นได้ว่า ด้วยคำสั่ง ? เราสามารถให้ดีเบสทำการคำนวณผลทางคณิตศาสตร์ต่าง ๆ ที่ต้องการได้

5.19.3 การแสดงข้อมูลชนิดตัวอักษร

คำสั่ง ? ยังสามารถใช้แสดงข้อมูลชนิดตัวอักษรได้อีกด้วย พิจารณาคำสั่งและผลลัพธ์ที่ได้ดังต่อไปนี้

การเก็บข้อมูลในตัวแปรด้วยคำสั่ง store และการกำหนดค่า

เมื่อต้องการเก็บข้อมูลในหน่วยความจำของคอมพิวเตอร์ ก็สามารถทำได้ด้วยการคีย์คำสั่ง store ซึ่งมีรูปแบบคำสั่งดังนี้

store <exp> to <var>

เมื่อ exp คือนิพจน์ซึ่งอาจเป็นค่าคงที่ชนิดตัวเลข ตัวอักษร หรือนิพจน์ทางคณิตศาสตร์ต่าง ๆ

var คือชื่อของตัวแปร

นอกจากนี้เราสามารถกำหนดค่าให้ตัวแปรได้ด้วยการใช้ = ดังนี้

. x=5

5

. Y=10

10

5.19.3 ตัวแปรชนิดตัวอักษร

ในการกำหนดข้อมูลชนิดตัวอักษรให้แก่ตัวแปร ก็สามารถทำได้ด้วยคำสั่ง store เช่นเดียวกัน แต่ข้อความที่เก็บไว้ในตัวแปรจะต้องอยู่ภายในเครื่องหมายคำพูด

5.19.4 การขอลู่อื่อตัวแปรด้วยคำสั่ง display memory

ตัวแปรที่ได้รับการกำหนดและถูกเก็บไว้ในหน่วยความจำทั้งหมดสามารถเรียกออกมาดูได้ด้วยคำสั่ง display memory

คำสั่ง display memory จะนำเอาชื่อตัวแปรที่มีอยู่ในหน่วยความจำแสดงบนจอภาพ พร้อมทั้งบอกชนิดของตัวแปรว่าเป็นชนิดตัวเลข (N) หรือตัวอักษร (C) และข้อมูลที่มีอยู่ในตัวแปรขณะนั้นด้วย

5.20 การใช้เอดิเตอร์ของดีเบส เขียนคำสั่งและโปรแกรม

ดีเบสเป็นโปรแกรมที่ยอมให้ผู้ใช้เขียนคำสั่งเรียงกันเป็นโปรแกรม โดยมีส่วนของเอดิเตอร์หรือส่วนที่จะช่วยในการแก้ไขโปรแกรมคล้าย ๆ กันจะเป็นเวิร์ดโปรเซสเซอร์ แต่คำสั่งที่ใช้กันนี้น้อยกว่าเวิร์ดโปรเซสเซอร์มาก เริ่มใช้เอดิเตอร์

ในส่วนของเอดิเตอร์ของดีเบส จะเรียกเข้าสู่งานโดยใช้คำสั่ง modify command ซึ่งดีเบสมีเอดิเตอร์ไว้สำหรับการสร้างโปรแกรม หรือเรียกโปรแกรมนั้นว่าคำสั่งใหม่ก็ได้ เป็นคำสั่งที่ผู้ใช้สร้างขึ้นโดยเอาคำสั่งพื้นฐานที่ดีเบสเข้าใจแล้วมาใช้

ลองดูวิธีการใช้เอดิเตอร์สร้างคำสั่งชื่อ addab โดยคำสั่ง modify command ในขณะที่เมื่อดีเบสขึ้นเครื่องหมายเตรียมพร้อม

```
.modify command
```

```
enter filename : addab
```

ดีเบสจะขึ้นข้อความว่า new file แล้วเกิดจอภาพว่างเปล่าพร้อมเคอร์เซอร์มาอยู่ที่มุมขวาบน ซึ่งหมายถึงดีเบส ซึ่งหมายถึงดีเบสให้กระดาษหนึ่งแผ่นกับเราเพื่อให้ตีโปรแกรม ลองตีโปรแกรม

เมื่อคีย์เสีรี่จะออกจากเอดิเตอร์นี้ เรามีทางออกได้ 2 ทางคือ

1. กด Ctrl-W หรือ Ctrk-End บลาให้ดีเบสเก็บสิ่งที่เราทำบนจอนี้ลงคิสค์ในชื่อ

ไฟล์ที่เราบอกไว้ตอนแรก

2. กด Ctrl-Q หรือ Esc บอกให้ดีเบสซกเล็ก ดีเบสจะถามว่ายกเลิกแน่หรือไม่ ถ้าตอบ Y ดีเบสจะซกเล็ก

ทั้งสองกรณีดังกล่าวดีเบสจะออกมาที่ . เสมอเมื่อจะเรียกคำสั่งที่สร้างขึ้น

การสร้างคำสั่งใหม่ดีเบสจะเก็บเสมือนหนึ่งเป็นไฟล์ ๆ หากให้ดีเบสแสดงผลชื่อไฟล์ออกมา เราจะเห็นชื่อไฟล์คำสั่งนั้นลงท้ายด้วย .PRG ซึ่งแสดงได้โดยใช้คำสั่ง

เมื่อจะเรียกใช้ในดีเบส จะใช้ DO แล้วตามด้วยชื่อโปรแกรม ซึ่งถ้าโปรแกรมอยู่ต่างโครง เราจะใช้ชื่อโครงหน้าหน้า เช่น DO B:ADDAB ลองดูการเรียกคำสั่งที่สร้างมาทำงาน

. DO B: ADDAB

5.20.1 คีย์ควบคุมของเอดิเตอร์

ดีเบสใช้คีย์ควบคุมในเอดิเตอร์ คล้ายกับเวิร์ดโปรเซสเซอร์อีกหลายตัว ซึ่งสามารถแก้ไขข้อความที่คีย์ได้แบบพลุกสุก โดยการเลื่อนเคอร์เซอร์ไปยังตำแหน่งต่างๆของจอภาพได้ อย่างสะดวก คีย์ควบคุมเมื่ออยู่ใน Modify command ประกอบด้วย

: หรือ Ctrl-E เลื่อนเคอร์เซอร์ขึ้นบนหนึ่งฟิลด์หรือหนึ่งบรรทัดที่แสดงบนจอ

: หรือ Ctrl-X เลื่อนเคอร์เซอร์ลงล่างหนึ่งฟิลด์หรือหนึ่งบรรทัดที่แสดงบนจอ

< หรือ Ctrl-S เลื่อนเคอร์เซอร์ไปทางซ้ายหนึ่งตัวอักษร

-> หรือ Ctrl-D เลื่อนเคอร์เซอร์ไปทางขวาหนึ่งตัวอักษร

Ctrl-Y ลบตัวอักษรออกโดยเริ่มจากจุดเคอร์เซอร์ไปทางขวา

Ins หรือ Ctrl-V สวิตช์ระหว่างโหมดการพิมพ์ทับกับโหมดตัวพิมพ์

Del หรือ Ctrl-G ลบตัวอักษรที่อยู่ตรงตำแหน่งของเคอร์เซอร์

Rubout ลบตัวอักษรที่อยู่ทางซ้ายของเคอร์เซอร์

Esc หรือ Ctrl-Q ยกเลิกการแก้ไขและกลับสู่สภาวะการเตรียมพร้อมของดีเบส โดยไม่เก็บที่แก้ไขลงดิสค์

Ctrl-N แทรกบรรทัดหนึ่งบรรทัด

Ctrl-T ลบตัวอักษรทางขวาออกหนึ่งคำ

Pgdn หรือ Ctrl-C สครอลลงข้างล่างหนึ่งช่องหน้าต่าง

Pgup หรือ Ctrl-R สครอลขึ้นข้างบนหนึ่งช่องหน้าต่าง

Ctrl-End หรือ Ctrl-W เก็บข้อมูลแก้ไขหรือเขียนลงในไฟล์

5.20.2 การแก้ไขโปรแกรม

สมมติว่าต้องการเพิ่มคำสั่ง set talk off และ set talk on และแก้ไขตัวเลข 7 เป็น 25 และ 5 เป็น 15 เราจะเรียกโปรแกรมเดิมออกมาแก้ไข

.Modify Command Addab

ขั้นตอนการทำงานมีดังต่อไปนี้

1. เคอร์เซอร์อยู่ที่ตรงมุมซ้ายบนสุด
2. กด ; มาถึงบรรทัด store 7 to a
3. กด Ctrl-N จะปรากฏบรรทัดว่าง 1 บรรทัด
4. พิมพ์ข้อความ set talk off แล้วกดรีเทิร์น
5. เคอร์เซอร์มาอยู่ที่บรรทัดถัดมา กด จนเคอร์เซอร์มาหยุดที่ตัวเลข 7 จากนั้นกดคีย์ Del เลข 7 จะหายไป กดคีย์ Ins จะปรากฏคำว่า Insert on บนจอภาพด้านบนแสดงว่าอยู่ในโหมดแทรก พิมพ์ตัวเลข 25 กดคีย์ Ins จะปรากฏว่าข้อความ Insert หายไป
6. กดรีเทิร์นมาที่บรรทัด B = 5 แล้วเลื่อนเคอร์เซอร์มาที่เลข 5 ด้วย จากนั้นกด Ins เพื่อเข้าโหมดแทรกพิมพ์เลข 1 กดรีเทิร์น ข้อความ Insert บนจอภาพหายไป
7. กดรีเทิร์นหรือ มาที่บรรทัด Return แล้วกด Ctrl - n จะมีบรรทัดแทรก แล้วพิมพ์ Set Talk On เมื่อแก้ไขเสร็จแล้วกด Ctrl - W จะปรากฏ ซึ่งดีเบสจะกลับมาพร้อมรับคำสั่งอีก

5.20.3 การสร้างไฟล์สำรองของดีเบส

ทุกครั้งที่มีการแก้ไข อาจมีปัญหาอันเนื่องมาจากระบบขัดข้องทางฮาร์ดแวร์ ซึ่งอาจจะทำให้ไฟล์โปรแกรมที่กำลังแก้ไขเสียหายได้ หรือในบางครั้งผู้แก้ไขงานโปรแกรมผิดไปจากรูปเดิมมาก และต้องการคงของเดิมไว้ แล้วยกเลิกสิ่งที่แก้ไขเพื่อคงของเดิมไว้ ดังนั้นดีเบสจึงสร้างไฟล์สำรองไว้ ลองดูไฟล์สำรองที่เกิดขึ้น

ลำดับการทำงานของดีเบส กล่าวคือ เมื่อสร้างไฟล์ครั้งแรกจะยังไม่มีไฟล์สำรอง

ดังนั้นเมื่อเรียกไฟล์มาแก้ไขจะปรากฏ x.Bak ซึ่งเหมือนเป็นไฟล์ก่อนการแก้ไขอยู่ เราสามารถนำกลับมาใช้ได้โดยเปลี่ยนชื่อให้เป็น x.prg ไปได้

5.20.4 การพิมพ์โปรแกรมดีเบส

ในดีเบสมีคำสั่งให้พิมพ์โปรแกรมออกมาโดยตรงโดยใช้คำสั่ง Type Addab ตามด้วย Ctrl-P ซึ่งเป็นการเปิดเครื่องพิมพ์

หากต้องการเลิกติดต่อกับเครื่องพิมพ์ให้ใช้ Ctrl - P อีกครั้ง หรือในคำสั่ง Type จะพิมพ์ออกทางเครื่องพิมพ์โดยเฉพาะ โดยใช้

.Type Addab.Prg To Print

คำว่า To Print เป็นการบอกว่าเป็นเอาผลที่พิมพ์ได้ออกทางเครื่องพิมพ์

5.21 การเซตสวิตช์ต่างๆของดีเบส

การใช้งานดีเบสก็ มีแผงสวิตช์ เป็นซอฟต์แวร์ที่จะให้ผู้ใช้เซตตามความต้องการ ลักษณะของแผงควบคุมของดีเบสจะเป็นสวิตช์ที่ NO และ OFF ในขณะที่เริ่มใช้ดีเบสหากไม่มีการเซตสวิตช์ เหล่านี้ ดีเบสจะเซตเองตามความเหมาะสม โดยดูได้จากคำสั่ง LIST STATUS

จะใช้คำสั่ง SET ซึ่งมีรูปแบบของคำสั่งเป็น

SET <ตัวหารามิเตอร์สวิตช์> ON/OFF

5.21.1 การใช้คำสั่ง SET

การใช้คำสั่ง SET มีรายละเอียดดังนี้

SET ALTERNATE TO และ SET ALTERNATE ON (OFF)

ดีเบสให้ข้อดีในเรื่องของการเซตอุปกรณ์เอาต์พุต แทนจอภาพหรือเครื่องพิมพ์ได้ไม่ยาก ทั้งนี้เพราะปัญหาที่สำคัญคือ ถ้าใช้งานแล้วให้ผลลัพธ์ปรากฏบนจอภาพ เราก็ไม่สามารถเรียกย้อนมาได้ หรือในขณะที่ทำงานในส่วนโปรแกรมที่ต้องการใช้เวลาานหลายชั่วโมง ซึ่งเราอาจจะต้องไปสระที่อื่นและให้คอมพิวเตอร์ทำงาน ผลลัพธ์บนจอเราไม่สามารถดูได้เพราะมันจะผ่านไป แต่ถ้าเราใช้ SET PRINT ON เข้าช่วย จะให้ผลลัพธ์ไปพิมพ์บนเครื่องพิมพ์ซึ่งก็เป็นข้อเสียที่ดีเบสทำงานช้าลงและหากเราไม่อยู่เครื่องพิมพ์อาจติดขัดเช่น กระดาษติด หรือกระดาษหมดต้องคอยเปลี่ยนกระดาษ ดีเบสจึงให้เราเลือกใช้คิสเกตต์ที่มีที่ว่างเหลือเป็นกระดาษ โดยดีเบสจะพิมพ์ผลลัพธ์ที่ปรากฏบนจอภาพลงคิสค์แล้วเราเรียกพิมพ์หรือเรียกดูที่หลังได้ ซึ่งอาจจะใช้แสดงผลลัพธ์ ซึ่งถ้าออกทางเครื่องพิมพ์จะเสียเวลา แต่ถ้าเก็บในคิสค์ก่อนแล้วค่อยสั่งพิมพ์ใน

ลักษณะไฟล์สำหรับพิมพ์ได้เร็ว และ พิมพ์ที่ชัดก็ได้

SET CARRY

ในการป้อนข้อมูลด้วยคำสั่ง INSERT และ APPEND นั้นดีเบสจะให้เรคอร์ดใหม่เพิ่มเติมเข้ามา โดยเรคอร์ดเดิมเข้ามานี้จะเป็นเรคอร์ดที่ว่างเปล่า เพื่อให้ป้อนข้อมูล แต่ในบางครั้งเราอาจต้องการข้อความที่เหมือนกับเรคอร์ดก่อนหน้า หรือการป้อนข้อมูลซ้ำๆกัน เพื่อลดการคีย์ข้อมูลบางฟิลด์ เราสามารถเซต CARRY ได้ โดยปกติดีเบสจะเซต CARRY OFF

SET COLOR

ในการใช้งานดีเบสกับจอภาพสี เราสามารถกำหนดสีสีนบนจอภาพได้ตามต้องการ ลักษณะของคำสั่ง SET จะอยู่ในรูป

SET COLOR TO <สีปกติ>, <สีขณะป้อนข้อมูล>, <สีกรอบ>, <สีพื้น>

SET CONSOLE

การติดต่อโต้ตอบของดีเบสจะแสดงผลกลับมาบนจอภาพ ซึ่งโดยปกติดีเบสจะเซต CONSOLE ON ไว้ กล่าวคือผลลัพธ์ต่างๆจะมาปรากฏบนจอภาพเสมอ แต่ถ้าเซต CONSOLE OFF จะทำให้ไม่มีอะไรมาปรากฏบนจอภาพเลย

SET DELETE

การกำหนดสภาวะการลบเรคอร์ดข้อมูลทีละตัวด้วยคำสั่ง DELETE แต่ยังไม่ได้ PACK ให้เสมือนการลบออกจริงๆคือ ค้นหาไม่พบหรือแน่แท้จะใช้คำสั่ง LIST DISPLAY ก็ไม่พบเช่นกัน

SET EJECT

ในภาวะปกติจะอยู่ในสภาวะ ON หมายถึง ให้เลื่อนหน้ากระดาษพิมพ์ 1 หน้าโดยอัตโนมัติ เมื่อใช้กับคำสั่ง REPORT และหากใช้ SET EJECT OFF จะทำให้ไม่มีการเลื่อนหน้ากระดาษ

SET ESCAPE

เราใช้คีย์ ESC เพื่อขัดจังหวะการทำงานของดีเบส ดังนั้นหากต้องการให้ยกเลิกการใช้คีย์ ESC เป็นคีย์ขัดจังหวะหยุดการทำงานก็ใช้ SET ESCAPE OFF

SET INDEX TO

คำสั่งนี้เป็นคำสั่งที่สำคัญมาก เช่น ไฟล์ NAMECARD เรามีการสร้าง INDEX ไว้ 2 ไฟล์ ไฟล์แรกใช้คีย์ ID ไฟล์ที่สอง ใช้คีย์ NAME สมมติว่าไฟล์แรกชื่อ NAMEID.NDX ไฟล์ที่ 2 ชื่อ NAMEAME.NDX เมื่อเวลาเรียกไฟล์ จะใช้

USE NAMECADR INDEX NAMEID, NAMENAME

ในการใช้คำสั่ง FIND หรือ SEEK จะมีการค้นหาในอินเด็กซ์ไฟล์ แต่จะตาม ID เพราะเราอินเด็กซ์ตาม ID ก่อนแต่ถ้ามีการ UPDATE จะ UPDATE ทั้งสองไฟล์ แต่ถ้าต้องการให้มีการใช้ FIND หรือ SEEK เพื่อการค้นหาคีย์ NAME เราจะต้องใช้คำสั่ง

SET INDEX TO NAMENAME, NAMEID

การค้นหาหลังจากคำสั่ง SET INDEX จะเป็นไปตามคีย์แรกที่กำหนด

SET PROCEDURE TO (ชื่อไฟล์)

ดีเบสทีพลัสมีการใช้โปรแกรมย่อยเป็นโมดูลโดยรวมโมดูลที่เรียกว่า Procedure หรือกระบวนการความหลากหลายโปรแกรมย่อยให้อยู่ในไฟล์เดียวกัน การใช้คำสั่ง SET PROCEDURE TO <ชื่อไฟล์> จึงเป็นการกำหนดไฟล์ที่จะเรียกโมดูลในไฟล์นั้นมาใช้ ปกติเราสามารถสร้าง Procedure ได้ 32 Procedure ในไฟล์การเรียกก็จะใช้ DO <ชื่อ Procedure>

บทที่ 6

การทำงานของโปรแกรม

อุปกรณ์ในการใช้งาน

- เครื่อง COMPUTER IBM PC XT/AT COMPATIBLE 2 FLOPPY DISK หรือ HARD DISK
- เครื่อง PRINTER ที่สามารถพิมพ์กระดาษความกว้าง 15"
- แผ่น DISKETTE สำหรับเก็บข้อมูล (DATA DISK)

การทำงานของโปรแกรม

ส่วนของตัวโปรแกรมได้มีการแบ่งออกเป็น 2 VERSIONS คือ

1. ใช้กับเครื่องที่ไม่มี HARD DISK

การใช้เครื่องที่ไม่มี HARD DISK ตัวเครื่องจะต้องมี 2 FLOPPY DISK ใน DRIVE A ควรจะเป็น DRIVE HIGH DENSITY (1.2 Mb) เพื่อเก็บ DOS, MAIN PROGRAM และ TABLE ไว้บนแผ่นเดียวกัน ถ้าเป็น DRIVE 360 Kb SPACE ใน DISK จะไม่พอเก็บสำหรับเก็บ Dos จะเก็บได้เฉพาะ Program เท่านั้น และ DRIVE B จะ HIGH หรือ LOW DENSITY ก็ได้เพื่อเป็น DRIVE ที่ใช้กับ DATA DISK สำหรับเครื่องที่มี HARD DISK นั้นจะไม่ต้องคำนึงถึง DATA DISK และในโปรแกรมจะไม่มีกรเรียกใช้เกี่ยวกับ DATA DISK ภายนอกเพราะได้อักรวมใน HARD DISK ตัวเดียวกันแล้ว

สำหรับเครื่องที่ไม่มี HARD DISK ตัวโปรแกรมได้แบ่งการเก็บข้อมูลออกเป็น 2 ส่วน ประกอบด้วย ส่วนแรก ที่เป็น TABLE เพื่อบอกความหมายของ CODE ต่าง ๆ ที่กำหนดขึ้น และรวมถึงรายละเอียดกับการอ้างอิงถึงตัวข้อมูลจริงที่เก็บไว้ ในส่วนที่สอง TABLE ส่วนนี้จะอยู่ใน DISK ที่เก็บ MAIN PROGRAM ที่ใช้ทำการ EXECUTE ส่วนในตัวข้อมูลจริงจะอยู่ใน DISK ที่ใช้เก็บข้อมูลอย่างเดี่ยวหรือเป็น DATA DISK ซึ่งจำนวนของ DATA DISK นั้นขึ้นอยู่กับกาหนด CODE ของสินค้าแต่ละชนิด และกำหนดในการเก็บข้อมูลของสินค้านั้นลงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA DISK เบอร์อะไร

ในการหาจำนวนของ CODE ที่เก็บไว้ใน DATA DISK แต่ละแผ่นนั้นตัวผู้ใช้จะต้องรู้ถึงจำนวนการเรียกใช้ CODE นั้นเพื่อการรับเข้าหรือการส่งออกใน 1 เดือนหรือก่อนที่จะปิด STOCK ซึ่งอาจจะนับได้จากจำนวนของ ROW ของ CODE ใน REPORT ที่ CLOSING ไปแล้วได้ ดังนั้นเมื่อรู้จำนวนความถี่ในการรับสินค้าเข้า และส่งสินค้าออกก็สามารถหาจำนวนของ CODE ที่จะเก็บไว้ใน DATA DISK โดยคำนวณได้ดังนี้

$$\text{จำนวน CODE ใน DISK} = \frac{80\% \text{ ของความจุใน DATA DISK (BYTE)}}{114 + (115 * \text{IN}) + (86 * \text{OUT})}$$

เมื่อ 114 คือ จำนวนความจุเป็น byte ต่อ 1 record ของจำนวนครั้งที่รับเข้ามา

115 คือ จำนวนความจุเป็น byte ต่อ 1 record ของจำนวนครั้งที่ส่งของออกไป

IN คือ จำนวนครั้งที่รับของเข้ามา

OUT คือ จำนวนครั้งที่ส่งของออกไป

ตัวอย่างเช่น ถ้าจำนวนในการรับสินค้าเข้าต่อเดือนเท่ากับ 100 ครั้ง และในการส่งสินค้าออก 100 ครั้งจำนวนความจุของ DATA DISK ขนาด 360 KB ดังนั้น ใน DATA DISK นี้จะเก็บสินค้าได้ดังนี้

$$\begin{aligned} \text{CODE NUMBER} &= \frac{.360 * 1024 * 0.80}{114 + (115 * 100) + (86 * 100)} \\ &= 14 \end{aligned}$$

ใน DATA DISK DISK จะสามารถเก็บสินค้าได้ 14 ชนิด ถ้าสมมุติว่ามีสินค้าทั้งหมด 200 ชนิดก็จะใช้ DATA DISK จำนวน 15 แผ่น

สินค้าที่เก็บไว้แต่ละชนิดนั้น จะสามารถเก็บความแตกต่างของการนำเข้าได้

4 อย่าง คือ สินค้าหนึ่งชนิดอาจจะมีการนำเข้ามา และมีราคาของค่าแรงหรือค่าวัสดุ แต่ละครั้งไม่เท่ากัน ซึ่งในโปรแกรมขอมให้มีความแตกต่างของค่าใช้จ่ายทั้งหมดต่อชิ้นได้ 4 อย่าง เช่น สินค้า A นำเข้าครั้งแรก ค่าแรง 10 บาท ค่าวัสดุ 10 บาท นำเข้าครั้งที่สอง ค่าแรง 10 บาท ค่าวัสดุ 15 บาท และถ้านำสินค้าเข้าครั้งที่สาม ค่าแรง 10 บาท ค่าวัสดุ 10 บาท เช่นนี้จะถือว่ามีความแตกต่างของค่าแรง และค่าวัสดุ เมื่อมีการนำสินค้าออกโดยตรวจจากความต้องการที่จะนำออก ถ้าจำนวนของความต้องการที่จะนำออกนั้นมีมากกว่าในการเก็บของครั้งแรกก็จะไปนำเอาส่วนที่ซึ่งขาดอีกในครั้งที่สองมาเสริม โดยการคิดค่าใช้จ่ายทั้งหมด โดยแยกระหว่างครั้งแรก และครั้งที่สองรวมกันออกมาให้

การ REPORT ในโปรแกรมจะแบ่งเป็น 2 ชนิด คือ

- MATERIAL REPORT
- STOCK REPORT

MATERIAL REPORT นั้นเป็นการรายงานจำนวนสินค้าแต่ละชนิดที่ยังมีอยู่ใน STOCK ทั้งหมดโดยสามารถให้นำแสดงออกทางจอภาพ หรือเครื่องพิมพ์ได้ และ STOCK REPORT นั้นจะแบ่งเป็นการ REPORT ปกติคือ แสดงการนำเข้าหรือการนำออกในแต่ละสินค้าว่าตั้งแต่หลังจาก CLOSING STOCK แล้วมีการนำเข้าหรือออกกี่ครั้ง และวันใดบ้าง รวมทั้งรายละเอียดของค่านวมค่าแรง วัสดุ และยอดที่เหลือในแต่ละครั้ง ส่วน STOCK REPORT อีกอย่างคือ การ CLOSING STOCK เมื่อต้องการจะตัดบัญชี และ CLEAR การนำเข้าหรือออกของทุกสินค้าในการ CLOSING STOCK นั้นจะขอมให้ CLOSE ได้ต่อเมื่อเครื่องพิมพ์พร้อมที่จะทำงานเท่านั้น เพราะในการ CLOSING แต่ละครั้งจะไม่มีเก็บรายละเอียดของที่ผ่านมาแล้ว ดังนั้นเมื่อมีการ CLOSE จึงควรที่จะเก็บเป็น HARDCOPY ไว้เป็นหลักฐานภายหลัง และนำยอดของสินค้าชกไปเป็น BALANCE ของครั้งใหม่ต่อไป

2. สำหรับเครื่องที่มี HARD DISK

ในการมีเครื่องที่มี HARD DISK จะไม่ต้องมาค่านึงถึง DATA DISK ดังนั้น CODE ของสินค้าทุกชนิดจะเก็บไว้ใน HARD DISK จึงไม่มีการถามถึงเบอร์ของ DISK ในส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของโปรแกรมจึงต้องมีการดัดแปลงแก้ไขในส่วนนี้ แต่การทำงานภายนอกและการคำนวณต่าง ๆ จะเหมือนกับทั่วไปใน FLOPPY DISK ซึ่งรวมทั้ง STRUCTURE ของ FILE ก็ยังคงเดิมแต่การปฏิบัติงานจะต้อง copy โปรแกรม main_hrd.exe รวมทั้ง data file ทั้งหมดด้วยลง harddisk

แนวทางปรับปรุงเพื่อนำไปใช้ในระบบ LAN

เมื่อนำไปใช้ในระบบ LAN ต้องนำโปรแกรมในส่วนที่ใช้กับ HARD DISK มาแก้ไข และเพิ่มเติมเพราะในระบบ LAN นั้นจะใช้ HARD DISK และ FILE ในลักษณะเป็น MULTI USER สิ่งที่จะต้องคำนึงถึงในการใช้งานพร้อมกัน ข้อมูลที่มีการใช้ร่วมกันในเวลาเดียวกันจะต้องทำการป้องกันไม่ให้ข้อมูลนั้นมีการผิดพลาดเกิดขึ้น ซึ่งจะกล่าวต่อไปจะเป็นวิธีการป้องกันในขนาดที่มีการใช้ข้อมูลเดียวกันในเวลาเดียวกัน

- USE DATABASE

ตามปกติของการ USE จะเป็นลักษณะไม่มีการ SHARE ของ FILE เกิดขึ้น คือเป็น SINGLE USER หรือเป็นลักษณะ EXCLUSIVE ON เมื่อ FILE นั้นถูกใช้งานอยู่ FILE นั้นจะถูก LOCK ไว้และโปรแกรมจะเรียกใช้อีกไม่ได้จนกว่าจะไม่มีใครใช้ FILE นั้น ดังนั้นจะต้องมีการ SET EXCLUSIVE OFF ก่อนที่จะ USE DATABASE

- APPEND

ในการ APPEND เมื่อใช้ใน MULTI USER จะต้องใช้ APPEND BLANK แล้ว LOCKED RECORD นั้น และทำการรื้อรับข้อมูลลงใน ตัวแปลแทนการลงใน RECORD และหลังจากลงในตัวแปลแล้วจึง REPLACE FIELD WITH VARIABLE นั้น การทำเช่นนี้เพื่อในกรณีที่มีการ APPEND พร้อมกัน เครื่องที่กระทำก่อนก็จะ LOCK FILE นั้นทันที และจะไม่ยอมให้เครื่องอื่นใช้ FILE นั้นอีก แต่ถ้าทำเช่นนี้จะเป็นการ LOCK เฉพาะ RECORD อย่างเดียว และเครื่องอื่นก็สามารถ APPEND พร้อมกันได้ ในขนาดเดียวกัน โดย LOCKED RECORD ถัดมาแทน และเมื่อทำการ REPLACE แล้วก็ UNLOCK ALL เพื่อยกเลิกการ LOCKED RECORD หรือ FILE นั้น

- EDIT RECORD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีขั้นตอนคล้ายกับการ APPEND คือ หลังจาก LOCATE ไปที่ RECORD ที่ต้องการจะ EDIT ก็ทำการเก็บข้อมูลของ FIELD นั้นลงตัวแปล และทำการ EDIT ที่ตัวแปล แทนการ EDIT ที่ FILE เมื่อ EDIT แล้วก็เก็บเข้าที่ RECORD นั้น โดยก่อนเก็บจะต้อง LOCKED RECORD นั้นและทำการ REPLACE ตัวแปลลงไป

- DELETE RECORD

หลังจากทำการ MARK RECORD แล้วจะต้องทำการ PACK DATABASE ในการ PACK แต่ละครั้ง FILE นั้นจะต้องไม่มีเครื่องใด LOCKED FILE หรือ RECORD อยู่จึงจะทำการ PACK ได้ ในระหว่างทำการ PACK DATA เครื่องนั้นจะ LOCKED FILE จนกว่าจะทำงานเสร็จ และ UNLOCK ALL FILE นั้น เพื่อยกเลิกการ LOCKED

เมื่อ DATABASE ถูก USE ใน MODE ของ EXCLUSIVE OFF ถ้ามีการกระทำกับ DATABASE นอกจากคำสั่งข้างต้นแล้ว DATABASE นั้นหรือ FILE นั้นจะไม่ถูก LOCK เพราะคำสั่งนอกจากข้างต้นนั้นจะเป็นการอ่านมาอย่างเดียวจะไม่มีการ UPDATE, DELETE, EDIT และ ADD ในการตรวจสอบการ LOCK จะทำได้โดยใช้ FUNCTION FLOCK() เพื่อทำการตรวจสอบการ LOCK ของ FILE ซึ่งจะ RETURN ค่าเป็น .T. (TURE) และ .F. (FALSE) และ FUNCTION RLOCK() จะทำการตรวจสอบ RECORD ที่กำลังชื่ออยู่ขณะนั้นว่ามี การ LOCK อยู่หรือไม่โดยจะ RETURN .T. และ .F. เช่นกัน

เทคนิคในการตรวจการ LOCK ของ DATABASE เมื่อพบว่า FILE หรือ RECORD นั้นถูก LOCK อยู่คือ RETURN .T. เราอาจจะทำการ LOOP รออยู่ประมาณ 5-10 วินาทีเพราะถ้าเราทำตามขั้นตอนข้างต้นสำหรับการ UPDATE, EDIT, DELETE และ ADD แล้ว ตัว DATABASE นั้นจะถูก LOCK อยู่ชั่วขณะเดียวเท่านั้นเพื่อทำการ REPLACE ข้อมูลลง FILE ดังนั้นในระยะเวลาที่ LOOP รอ นั้นเพียงพอสำหรับการทำคำสั่ง REPLACE ทำงานเสร็จ ในการตรวจการ LOCK ควรทำเป็นลักษณะ PROCEDURE

การแปลง PRG. PROGRAM เป็น EXE. PROGRAM

ใน SOURCE PROGRAM นั้นเขียนด้วย DBASE III PLUS ซึ่งมีความยุ่งยากในการเรียกใช้งาน และไม่สะดวกในการใช้งานเพราะจะต้องเรียก Program Dbase III

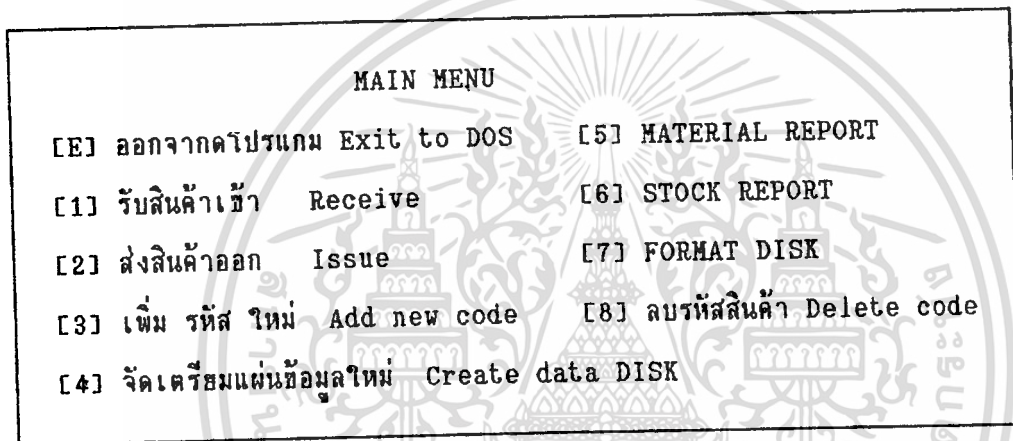
plus และ overlay ด้วย ดังนั้น จึงได้มีการนำ UTILITY PROGRAM มา COMPILE PRG เป็น EXE เพื่อเรียก EXECUTE ได้ในตัวเอง COMPILE ที่รอเป็น CLIPPER VERSION 5.0 คำสั่งที่ใช้ใน CLIPPER นั้น COMPATIBLE กับ DBASE III และนอกเหนือจากคำสั่งใน DBASE แล้วยังมีคำสั่งที่เป็น Function สำเร็จที่สามารถเรียกใช้งานได้สะดวกและประโยชน์อีกมากในการเรียกใช้งานซึ่งสามารถดูได้จาก source program



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเริ่มการทำงานของโปรแกรมอย่างไร ?

ใส่แผ่น SYSTEM DISK ที่ DRIVE A: แล้วเปิดเครื่องหรือ BOOT เครื่องใหม่จากนั้นจะ
และเรียกโปรแกรมชื่อ MAIN_NOR จะทำการ Run โปรแกรมเข้าสู่ MAIN MENU ดังรูป
(แต่ถ้าใช้กับ Hard Disk จะเรียกใช้โปรแกรม MAIN_HRD และ Menu ต่างๆจะไม่
ถามถึง Disk Number ด้วย เพราะได้เก็บ Data รวมอยู่บน Hard Disk เรียบร้อยแล้ว)



ที่ MAIN MENU นี้จะมีหัวข้อให้เลือก การทำงานแต่ละอย่างโดยการกด ตัวเลขที่อยู่หน้า
ข้อนั้น การทำงานของแต่ละข้อมีดังนี้

[E] ออกจากโปรแกรม

เป็นการสิ้นสุดของการทำงาน ซึ่งจะออกไปสู่ DOS PROMPT

[1] รับสินค้าเข้า

เป็นการเก็บรายละเอียดของสินค้าที่นำเข้ามาเก็บใน STOCK ซึ่งเราจะต้องรู้ รหัส
ของสินค้านั้น ถ้าไม่รู้รหัสสามารถดูได้โดยการ กด key F1 เพื่อขอ code ที่มีอยู่ทั้งหมดได้
และออกจากการขอความช่วยเหลือโดยการกด Esc และเมื่อผู้ใช้ทำการ คีย์รหัสสินค้าลงไป

RECEIV MENU	+-----+
	รหัสสินค้า:
รหัสสินค้า Code NO. _____	
	Press F1 for help
	กด key F1 เพื่อแสดง code

โปรแกรมจะทำการตรวจว่ารหัสสินค้านั้นมีอยู่ใน STOCK หรือไม่ ถ้าไม่พบจะแสดงข้อความออกมา (การกำหนดรหัสสินค้าทำได้โดยเลือกข้อ [5])

ไม่มี รหัสสินค้า อยู่ใน STOCK
 กลับไปยัง MAIN MENU เพื่อ สร้างรหัสสินค้าใหม่
 Find not found this code ! Return to MAIN MENU.

แต่ถ้ารหัสสินค้านั้นมีการสร้างไว้แล้ว โปรแกรมจะตรวจสอบว่าสินค้านั้นอยู่ที่ DATA DISK เบอร์อะไรและจะแสดง MENU ให้ใส่ DISK เบอร์นั้นเพื่อทำงานต่อไป

ใส่ DISK เบอร์ : ___ ลงใน DRIVE [B:]
 Insert DISK No. __ into DRIVE [B:]
 [C] ทำงานขึ้นต่อไป Continue
 [E] ออกไป MAIN MENU
 ข้อที่เลือก: Select ==> __

เมื่อใส่แผ่น DATA DISK เบบร์นั้นแล้วให้กด [C] เพื่อทำงานขึ้นไป ถ้าแผ่นนั้นผิด เบบร์จะ แสดงข้อความออกมา และให้ใส่เบบบร์ให้ถูกต้อง หรือต้องการยกเลิกข้อ [E] เพื่อ กลับไป MAIN MENU

หลังจากที่ใส่ DATA DISK ถูกต้องแล้วจะเข้าสู่ MENU การบ้องรายละเอียดสินค้าที่นำ เข้ามา

RECEIV MENU

DISK # : ____

รหัสสินค้า: _____

รหัสสินค้า Code No.	_____	จำนวน QTY	_____
DATE	<u>mm/dd/yy</u>	ค่าแรงต่อชิ้น JOB_PAY	_____ X ← A เหมาท้ทั้งหมด
JOB_NBR	_____	ค่าวัสดุต่อชิ้น Price_AVE	_____
D/O No.	_____	จำนวนเงินทั้งหมด TOTAL	_____
Receive form	_____		
Comment	_____		

การคิดค่าแรงต่อชิ้น จะมีอยู่สองอย่างคือ การเหมาท้ทั้งหมดหรือการคิดต่อชิ้นจริงๆ ซึ่งหลัง จากใส่จำนวนค่าแรงแล้วจะมีการถามว่าค่าแรงนั้นเป็นแบบเหมามาหรือคิดต่อชิ้น ถ้าเลือก A จะ เป็นการเหมาท้ทั้งหมด และการคิดจำนวนเงินทั้งหมดจะคิดจาก ค่าใช้จากทั้งหมดต่อชิ้น คุณ จำ นวนของสินค้า

หลังจากป้อนข้อมูลถึง หมายเหตุ จะมี MANU ถาม ต้องการกลับไปทำอีกหรือเปล่า ? ซึ่ง จะเป็นการกลับไปเริ่มแรกการรับสินค้าเข้าอีกครั้้ง

ในการรับสินค้าเข้ามา แต่ละครั้้งอาจจะมีราคาของค่าแรงและค่าวัสดุต่อชิ้น ไม่เท่ากัน หมายความว่าในสินค้าที่สั่งเข้ามาใหม่ จำนวนค่าใช้จ่ายทั้งหมดในแต่ละ LOT ไม่เหมือนกัน เวลานำมาคิดค่าใช้จ่าย จะต้องแยกกันแต่ละ LOT ของการสั่งเข้ามา ดังนั้นโปรแกรมจึง

อนุญาตให้มีความแตกต่างกันของค่าแรงหรือค่าวัสดุต่อชิ้นต่างกันได้ 4 อย่าง และถ้ามีความแตกต่างเกินจากนี้จะแสดงข้อความขึ้นมา

[2] ส่งสินค้าออก

ข้อนี้เป็นการป้อนรายละเอียดข้อมูลของสินค้าออก เมื่อเลือกข้อนี้จะแสดง MENU ให้ใส่รหัสสินค้า ซึ่งจะมีลักษณะคล้ายกับ การรับสินค้าเข้า และจะนำ CODE ที่มีการส่งสินค้าออกไปหาในข้อมูล ถ้าไม่พบก็จะ แสดงข้อความออกมาปรากฏให้เห็น และกลับไปยัง MAIN MENU หลังจากค้นหา CODE นั้นพบในข้อมูลก็จะให้ใส่ DISK เบอร์ตามที่ระบุ เมื่อใส่ DATA DISK หลังจากใส่ DATA DISK ถูกต้องแล้วจะแสดง MENU ดังรูป

ISSUE MENU		DISK # : _____
		รหัสสินค้า: _____
รหัสสินค้า Code No. _____	จำนวนสินค้าที่มีอยู่ใน STOCK : _____	
DATE <u>mm/dd/yy</u>	จำนวน	QTY _____
JOB_NBR _____		
INVOICE No. _____	จำนวนเงินทั้งหมด	Total _____
ร้านส่งออก Issue to _____		
Comment _____		

ถ้าจำนวนที่สินค้าที่ส่งออกมีมากกว่าใน STOCK ก็แสดงผลออกมา และกลับไปให้ใส่จำนวนของสินค้าที่สั่งใหม่หลังจากถูกต้องแล้วก็กลับไป MAIN MENU

[3] เพิ่ม รหัส ใหม่

เมื่อมีสินค้าชนิดใหม่ จะต้องป้อนข้อมูลรายละเอียดของสินค้านั้นลงใน DATA BASE บอกรายละเอียด กับสินค้า หมายถึง และกำหนดรหัสสินค้าว่าจะเก็บที่ DISK เบอร์อะไร ตามที่เราทำการเตรียมแผ่นข้อมูล

เพิ่ม รหัส ใหม่ ADD NEW CODE

เบอร์ของ DISK No. ---
 รหัสสินค้า Code No. -----
 ชื่อสินค้า Commodity -----
 หมายเหตุ Comment -----

หลังจากป้อนข้อมูลจนถึงหมายเหตุแล้วปรากฏข้อความถามเพื่อความแน่ใจอีกครั้งที่จะเก็บรหัสสินค้านี้ลงในแผ่นข้อมูล

[4] จัดเตรียมแผ่นข้อมูลใหม่

ข้อนี้จะเป็นการสร้างแผ่นเก็บข้อมูล หรือ DATA DISK ใหม่ แผ่นที่จะนำมาสร้างใหม่นี้จะต้องผ่านการ FORMAT มาก่อน(จากข้อ [7]) ในการกำหนดเบอร์จะใส่หมายเลขได้สามหลักซึ่งควรจะกำหนดให้ใส่ตัวเลขครบทั้งสามหลัก เพื่อจะได้เป็นการตรวจสอบได้ง่ายและในการกำหนดเลขจะกำหนดซ้ำกันไม่ได้ หลังการนั้นก็ให้ใส่แผ่นข้อมูลที่ต้องการจะสร้างลงใน DRIVE B และกด KEY ENTER

--- สร้าง DATA DISK ----

ใส่ DISK ลงใน DRIVE B:

ต้องการสร้าง DISK เบอร์ ---

Insert DISK into DRIVE b:

What is DISK No. ?

ในการสร้าง DISK นี้จะมีการถาม DISK ที่จะสร้างเป็นเบอร์อะไร ถ้าหากมี DISK นั้นอยู่แล้วจะแสดงข้อความออกมาเพื่อจะไปกำหนดใหม่หรือยกเลิกออกไป MAIN MENU

[5] MATERIAL REPORT

เป็นการพิมพ์ STATUS ของคลังสินค้าที่เหลืออยู่ในปัจจุบัน เมื่อเลือกข้อนี้จะให้ใส่แผ่น DATA DISK ตามเบอร์ที่ระบุมาให้ ถ้าแผ่น DATA DISK ผิดเบอร์ก็จะแสดงข้อความออกมาบอกและให้ใส่แผ่นใหม่อีกครั้ง

```

ใส่ DISK เบอร์ : ___ ลงใน DRIVE [B:]
Insert DISK No. ___ into DRIVE [B:]
[ C ] ทำงานต่อไป Continue
[ E ] ออกไป MAIN MENU
ข้อที่เลือก: Select ==> ___

```

การเลือก [E] นั้นจะเป็นการยกเลิกการทำงานและออกไปสู่ MAIN MENU ดังนั้นหลังจากใส่ DATA DISK ตามเบอร์ และเลือก [C] ถ้าใส่แผ่นถูกต้องจะขึ้นข้อความให้ลเพื่อการค้าจำนวนและหลังจากเสร็จแล้วก็ให้ใส่แผ่นต่อไปอีก จนกระทั่งหมดจำนวนแผ่น DATA DISK ที่ใช้ จะแสดง MENU ถามการแสดงผลว่าจะให้แสดงทาง เครื่องพิมพ์(PRINTER) หรือที่จอภาพ (DISPLAY)

```

ทำงานเสร็จแล้ว           Successful...
ต้องการจะแสดงผลทาง Display หรือ Printer ? [D/P]
You want to Display or Printer ? [D/P]

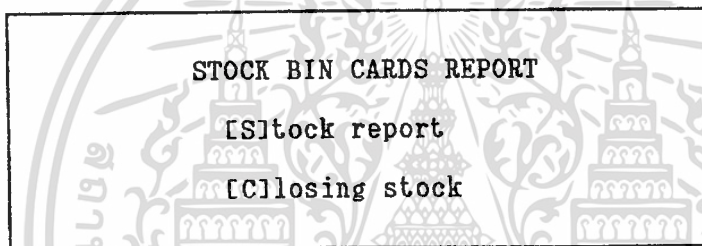
```

ถ้าเลือก [P] จะเป็นการแสดงผลทางเครื่องพิมพ์ ซึ่งเครื่องพิมพ์จะต้องพร้อมที่จะทำงาน ถ้าเครื่องพิมพ์ไม่พร้อมที่จะทำงานก็จะแสดงข้อความออกมาทาง จอภาพและเมื่อกด KEY

ESC จะเป็นการยกเลิกการพิมพ์ออกไป MAIN MENU แต่ถ้ากด KEY อื่นๆ จะไปตรวจเครื่องพิมพ์อีกจนกว่าจะพร้อมที่จะทำงาน หรือ ถ้าเลือก [D] จะแสดงผลทางจอภาพ ในการดูที่จอภาพนี้จะเลื่อนภาพไปทาง ซ้าย ขวา บน ล่าง ได้โดยการกด ปุ่มที่เป็นลูกศรตามที่ต้องการ เมื่อต้องการออกจากการแสดงผลให้กดปุ่ม [Esc] ก็จะกลับเข้าสู่ MAIN MENU

[6] STOCK REPORT

ข้อนี้จะเป็นการรายงาน จำนวนสินค้าที่รับเข้ามาและส่งออก ซึ่งจะมีการรายงานสองแบบ คือเมื่อเลือก STOCK REPORT จาก MAIN MENU ก็จะปรากฏ MENU ย่อยให้เลือกอีก



แบบแรก [S] จะเป็นการแสดงให้ทราบถึงสินค้าทั้งหมดที่มีการรับเข้าและส่งออกในปัจจุบัน คือจะมีการรายงานการรับ ส่งตั้งแต่แรกจนกระทั่งถึงปัจจุบันซึ่งถ้ามีการ CLOSING STOCK ไปจะทำให้ลบข้อมูลของการ รับและส่งออกหมดและเริ่มเก็บใหม่หลังจาก CLOSING STOCK และจำนวนสินค้าที่ยังเหลืออยู่ใน STOCK จะเลือกให้แสดงออกทางจอภาพหรือเครื่องพิมพ์เช่นเดียวกับการทำ MATERIAL REPORT

แบบสอง [C] จะเหมือนกับแบบแรก แต่แบบที่สองนี้จะมีการ ปิด STOCK คือจะ ลบจำนวนของ สินค้าที่มีการนำเข้าและส่งออก แต่จะเก็บจำนวนสินค้าที่ยังเหลืออยู่ ไว้เป็นยอดที่ยกมาในครั้งใหม่ ซึ่งการแสดงผลจะแสดงออกทางเครื่องพิมพ์อย่างเดียว ในการเลือกแบบนี้ ต้องเตรียมเครื่องพิมพ์ให้พร้อมทุกครั้งจึงจะสามารถ CLOSING ได้ถ้าเครื่องพิมพ์ไม่พร้อม จะกลับไป MAIN MENU

ในการเลือก STOCK REPORT จะมีการให้ใส่แผ่น DATA DISK จนกระทั่ง ครบทั้งหมดตามเบอร์ของการ สร้าง DISK

[7] FORMAT DISK

แผ่น DISK ที่จะนำไปเก็บข้อมูล หรือเป็น DATA DISK จะต้องนำมาทำการ FORMAT ก่อนโดยนำแผ่น DISK นั้นใส่ไว้ที่ DRIVE B: แล้วกด KEY 'Y' เพื่อเริ่มต้นการ FORMAT แผ่น

----- FORMAT DISK -----
ใส่ DISK ใหม่ลงใน DRIVE B:
Insert DISK into DRIVE B: OK ? _

[8] DELETE CODE

เป็นการลบรหัสสินค้าออกจาก DATA DISK เพื่อในกรณีที่ รหัสสินค้านั้นไม่มีการใช้งาน หรือมีการเพิ่ม รหัสสินค้าผิดแล้วต้องการจะลบออก ในการลบออกนั้น สินค้าจะต้องไม่มีจำนวน เหลืออยู่ใน Stock

เริ่มเก็บข้อมูล อย่างไร ?

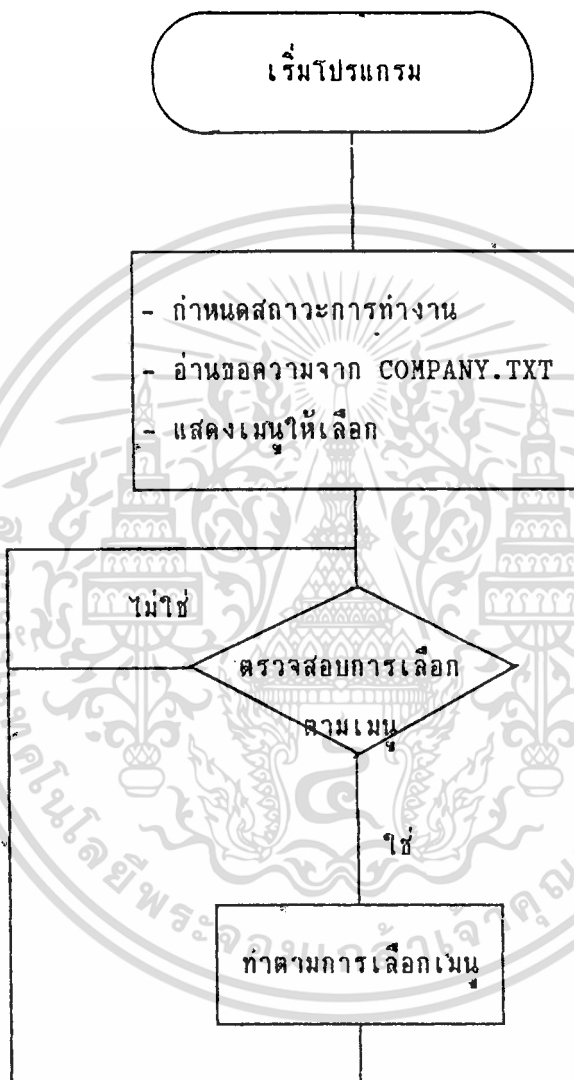
1. เตรียมแผ่นที่จะนำมาเป็น DATA DISK ต้องเป็นแผ่นที่มีคุณภาพที่ดี เพราะจะทำให้ข้อมูลนั้นไม่เสียหายได้ง่าย
2. นำแผ่น DATA DISK นั้นไป FORMAT โดยเลือกข้อ [7]
3. หลังจาก FORMAT DATA DISK ทุกแผ่นแล้วนำ แผ่นเหล่านั้นมาใส่ร้าน DISK ใหม่ โดยเลือกข้อ [6] ซึ่ง DISK แต่ละแผ่นจะต้องถูกกำหนด เบอร์ของ DISK หลังจากกำหนด แล้วเขียน เบอร์ติดไว้ที่ DISK แต่ละแผ่น
4. นำสินค้าที่จะเก็บมาตั้ง รหัสสินค้าโดยเลือกข้อ [5] การสร้างรหัส หรือเพิ่มรหัส สินค้า และกำหนดว่าสินค้าจะอยู่ที่ DISK เบอร์ใด ในการกำหนดเบอร์ DISK นั้นจะต้องให้ ตรงกับการกำหนดไว้ในข้อ 3 ถ้าไม่ตรงจะแสดงข้อความบอกออกมา จะยกเลิกได้โดยการกด KEY ESC
5. เมื่อมีสินค้ารับเข้ามา เลือกข้อ [1] หรือสินค้าส่งออก เลือกข้อ [2] และก็จะแสดง MENU ถามรหัสสินค้าที่เราต้องการ และจะตรวจสอบดูว่ารหัสนั้นมีอยู่ในข้อมูลหรือไม่ ถ้า ไม่มีก็จะแสดงข้อความบอกออกมาและให้ใส่ข้อมูลใหม่ หรือยกเลิก ถ้ามีข้อมูลอยู่ใน STOCK ก็จะแสดง MENU เพื่อให้ใส่รายละเอียดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า :

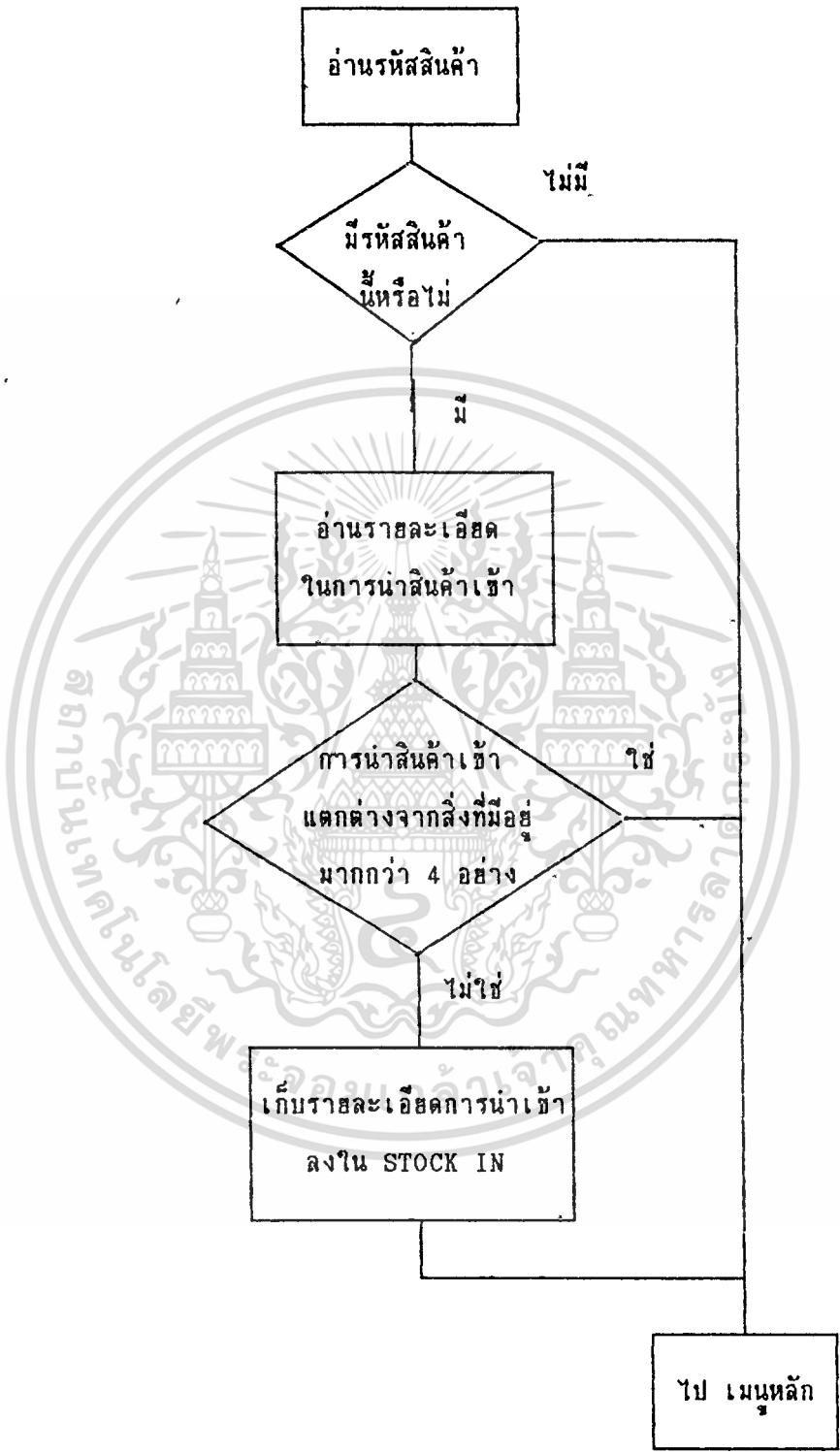
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

แผนผังการทำงานและโปรแกรม



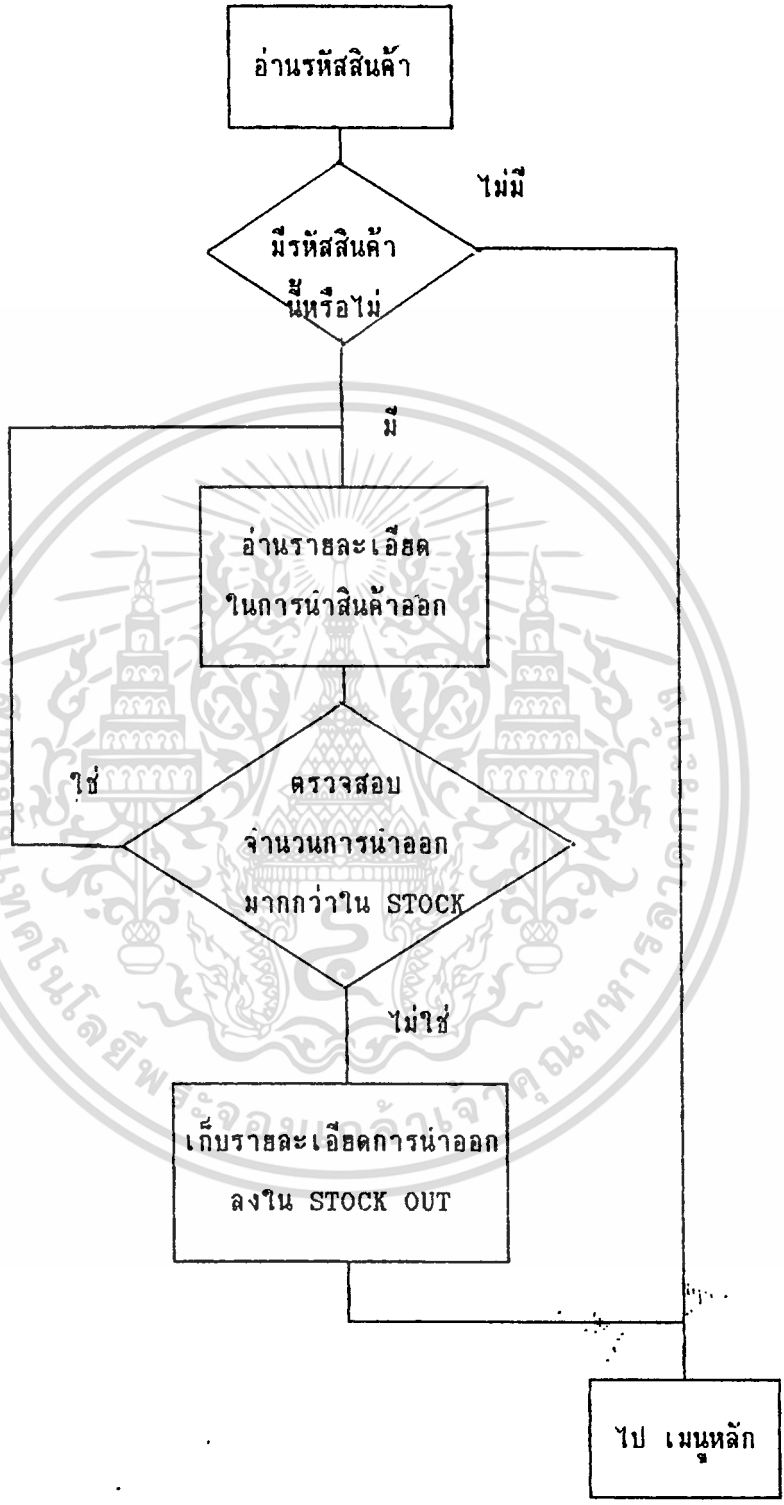
FLOW CHART แสดงเมนูหลัก



FLOW CHART แสดงการรับสินค้าเข้า

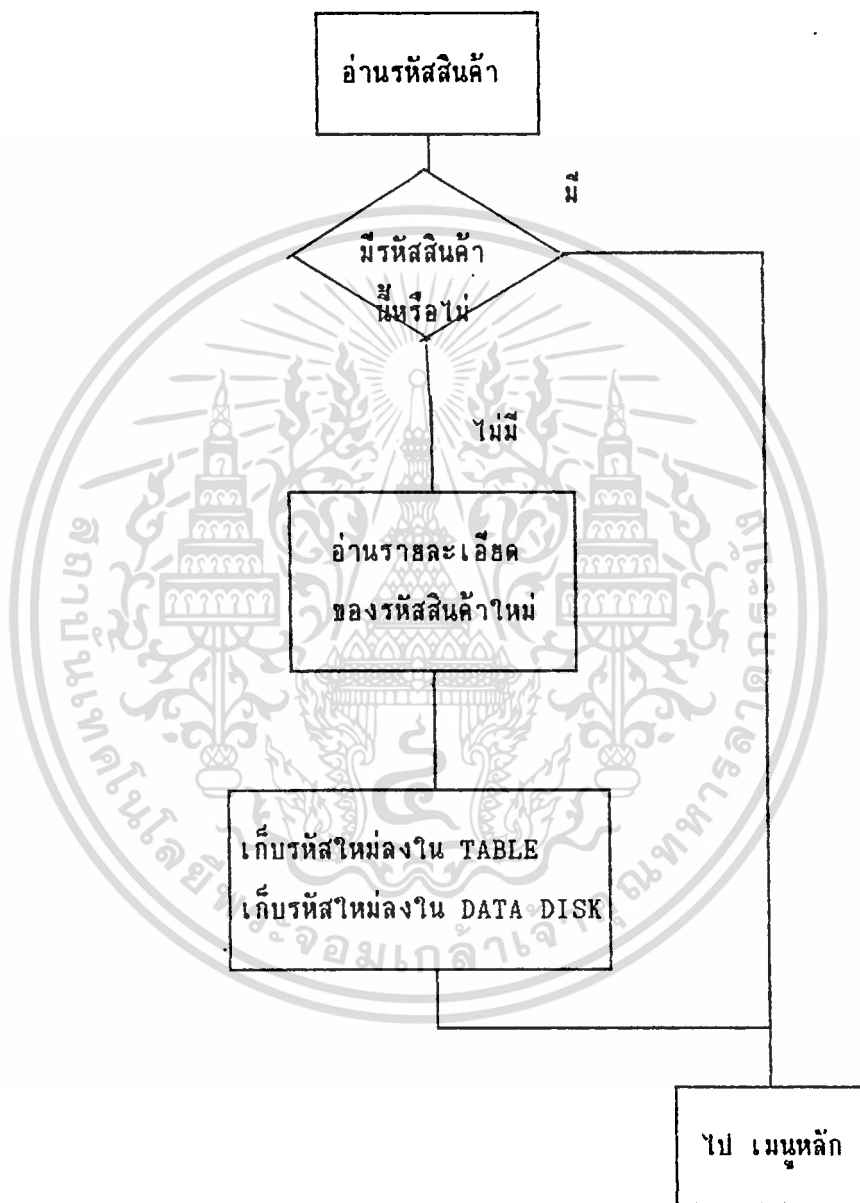
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



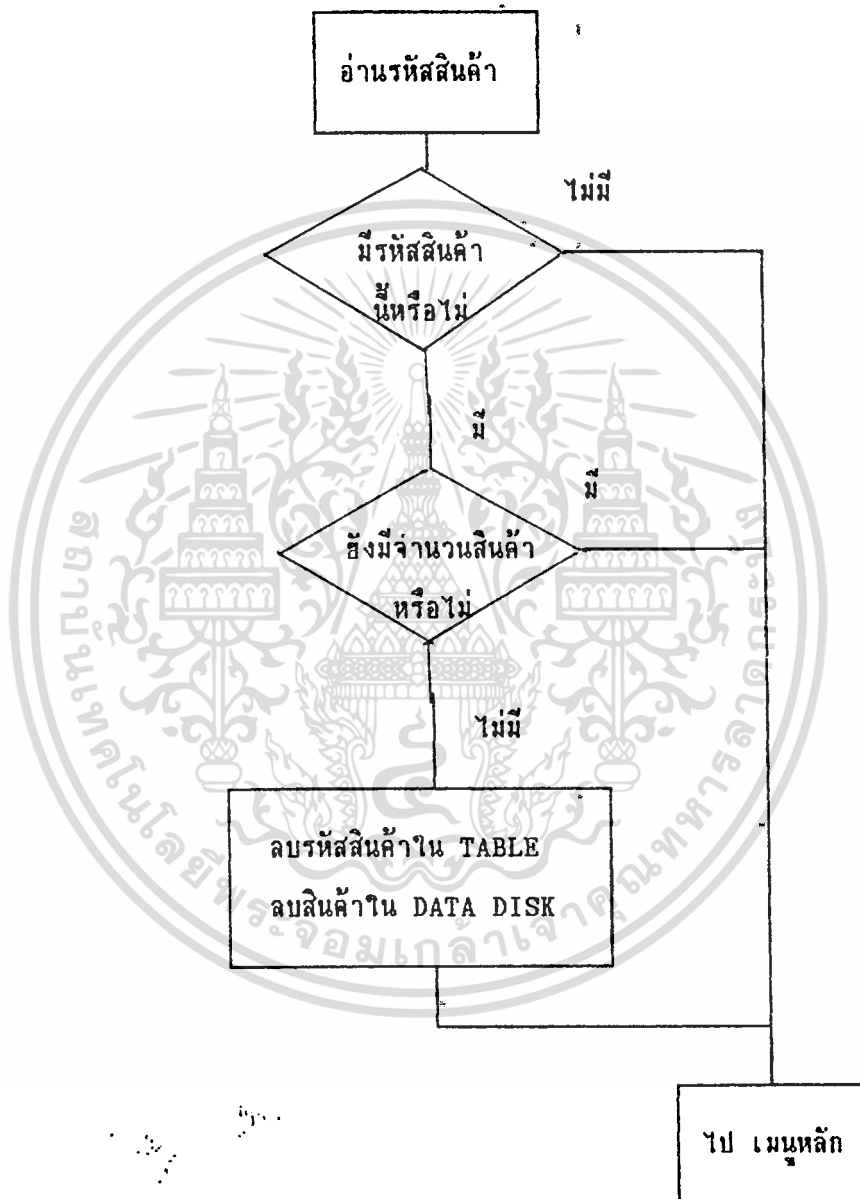
FLOW CHART แสดงการส่งสินค้าออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



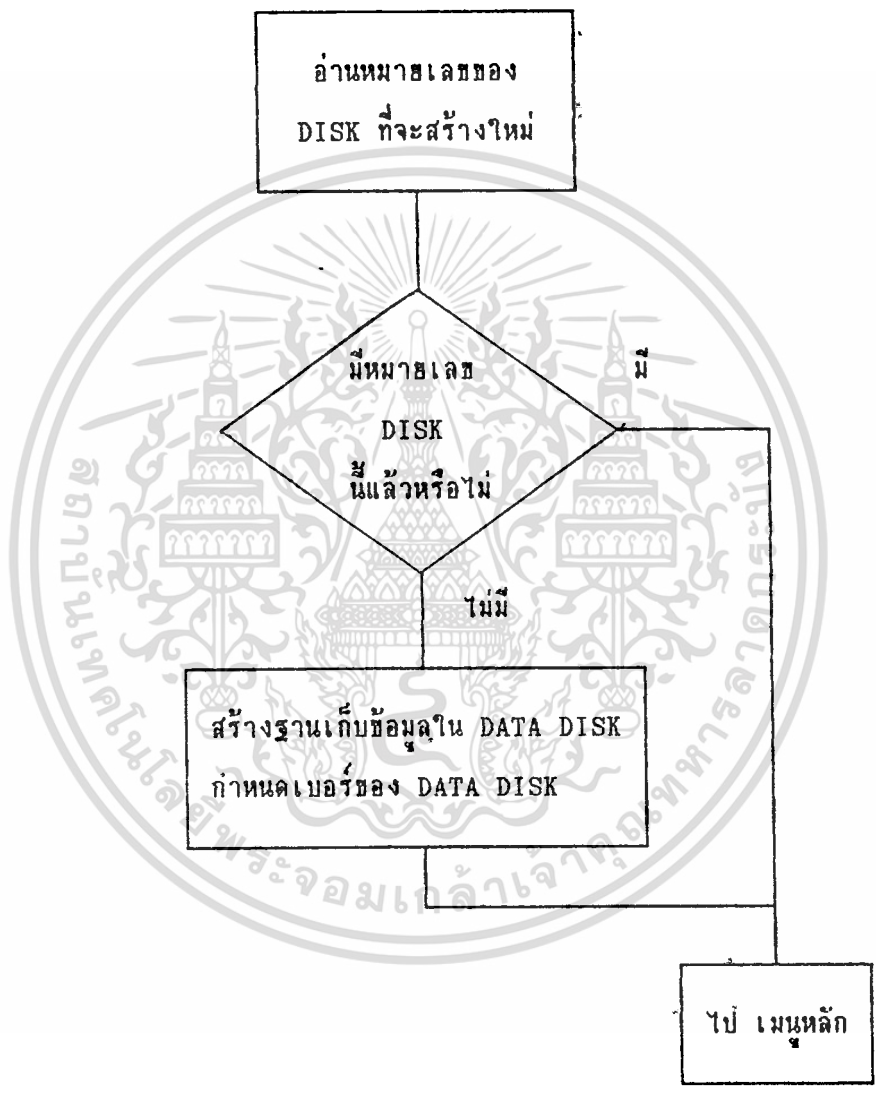
FLOW CHART แสดงการเพิ่มรหัสใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



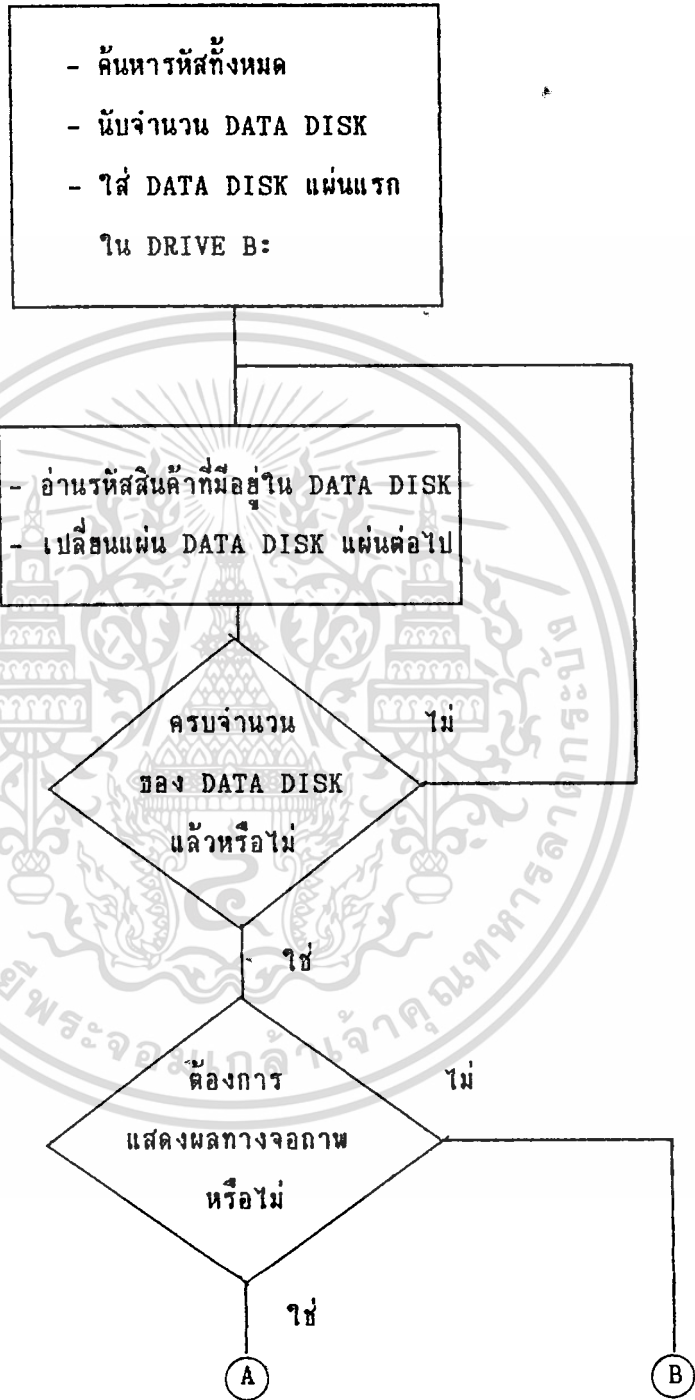
FLOW CHART แสดงการลบสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

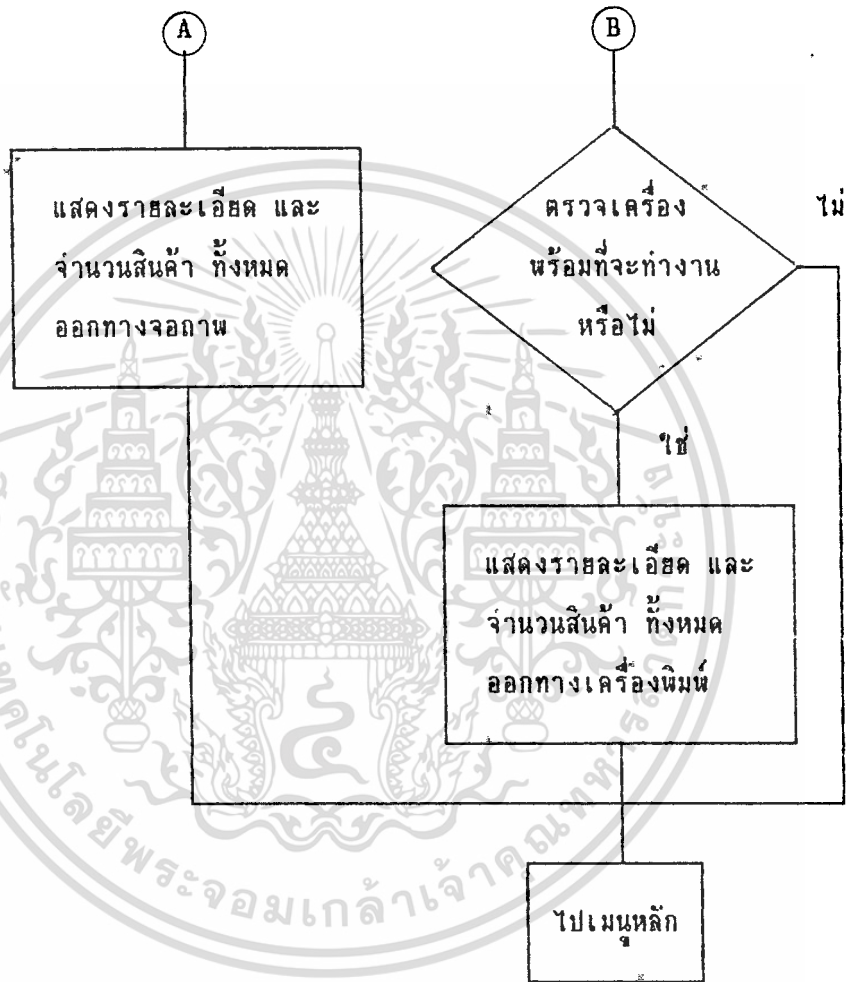


FLOW CHART แสดงการจัดเตรียมแผ่นข้อมูลใหม่

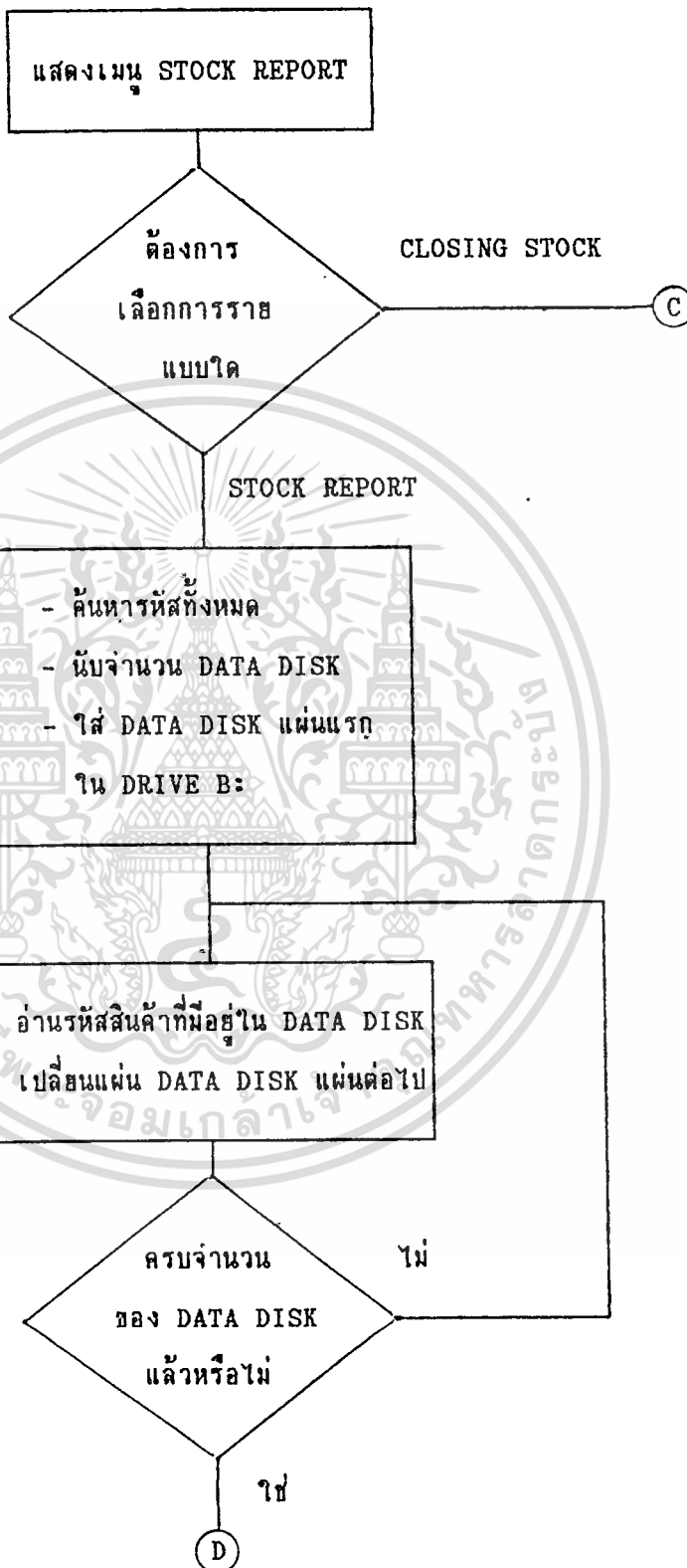
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FLOW CHART แสดงการรายงานจำนวนสินค้าใน STOCK (1 of 2)



FLOW CHART แสดงการรายงานจำนวนสินค้าใน STOCK (2 of 2)

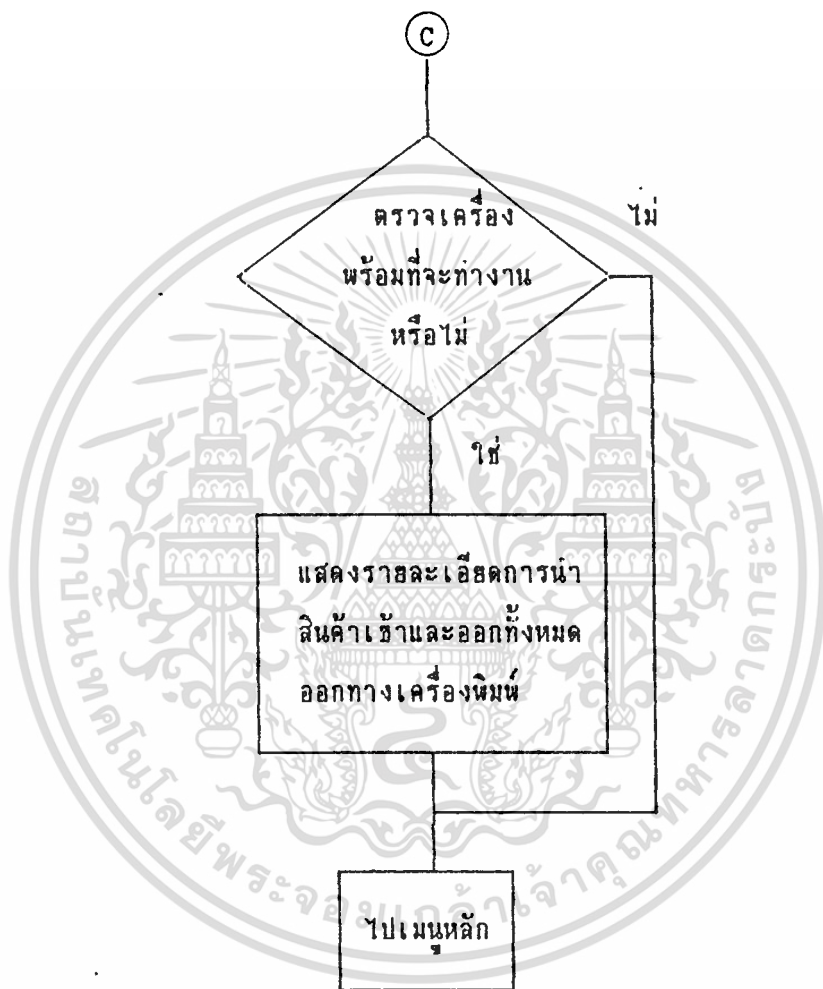


FLOW CHART แสดงการรายงานสินค้านำเข้าและนำออกใน STOCK (1 of 3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FLOW CHART แสดงการรายงานสินค้านำเข้าและนำออกใน STOCK (2 of 3)



FLOW CHART แสดงการรายงานสินค้านำเข้าและนำออกใน STOCK (3 of 3)

System: Stock Inventory Control

Author: KMITL'S

03/28/91 15:25:46

System Summary:

This system has:-

- 1807 lines of code-
- 0 program files
- 1 procedure files
- 13 procedures and functions
- 11 databases
- 3 index files
- 0 report forms
- 0 format files
- 0 label forms
- 0 memory variable files
- 321 cross-referenced tokens

See the tree diagram for programs, procedures, functions and format files

	Index	Report	Label	Memory
Databases	Files	Forms	Forms	Files

TABLE1.DBF	TABLE1.NTX			
B:DISK.DBF	STOCK.NTX			
B:STIN.DBF	BIN.NTX			
B:STOCK.DBF				
B:STOUT.DBF				
DISK.DBF				
STOCK.DBF				
STIN.DBF				
STOUT.DBF				
TEMP.DBF				
BIN.DBF				

FoxDoc created the following documentation files: -

C:\TEMP\STATS.DOC
 C:\TEMP\TREE.DOC
 C:\TEMP\FILELIST.DOC
 C:\TEMP\NDXSUMRY.DOC
 C:\TEMP\DATADICT.DOC
 C:\TEMP\ERROR.DOC

Action diagram files

UPDATE.BAT to update program source files in C:\TEMP
 BACKDBF.BAT to backup databases, indexes and memory files
 BACKPRG.BAT to backup program files, report forms and format files
 PRINTDOC.BAT to print documentation files

System: Stock Inventory Control

Author: KNITL'S

03/28/91 15:25:43

Tree Diagram

MAIN_NOR.PRG

—BOX_ (procedure in MAIN_NOR.PRG)—

—RECMENU (procedure in MAIN_NOR.PRG)

TABLE1.DBF (database)

B:DISK.DBF (database)

B:STIN.DBF (database)

B:STOCK.DBF (database)

TABLE1.NTX (index)

—BOX_ (procedure in MAIN_NOR.PRG)

—SHOW_TABLE (procedure in MAIN_NOR.PRG)

TABLE1.DBF (database)

TABLE1.NTX (index)

—BOX_ (procedure in MAIN_NOR.PRG)

—ISUMENU (procedure in MAIN_NOR.PRG)

TABLE1.DBF (database)

B:DISK.DBF (database)

B:STOUT.DBF (database)

B:STOCK.DBF (database)

TABLE1.NTX (index)

—BOX_ (procedure in MAIN_NOR.PRG)

—SHOW_TABLE (procedure in MAIN_NOR.PRG)

TABLE1.DBF (database)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE1.NTX (index)

BOX_ (procedure in MAIN_NOR.PRG)

MATERIAL (procedure in MAIN_NOR.PRG)

TABLE1.DBF (database)

B:DISK.DBF (database)

B:STOUT.DBF (database)

B:STIN.DBF (database)

B:STOCK.DBF (database)

TABLE1.NTX (index)

STOCK.NTX (index)

BOX_ (procedure in MAIN_NOR.PRG)

STOCK_RE (procedure in MAIN_NOR.PRG)

TABLE1.DBF (database)

TEMP.DBF (database)

B:DISK.DBF (database)

B:STOCK.DBF (database)

BIN.DBF (database)

B:STIN.DBF (database)

B:STOUT.DBF (database)

TABLE1.NTX (index)

STOCK.NTX (index)

BIN.NTX (index)

BOX_ (procedure in MAIN_NOR.PRG)

STOCK_CL (procedure in MAIN_NOR.PRG)

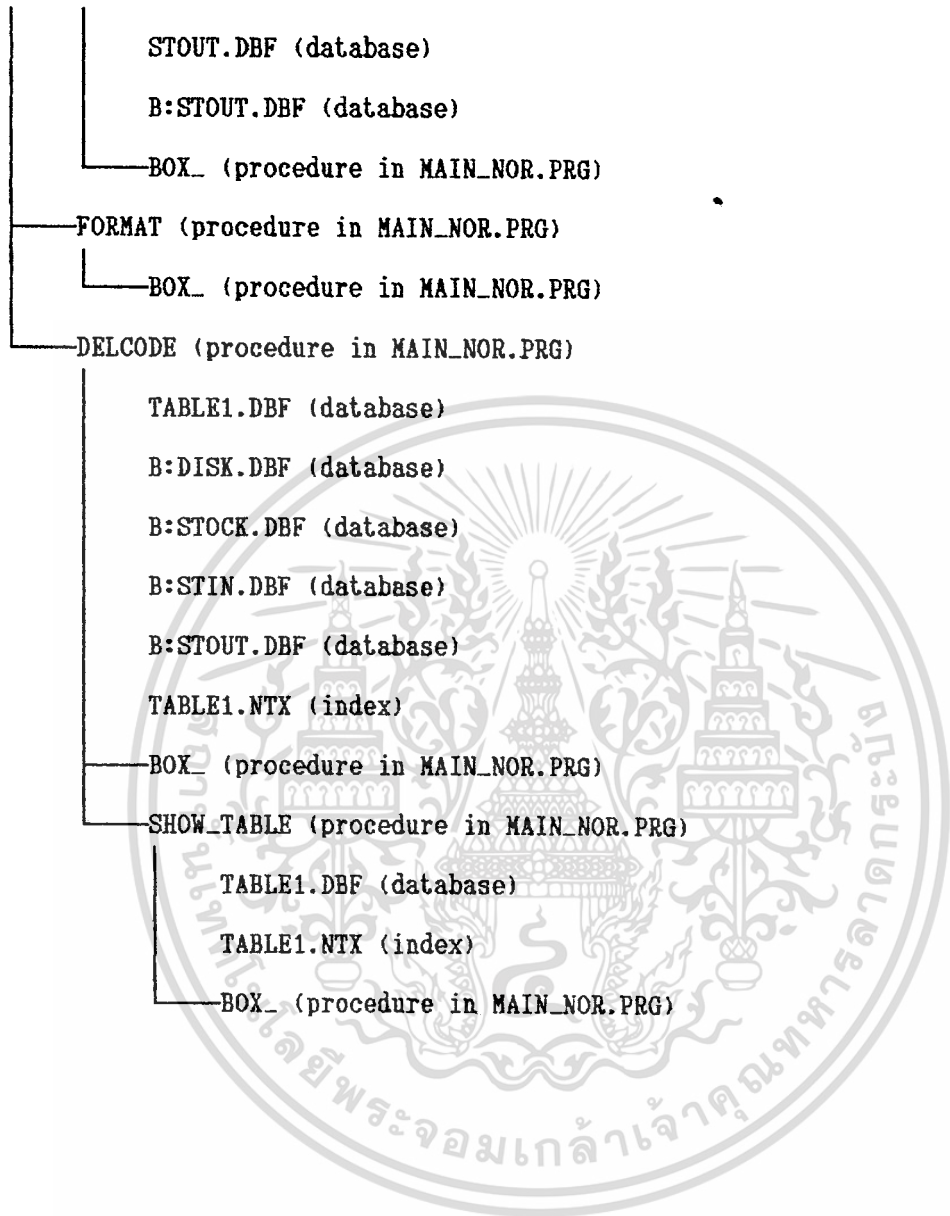
TABLE1.DBF (database)

TEMP.DBF (database)

B:DISK.DBF (database)

B:STOCK.DBF (database)

BIN.DBF (database)
 B:STIN.DBF (database)
 B:STOUT.DBF (database)
 TABLE1.NTX (index)
 STOCK.NTX (index)
 BIN.NTX (index)
 BOX_ (procedure in MAIN_NOR.PRG)
 TABLE_RL (procedure in MAIN_NOR.PRG)
 TABLE1.DBF (database)
 TABLE1.NTX (index)
 BOX_ (procedure in MAIN_NOR.PRG)
 CRCODE (procedure in MAIN_NOR.PRG)
 TABLE1.DBF (database)
 B:DISK.DBF (database)
 B:STOCK.DBF (database)
 TABLE1.NTX (index)
 BOX_ (procedure in MAIN_NOR.PRG)
 SHOW_TABLE (procedure in MAIN_NOR.PRG)
 TABLE1.DBF (database)
 TABLE1.NTX (index)
 BOX_ (procedure in MAIN_NOR.PRG)
 CRDISK (procedure in MAIN_NOR.PRG)
 DISK.DBF (database)
 B:DISK.DBF (database)
 STOCK.DBF (database)
 B:STOCK.DBF (database)
 STIN.DBF (database)
 B:STIN.DBF (database)



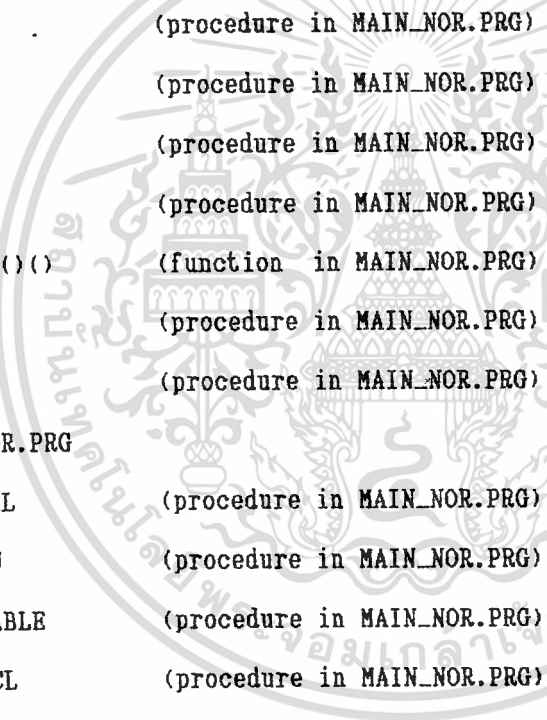
System: Stock Inventory Control

Author: KMITL'S

03/28/91 15:25:46

File List.

Programs and procedures:-



BOX_	(procedure in MAIN_NOR.PRG)
CRCODE	(procedure in MAIN_NOR.PRG)
CRDISK	(procedure in MAIN_NOR.PRG)
DELCODE	(procedure in MAIN_NOR.PRG)
ERR_PRN()()	(function in MAIN_NOR.PRG)
FORMAT	(procedure in MAIN_NOR.PRG)
ISUMENU	(procedure in MAIN_NOR.PRG)
MAIN_NOR.PRG	
MATERIAL	(procedure in MAIN_NOR.PRG)
RECMENU	(procedure in MAIN_NOR.PRG)
SHOW_TABLE	(procedure in MAIN_NOR.PRG)
STOCK_CL	(procedure in MAIN_NOR.PRG)
STOCK_RE	(procedure in MAIN_NOR.PRG)
TABLE_RL	(procedure in MAIN_NOR.PRG)

Procedure files:

MAIN_NOR.PRG

Databases:

B:DISK.DBF

B:STIN.DBF

B:STOCK.DBF

B:STOUT.DBF

BIN.DBF

DISK.DBF

STIN.DBF

STOCK.DBF

STOUT.DBF

TABLE1.DBF

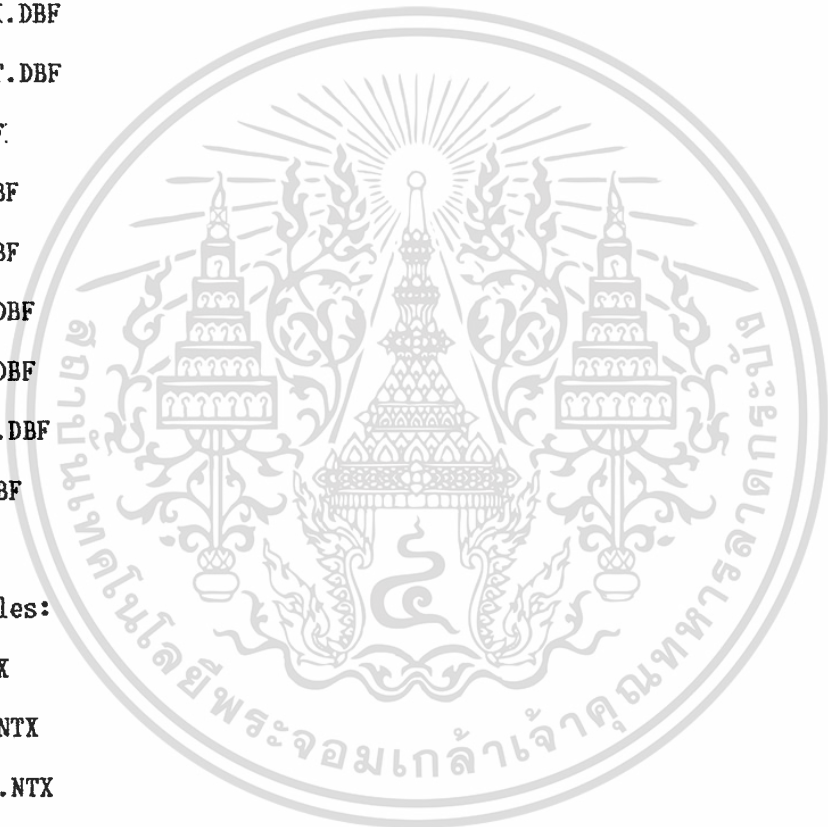
TEMP.DBF

Index files:

BIN.NTX

STOCK.NTX

TABLE1.NTX



System: Stock Inventory Control

Author: KMITL'St

03/28/91 15:25:30

Index Parameter Summary

3 index files in the system-

TABLE1.NTX

STOCK.NTX

BIN.NTX

TABLE1.NTX -- Indexed on: code-

Last updated: 03/27/91 at 1:08

This index file appears to be associated with database(s):

: TABLE1.DBF

Used by: RECMENU (procedure in MAIN_NOR.PRG)

: ISUMENU (procedure in MAIN_NOR.PRG)

: MATERIAL (procedure in MAIN_NOR.PRG)

: STOCK_RE (procedure in MAIN_NOR.PRG)

: STOCK_CL (procedure in MAIN_NOR.PRG)

: TABLE_RL (procedure in MAIN_NOR.PRG)

: CRCODE (procedure in MAIN_NOR.PRG)

: DELCODE (procedure in MAIN_NOR.PRG)
 : SHOW_TABLE (procedure in MAIN_NOR.PRG)

STOCK.NTX -- Indexed on: code-

Last updated: 03/28/91 at 12:15-

This index file appears to be associated with database(s):

: B:STOCK.DBF

Used by: MATERIAL (procedure in MAIN_NOR.PRG)

: STOCK_RE (procedure in MAIN_NOR.PRG)

: STOCK_CL (procedure in MAIN_NOR.PRG)

BIN.NTX -- Indexed on: code+day-

Last updated: 03/28/91 at 12:15-

This index file appears to be associated with database(s):

: BIN.DBF

Used by: STOCK_RE (procedure in MAIN_NOR.PRG)

: STOCK_CL (procedure in MAIN_NOR.PRG)

System: Stock Inventory Control

Author: KMITL'St

03/28/91 15:25:31

Database Structure Summary

11 databases in the system-

TABLE1.DBF

B:DISK.DBF

B:STIN.DBF

B:STOCK.DBF

B:STOUT.DBF

DISK.DBF

STOCK.DBF

STIN.DBF

STOUT.DBF

TEMP.DBF

BIN.DBF

Structure for database : TABLE1.DBF

Number of data records : 2

Last updated : 03/28/91 at 12:13

Field	Field name	Type	Width	Dec	Start	End
1	CODE	Character	15		1	15
2	NAME	Character	40		16	55
3	COMMENT	Character	15		56	70
4	DISK_NBR	Character	3		71	73
5	STATUS	Character	1		74	74
** Total **			75			

This database appears to be associated with index file(s):

: TABLE1.NTX (code)

Used by: RECMENU (procedure in MAIN_NOR.PRG)
 : ISUMENU (procedure in MAIN_NOR.PRG)
 : MATERIAL (procedure in MAIN_NOR.PRG)
 : STOCK_RE (procedure in MAIN_NOR.PRG)
 : STOCK_CL (procedure in MAIN_NOR.PRG)
 : TABLE_RL (procedure in MAIN_NOR.PRG)
 : CRCODE (procedure in MAIN_NOR.PRG)
 : DELCODE (procedure in MAIN_NOR.PRG)
 : SHOW_TABLE (procedure in MAIN_NOR.PRG)

Structure for database : B:DISK.DBF

Number of data records : 1

Last updated : 03/28/91 at 10:27

Field	Field name	Type	Width	Dec	Start	End
1	DNBR	Character	3		1	3
** Total **			4			

FoxDoc did not find any associated index files

Used by: RECMENU (procedure in MAIN_NOR.PRG)
 : ISUMENU (procedure in MAIN_NOR.PRG)
 : MATERIAL (procedure in MAIN_NOR.PRG)
 : STOCK_RE (procedure in MAIN_NOR.PRG)
 : STOCK_CL (procedure in MAIN_NOR.PRG)
 : CRCODE (procedure in MAIN_NOR.PRG)
 : CRDISK (procedure in MAIN_NOR.PRG)
 : DELCODE (procedure in MAIN_NOR.PRG)

Structure for database : B:STIN.DBF

Number of data records : 0

Last updated : 03/28/91 at 10:27

Field	Field name	Type	Width	Dec	Start	End
1	CODE	Character	15		1	15
2	DAY	Character	8		16	23
3	JOB_NBR	Character	6		24	29
4	JOB_PAY	Numeric	7	2	30	36
5	INV_NBR	Character	8		37	44
6	RECEIVE	Character	40		45	84
7	QTY_IN	Numeric	5		85	89
8	PRICE_AVE	Numeric	7	2	90	96
9	TOTAL	Numeric	8	2	97	104
10	COMMENT	Character	10		105	114
** Total **			115			

FoxDoc did not find any associated index files

Used by: RECMENU (procedure in MAIN_NOR.PRG)
 : MATERIAL (procedure in MAIN_NOR.PRG)
 : STOCK_RE (procedure in MAIN_NOR.PRG)
 : STOCK_CL (procedure in MAIN_NOR.PRG)
 : CRDISK (procedure in MAIN_NOR.PRG)
 : DELCODE (procedure in MAIN_NOR.PRG)

Structure for database : B:STOCK.DBF

Number of data records : 0

Last updated : 03/28/91 at 10:27

Field	Field name	Type	Width	Dec	Start	End
1	CODE	Character	15		1	15
2	Q1	Numeric	6		16	21
3	Q2	Numeric	6		22	27
4	Q3	Numeric	6		28	33
5	Q4	Numeric	6		34	39
6	P1	Numeric	7	2	40	46
7	P2	Numeric	7	2	47	53
8	P3	Numeric	7	2	54	60
9	P4	Numeric	7	2	61	67
10	JP1	Numeric	7	2	68	74
11	JP2	Numeric	7	2	75	81
12	JP3	Numeric	7	2	82	88
13	JP4	Numeric	7	2	89	95
14	QTY_BAL	Numeric	7		96	102
15	TOT_BAL	Numeric	11	2	103	113

** Total **

114

This database appears to be associated with index file(s):

: STOCK.NTX (code)d

Used by: RECMENU (procedure in MAIN_NOR.PRG)

: ISUMENU (procedure in MAIN_NOR.PRG)

: MATERIAL (procedure in MAIN_NOR.PRG)

: STOCK_RE (procedure in MAIN_NOR.PRG)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

: STOCK_CL (procedure in MAIN_NOR.PRG)
 : CRCODE (procedure in MAIN_NOR.PRG)
 : CRDISK (procedure in MAIN_NOR.PRG)
 : DELCODE (procedure in MAIN_NOR.PRG)

Structure for database : B:STOUT.DBF

Number of data records : 0

Last updated : 03/28/91 at 10:27

Field	Field name	Type	Width	Dec	Start	End
1	CODE	Character	15		1	15
2	DAY	Character	8		16	23
3	JOB_NBR	Character	6		24	29
4	ISSUED	Character	25		30	54
5	INV_NBR	Character	8		55	62
6	QTY_OUT	Numeric	5		63	67
7	COMMENT	Character	10		68	77
8	TOTAL	Numeric	8	2	78	85
** Total **			86			

FoxDoc did not find any associated index files

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Used by: ISUMENU (procedure in MAIN_NOR.PRG)
 : MATERIAL (procedure in MAIN_NOR.PRG)
 : STOCK_RE (procedure in MAIN_NOR.PRG)
 : STOCK_CL (procedure in MAIN_NOR.PRG)
 : CRDISK (procedure in MAIN_NOR.PRG)
 : DELCODE (procedure in MAIN_NOR.PRG)

 Structure for database : DISK.DBF-

Number of data records : 0

Last updated : 03/28/91 at 10:11

Field	Field name	Type	Width	Dec	Start	End
1	DNBR	Character	3		1	3
** Total **			4			

FoxDoc did not find any associated index files

Used by: CRDISK (procedure in MAIN_NOR.PRG)

 Structure for database : STOCK.DBF--

Number of data records : 0-

Last updated : 03/28/91 at 10:56

Field	Field name	Type	Width	Dec	Start	End
1	CODE	Character	15		1	15

2	Q1	Numeric	6		16	21
3	Q2	Numeric	6		22	27
4	Q3	Numeric	6		28	33
5	Q4	Numeric	6		34	39
6	P1	Numeric	7	2	40	46
7	P2	Numeric	7	2	47	53
8	P3	Numeric	7	2	54	60
9	P4	Numeric	7	2	61	67
10	JP1	Numeric	7	2	68	74
11	JP2	Numeric	7	2	75	81
12	JP3	Numeric	7	2	82	88
13	JP4	Numeric	7	2	89	95
14	QTY_BAL	Numeric	7		96	102
15	TOT_BAL	Numeric	11	2	103	113
** Total **			114			

FoxDoc did not find any associated index files

Used by: CRDISK

(procedure in MAIN_NOR.PRG)

r

Structure for database : STIN.DBF

Number of data records : 0

Last updated : 03/28/91 at 10:57

Field	Field name	Type	Width	Dec	Start	End
1	CODE	Character	15		1	15
2	DAY	Character	8		16	23
3	JOB_NBR	Character	6		24	29
4	JOB_PAY	Numeric	7	2	30	36
5	INV_NBR	Character	8		37	44
6	RECEIVE	Character	40		45	84
7	QTY_IN	Numeric	5		85	89
8	PRICE_AVE	Numeric	7	2	90	96
9	TOTAL	Numeric	8	2	97	104
10	COMMENT	Character	10		105	114
** Total **			115			

FoxDoc did not find any associated index files

Used by: CRDISK (procedure in MAIN_NOR.PRG)

Structure for database : STOUT.DBF

Number of data records : 0

Last updated : 03/28/91 at 10:53

Field	Field name	Type	Width	Dec	Start	End
1	CODE	Character	15		1	15
2	DAY	Character	8		16	23
3	JOB_NBR	Character	6		24	29
4	ISSUED	Character	25		30	54
5	INV_NBR	Character	8		55	62
6	QTY_OUT	Numeric	5		63	67
7	COMMENT	Character	10		68	77
8	TOTAL	Numeric	8	2	78	85
** Total **			86			

FoxDoc did not find any associated index files

Used by: CRDISK (procedure in MAIN_NOR.PRG)

Structure for database : TEMP.DBF-

Number of data records : 2-

Last updated : 03/28/91 at 10:09

Field	Field name	Type	Width	Dec	Start	End
1	CODE	Character	15		1	15
2	NAME	Character	40		16	55
3	COMMENT	Character	15		56	70
4	DISK_NBR	Character	3		71	73
5	STATUS	Character	1		74	74
** Total **			75			

Used by: STOCK_RE (procedure in MAIN_NOR.PRG)

: STOCK_CL (procedure in MAIN_NOR.PRG)

Structure for database : BIN.DBF

Number of data records : 1

Last updated : 03/28/91 at 12:15

Field	Field name	Type	Width	Dec	Start	End
1	CODE	Character	15		1	15
2	DAY	Character	8		16	23
3	JOB_NBR	Character	6		24	29
4	INV_NBR	Character	8		30	37
5	RECEIVE	Character	40		38	77
6	QTY_IN	Numeric	5		78	82
7	PRICE_AVE	Numeric	7	2	83	89
8	TOTAL	Numeric	8	2	90	97
9	JOB_PAY	Numeric	7	2	98	104
10	ISSUED	Character	25		105	129
11	QTY_OUT	Numeric	5		130	134
**	Total **		135			

This database appears to be associated with index file(s):

: BIN.NTX (code+day)1

Used by: STOCK_RE (procedure in MAIN_NOR.PRG)

: STOCK_CL (procedure in MAIN_NOR.PRG)

System: Stock Inventory Control

Author: KMITL'S

03/28/91 15:25:35

Data Dictionary

Field Name	Type	Len	Dec	Database
CODE	C	15	0	TABLE1.DBF
				BIN.DBF
				TEMP.DBF
				STOCK.DBF
				B:STOUT.DBF
				B:STOCK.DBF
				B:STIN.DBF
				STIN.DBF
				STOUT.DBF
				COMMENT
COMMENT	C	10	0	STOUT.DBF
				B:STIN.DBF
				STIN.DBF
				B:STOUT.DBF
COMMENT	C	15	0	TEMP.DBF
DAY	C	8	0	BIN.DBF
				STIN.DBF
				STOUT.DBF
				B:STIN.DBF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

				B:STOUT.DBF
DISK_NBR	C	3	0	TABLE1.DBF
				TEMP.DBF
DNBR	C	3	0	DISK.DBF
				B:DISK.DBF
INV_NBR	C	8	0	STOUT.DBF
				STIN.DBF
				B:STIN.DBF
				BIN.DBF
				B:STOUT.DBF
ISSUED	C	25	0	B:STOUT.DBF
				STOUT.DBF
				BIN.DBF
JOB_NBR	C	6	0	B:STIN.DBF
				BIN.DBF
				STOUT.DBF
				B:STOUT.DBF
				STIN.DBF
JOB_PAY	N	7	2	STIN.DBF
				B:STIN.DBF
				BIN.DBF
JP1	N	7	2	STOCK.DBF
				B:STOCK.DBF
JP2	N	7	2	STOCK.DBF
				B:STOCK.DBF
JP3	N	7	2	STOCK.DBF
				B:STOCK.DBF
JP4	N	7	2	STOCK.DBF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

				B:STOCK.DBF
NAME	C	40	0	TEMP.DBF
				TABLE1.DBF
P1	N	7	2	STOCK.DBF
				B:STOCK.DBF
P2	N	7	2	STOCK.DBF
				B:STOCK.DBF
P3	N	7	2	B:STOCK.DBF
				STOCK.DBF
P4	N	7	2	STOCK.DBF
				B:STOCK.DBF
PRICE_AVE	N	7	2	BIN.DBF
				STIN.DBF
				B:STIN.DBF
Q1	N	6	0	B:STOCK.DBF
				STOCK.DBF
Q2	N	6	0	STOCK.DBF
				B:STOCK.DBF
Q3	N	6	0	STOCK.DBF
				B:STOCK.DBF
Q4	N	6	0	STOCK.DBF
				B:STOCK.DBF
QTY_BAL	N	7	0	STOCK.DBF
				B:STOCK.DBF
QTY_IN	N	5	0	STIN.DBF
				BIN.DBF
				B:STIN.DBF
QTY_OUT	N	5	0	STOUT.DBF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

				B:STOUT.DBF
				BIN.DBF
RECEIVE	C	40	0	BIN.DBF
				B:STIN.DBF
				STIN.DBF
STATUS	C	1	0	TEMP.DBF
				TABLE1.DBF
TOTAL	N	8	2	B:STIN.DBF
				STIN.DBF
				B:STOUT.DBF
				BIN.DBF
				STOUT.DBF
TOT_BAL	N	11	2	B:STOCK.DBF
				STOCK.DBF

System: Stock Inventory Control

Author: KNITL'S

03/28/91 15:25:36

Procedure and Function Summary

1 files containing procedure in the system-

MAIN_NOR.PRG

MAIN_NOR.PRG -- Last updated: 03/28/91 at 12:30

Contains: SHOW_TABLE

Called by: RECMENU (procedure in MAIN_NOR.PRG)

Called by: ISUMENU (procedure in MAIN_NOR.PRG)

Called by: CRCODE (procedure in MAIN_NOR.PRG)

Called by: DELCODE (procedure in MAIN_NOR.PRG)

Calls: BOX_ (procedure in MAIN_NOR.PRG)

Contains: TABLE_RL

Called by: MAIN_NOR.PRG

Calls: BOX_ (procedure in MAIN_NOR.PRG)

Contains: ERR_PRN()()

Contains: BOX_

Called by: MAIN_NOR.PRG

Called by: RECMENU (procedure in MAIN_NOR.PRG)

Called by: ISUMENU (procedure in MAIN_NOR.PRG)

Called by: MATERIAL (procedure in MAIN_NOR.PRG)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Called by: STOCK_RE (procedure in MAIN_NOR.PRG)

Called by: STOCK_CL (procedure in MAIN_NOR.PRG)

Called by: TABLE_RL (procedure in MAIN_NOR.PRG)

Called by: CRCODE (procedure in MAIN_NOR.PRG)

Called by: CRDISK (procedure in MAIN_NOR.PRG)

Called by: FORMAT (procedure in MAIN_NOR.PRG)

Called by: DELCODE (procedure in MAIN_NOR.PRG)

Called by: SHOW_TABLE (procedure in MAIN_NOR.PRG)

Contains: RECMENU r

Called by: MAIN_NOR.PRG .:

Calls: BOX_ (procedure in MAIN_NOR.PRG)

Calls: SHOW_TABLE (procedure in MAIN_NOR.PRG)

Contains: ISUMENU r

Called by: MAIN_NOR.PRG .:

Calls: BOX_ (procedure in MAIN_NOR.PRG)

Calls: SHOW_TABLE (procedure in MAIN_NOR.PRG)

Contains: DELCODE r

Called by: MAIN_NOR.PRG .:

Calls: BOX_ (procedure in MAIN_NOR.PRG)

Calls: SHOW_TABLE (procedure in MAIN_NOR.PRG)

Contains: CRCODE r

Called by: MAIN_NOR.PRG .:

Calls: BOX_ (procedure in MAIN_NOR.PRG)

Calls: SHOW_TABLE (procedure in MAIN_NOR.PRG)

Contains: CRDISK r

Called by: MAIN_NOR.PRG .:

Calls: BOX_ (procedure in MAIN_NOR.PRG)

Contains: FORMAT r

Called by: MAIN_NOR.PRG

Calls: BOX_ (procedure in MAIN_NOR.PRG)

Contains: MATERIAL r

Called by: MAIN_NOR.PRG :

Calls: BOX_ (procedure in MAIN_NOR.PRG)

Contains: STOCK_RE r

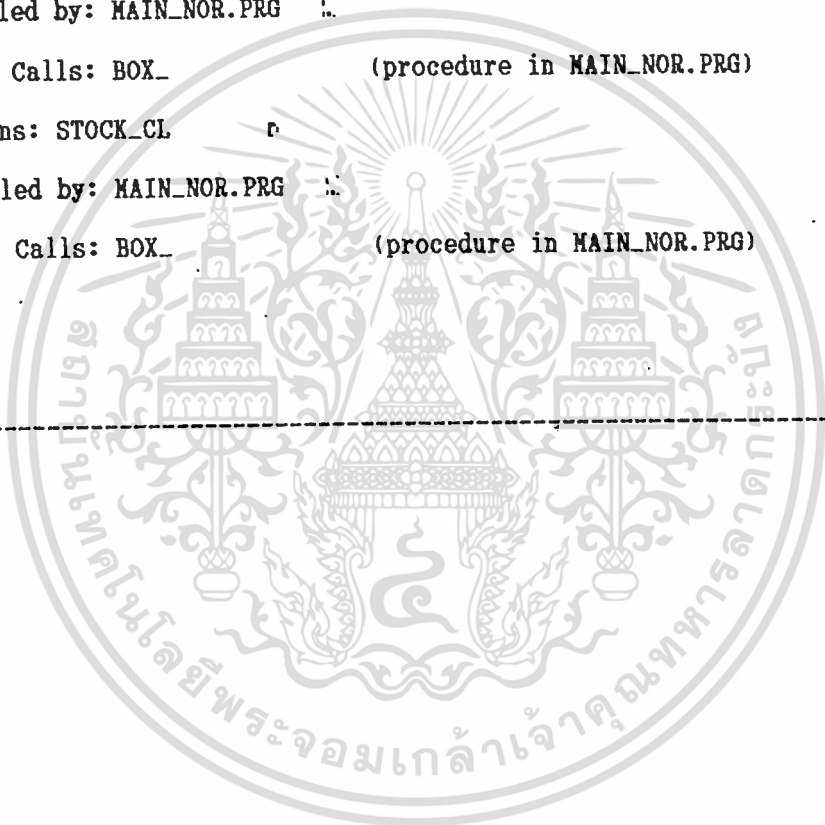
Called by: MAIN_NOR.PRG :

Calls: BOX_ (procedure in MAIN_NOR.PRG)

Contains: STOCK_CL r

Called by: MAIN_NOR.PRG :

Calls: BOX_ (procedure in MAIN_NOR.PRG)



```

*****
*:
*:
*:      Program: MAIN_NOR.PRG
*:
*:
*:      System: Stock Inventory Control
*:
*:      Author: KMITL'S
*:
*:      Copyright (c) 28, 1991, V2.0
*:
*:      Last modified: 03/28/91      12:30
*:
*:
*:  Procs & Fncts: BOX_
*:
*:          : RECMENU
*:
*:          : ISUMENU
*:
*:          : MATERIAL
*:
*:          : STOCK_RE
*:
*:          : STOCK_CL
*:
*:          : TABLE_RL
*:
*:          : CRCODE
*:
*:          : CRDISK
*:
*:          : FORMAT
*:
*:          : DELCODE
*:
*:          : SHOW_TABLE
*:
*:          : ERR_PRN()()
*:
*:
*:  Calls: BOX_      (procedure in MAIN_NOR.PRG)
*:
*:          : RECMENU      (procedure in MAIN_NOR.PRG)
*:
*:          : ISUMENU      (procedure in MAIN_NOR.PRG)
*:
*:          : MATERIAL      (procedure in MAIN_NOR.PRG)
*:
*:          : STOCK_RE      (procedure in MAIN_NOR.PRG)
*:
*:          : STOCK_CL      (procedure in MAIN_NOR.PRG)
*:
*:          : TABLE_RL      (procedure in MAIN_NOR.PRG)
*:
*:          : CRCODE      (procedure in MAIN_NOR.PRG)
*:
*:          : CRDISK      (procedure in MAIN_NOR.PRG)
*:
*:          : FORMAT      (procedure in MAIN_NOR.PRG)
*:
*:          : DELCODE      (procedure in MAIN_NOR.PRG)
*:
*:
*:
*:      Documented: 03/28/91 at 15:20      FoxDoc version 1.0
*:
*****
* *****
* *****
*
*      MAIN PROGRAM
* *****
*
* #include "Inkey.ch"
*
* clear      && Initian memory and data base
*
* close all
*
* clear all
*
* set echo off      && Set parameter the operation system
*
* set talk off
*
* set safety off
*
* set status off
*
* set escap off
*
* set scor off

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set procedure to main_nor
set func 2 to ''          && Clear function key
set func 3 to ''
set func 4 to ''
set func 5 to ''
set func 6 to ''
set func 7 to ''
set func 8 to ''
set func 9 to ''
set func 10 to ''
*on error do err_fix with error(),message()

*errorblock( { !objerror; err_fix( objerror ) } )
if file("company.txt")
  nheader=fopen("company.txt")      && Read company name from text file
  head = rtrim( freadstr(nheader,40) )
  fclose(nheader)
  num=len(head)
else
  head="COMPANY NAME"
  num=12
endif
do while .t. && (1)
  close data
  ch = ' '
  clear          && Display main menu
  set color to /w
  @ 4, 32 say ' MAIN MENU '
  set color to
  @ 2, 38-(num/2) say head
  do box_ with 3,7,17,69
  @ 6, 10 say '[E] ออกจากโปรแกรม Exit to DOS'

  @ 7, 10 say '      '
  @ 8, 10 say '[1] รับสินค้าเข้า Receive'
  @ 9, 10 say '      '
  @ 10, 10 say '[2] ส่งสินค้าออก Issue'
  @ 11, 10 say '      '
  @ 12, 10 say '[3] เพิ่ม รหัส ใหม่ Add new CODE'
  @ 13, 10 say '      '
  @ 14, 10 say '[4] จัดเตรียมแผ่นข้อมูลใหม่ Create data DISK'
  @ 15, 10 say '      '

  @ 6, 43 say '[5] MATERIAL REPORT'
  @ 8, 43 say '[6] STOCK REPORT'
  @ 10, 43 say '[7] FORMAT DISK'
  @ 11, 43 say '      '
  @ 12, 43 say '[8] ลบรหัสสินค้า Delete CODE'
  @ 15, 28 say '      '

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@ 16, 28 say 'ขอฟทเลอก Select : ' get ch pict '!
read

do case                                && Check select choice from menu
case ch = 'E'
    quit
case ch = '1'
    do recmenu
case ch = '2'

do isumenu
case ch = '5'
do material with head
case ch = '6'
    @ 7,10 clear to 16,55                && Display stock report menu
do box_ with 7,10 ,16,55
    @ 9,20 say ' STOCK BIN CARDS REPORT '
    @ 11,20 say ' [S] Stock report'
    @ 13,20 say ' [C] Closing stock'
    @ 15,20 say ' [T] Table code report'
do while .t.
    key = 0
do while key = 0
    key = inkey()
enddo
do case                                && Check select report
case key = 83 .or. key = 115           && Select 'S'
do stock_re with head
case key = 99 .or. key = 67           && Select 'C'
do stock_cl
case key = 84 .or. key = 116
do table_rl
endcase
exit
enddo
case ch = '3'
do crcode
case ch = '4'
do crdisk
case ch = '7'
do format
case ch = '8'
do delcode
endcase
enddo
close all
set status on
return && QUIT
*****
*!*****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า :

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*!
*!      Procedure: SHOW_TABLE
*!
*!      Called by: RECMENU      (procedure in MAIN_NOR.PRG)
*!                  : ISUMENU      (procedure in MAIN_NOR.PRG)
*!                  : CRCODE      (procedure in MAIN_NOR.PRG)
*!                  : DELCODE      (procedure in MAIN_NOR.PRG)
*!
*!      Calls: BOX_      (procedure in MAIN_NOR.PRG)
*!
*!      Uses: TABLE1.DBF
*!
*!      Indexes: TABLE1.NDX
*!
*!*****
procedure show_table
use table1 new
index on code to table1
fsize = (recc()*60)+3
set console off
list code, name to file ptable all off
set console on

handle = fopen( "ptable.txt" )
fbuff = space( fsize )

fread( handle, @fbuff, fsize )
nscr = savescreen(10, 10, 22, 70)
@ 10,10 clear to 22,70
do box_ with 10,10,22,70
set color to /w
@ 11,11 clear to 21,69
@ 11,13 say "CODE" NAME"
set color to
set cursor off
memoedit( fbuff, 12, 11, 21, 69, .F. )
set cursor on
restscreen(10, 10, 22, 70, nscr)

return
*****
*!*****
*!
*!      Procedure: TABLE_RL
*!
*!      Called by: MAIN_NOR.PRG
*!
*!      Calls: BOX_      (procedure in MAIN_NOR.PRG)
*!
*!      Uses: TABLE1.DBF
*!

```

```

*!           Indexes: TABLE1.NDX
*!
*!*****
procédure table_rl
set color to /w
@ 16,10 clear to 22,66

@ 18,15 say 'Please wait for Process'
@ 19,15 say '      '
@ 20,15 say 'โปรดรอ สกคร'
@ 21,15 say '      '
set color to

use table1 new
index on code to table1
fsize = (recc()*60)+3
set console off
list code, name to file ptable all off
set console on

set color to /w
@ 13,10 clear to 22,63
@ 16,23 say '      '
@ 17,23 say 'ทำงานสำเร็จแล้ว Successfully.'
@ 18,14 say '      '
@ 19,14 say 'ต้องการจะแสดงผลทาง display หรือ printer ? [D/P]'
@ 21,14 say 'You want to Display or Printer ? [D/P]'
set color to

do while .t.    && (5)
  key= 0
  do while key=0
    key=inkey()
  enddo
  do case
  case key=68.or.key=100    && D

    handle = fopen( "ptable.txt" )
    fbuff = space( fsize )

    fread( handle, @fbuff, fsize )
    nscr = savescreen(10, 10, 22, 70)
    @ 10,10 clear to 22,70
    do box_ with 10,10,22,70
    set color to /w
    @ 11,11 clear to 21,69
    @ 11,13 say "CODE           NAME"
    set color to
    set cursor off

```

```

memoedit( fbuff, 12, 11, 21, 69, .P. )
set cursor on
restscreen(10, 10, 22, 70, nscr)

exit
case key=80.or.key=112  && P
if err_prn()
set console off
set print on
?" -CODE-          -NAME-"
type ptable.txt
eject
set print off
set cons on
endif
exit
endcase
enddo      && (5)

return
*****
*!*****
*!
*!      Function: ERR_PRN()
*!
*!*****
FUNCTION err_prn()

err_dis = savescreen(22,0,24,79)
pass = .t.
do while !isprinter()
set color to /+w
@ 22,0 clear to 24,79      && Display message retry when error

@ 22,10 say ' ๘ ๘ ๘ ๘ ๘ '
@ 23,10 say 'เครื่องพิมพ์ ไม่พร้อมที่จะทำงาน '
@ 24,10 say 'Printer LPT1 OFFLINE      Press Enter key to try again.'
* @ 24,10 say objerror:decription+ ' Press Enter key to try again.'
set color to
do while .t.
key=0
do while key=0
key=inkey()
enddo
do case
                                && Check key is select
case key=13                      && Press ENTRY key
restscreen(22,0,24,79,err_dis)
exit
case key=27                      && Press ESC key
restscreen(22,0,24,79,err_dis)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return (.f.)
    endcase
enddo
enddo
return (.t.)

```

```

*****
*!*****
*!
*!      Procedure: BOX_
*!
*!      Called by: MAIN_NOR.PRG
*!          : RECMENU          (procedure in MAIN_NOR.PRG)
*!          : ISUMENU          (procedure in MAIN_NOR.PRG)
*!          : MATERIAL         (procedure in MAIN_NOR.PRG)
*!          : STOCK_RE         (procedure in MAIN_NOR.PRG)
*!          : STOCK_CL         (procedure in MAIN_NOR.PRG)
*!          : TABLE_RL        (procedure in MAIN_NOR.PRG)
*!          : CRCODE           (procedure in MAIN_NOR.PRG)
*!          : CRDISK           (procedure in MAIN_NOR.PRG)
*!          : FORMAT           (procedure in MAIN_NOR.PRG)
*!          : DELCODE          (procedure in MAIN_NOR.PRG)
*!          : SHOW_TABLE       (procedure in MAIN_NOR.PRG)
*!
*!*****
procedure box_
parameter y1,x1,y2,x2          && Write box in font thai
@ y1,x1,y2,x2 box chr(149)+chr(158)+chr(151)+chr(159)+chr(157)+;
chr(158)+chr(155)+chr(159)
return
*****
*!*****
*!
*!      Procedure: RECMENU
*!
*!      Called by: MAIN_NOR.PRG
*!
*!          Calls: BOX_          (procedure in MAIN_NOR.PRG)
*!          : SHOW_TABLE        (procedure in MAIN_NOR.PRG)
*!
*!          Uses: TABLE1.DBF
*!          : B:DISK.DBF
*!          : B:STIN.DBF
*!          : B:STOCK.DBF
*!
*!          Indexes: TABLE1.NDX
*!
*!*****
procedure recmenu
kcode= '

```

```

clear                                && Display receive code menu
@ 3, 9 say "RECEIVE MENU"
@ 5, 3 say '  '
@ 6, 3 say 'รหัสค้นหา Code No.'
do box_ with 1, 2, 18, 79 && (1)
do box_ with 2, 3, 4, 28 && (3)
@ 8, 20 say 'Press F1 for help'
@ 9, 20 say '  '
@ 10, 20 say 'กด key F1 เพื่อแสดง code'

set key K_F1 to show_table
@ 6, 20 get kcode pict '@!'          && Read code from keyboard
read
set key K_F1
do while .t. && (2)
  use tabel new
  index on code to tabel
  @ 2,30 say '+-----+'
  @ 3,30 say '  '
  @ 4,30 say 'รหัสค้นหา: '+kcode
  seek kcode                          && Serach code from tebale data
  if found() && (IF 2)                  && Check code in table data
    cn = 'X'
    dk = disk_nbr
    sta= status
    na = name
    cm = comment
    @ 7,10 clear to 16,55
    do while .t. && (3)
      do box_ with 7,10, 16,55          && Display disk number of code
      @ 8,12 say '  '
      @ 9,12 say 'ใส่ DISK (เบอร์ :'+dk+' )ลงใน DRIVE [B:]'
      @ 10,12 say 'Insert DISK No. '+dk+' into DRIVE [B:]'
      @ 11,12 say '  '
      @ 12,12 say ' [C] ทำงานจนตอไป Continue'
      @ 13,12 say ' [E] ออกไป MAIN MENU'
      @ 14,12 say '  '
      @ 15,12 say 'ขอเลือก: Select ==>' get cn pict '! '
      read
      do case                            && Check select key
        case cn = 'C' .or. cn = 'c'
          if .not.file("b:disk.dbf")      && Check disk is data disk?
            set color to /w
            @ 13,35 clear to 15,54
            @ 13,36 say '  '
            @ 14,36 say 'ใส่ DISK ผิดเบอร์ '
            @ 15,36 say 'Insert disk wrong'
            set color to

```

```

loop && (3)                                && If not match loop for
endif                                        && try again

use b:disk new
store dnbr to dn

if dn = dk && (IF 3)                        && Check disk number is
@ 2,30 say 'DISK # : '+dk                 && correct ?
select 1
use b:stin
select 2
use b:stock
locate for kcode=code                      && Find code in stock
if found() && (IF 4)                       && Check is found?
xq1 =q1
xq2 =q2
xq3 =q3
xq4 =q4
xp1 =p1
xp2 =p2
xp3 =p3
xp4 =p4
xjpl=jpl
xjp2=jp2
xjp3=jp3
xjp4=jp4
select 1
append blank
day_=ctod(" / / ")
@ 7,10 clear to 18,55                       && Display receive menu
do box_ with 1, 2, 18, 79
@ 6, 20 say kcode
@ 8, 5 say 'DATE'
@ 10, 5 say "JOB_NBR"
@ 12, 5 say "D/O No."
@ 14, 5 say 'Receive from'
@ 16, 5 say 'Comment'
@ 6, 40 say "จำนวน QTY"
@ 7, 31 say ' '
@ 8, 31 say "ค่าแรงต่อชิ้น JOB_PAY"
@ 9, 31 say ' '
@ 10, 31 say "ค่าสต็อคต่อชิ้น Price_AVE"
@ 11, 31 say ' '
@ 12, 31 say "จำนวนเงินทั้งหมด Total"
@ 8, 18 get day_
set color to /w
@ 8, 18 say 'mm/dd/yy'
set color to
@ 10, 18 get stin->job_nbr pict '@!'
@ 12, 18 get stin->inv_nbr pict '@!'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@ 14, 18 get stin->receive pict '@!'
set color to /w
@ 16, 18 say '      '
@ 6, 52 say "      "
@ 8, 52 say '      '
@ 8, 61 say 'X'
@ 10, 52 say '      '
@ 12, 52 say '      '
set color to
read
if day_=ctod(" / / ")          && Check data not input
return                          && return to main menu
endif
qt = 0
jpx= 0
@ 6, 52 get qt pict "99999"
read                            && Save data to file
repl stin->qty_in with qt, code with kcode,stin->day with dtoc(day_)
ty = 'X'
jp = 0
@ 8, 52 get jp pict '9999.99'
@ 8, 61 get ty pict '!'
read
do case                          && Check kind of job pay
case jp > 0 .and. ty = 'A'
jpx = iif(qt>0,jp/qt,0)
case jp > 0 .and. ty <> 'A'
jpx = jp
case jp = 0
jpx = jp
endcase
pa = 0
ta = 0
@ 10, 52 get pa pict '9999.99'
read
if pa > 0                          && Check prince average
ta = (qt * pa)+(qt * jpx)
set color to /w
@ 12, 52 say ta pict '99999.99'
set color to
else
@ 12, 52 get ta pict '99999.99'
read
pa = iif(qt>0,(ta - (qt * jpx)) / qt,0)
set color to /w
@ 10, 52 say pa pict '9999.99'
set color to
endif
repl price_ave with pa, total with ta,job_pay with jpx

```

```

@ 16, 18 get stin->comment
read
select 2
do case          && Check different of commodity
case xp4=pa .and. xjp4=jpx
    xq4 = xq4+qt
case xp3=pa .and. xjp3=jpx
    xq3 = xq3+qt
case xp2=pa .and. xjp2=jpx
    xq2 = xq2+qt
case .t.
    if xq1 # 0
        select 1 && stin
        go bott
        dele
        pack
        set color to /w
        @ 8,11 clear to 15,54
        @ 8,15 say '      '
        @ 9,15 say 'ไม่สามารถเก็บสินค้าที่แตกต่างกัน 4 ชนิด'
        @ 10,20 say '      '
        @ 11,20 say 'กดปุ่มใด ๆ เพื่อ ยกเลิก '
        @ 12,20 say '      '
        @ 13,20 say "Can't keep this commodity"
        @ 14,20 say '      over than 4 kind.'
        set color to
        key = 0
        do while key = 0
            key = inkey()
        enddo
        return
    endif
    xq1 = xq2
    xq2 = xq3
    xq3 = xq4
    xq4 = qt
    xp1 = xp2
    xp2 = xp3
    xp3 = xp4
    xp4 = pa
    xjp1 = xjp2
    xjp2 = xjp3
    xjp3 = xjp4
    xjp4 = jpx
endcase          && Save data to stock receive file
repl q1 with xq1,q2 with xq2,q3 with xq3,q4 with xq4,;
p1 with xp1,p2 with xp2,p3 with xp3,p4 with xp4,;
j1 with xjp1,j2 with xjp2,j3 with xjp3,j4 with xjp4

endif && (IF 4)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        exit  && (3)                && If not found of check code
    else
        set color to /w
        @ 13,35 clear to 15,54
        @ 13,36 say '      ^      '
        @ 14,36 say 'ใส่ DISK ผิดเบอร์ '
        @ 15,36 say 'Insert disk wrong'
        set color to
        loop  && (3)
    endif  && (IF 3)
    case cn = 'E' .or. cn = 'e'
        return  && (3)
    endcase
    enddo  && (3)
else && (IF 2)                && Code not found ip stock
    cn = 'Y'
    @ 9,10 clear to 15,65      && Display not found code menu
    do box_ with 9,10, 15,65
    @ 10,12 say '      ^ ^ ^ ^ ^ '
    @ 11,12 say '      โปรแกรมสินค้า อยู่ใน STOCK '
    @ 12,12 say '      ^ ^ ^ ^ ^ '
    @ 13,12 say '      กลับไปยัง MAIN MENU เพื่อ สร้างรหัส สินค้าใหม่ '
    @ 14,12 say 'Find not found this code ! Return to MAIN MENU.'
    do while inkey()=0        && Ckeck key for waiting and press
    enddo                      && to return
        exit  && (2)
    endif && (IF 2 )
    exit  && (2)
enddo  && (2)

return
*****
*!*****
*!
*!      Procedure: ISUMENU
*!
*!      Called by: MAIN_NOR.PRG
*!
*!      Calls: BOX_                (procedure in MAIN_NOR.PRG)
*!              : SHOW_TABLE      (procedure in MAIN_NOR.PRG)
*!
*!      Uses: TABLE1.DBF
*!              : B:DISK.DBF
*!              : B:STOUT.DBF
*!              : B:STOCK.DBF
*!
*!      Indexes: TABLE1.NDX
*!
*!*****
procedure isumenu

```

```

kcode= '
clear                                && Display select code for issue menu
@ 3, 9 say "ISSUE MENU"
@ 5, 3 say '   '
@ 6, 3 say "รหัสสินค้า Code No."
do box_ with 1, 2 , 18, 79 && (1)
do box_ with 2, 3 , 4, 28 && (3)
@ 8, 20 say 'Press F1 for help'
@ 9, 20 say '   '
@ 10, 20 say 'กด key F1 เพื่อแสดง code'

set key K_F1 to show_table
@ 6, 20 get kcode pict '!!!!!!!!!!!!!!'
read
set key K_F1
do while .t. && (2)
  use table1 new
  index on code to table1
  @ 2,30 say '+-----+'
  @ 3,30 say '   '
  @ 4,30 say 'รหัสสินค้า :'+kcode
  seek kcode                                && Find code from table
  if found() && (IF 2)                       && Check code is correct?
    cn = 'X'
    dk = disk_nbr
    sta= status
    na = name
    cm = comment
    @ 7,10 clear to 16,55
    do while .t. && (3)
      do box_ with 7,10, 16,55                && Display insert disk number
      @ 8,12 say '   '
      @ 9,12 say '   รหัส DISK เบอร์ :'+dk+' ลงใน DRIVE [B:]'
      @ 10,12 say ' Insert DISK No. '+dk+' into DRIVE [B:]'
      @ 11,12 say '   '
      @ 12,12 say '   [C] ทำงานจนต่อไป Continue'
      @ 13,12 say '   [E] ออกไป MAIN MENU'
      @ 14,12 say '   '
      @ 15,12 say ' ขอเลือก: Select ==>' get cn pict '!'

    read
    do case
      case cn = 'C' .or. cn = 'c'
        if .not.file("b:disk.dbf")           && Check disk is data disk?
          set color to /w
          @ 13,35 clear to 15,54
          @ 13,36 say '   '
          @ 14,36 say '   รหัส DISK ผิดเบอร์ '

```

```

@ 15,36 say 'Insert disk wrong'
set color to
loop && (3)                && Loop when disk not correct
endif

use b:disk new
store dnbr to dn
if dn = dk && (IF 3)        && Check disk number of correct
@ 2,30 say 'DISK # : '+dk
select 1
use b:stout
select 2
use b:stock
locate for kcode=code      && Find code in stock
if found() && (IF 4)        && Check code is found?
    xq1 =q1
    xq2 =q2
    xq3 =q3
    xq4 =q4
    xp1 =p1
    xp2 =p2
    xp3 =p3
    xp4 =p4
    xjp1=jp1
    xjp2=jp2
    xjp3=jp3
    xjp4=jp4

select 1
append blank
qt = 0
day=ctod(" / / ")
@ 7,10 clear to 18,55
do box_ with 1, 2, 18, 79 && (1) && Display issue
@ 6, 20 say kcode          && menu
@ 8, 6 say 'DATE'
@ 10, 6 say "JOB_NBR"
@ 12, 6 say "INVOICE No."
@ 13, 6 say ' '
@ 14, 6 say "รานส่งออก Issue to"
@ 16, 6 say 'Comment'
@ 8, 40 say "จำนวน QTY"
@ 5, 40 say ' '
@ 6, 40 say 'จำนวนสินค้าที่มีอยู่ STOCK : '+str(xq1+xq2+xq3+xq4,5)
@ 7, 35 say ' '
@ 11, 35 say ' '
@ 12, 35 say "จำนวนเงินทั้งหมด Total"
@ 8, 20 get day..
set color to /w
@ 8, 20 say 'mm/dd/yy'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@ 16, 25 say ' '
@ 12, 57 say ' . '
set color to
@ 10, 20 get job_nbr pict '@!'
@ 12, 20 get inv_nbr pict '@!'
@ 14, 25 get issued pict '@!'

do while .t.
  @ 8, 52 get qt pict '@9'
  read
  if day=_ctod(" / / ")      && Check data not input
    return                    && to return
  endif

  if qt > (xq1+xq2+xq3+xq4)    && Check qty request more
    set color to +w           && in stock?
    @ 9,42 say ' '
    @ 10,42 say 'จำนวนที่ส่ง มากกว่าที่ใน STOCK'
    set color to
    loop
  endif
  @ 9,42 say '
  @ 10,42 say '
  exit
enddo
tall = 0
xqt = qt
do case
case qt > xq1      && Check get qty request in stock
  tall= tall+(xp1*xq1)+(xjpl*xq1)
  qt = qt-xq1
  xq1 = 0
do case
case qt > xq2
  tall= tall+(xp2*xq2)+(xjp2*xq2)
  qt = qt-xq2
  xq2 = 0
do case
case qt > xq3
  tall= tall+(xp3*xq3)+(xjp3*xq3)
  qt = qt-xq3
  xq3 = 0
do case
case qt > xq4
  tall= tall+(xp3*xq3)+(xjp3*xq3)
  qt = qt-xq3
  xq3 = 0
case qt <= xq4
  xq4 = xq4-qt
  tall=tall+(xp4*qt)+(xjp4*qt)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        qt = 0
        endcase
        case qt <= xq3
            xq3 = xq3-qt
            tall= tall+(xp3*qt)+(xjp3*qt)
            qt = 0
        endcase
        case qt <= xq2
            xq2 = xq2-qt
            tall= tall+(xp2*qt)+(xjp2*qt)
            qt = 0
        endcase
        case qt <= xq1
            xq1 = xq1-qt
            tall= tall+(xp1*qt)+(xjp1*qt)
            qt = 0
        endcase
        set color to /w
        @ 12, 57 say tall pict '99999.99'
        set color to
        @ 16, 25 get stout->comment
        read
        && STOUT
        repl qty_out with xqt,code with kcode,;
        total with tall,day with dtoc(day_)

        select 2
        repl q1 with xq1,q2 with xq2,q3 with xq3,q4 with xq4

        endif && (IF 4)
        exit && (3)
    else
        set color to /w
        @ 13,35 clear to 15,54
        @ 13,36 say ' '
        @ 14,36 say 'ใส่ DISK ผิดเบอร์ '
        @ 15,36 say 'Insert disk wrong'
        set color to

        loop && (3)
        endif && (IF 3)
        case cn = 'E' .or. cn = 'e'
            return && (3)
        endcase
    enddo && (3)
else && (IF 2)
    && Display not found code in table
    cn = 'Y'
    @ 10,10 clear to 15,65
    do box_ with 9,10, 15,65
    @ 10,12 say ' '

```

```

@ 11,12 say '      โหมม รหัสสินค้า นอขยำน STOCK '
@ 12,12 say '      '
@ 13,12 say '      กลับไปยัง MAIN MENU เพื่อ สร้างรหัส สินค้าใหม่ '
@ 14,12 say 'Find not founhd this code ! Return to MAIN MENU.'
do while inkey()=0
enddo
exit   && (2)
endif && (IF 2 )
exit && (2)
enddo && (2)

return
*****
*!*****
*!
*!      Procedure: DELCODE
*!
*!      Called by: MAIN_NOR.PRG
*!
*!      Calls: BOX_      (procedure in MAIN_NOR.PRG)
*!              : SHOW_TABLE      (procedure in MAIN_NOR.PRG)
*!
*!      Uses: TABLE1.DBF
*!              : B:DISK.DBF
*!              : B:STOCK.DBF
*!              : B:STIN.DBF
*!              : B:STOUT.DBF
*!
*!      Indexes: TABLE1.NDX
*!
*!*****
procedure delcode
clear
@ 2, 9 say '      '
@ 3, 9 say '      ลบ รหัส สินค้า DELETE CODE'
do box_ with 1, 2 , 17, 79
do box_ with 4, 3 , 16, 78
@ 10, 25 say 'Press F1 for help'
@ 11, 25 say '      '
@ 12, 25 say 'กด key F1 เพื่อลบตง code'

cx = '
xx = '
set key K_F1 to show_table
@ 7, 05 say '      '
@ 8, 05 say 'รหัสสินค้า Code No. ' get cx pict '@!' && Read code for delete
read
set key K_F1
if cx=' '
return

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

endif
use tabel1 new
index on code to tabel1
seek cx          && Find code from table
if .not.found()
  @ 10, 20 clear to 18, 75
  do box_ with 10, 20, 18, 75
  @ 12, 30 say '          . . . . . ๕'
  @ 13, 30 say "          ไม่พบรหัสสินค้า FIND NOT FOUND THIS CODE"

  @ 14, 30 say '          . . . . . ๕'
  @ 15, 30 say 'กดปุ่มใด ๆ เพื่อ ยกเลิก          Any key to EXIT'
  @ 16, 30 say '          '

  do while inkey()=0
  enddo
  return
endif
dn=disk_nbr          && If found code insert data disk
do while .t.
  @ 6, 20 clear to 14, 75
  do box_ with 7, 20 ,11, 75
  @ 8, 30 say 'Insert DISK No. '+dn+' into drive B:'
  @ 9, 30 say '          '
  @ 10, 30 say 'ใส่ DISK เบอร์ :'+dn+' ลงใน DRIVE [B:]'
  do while .t.
    key=0
    do while key=0
      key=inkey()
    enddo
    do case
      case key=27
        return
      endcase
    exit
  enddo

  if .not.file("b:disk.dbf")          && Check data disk is correct
    set color to /w
    @ 13,35 clear to 15,54
    @ 13,36 say '          . . . . . ๕'
    @ 14,36 say 'ใส่ DISK ผิดเบอร์ '
    @ 15,36 say 'Insert disk wrong'
    set color to

  loop
endif

```

```

use b:disk new
store dnbr to xdn
if xdn = dn
  use b:stock new  && [B:]
  locate for code = cx          && Find code from stock
  if q1+q2+q3+q4 > 0          && Check stock is empty?
    @ 7, 20 clear to 13, 75
    do box_ with 7, 20 ,13, 75
    @ 8, 30 say '      '
    @ 9, 30 say ' ยงมสคอาขาน STOCK ไมสามารถ ลบรหัสคานาค '
    @ 10, 30 say '      '
    @ 11, 30 say "Stock not empty, Can't delete this code. "
    do while inkey()=0
    enddo
    return
  endif
  @ 7, 20 clear to 13, 75
  @ 7, 05 say '      '
  @ 8, 05 say 'รหัสสินค้า Code No. ' +cx
  @ 13,05 say '      '
  @ 14,05 say ' กดตกลงแล้วหรือไม Are you sure? [Y/N]'
  @ 15,05 say '      '
  do while .t.
    key=0
    do while key=0
      key=inkey()
    enddo
    do case
      && Confirm for make sure to delete
      case key=89.or.key=121 && Press key 'Y'
        @ 7, 20 clear to 11, 75
        do box_ with 7, 20, 11, 75
        @ 8, 30 say '      '
        @ 9, 30 say "   โปรด รอสักคร.....Wait a minute"
        @ 10, 30 say '      '

        delete all for code=cx && Delete code in data base
        pack
        if file("B:STIN.DBF")
          use b:stin new
          delete all for code=cx
          pack
        endif
        if file("B:STOUT.DBF")
          use b:stout new
          delete all for code=cx
          pack
        endif
        use tabel new
        delete all for code=cx

```

```

pack
exit
case key=78.or.key=110  && N
return
endcase
enddo
exit
else
xx= ' ' && If data disk not correct the number
@ 6, 22 clear to 14, 73
do box_ with 6, 22 , 14, 73
@ 7, 30 say ' '
@ 8, 30 say " DISK เบอร์ "+xdn+" ใช้งานได้"
@ 9, 30 say ' '
@ 10, 30 say "ใส่ DISK เบอร์ ':'+dn+' ลงใน DRIVE [B:]' get xx pict '!'"
@ 11,30 say ' '
@ 12,30 say "ปิดบม X เพื่อยกเลิก หรือ บมใด ๆ เพื่อทำงานต่อ"
@ 13,30 say ' '
read
if xx="X"
return
endif
use
loop
endif
enddo
set color to /w
@ 7, 20 clear to 10, 75
do box_ with 7, 20, 10, 75
@ 8, 30 say ' '
@ 9, 30 say "การทำงานเสร็จ....Successfully"
set color to
key=0
do while key=0
key=inkey()
enddo
return
*****
*!*****
*!
*! Procedure: CRCODE
*!
*! Called by: MAIN_NOR.PRG
*!
*! Calls: BOX_ (procedure in MAIN_NOR.PRG)
*! : SHOW_TABLE (procedure in MAIN_NOR.PRG)
*!
*! Uses: TABLE1.DBF
*! : B:DISK.DBF
*! : B:STOCK.DBF

```

```

*!
*!      Indexes: TABLE1.NDX
*!
*!*****
procedure crcode
ch = 'X'
clear                                && Display add new code menu
@ 2, 9 say ' '
@ 3, 9 say "พิมพ์ รหัส พิมพ์ ADD NEW CODE"
do box_ with 1, 2 , 17, 79 && (1)
do box_ with 4, 3 , 16, 78 && (7)
xx = ' '
dn = ' '
cx = ' '
na = ' '
co = ' '
st = ' '
do while .t.
@ 13,05 say ' '
@ 14,05 say ' '
@ 15,05 say ' '
@ 6, 50 say 'Press F1 for help'
@ 7, 50 say ' '
@ 8, 50 say 'กด key F1 พิมพ์ code'

set key K_F1 to show_table
@ 5, 05 say ' '
@ 6, 05 say 'เบอร์ของ DISK No. ' get dn pict '@9'
@ 7, 05 say ' '
@ 8, 05 say 'รหัสสินค้า Code No. ' get cx pict '@!'
@ 9, 05 say ' '
@ 10,05 say 'ชื่อสินค้า Commodity ' get na pict '@!'
@ 12,05 say 'หมายเหตุ Comment ' get co pict '@!'
@ 13,05 say ' '
read                                && read data of the new code
set key K_F1
if dn=' '
return
endif
@ 13,05 say ' '
@ 14,05 say ' กดตกลงแล้วใช่ไหม Are you sure? [Y/N]'
@ 15,05 say ' '

do while .t.
key=0
do while key=0
key=inkey()
enddo
do case
&& Confirm for make sure to write a new code
case key=89.or.key=121 && Press key 'Y'

```

```

        key=0
        exit
    case key=78.or.key=110    && Press key 'N'
        exit
    endcase
enddo
    if key=0
        exit
    endif
loop
enddo

@ 7, 20 clear to 11, 75
do box_ with 7, 20, 11, 75
@ 8, 30 say ' '
@ 9, 30 say "   โปรด รอสักครู่.....Wait a minute"
@ 10, 30 say ' '
use tabel new
index on code to tabel
seek cx    && Check code is already in stock?
if found()
    set color to /w
    @ 6, 20 clear to 13, 75
    @ 7, 30 say ' '
    @ 8, 30 say "มีรหัสสินค้าอยู่แล้ว ...."
    @ 9, 30 say ' '    This code is already..
    @ 10, 30 say ' '
    @ 11, 30 say 'กดปุ่มใด ๆ เพื่อ ยกเลิก    Any key to EXIT'
    @ 12, 30 say ' '
    set color to

do while inkey()=0
enddo
return
endif
@ 7, 20 clear to 11, 75
do box_ with 7, 20 ,11, 75
@ 8, 30 say 'Insert DISK No. '+dn+' into drive B:'
@ 9, 30 say ' , '
@ 10, 30 say 'ใส่ DISK เบอร์ :'+dn+' ลงใน DRIVE [B:]'
do while inkey()=0
enddo
do while .t.
    if .not.file("b:disk.dbf")    && Check data disk number?
        set color to /w
        @ 13,35 clear to 15,54
        @ 13,36 say ' , '
        @ 14,36 say 'ใส่ DISK ผิดเบอร์ '
        @ 15,36 say 'Insert disk wrong'
        set color to

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do while .t.
  key=0
  do while key=0
    key=inkey()
  enddo
  do case
  case key=27
    return
  endcase
  exit
enddo
loop
endif

use b:disk new
store dnbr to xdn
if xdn = dn
  use table1 new
  append blank      && Save data to file
  repl code with cx, name with na, comment with co, disk_nbr;
  with dn, status with st
  use b:stock new  && [B:]
  append blank
  repl code with cx
  exit
else
  @ 6, 22 clear to 14, 73      && Display data disk not correct
  do box_ with 6, 22, 14, 73
  @ 7, 30 say ' '
  @ 8, 30 say " DISK เบอร์ "+xdn+" ใช้ไม่ได้"
  @ 9, 30 say ' '
  @ 10, 30 say "ใส่ DISK เบอร์ ':'+dn+' ลงใน DRIVE [B:]' get xx pict '!'"
  @ 11,30 say ' '
  @ 12,30 say "กดปุ่ม X เพื่อยกเลิก หรือ ปุ่มใด ๆ เพื่อทำงานต่อ"
  @ 13,30 say ' '
  read
  if xx="X"
    return
  endif
  use
  loop
endif
enddo

@ 7, 20 clear to 10, 75
do box_ with 7, 20, 10, 75
@ 8, 30 say ' '
@ 9, 30 say "      การทำงานเสร็จ....Successfully"
key=0
do while key=0
  key=inkey()

```

```

enddo
return

```

```

*****
*!*****
*!
*!      Procedure: CRDISK
*!
*!      Called by: MAIN_NOR.PRG
*!
*!      Calls: BOX_          (proceduré in MAIN_NOR.PRG)
*!
*!      Uses: DISK.DBF
*!            : B:DISK.DBF
*!            : STOCK.DBF
*!            : B:STOCK.DBF
*!            : STIN.DBF
*!            : B:STIN.DBF
*!            : STOUT.DBF
*!            : B:STOUT.DBF
*!
*!*****
procedure crdisk
ch = ' '
dn = ' '
@ 5, 37 clear to 16, 79 && (4)
do box_ with 5, 37, 16, 79      && (4)
@ 06, 45 say '      '
@ 07, 45 say '--- สร้าง DATA DISK ---'
@ 8, 46 say ' , '
@ 9, 46 say 'ใส่ DISK ลงใน DRIVE B:'
@ 11, 50 say '      '
@ 12, 50 say 'ต้องการสร้าง DISK เบอร์ 'get dn pict '999'
@ 14, 38 say 'Insert DISK into Drive b:'
@ 15, 61 say 'What is DISK No.?'
read          && Read disk number for make disk
if dn= ' '
return
endif
if file("b:disk.dbf")          && Check disk is data disk already
set color to /w
@ 6, 20 clear to 13, 75
@ 7, 30 say '      '
@ 8, 30 say "มี DISK เบอร์นอยแล้ว ....."
@ 9, 30 say '      ' This disk is already..'
@ 10, 30 say '      '
@ 11, 30 say 'กดแป้นใด ๆ เพื่อ ยกเลิก Any key to EXIT'
@ 12, 30 say '      '
set color to
do while inkey()=0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        enddo
        return
    endif

    set color to /w
    @ 15, 20 clear to 19, 75
    @ 16, 30 say '
    @ 17, 30 say "        โปรด รอสักครู่ .....Wait a minute"
    @ 18, 30 say '
    set color to

    set echo off
    @ 18,1 say ' '
    use disk new                                && Begin make data disk
    copy stru to b:disk.dbf
    use b:disk new
    append blank
    repl dnbr with dn
    use stock new
    copy stru to b:stock
    use stin new
    copy stru to b:stin
    use stout new
    copy stru to b:stout
    use
    @ 15, 20 clear to 19, 75
    do box_ with 15, 20 , 19, 75
    @ 16, 30 say '
    @ 17, 30 say "        การทำงานเสร็จ....Successfully"

    do while inkey()!=0
    enddo

    return
    *****
    *!*****
    *!
    *!      Procedure: FORMAT
    *!
    *!      Called by: MAIN_NOR.PRG
    *!
    *!      Calls: BOX_          (procedure in MAIN_NOR.PRG)
    *!
    *!*****
    procedure format
    clear
    ch = 'Y'
    do box_ with 3,10,10,53
    @ 4, 17 say '--- FORMAT DISK ---'
    @ 5, 17 say ' , , '

```

```

@ 6, 17 say 'ใส่ DISK ลงใน DRIVE B: '
@ 8, 17 say 'Insert DISK into DRIVE B:'
@ 8, 43 say 'ok ? ' get ch pict '!'
read
if ch#'Y'                                && Confirm for need the format new disk
    return
endif
set color to /w
@ 15, 20 clear to 19, 75
@ 16, 30 say '
@ 17, 30 say "   โปรด รอสักคร ..Wait a minute"
@ 18, 30 say '
set color to
clear
set echo off
!format b:                                && execute the dos command
clear
@ 17, 20 clear to 20, 75
do box_ with 17, 20 , 20, 75
@ 18, 30 say '
@ 19, 30 say "   การทำงานเสร็จ...Successfully"
key=0
do while key=0
    key=inkey()
enddo
return
*****
*!*****
*!
*! Procedure: MATERIAL
*!
*! Called by: MAIN_NOR.PRG
*!
*! Calls: BOX_ (procedure in MAIN_NOR.PRG)
*!
*! Uses: TABLE1.DBF
*!       : B:DISK.DBF
*!       : B:STOUT.DBF
*!       : B:STIN.DBF
*!       : B:STOCK.DBF
*!
*! Indexes: TABLE1.NDX
*!          : STOCK.NDX
*!
*!*****
procedure material
parameter head
set color to /w
@ 9,20 clear to 13,42
do box_ with 9,20 ,- 13,42

```

```

@ 10,24 say '
@ 11,24 say '   โปรด รอสมัคร '
@ 12,24 say 'wait a minute '
set color to
close data

select 1
use table1
index on disk_nbr+code to table1          && Index field for disk number
go top
num=len(head)
set cons off
set alte to temp.txt                      && Open temporary file for write
set alte on                               && report formate

?space(60)+'MATERIAL AND FINISHED PRODUCTS REPORT'   && write head report
?' '+head+space(58-num)+repli('-',37)+space(35)+'DATE : '+str(day(date()),2);
+' '+cmonth(date())+' '+str(year(date()),4)
?repli('-',156)
?'
; PREVIOUS BALANCE ; RECEIVED ; ISSUED ;
; BALANCE ;
?' PARTS No. ; DESCRIPTION ;'+repli('-',95)
?' ; QTY ;
; VALUE ; QTY ; VALUE ; QTY ; VALUE ; QTY ;
; VALUE ;
?repli('-',156)

do while .not.eof() && (1)
select 1
dk=disk_nbr
cn=' '
@ 7,10 clear to 16,63
do while .t. && (2)

do box_ with 7,10, 16,55          && Display insert disk number
@ 8,12 say '
@ 9,12 say '   ใส่ DISK เบอร์ :'+dk+' ลงใน DRIVE [B:]'
@ 10,12 say ' Insert DISK No. '+dk+' into DRIVE [B:]'
@ 11,12 say '   '
@ 12,12 say '   [C] ทำงานจนต่อไป Continue'
@ 13,12 say '   [E]  ออกไป MAIN MENU'
@ 14,12 say '   '
@ 15,12 say '   ขอตเลือก: Select ==>' get cn pict ''

read
do case
case cn = 'C' .or. cn = 'c'
if .not.file("b:disk.dbf")          && Check data disk is correct?

```

```

set color to /w
@ 13,35 clear to 15,54
@ 13,36 say ' .      ^      '
@ 14,36 say 'ใส่ DISK ผิดเบอร์ '
@ 15,36 say 'Insert disk wrong'
set color to
loop
endif
select 5
use b:disk
if dnbr # dk                                && Check disk number is correct
set color to /w
@ 13,35 clear to 15,54
@ 13,36 say ' .      ^      '
@ 14,36 say 'ใส่ DISK ผิดเบอร์ '
@ 15,36 say 'Insert disk wrong'
set color to
loop && (2)
endif
select 4
use b:stout
select 3
use b:stin
select 2
use b:stock
index on code to stock
exit
case cn = 'E' .or. cn = 'e'
set conso on
set alter off                                && Close temporary file
return
endcase
enddo && (2)
set color to */w
@ 7,10 clear to 16,63
@ 10,23 say ' .      .      '
@ 11,23 say ' กำลัง ทำงานพอย รอสักคร '
@ 12,23 say '      '
@ 13,20 say 'wait a minute'
set color to
sele 1 && TABLE1
do while dk=disk_nbr .and. .not. eof()      && (3)
part=code
desc=name
select 3 && STIN
sum qty_in,total to rec_qty,rec_tot for code=part
select 4 && STOUT
sum qty_out,total to iss_qty,iss_tot for code=part
select 2 && STOCK

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

seek part
sto_qty=q1+q2+q3+q4
sto_tot=(q1*(p1+jp1))+(q2*(p2+jp2))+(q3*(p3+jp3))+(q4*(p4+jp4))
pre_qty=qty_bal
pre_tot=tot_bal
? '+part+' ; '+desc+' ; '+str(pre_qty,7)+' ? '+str(pre_tot,11,2);
+ ; '+str(rec_qty,7)+' ; '+str(rec_tot,11,2)+' ; '+str(iss_qty,7);
+ ; '+str(iss_tot,11,2)+' ; '+str(sto_qty,7)+' ; '+str(sto_tot,11,2)

select 1      && TABLE1
skip
enddo      && (3)
enddo      && (1)
?repli('-',156)
?
set console on
set alter off
set color to /w
@ 7,10 clear to 16,63
@ 10,23 say ' '
@ 11,23 say 'ทำงานสำเร็จแล้ว Successfully.'
@ 12,14 say 'v '
@ 13,14 say 'ต้องการจะแสดงผลทาง display ทาง printer ? [D/P]'
@ 15,14 say 'You want to Display or Printer ? [D/P]'
set color to
@ 23,1 say ''
do while .t.      && (5)
key= 0
do while key=0
key=inkey()
enddo
do case
case key=68.or.key=100      && D
set alter to
!browse temp.txt      && Execute file for display
exit
case key=80.or.key=112      && P
if err_prn()
set console off
set print on
close alter
?chr(15)
type temp.txt
?chr(18)
?eject

set print off
set console on
endif
exit

```

```

endcase
enddo      && (5)
set alter to
close alter
close data
return
*****
*!*****
*!
*!      Procedure: STOCK_RE
*!
*!      Called by: MAIN_NOR.PRG
*!
*!      Calls: BOX_          (procedure in MAIN_NOR.PRG)
*!
*!      Uses: TABLE1.DBF
*!            : TEMP.DBF
*!            : B:DISK.DBF
*!            : B:STOCK.DBF
*!            : BIN.DBF
*!            : B:STIN.DBF
*!            : B:STOUT.DBF
*!
*!      Indexes: TABLE1.NDX
*!               : STOCK.NDX
*!               : BIN.NDX
*!
*!*****
procedure stock_re
parameter head
set color to /w
@ 9,20 clear to 13,42
do box_ with 9,20, 13,42
@ 10,24 say '
@ 11,24 say '      โปรด รอสักคร '
@ 12,24 say 'wait a minute '
set color to

use table1 new
index on disk_nbr to table1
total to temp on disk_nbr          && Find number of data disk
sele 2
use temp
go top
num=len(head)
set cons off
set alte to temp.txt              && Open temporary file
set alte on
?space(100)+'STOCK BIN CARDS REPORT' && Write head of report
?' '+head+space(97-num)+repli('=' ,22)+space(75)+'DATE : '+str(day(date()),2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

loop
endif
sele 1
use b:stock
index on code to stock

exit
case cn = 'E' .or. cn = 'e'
set alter to
set conso on
set alter off
return
endcase
enddo      && (2)
set color to */w
@ 7,10 clear to 16,63
@ 10,23 say ' '
@ 11,23 say ' ภาลง หนางานอย รอดคคร '
@ 12,23 say ' '
@ 14,23 say 'Wait a minute'
set color to
sele 3
use bin
dele all
pack
append from b:stin      && Write data to temporary data base
append from b:stout
index on code+day to bin
go top
do while .not. eof()      && (3)
sele 3      && BIN
l_code=code
sele 1      && STOCK
seek l_code
bal_qty=qty_bal
bal_tot=tot_bal
sele 3      && BIN
do while code = l_code .and. .not.eof()      && (4)
? ' '+day+ ' | '+code+ ' | '+job_nbr+ ' | '+receive+ ' | ' ;
+iif(qty_in>0,inv_nbr,spac(8))+ ' | '+str(qty_in,5)+ ' | ' ;
+str(price_ave,7,2)+ ' | '+str(job_pay,7,2)+ ' | ' ;
+str(iif(qty_in>0,total,0),8,2)+ ' | '
??issued+ ' | '+iif(qty_out>0,inv_nbr,spac(8))+ ' | '+str(qty_out,5);
+ ' | '+str(iif(qty_out>0,total/qty_out,0),7,2)+ ' | ' ;
+str(iif(qty_out>0,total,0),8,2)+ ' | '
bal_qty=bal_qty-qty_out+qty_in
bal_tot=bal_tot+iif(qty_in>0,total,-total)
??str(bal_qty,7)+ ' | '+str(bal_tot,11,2)
skip
enddo      && (4)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

?repli('-',222)
  sele 3  && BIN
enddo    && (3)
  sele 2  && TEMP
  skip
enddo    && (1)
?
set conso on
set alter off
set color to /w
@ 7,10 clear to 16,63
@ 10,23 say '          '
@ 11,23 say 'ทำงานสำเร็จแล้ว Successfully.'
@ 12,14 say '          '
@ 13,14 say 'ต้องการจะแสดงผลทาง display หรือ printer ? [D/P]'
@ 15,14 say 'You want to Display or Printer ? [D/P]'
set color to

@ 23,0 say ''
do while .t.  && (5)
  key= 0
  do while key=0
    key=inkey()
  enddo
  do case
  case key=68.or.key=100  && D
    set alter to
    !browse temp.txt
    exit
  case key=80.or.key=112  && P
    if err_prn()
      set print on
      close alter
      ?chr(15)
      type temp.txt
      ?chr(18)
      eject
      set print off
      set cons on
    endif
    exit
  endcase
enddo    && (5)
set alter to
close alter
close data
return
*****
*!*****
*!

```

```

*!      Procedure: STOCK_CL
*!
*!      Called by: MAIN_NOR.PRG
*!
*!      Calls: BOX_          (procedure in MAIN_NOR.PRG)
*!
*!      Uses: TABLE1.DBF
*!            : TEMP.DBF
*!            : B:DISK.DBF
*!            : B:STOCK.DBF
*!            : BIN.DBF
*!            : B:STIN.DBF
*!            : B:STOUT.DBF
*!
*!      Indexes: TABLE1.NDX
*!            : STOCK.NDX
*!            : BIN.NDX
*!
*!*****
procedure stock_cl
@ 18,20 say 'Enter PASSWORD'
set color to ,bn/bn
pass=' '
@ 18,35 get pass          && Read password for access
read
set color to
if upper(pass)='SUPERMAN' && Compare password is correct
return
endif

&& Check status printer
if !err_prn()
return
endif

set color to /w
@ 9,20 clear to 13,42
do box_ with 9,20, 13,42
@ 10,24 say '
@ 11,24 say '   โปรด รอสักคร '
@ 12,24 say 'wait a minute   '

set color to
use table1 new
index on disk_nbr to table1
total to temp on disk_nbr          && FInd number of disk in stock
use
sele 4
use table1
index on code to table1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sele 2
use temp
go top
page=1
set cons off
set print on

do while .not. eof()   && (1)
  dk=disk_nbr
  cn=' '
  @ 7,10 clear to 16,55
  do while .t.   && (2)

    do box_ with 7,10, 16,55
    @ 8,12 say '          '
    @ 9,12 say '      ใส่ DISK เบอร์ :'+dk+' ลงใน DRIVE [B:]'
    @ 10,12 say ' Insert DISK No. '+dk+' into DRIVE [B:]'
    @ 11,12 say '          '
    @ 12,12 say '      [C] ทำงานจนต่อไป Continue'
    @ 13,12 say '      [E]  ออกไป MAIN MENU'
    @ 14,12 say '          '
    @ 15,12 say '  ยกดเลือก: Select ==>' get cn pict '! '

  read
  do case
  case cn = 'C' .or. cn = 'c'
    if .not.file("b:disk.dbf")
      set color to /w
      @ 13,35 clear to 15,54
      @ 13,36 say '          '
      @ 14,36 say 'ใส่ DISK ผิดเบอร์ '
      @ 15,36 say 'Insert disk wrong'
      set color to
      loop
    endif
  select 5
  use b:disk new
  if dnbr # dk
    set color to /w
    @ 13,35 clear to 15,54
    @ 13,36 say '          '
    @ 14,36 say 'ใส่ DISK ผิดเบอร์ '
    @ 15,36 say 'Insert disk wrong'
    set color to

  loop
  endif
sele 1
use b:stock
index on code to stock

```

```

exit
case cn = 'E' .or. cn = 'e'
  set print off
  set alter to
  set conso on
  set alter off
  return
endcase
enddo      && (2)

set color to */w
@ 8,11 clear to 15,54
@ 10,23 say '      '
@ 11,23 say ' ภาลง ทางการอย รอดศกร '
@ 12,23 say '      '
@ 14,23 say 'Wait a minute'

set color to
sele 3
use bin
dele all
pack
append from b:stin      && Write data to temporary data base
append from b:stout
index on code+day to bin
go top
do while .not. eof()      && (3)
  sele 3      && BIN
  l_code=code
  sele 1      && STOCK
  seek l_code
  bal_qty=qty_bal
  bal_tot=tot_bal
  sto_qty=q1+q2+q3+q4
  sto_tot=(q1*(p1+jp1))+(q2*(p2+jp2))+(q3*(p3+jp3))+(q4*(p4+jp4))
  repl qty_bal with sto_qty,tot_bal with sto_tot
  sele 4      && TABLE1
  seek l_code
  name_=name
  sele 3      && BIN

?chr(15)+space(100)+'STOCK BIN CARDS REPORT'+space(74)+'Page  ';
  +str(page,2)
?space(100)+repli('=' ,22)
?space(10)+'COMMODITY NAME : '+name_+space(10)+'STOCK No. : ';
  +l_code+space(60)+'DATE : '+str(day(date()),2)+' '+cmonth(date());
  + ' '+str(year(date()),4)
?repli('-',204)
?' DATE      ;      ;

```

บทสรุป

โครงการวิจัยนี้เป็นการพัฒนา Package Software dBASE ซึ่งเป็นโปรแกรมที่ใช้ Manage Database ต่างๆซึ่งสามารถเก็บข้อมูล เพิ่มข้อมูล ลบข้อมูล หรือจะปรับปรุงเปลี่ยนแปลงข้อมูลต่างๆได้ง่าย และสามารถนำข้อมูลมาใช้งานทางด้านงานเอกสารต่างๆ เช่น จดหมายเวียน ใบสิ่งสินค้า ใบเสร็จ ใบรับ-ส่งสินค้า ใบแสดงรายการสินค้า รายชื่อพนักงาน ซึ่งสามารถเรียกดูได้ และนำมาใช้ได้สะดวกและยังทำให้การทำงานของหน่วยงานมีระบบและทันต่อเหตุการณ์ในปัจจุบันอีกด้วย

หลักสำคัญในการออกแบบโปรแกรมนี้คือต้องการให้ประหยัดในการใช้ Hardware มากที่สุดเท่าที่จะทำได้และส่วนของโปรแกรมนี้สามารถนำไปใช้กับ Package Software อื่นๆได้อีก เช่น Lotus Clipper Foxbase ซึ่งสามารถประหยัด เนื้อที่ในการเก็บโปรแกรม นอกจากนี้ยังพัฒนาไปใช้ในระบบ Multi user ในการ Share Database กันระหว่างหน่วยงานได้อีกด้วย

บทวิจารณ์

สำหรับโปรแกรมนี้ สามารถจะพัฒนาต่อไปได้อีกเพื่อให้สามารถใช้งานได้อย่างมีประสิทธิภาพให้ดียิ่งขึ้น ในการเก็บข้อมูลที่มีความซับซ้อนมากกว่านี้และใช้เป็นระบบ Multiuser ซึ่งเมื่อข้อมูลอยู่ในระบบแล้วจะต้องคำนึงถึงการใช้งานที่พร้อมๆกันหลาย user ด้วยและถ้าเป็นระบบข้อมูลที่เป็นความลับ (Confidential Data) หรือเป็นข้อมูลที่สำคัญมากควรจะต้องออกแบบระบบรักษาความปลอดภัย (Security System) และการสำรองข้อมูลเอาไว้ด้วย (Backup) พร้อมทั้งกำหนดผู้ใช้แต่ละคนว่าจะสามารถทำอะไรกับข้อมูลได้บ้าง (Privilage Assign)



กิตติกรรมประกาศ

โครงการนี้ได้จัดทำขึ้นเพื่อสนับสนุนการนำเครื่องคอมพิวเตอร์ส่วนบุคคลมาใช้งานให้เกิดประโยชน์สูงสุด ซึ่งได้รับความช่วยเหลือจากทางหลายบริษัทที่ให้การสนับสนุนโครงการในด้านข้อมูลเกี่ยวกับเรื่องต่างๆคือ บริษัท Dapper จำกัด ได้ให้ความช่วยเหลือทางข้อมูลและรายละเอียดทางการตลาด การจัดเก็บ Stock ของสินค้า พร้อมทั้งข้อมูลในการทดลองครั้งนี้ด้วย และอีกบริษัท คือ บริษัท National (Semiconductor) Bangkok โดย คุณ วราวุธ สุภักดีโนดม (Cim/Mis Manager) ได้ให้ความช่วยเหลือทางการวางแผนทาง Databaseและการพัฒนาทางด้าน LAN และ คุณ มุกดา อองคสุวรรณ (System Analysis) ได้ช่วยทางด้านการวิเคราะห์ระบบและทางด้าน Relational Database ทางคณะผู้จัดทำขอขอบพระคุณมา ณ ที่นี้ เป็นอย่างสูง