

บทที่ A

กรณีศึกษาระบบงานขาย และสินค้าคงคลัง

จุดประสงค์เชิงพฤติกรรม

1. เข้าใจระบบขาย และสินค้าคงคลังในมุมมองของ Programmer มากขึ้น
2. เข้าใจการสร้างระบบงานคอมพิวเตอร์ โดยพิจารณาจากความต้องการของผู้ใช้
3. มีประสบการณ์ และมองเห็นแนวการออกแบบเมนูทั้งระบบงาน
4. เข้าใจแนวการเขียน และการแปลโปรแกรมในอีกรูปแบบหนึ่ง
5. เข้าใจการทำรายงาน และการทำซ้ำจากกรณีศึกษานี้มากขึ้น

หัวข้อในบทเรียน

- 6.1 ศึกษาความต้องการ
- 6.2 โครงสร้างข้อมูล
- 6.3 วิเคราะห์ระบบงานเบื้องต้น
- 6.4 โปรแกรมในระบบงานขาย และสินค้าคงคลัง
 - 6.4.1 โปรแกรมต่าง ๆ ใน MENU.PRG : สารบัญของโปรแกรม
 - 6.4.2 โปรแกรมต่าง ๆ ใน SALE1.PRG : รวมการขายหลายรูปแบบ
 - 6.4.3 โปรแกรมต่าง ๆ ใน SALE2.PRG : รวมโปรแกรมด้านการปรับปรุงข้อมูล
 - 6.4.4 โปรแกรมต่าง ๆ ใน SALE3.PRG : รวมโปรแกรมรายงานยอดขาย
 - 6.4.5 โปรแกรมต่าง ๆ ใน SALE4.PRG : รวมโปรแกรมด้านสินค้าคงคลัง
 - 6.4.6 โปรแกรมต่าง ๆ ใน SALE5.PRG : รวมรายงานอื่น ๆ ที่เกิดขึ้น

ตัวอย่างโปรแกรมเกี่ยวกับระบบงานขาย และสินค้าคงคลัง 37 ตัวอย่าง

หมายเหตุ

ตัวอย่างนี้อาจขัดกับความเป็นจริงอยู่มาก เพราะโปรแกรมขาย หรือสินค้าคงคลังนั้น จะมีความซับซ้อนอย่างมาก ถึงมากที่สุด กรณีศึกษานี้เกิดจากประสบการณ์ในการเป็นลูกค้าเป็นหลักใหญ่ ซึ่งต้องการสื่อให้เห็นแนวการเขียนโปรแกรมด้านการปรับปรุง การทำรายงานให้มีความหลากหลาย และซับซ้อนที่แตกต่างกันไป .. หากจะนำไปใช้จริง ต้องศึกษาความต้องการของผู้ที่จะนำไปใช้ และพัฒนาต่ออีกมาก และโปรแกรมทั้งหมดในหนังสือ นักศึกษาสามารถ Download ไปศึกษา ประกอบการเรียนภาคปฏิบัติ

บทที่ 6 กรณีศึกษาระบบงานขาย และสินค้าคงคลัง

เนื่องจากสังคมทุกวันนี้ เป็นสังคมของนายทุน มีการซื้อขายแลกเปลี่ยนในสังคมตลอดเวลา ธุรกิจที่หวังผลกำไรจึงอยู่ได้ด้วยการขาย และการจัดเก็บสินค้าไว้ในคลัง เพื่อให้การขายได้เงินมาอย่างไม่ตกหล่น และการจัดเก็บสินค้ามีประสิทธิภาพไม่มากไม่น้อยเกินไปนัก การค้ายุคใหม่จึงนำคอมพิวเตอร์มาใช้ในหน่วยงานกันทั่วหน้า

โปรแกรมคอมพิวเตอร์ที่มีขายอยู่ในท้องตลาด ถูกเขียนขึ้นให้ครอบคลุมกับงานได้ในหลาย ๆ ลักษณะมีความยืดหยุ่นสูง แต่มีบางบริษัทที่มีรูปแบบ และความต้องการเฉพาะของตน จึงไม่อาจนำโปรแกรมที่มีขายในท้องตลาดมาใช้ได้อย่างเต็มที่ บริษัทเหล่านั้นมักจ้างนักเขียนโปรแกรม มาพัฒนาระบบคอมพิวเตอร์ให้ตน ตามความต้องการที่ไม่จำกัด

& 6.1 ศึกษาความต้องการ

ผู้เขียนไม่ได้เป็นพ่อค้าโดยอาชีพ จึงศึกษาระบบนี้โดยอาศัยประสบการณ์ตรงของการเป็นผู้ซื้อมาพัฒนาระบบ โดยมองตามความน่าจะเป็น ในเรื่องของงานขาย การจัดทำรายงานตลอดจนการควบคุมสินค้าคงคลัง และจะเน้นการขายส่ง โดยไม่มีการขายหน้าร้าน นั่นคือรับคำสั่งซื้อทางโทรศัพท์ จดหมาย หรือโทรสาร เพื่อให้เข้าใจระบบได้อย่างง่าย ๆ

6.1.1 บุคคลที่เกี่ยวข้องกับงาน

ลำดับ	ผู้เกี่ยวข้อง	ความสัมพันธ์ต่อข้อมูล
1	ลูกค้า	สั่งซื้ออยู่ที่บ้าน หรือที่ทำงาน
2	ลูกจ้าง หรือพนักงานขาย	ป้อนข้อมูลเข้าเครื่องเมื่อมีการสั่งซื้อ หรือฝ่ายอื่น ๆ
3	เจ้าของกิจการ	รับรายงานการขาย ต่าง ๆ
4	ผู้ผลิตสินค้า	ส่งสินค้าเข้าร้าน
5	บริษัทรับส่งของ	เพราะบริษัทนี้ไม่ส่งของตัวเอง แต่จ้างบริษัทรับส่งของมาช่วยรับภาระ

6.1.2 ระบบงานที่เป็นอยู่ และสิ่งที่ผู้ใช้ต้องการ

ผู้เขียนต้องการให้ระบบนี้ เป็นตัวอย่างการพัฒนาระบบที่ทำให้ผู้อ่านเข้าใจได้ง่าย ๆ จึงควบคุมขอบเขตของงานให้อยู่ในเรื่องการขายส่ง โดยซื้อสินค้าจากผู้ผลิต รอรับใบสั่งซื้อ แล้วส่งสินค้าให้ลูกค้าโดย จ้างบริษัทขนส่งมารับผิดชอบ และระบบนี้จะไม่มองถึงหนี้สิน เพราะความคิดที่ว่าลูกค้าทุกคนเป็นคนดี เมื่อรับของแล้วจะส่งเงินเข้าบัญชีของบริษัททันที และไม่มีปัญหาการส่งคืนสินค้า หรือเช็คค้างแน่นอน

ผู้เขียนได้จำลองงานต่าง ๆ ที่ระบบนี้ควรมี เป็นข้อ ๆ ด้วยภาษาของผู้ใช้ที่ไม่มีความรู้ในคอมพิวเตอร์และการเขียนโปรแกรมมากเท่าใด จึงได้รายการของความต้องการดังต่อไปนี้¹³

¹³ ผู้เขียนเคยอ่านหนังสือ 3 เล่ม คือ การประยุกต์คอมพิวเตอร์ด้านการเงิน และการบัญชี ประกอบด้วยระบบบัญชีสินค้าคงคลัง ระบบบัญชีเงินเดือน และระบบบัญชีแยกประเภท ของบริษัท คอมพิวเตอร์เทคโนโลยี จำกัด พ.ศ.2538 ซึ่งมีหลายส่วนที่มีอิทธิพลต่อการออกแบบระบบกรณีศึกษาในบทนี้

ความต้องการของผู้ใช้

1. ขาย ด้วยตัวแปรชุดเดียว
2. ขาย ด้วยตัวแปรหลายชุด
3. ขาย ด้วยตัวแปรอาเรย์
4. พิมพ์รายงานการขาย ตามวันที่
5. พิมพ์สรุปยอดขายของใบสั่งสินค้า ตามวันที่
6. พิมพ์ข้อมูลใบสั่งสินค้า ตามรหัสพนักงานขาย
7. พิมพ์รายการสินค้าที่ ถึงจุดสั่งซื้อ เรียงตามรหัสสินค้า
8. พิมพ์สรุปยอดขายสินค้าแต่ละชนิด ตามเดือน
9. พิมพ์สรุปยอดขายสินค้าแต่ละประเภท ตามเดือน
10. ปรับปรุงแฟ้มต่าง ๆ อย่างง่าย ๆ
11. พิมพ์ข้อมูลจากแฟ้มอย่างง่าย ๆ
12. ยกเลิกใบสั่งสินค้า
13. ยกเลิกสินค้าในใบสั่งสินค้าบางรายการ
14. ปรับปรุงใบสั่งสินค้าให้ถูกต้อง
15. พิมพ์ข้อมูลการขายสินค้าในรอบเดือน ตามประเภทสินค้า
16. พิมพ์สรุปค่าขนส่งให้บริษัทขนส่ง ตามบริษัทขนส่ง
17. พิมพ์สรุปยอดขายสินค้า ตามผู้ผลิตสินค้า
18. พิมพ์สรุปยอดขายสินค้า ตามผู้ผลิตสินค้า และเดือน
19. สั่งเพิ่มเงินเดือนพนักงานทุกคน ตามเปอร์เซ็นต์ที่ต้องการ
20. พิมพ์สถิติการใช้บริการบริษัทขนส่งของลูกค้าแต่ละราย ตามช่วงวันที่
21. พิมพ์รายการสินค้าที่ขายดี จากมากไปน้อย ตามช่วงวันที่
22. พิมพ์ใบสั่งสินค้าที่ ส่งไม่ทันกำหนด ตามเดือน
23. พิมพ์จดหมายขอโทษ ที่ส่งสินค้าไม่ทันตามกำหนด ตามเลขที่ใบสั่งสินค้าที่ต้องการ
24. พิมพ์รายการสินค้าที่ เหลือไม่ถึง 10 เปอร์เซ็นต์ จะถึงจุดสั่งซื้อ
25. พิมพ์รายชื่อลูกค้า ที่พนักงานแต่ละคนรับผิดชอบ ตามช่วงวันที่
26. เพิ่มยอดสินค้าเข้ามาในคลังสินค้า
27. รับสินค้ารายการใหม่เข้ามาในคลังสินค้า
28. พิมพ์รายการสินค้าที่กำลังสั่งซื้อ เพิ่มปัจจุบัน
29. พิมพ์เลขที่ใบสั่งสินค้า ที่มีกำหนดส่งไม่เกิน 10 วันจากวันนี้
30. พิมพ์รายการสินค้าที่ถึงจุดสั่งซื้อ ตามผู้ผลิตสินค้า และยังไม่ได้สั่งซื้อเพิ่ม

& 6.2 โครงสร้างข้อมูล

โครงสร้างข้อมูลต่อไปนี้ ถูกออกแบบให้รองรับงานตามความต้องการข้างต้น โดยแยกเพิ่มออกตามลักษณะเฉพาะของข้อมูล และเชื่อมเพิ่มเข้าด้วยกันด้วยฟิลด์หลักหลาย ๆ ฟิลด์จนเป็นฐานข้อมูลที่เก็บข้อมูลได้ไม่ซ้ำซ้อน มีความกระชับและเรียกข้อมูลมาใช้ร่วมกันได้

เพิ่มที่เกี่ยวข้องกับระบบงานขายและสินค้าคงคลัง

1. เพิ่มสั่งสินค้า ORDER.DBF

แฟ้มดรรชนี	ORDER1.NTX	จัดเรียงตาม	ORDER,CUST,EMPL
1.1 รหัสใบสั่ง		*ORDER	9(5)
1.2 รหัสลูกค้า		*CUST	9(4)
1.3 รหัสพนักงาน		*EMPL	9(4)
1.4 สถานที่จัดส่ง		SHIPADDR	X(40)
1.5 วันที่สั่งสินค้า		DATEORD	X(8)
1.6 วันที่ต้องการสินค้า		DATEREQ	X(8)
1.7 วันที่จัดส่งสินค้า		DATESHIP	X(8)
1.8 บริษัทจัดส่ง		SHIPPER	9(2)
1.9 ค่าขนส่ง		FREIGHT	9(5)V99

ตัวอย่างข้อมูล

```
95001 1009 3804 444 พระบาท อ.เมือง ลำปาง 52000 12/09/95 25/09/95 81 18/09/95
95002 1012 3844 43 บ้านผา อ.เกาะคา ลำปาง 52130 19/09/95 26/09/95 83 30/09/95
95003 1021 3801 21 บ้านทอง อ.เมือง เลย 44000 21/09/95 28/09/95 81 09/10/95
95004 1033 3816 5 ต.คลองคำ อ.เมือง ตาก 63000 25/09/95 06/10/95 82 15/10/95
```

2. เพิ่มสั่งสินค้าละเอียด ORDERD.DBF

แฟ้มดรรชนี	ORDERD1.NTX	จัดเรียงตาม	ORDER,PRO
	ORDERD2.NTX	จัดเรียงตาม	PRO,ORDER
2.1 รหัสใบสั่ง		*ORDER	9(5)
2.2 รหัสสินค้า		*PRO	9(4)
2.3 ราคาขาย		OPRICE	9(5)V99
2.4 ปริมาณขาย		OQUANT	9(4)
2.5 ส่วนลด		ODISC	9(4)V99

ตัวอย่างข้อมูล

```
95001 1105 12.00 20 40.00
95001 1116 200.50 100 50.00
95002 1111 81.00 30 30.00
95002 1121 950.00 10 0.00
```

3. เพิ่มสินค้า PRO.DBF

เพิ่มดรรชนี	PRO1.NTX	จัดเรียงตาม	PRO
3.1 รหัสสินค้า	*PRO		9(4)
3.2 รหัสผู้ผลิต	SUP		9(3)
3.3 รหัสประเภท	CAT		9(2)
3.4 ชื่อสินค้า	PRONAME		X(40)
3.5 หน่วยที่ใช้	PROUNIT		X(10)
3.6 ราคาทุน	PROPRICE		9(5)V99
3.7 จำนวนเหลือ	PROREST		9(4)
3.8 จำนวนที่สั่งซื้อเพิ่ม	PROONORD		9(4)
3.9 จุดสั่งซื้อ	REORDER		9(3)

ตัวอย่างข้อมูล

1101 511 14 ปากกา โหล 25.00 50 0 10
 1102 509 13 กระดาษ รีม 75.00 141 300 150

4. เพิ่มพนักงาน EMPL.DBF

เพิ่มดรรชนี	EMPL1.NTX	จัดเรียงตาม	EMPL
4.1 รหัสพนักงาน	*EMPL		9(4)
4.2 ชื่อพนักงาน	EMPLNAME		X(40)
4.3 ที่อยู่พนักงาน	EMPLADDR		X(40)
4.4 เงินเดือน	EMPLSAL		9(6)V99
4.5 วันเกิด	EMPLBORN		X(8)

ตัวอย่างข้อมูล

3801 นาย สมชาย ดีดี 44 ต.สบสงค์ อ.ด่านเกวียน เลย 48000 6500.00 13/06/69
 3802 น.สงงาน การดี 210 ต.คงคง อ.รักคำ ลำพูน 51000 8900.00 21/01/54
 3803 นาย ดง ไม้ดีก 158 ต.ฐานะ อ.มะระ ลำปาง 52000 10000.00 09/12/65

5. เพิ่มลูกค้า CUST.DBF

เพิ่มดรรชนี	CUST1.NTX	จัดเรียงตาม	CUST
5.1 รหัสลูกค้า	*CUST		9(4)
5.2 ชื่อลูกค้า	CUSTNAME		X(40)
5.3 ที่อยู่ลูกค้า	CUSTADDR		X(40)

ตัวอย่างข้อมูล

1001 นาย ไข่คำ ดงดีก 488 ต.ป่าดง อ.บ้านสัน สมุทรสงคราม 16000
 1002 นาย คำดี มีไชย 12 ต.คงคา อ.บ้านสัน สมุทรสงคราม 16000
 1003 น.ส.เด็กขาว กากา 1000 ต.ฐานะ อ.มะระ ลำปาง 52000

6. เพิ่มบริษัทรับส่งของ SHIPPER.DBF

เพิ่มดรรชนี SHIPPER1.NTX จัดเรียงตาม SHIPPER

- | | | |
|----------------------|----------|-------|
| 6.1 รหัสบริษัทรับส่ง | *SHIPPER | 9(2) |
| 6.2 ชื่อบริษัทรับส่ง | SHIPNAME | X(40) |
- ตัวอย่างข้อมูล
- | | |
|----|----------------------|
| 81 | แสงตะวันขนส่ง |
| 82 | แสงห้านขนส่ง |
| 83 | คำดีร์กับบริการขนส่ง |

7. เพิ่มประเภทสินค้า CAT.DBF

เพิ่มดรรชนี CAT1.NTX จัดเรียงตาม CAT

- | | | |
|----------------------|---------|-------|
| 7.1 รหัสประเภท | *CAT | 9(2) |
| 7.2 ชื่อประเภทสินค้า | CATNAME | X(40) |
- ตัวอย่างข้อมูล
- | | |
|----|--------------|
| 11 | เครื่องดื่ม |
| 12 | เครื่องเขียน |
| 13 | ผักและผลไม้ |

8. เพิ่มผู้ผลิต SUP.DBF

เพิ่มดรรชนี SUP1.NTX จัดเรียงตาม SUP

- | | | | |
|-----------------|---------|-------|-----------------------|
| 8.1 รหัสผู้ผลิต | *SUP | 9(3) | ตัวอย่างข้อมูล |
| 8.2 ชื่อผู้ผลิต | SUPNAME | X(40) | 501 กาแฟเขาช่อง จำกัด |
| | | | 502 ซีพี ประเทศไทย |

& 6.3 วิเคราะห์ระบบงานเบื้องต้น

การวิเคราะห์ระบบงานครั้งนี้ เน้นการจัดทำโปรแกรม จึงศึกษาความต้องการ แล้วนำความต้องการ มาจัดกลุ่มเป็นระบบเล็ก ๆ จากนั้นนำมาเป็นแนวในการจัดทำเมนู เพื่อแบ่งกลุ่มของโปรแกรม ทำให้มองภาพได้ชัดเจนโดยมีความสัมพันธ์โดยตรงมาจาก สิ่งที่ต้องการ

การวิเคราะห์ระบบงานแบบนี้ไม่สามารถนำไปใช้งานจริงได้ เพราะยังขาดสิ่งสำคัญต่าง ๆ อีกมาก เช่นการ ศึกษาความเป็นไปได้ของระบบ การจัดทำผังงาน การศึกษาทิศทางการใช้และการเปลี่ยนแปลงของข้อมูล เป็นต้น

6.3.1 จัดระบบย่อยจากความต้องการของผู้ใช้

ระบบขาย

1. ขาย ด้วยตัวแปรชุดเดียว
2. ขาย ด้วยตัวแปรหลายชุด
3. ขาย ด้วยตัวแปรอาเรย์

ระบบปรับปรุง

1. ปรับปรุงใบสั่งสินค้าให้ถูกต้อง
2. ยกเลิกใบสั่งสินค้า
3. ยกเลิกสินค้าในใบสั่งสินค้าบางรายการ
4. สั่งเพิ่มเงินเดือนพนักงานทุกคน ตามเปอร์เซ็นต์ที่ต้องการ
5. ปรับปรุงแฟ้มต่าง ๆ อย่างง่าย ๆ

ระบบรายงานยอดขาย

1. พิมพ์สรุปยอดขายของใบสั่งสินค้า ตามวันที่
2. พิมพ์สรุปยอดขายสินค้าแต่ละชนิด ตามเดือน
3. พิมพ์สรุปยอดขายสินค้าแต่ละประเภท ตามเดือน
4. พิมพ์สรุปยอดขายสินค้า ตามผู้ผลิตสินค้า
5. พิมพ์สรุปยอดขายสินค้า ตามผู้ผลิตสินค้า และเดือน
6. พิมพ์รายงานการขาย ตามวันที่
7. พิมพ์ข้อมูลใบสั่งสินค้า ตามรหัสพนักงานขาย
8. พิมพ์ข้อมูลการขายสินค้าในรอบเดือน ตามประเภทสินค้า

ระบบควบคุมสินค้าคงคลัง

1. เพิ่มยอดสินค้าเข้ามาในคลังสินค้า
2. รับสินค้ารายการใหม่เข้ามาในคลังสินค้า
3. พิมพ์รายการสินค้าที่ ถึงจุดสั่งซื้อ เรียงตามรหัสสินค้า
4. พิมพ์รายการสินค้าที่ เหลือไม่ถึง 10 เปอร์เซ็นต์ จะถึงจุดสั่งซื้อ
5. พิมพ์รายการสินค้าที่กำลังสั่งซื้อ เพิ่มปัจจุบัน
6. พิมพ์รายการสินค้าที่ถึงจุดสั่งซื้อ ตามผู้ผลิตสินค้า และยังไม่สั่งซื้อเพิ่ม
7. พิมพ์รายการสินค้าที่ขายดี จากมากไปน้อย ตามช่วงวันที่

ระบบรายงานอื่น

1. พิมพ์ข้อมูลจากแฟ้มอย่างง่าย ๆ
2. พิมพ์สรุปค่าขนส่งให้บริษัทขนส่ง ตามบริษัทขนส่ง
3. พิมพ์สถิติการใช้บริการบริษัทขนส่งของลูกค้าแต่ละราย ตามช่วงวันที่
4. พิมพ์ใบสั่งสินค้าที่ ส่งไม่ทันกำหนด ตามเดือน
5. พิมพ์จดหมายขอโทษ ที่ส่งสินค้าไม่ทันตามกำหนด ตามเลขที่ใบสั่งสินค้าที่ต้องการ
6. พิมพ์รายชื่อลูกค้า ที่พนักงานแต่ละคนรับผิดชอบ ตามช่วงวันที่
7. พิมพ์เลขที่ใบสั่งสินค้า ที่มีกำหนดส่งไม่เกิน 10 วันจากวันนี้

6.3.2 ออกแบบเมนูตามระบบย่อย

เมื่อได้มองเห็นกลุ่มงานซึ่งถูกแยกเป็นระบบย่อยแล้ว เพื่อให้การออกแบบเมนูง่ายขึ้น ในกรณีนี้จึงขอออกแบบตัวเลือกตามระบบย่อย แต่ควรแต่งคำพูดให้กระชับขึ้น เพื่อจะได้ไม่ยาวจนเกินไปในการทำเมนู

เมนูหลัก

1. ระบบขาย
2. ระบบปรับปรุง
3. ระบบรายงานยอดขาย
4. ระบบควบคุมสินค้าคงคลัง
5. ระบบรายงานอื่น
6. เลิกการทำงาน

1. เมนูระบบขาย

1. ขาย ด้วยตัวแปรชุดเดียว
2. ขาย ด้วยตัวแปรหลายชุด
3. ขาย ด้วยตัวแปรอาเรย์
4. กลับเมนูหลัก

2. เมนูระบบปรับปรุง

1. ปรับปรุงใบสั่งสินค้า
2. ยกเลิกใบสั่งสินค้าทั้งใบ
3. ยกเลิกสินค้าในใบสั่งสินค้า
4. เพิ่มเงินเดือนพนักงาน
5. ปรับปรุงแฟ้มต่าง ๆ
6. กลับเมนูหลัก

3. ระบบรายงานยอดขาย

1. พิมพ์สรุปรายยอดขายของสินค้า ตามเดือน
2. พิมพ์สรุปรายยอดขายของสินค้า ตามผู้ผลิตสินค้า
3. พิมพ์สรุปรายยอดขายของสินค้า ตามผู้ผลิตสินค้า และเดือน
4. พิมพ์สรุปรายยอดขายของสินค้าแต่ละประเภท ตามเดือน
5. พิมพ์สรุปรายยอดขายของใบสั่งสินค้า ตามวันที่
6. พิมพ์สรุปรายยอดขายของใบสั่งสินค้า ตามรหัสพนักงานขาย
7. พิมพ์สรุปรายยอดขายสินค้า ตามประเภทสินค้า
8. พิมพ์รายการขายสินค้าแต่ละรายการ ตามวันที่
9. กลับเมนูหลัก

4. ระบบควบคุมสินค้าคงคลัง

1. ซื้อสินค้าเพิ่ม เมื่อสินค้าถึงจุดสั่งซื้อ
2. ซื้อสินค้ารายการใหม่ มาจำหน่าย
3. พิมพ์รายการสินค้าที่ ถึงจุดสั่งซื้อ เรียงตามรหัสสินค้า
4. พิมพ์รายการสินค้าที่ ถึงจุดสั่งซื้อ ตามผู้ผลิต และยังไม่ได้สั่งซื้อเพิ่ม
5. พิมพ์รายการสินค้าที่ เหลือไม่ถึง 10 เปอร์เซ็นต์ จะถึงจุดสั่งซื้อ
6. พิมพ์รายการสินค้าที่ กำลังสั่งซื้อเพิ่มปัจจุบัน
7. พิมพ์รายการสินค้าที่ ขายดี จากมากไปน้อย ตามช่วงวันที่
8. กลับเมนูหลัก

5. ระบบรายงานอื่น

1. พิมพ์สรุปค่าขนส่งให้บริษัทขนส่ง ตามเดือน
2. พิมพ์สถิติการใช้บริษัทขนส่งของลูกค้าแต่ละราย ตามช่วงวันที่
3. พิมพ์จดหมายขอโทษ ที่ส่งสินค้าไม่ทันกำหนด โดยระบุเลขที่ใบสั่งซื้อสินค้า
4. พิมพ์รายชื่อลูกค้า ที่พนักงานแต่ละคนรับผิดชอบ ตามช่วงวันที่
5. พิมพ์เลขที่ใบสั่งซื้อสินค้า ที่ส่งไม่ทันกำหนด ตามเดือน
6. พิมพ์เลขที่ใบสั่งซื้อสินค้า ที่มีกำหนดส่งไม่เกิน 10 วันจากวันนี้
7. พิมพ์ข้อมูลจากแฟ้มอย่างง่าย ๆ
8. กลับเมนูหลัก

6.3.3 หลักการตั้งชื่อโปรแกรม

สำหรับระบบงานขาย และสินค้าคงคลัง ได้ตั้งชื่อโปรแกรมตามลำดับในเมนู เพื่อสื่อให้เข้าใจได้ง่ายโดยมองลำดับของตัวเลือกตามเมนู สำหรับชื่อโปรแกรมกำหนดไว้ 6 หลัก

หลักที่ 1 ให้เป็นตัว S ซึ่งย่อมาจาก SALE

หลักที่ 2 และ 3 เป็นลำดับในเมนูหลัก

หลักที่ 4 และ 5 เป็นลำดับในเมนูย่อย

โปรแกรมต่าง ๆ ในระบบทะเบียนมีดังนี้

MENU.PRГ : เมนูหลักสำหรับระบบงานขาย และสินค้าคงคลัง ประกอบด้วย 10 โปรแกรมย่อย

MENUALL : โปรแกรมหลักที่เรียกเมนูต่าง ๆ มาให้เลือก

PRESSDOWN : กระทำเมื่อกดปุ่มลูกศรลง เพื่อปรับปรุ่่งค่าสูงสุด และต่ำสุดของตัวเลือก

PRESSUP : กระทำเมื่อกดปุ่มลูกศรขึ้น เพื่อปรับปรุ่่งค่าสูงสุด และต่ำสุดของตัวเลือก

WRITEMENU : เขียนเมนูใหม่ทุกครั้งที่กดลูกศร

ALERTNEW : ใช้แทนคำสั่ง ALERT ทำให้ได้ตัวเลือกมากกว่า 4 และเปลี่ยนสีได้

ASKPRT : สอบถามการปลายทางของผลลัพธ์

ASKUPD : ถามย้ำการปรับปรุงข้อมูล

HELPPFORM : แสดงข้อความช่วยเหลืออย่างง่าย ๆ เมื่อกดปุ่มฟังก์ชัน F1

HELPPREADVAR : แสดงค่าที่เป็นไปได้ของฟิลด์ที่กำลังรอรับค่า

QUITPROC : สั่งเลิกทำงานของโปรแกรมทั้งหมด

SALE1.PRG : ระบบขาย ประกอบด้วย 3 โปรแกรมย่อย

S0101 : ขาย ด้วยตัวแปรชุดเดียว

S0102 : ขาย ด้วยตัวแปรหลายชุด

S0103 : ขาย ด้วยตัวแปรอาเรย์

SALE2.PRG : ระบบปรับปรุง ประกอบด้วย 5 โปรแกรมย่อย

S0201 : ปรับปรุงใบสั่งสินค้า

S0202 : ยกเลิกใบสั่งสินค้าทั้งใบ

S0203 : ยกเลิกสินค้าในใบสั่งสินค้า

S0204 : เพิ่มเงินเดือนพนักงาน

S0205 : ปรับปรุงแฟ้มต่าง ๆ

SALE3.PRG : ระบบรายงานยอดขาย ประกอบด้วย 8 โปรแกรมย่อย

S0301 : พิมพ์สรุปยอดขายของสินค้า ตามเดือน

S0302 : พิมพ์สรุปยอดขายของสินค้า ตามผู้ผลิตสินค้า

S0303 : พิมพ์สรุปยอดขายของสินค้า ตามผู้ผลิตสินค้า และเดือน

S0304 : พิมพ์สรุปยอดขายของสินค้าแต่ละประเภท ตามเดือน

S0305 : พิมพ์สรุปยอดขายของใบสั่งสินค้า ตามวันที่

S0306 : พิมพ์สรุปยอดขายของใบสั่งสินค้า ตามรหัสพนักงานขาย

S0307 : พิมพ์สรุปยอดขายสินค้า ตามประเภทสินค้า

S0308 : พิมพ์รายการขายสินค้าแต่ละรายการ ตามวันที่

SALE4.PRG ระบบควบคุมสินค้าคงคลัง ประกอบด้วย 7 โปรแกรมย่อย

S0301 : ชื่อสินค้าเพิ่ม เมื่อสินค้าถึงจุดสั่งซื้อ

S0302 : ชื่อสินค้ารายการใหม่ มาจำหน่าย

S0303 : พิมพ์รายการสินค้าที่ ถึงจุดสั่งซื้อ เรียงตามรหัสสินค้า

S0304 : พิมพ์รายการสินค้าที่ ถึงจุดสั่งซื้อ ตามผู้ผลิต และยังไม่ได้สั่งเพิ่ม

S0305 : พิมพ์รายการสินค้าที่ เหลือไม่ถึง 10 เปอร์เซนต์ จะถึงจุดสั่งซื้อ

S0306 : พิมพ์รายการสินค้าที่ กำลังสั่งซื้อเพิ่มปัจจุบัน

S0307 : พิมพ์รายการสินค้าที่ ขายดี จากมากไปน้อย ตามช่วงวันที่

SALE5.PRG ระบบรายงานอื่น ประกอบด้วย 7 โปรแกรมย่อย

S0301 : พิมพ์สรุปค่าขนส่งให้บริษัทขนส่ง ตามเดือน

S0302 : พิมพ์สถิติการใช้บริษัทขนส่งของลูกค้าแต่ละราย ตามช่วงวันที่

S0303 : พิมพ์จดหมายขอโทษ ที่ส่งสินค้าไม่ทันกำหนด โดยระบุเลขที่ใบส่งสินค้า

S0304 : พิมพ์รายชื่อลูกค้า ที่พนักงานแต่ละคนรับผิดชอบ ตามช่วงวันที่

S0305 : พิมพ์เลขที่ใบส่งสินค้า ที่ส่งไม่ทันกำหนด ตามเดือน

S0306 : พิมพ์เลขที่ใบส่งสินค้า ที่มีกำหนดส่งไม่เกิน 10 วันจากวันนี้

S0307 : พิมพ์ข้อมูลจากแฟ้มอย่างง่าย ๆ

6.3.4 วิธีการแปลโปรแกรม

การแปลโปรแกรมสำหรับกรณีศึกษาครั้งนี้ ใช้คำสั่ง RMAKE ในการแปลโปรแกรม โดยระบุชื่อโปรแกรมหลัก และโปรแกรมย่อยทั้งหมด ไว้ในแฟ้ม MENU.RMK

การสร้างแฟ้ม MENU.RMK

// MENU.RMK

// =====

.PRG.OBJ:

CLIPPER \$< /M /N

.OBJ.EXE:

SET RTLINKCMD=/POSI

RTLINK \$*, \$@;

MENU.OBJ: MENU.PRG

SALE1.OBJ: SALE1.PRG

SALE2.OBJ: SALE2.PRG

SALE3.OBJ: SALE3.PRG

SALE4.OBJ: SALE4.PRG

SALE5.OBJ: SALE5.PRG

MENU.EXE: MENU.OBJ SALE1.OBJ SALE2.OBJ SALE3.OBJ SALE4.OBJ SALE5.OBJ

& 6.4 โปรแกรมในระบบงานขาย และสินค้าคงคลัง

โปรแกรมของระบบนี้ ได้จัดทำเป็นโปรแกรมย่อยไว้ภายนอก โดยมีโปรแกรม MENU.PRG คอยเรียกโปรแกรมเล็ก ๆ ที่อยู่โปรแกรมย่อยอีกครั้งหนึ่ง สำหรับโปรแกรมย่อยภายนอกนั้น แต่ละโปรแกรมย่อยคือระบบย่อยระบบหนึ่ง เพราะทำให้ง่ายต่อการปรับปรุงแก้ไขและค้นหา หากประมวลผลตามตัวเลือกแล้วผิดพลาด

โปรแกรมย่อยจากภายนอกมี 5 โปรแกรมคือ SALE1.PRG จนถึง SALE5.PRG รวมโปรแกรมทั้งหมดในระบบนี้เป็น 6 โปรแกรม โดยให้โปรแกรมย่อยทั้ง 5 นี้ควบคุมระบบย่อย 5 ระบบ

1. ระบบขาย
2. ระบบปรับปรุง
3. ระบบรายงาน
4. ระบบคงคลัง
5. ระบบอื่น

6.4.1 โปรแกรมต่าง ๆ ใน MENU.PRG

โปรแกรมใน MENU.PRG นี้มีโปรแกรมย่อยทั้งหมด 10 โปรแกรม โดยรวมโปรแกรมที่เกี่ยวกับเมนู และโปรแกรมสนับสนุน เช่น โปรแกรมสอบถามปลายทางของผลลัพธ์ โปรแกรมถามย้ำการปรับปรุงข้อมูล โปรแกรมแสดงส่วนช่วยเหลือแบบต่าง ๆ โปรแกรมสั่งเลิกการทำงาน เป็นต้น

โปรแกรมเมนูในโปรแกรม MENU.PRG นี้ ใช้หลักการในการตรวจสอบการกดลูกศร ทำให้เมนูแบบนี้สามารถแสดงนาฬิกาที่เดิน ขณะรอเลือกเมนูได้ เมนูแบบนี้ให้ประสิทธิภาพในการใช้การ สามารถเพิ่มการทำงานต่าง ๆ ได้อีกมาก โดยโปรแกรมเมนูนี้เรียกโปรแกรมย่อยจากภายในอีก 3 โปรแกรมคือ PRESSDOWN, PRESSUP และ WRITEMENU การนำโปรแกรมนี้ไปปรับปรุงแก้ไขและนำไปใช้ทำได้โดยง่าย เพราะโปรแกรมนี้จะตรวจสอบการกดปุ่มลูกศร แล้วทำการจอภาพใหม่ทุกครั้ง แต่ถ้ากดปุ่ม ENTER จะประมวลผลตามเงื่อนไขที่ระบุตามตัวเลือกนั้น

ในหัวข้อนี้จะแสดงตัวอย่างโปรแกรมต่าง ๆ ที่มีในโปรแกรม MAIN.PRG ทั้งหมด 10 โปรแกรม โดยเริ่มจากตัวอย่างที่ 6.1 ถึง ตัวอย่างที่ 6.8 (บางโปรแกรมสั้นและมีการทำงานคล้ายกัน จึงนำมารวมในตัวอย่างเดียวกัน)

: ตัวอย่างที่ 6.1

& ผลลัพธ์

// โปรแกรมหลัก ของระบบขาย และสินค้าคงคลัง เพื่อจัดทำเมนูหลักและเมนูย่อยใน

// โปรแกรมเดียวกัน เมนูมีลักษณะเป็นแบบตั้งลง และมีนาฬิกาวิ่งตลอดเวลา

// ซึ่งเมนูแบบนี้สามารถปรับปรุง แก้ไขให้มีประสิทธิภาพได้อีกมากมาย

// และมีการเรียกโปรแกรมย่อยเพื่อจัดทำเมนูอีก 3 โปรแกรมคือ

// 1. ทำหน้าที่เปลี่ยนค่าเมื่อกดลงด้วย โปรแกรมย่อย PRESSDOWN

// 2. ทำหน้าที่เปลี่ยนค่าเมื่อกดขึ้นด้วย โปรแกรมย่อย PRESSUP

// 3. ทำหน้าที่เขียนเมนูใหม่เมื่อเลื่อนตัวเลือกด้วย โปรแกรมย่อย WRITEMENU

PROCEDURE MENUALL

SET KEY 28 TO HELPFORM // ควบคุมปุ่ม F1 เพื่อแสดงข้อมูลเกี่ยวกับผู้เขียนโปรแกรม
 SET KEY -1 TO HELPREADVAR // ควบคุมปุ่ม F2 เพื่อเลือกข้อมูลได้อย่างง่าย ๆ
 SET KEY -9 TO QUITPROC // ควบคุมปุ่ม F10 เพื่อถามการเลิกทำงาน
 SET WRAP ON
 SET DELIMITERS TO '['']
 SET DELIMITERS ON
 SET EPOCH TO 1960
 SET CURSOR OFF
 ARMAIN = 6
 ARCM = ARRAY(ARMAIN)
 ARCM[1] = [1.ระบบขาย]
 ARCM[2] = [2.ระบบปรับปรุง]
 ARCM[3] = [3.ระบบรายงาน]
 ARCM[4] = [4.ระบบคงคลัง]
 ARCM[5] = [5.ระบบอื่น]
 ARCM[6] = [6.เลิกงาน]
 ARS1 = 4
 ARC1 = ARRAY(ARS1)
 ARC1[1] = [1. ขาย ด้วยตัวแปรชุดเดียว]
 ARC1[2] = [2. ขาย ด้วยตัวแปรหลายชุด]
 ARC1[3] = [3. ขาย ด้วยตัวแปรอาเรย์]
 ARC1[4] = [4. กลับเมนูหลัก]
 ARS2 = 6
 ARC2 = ARRAY(ARS2)
 ARC2[1] = [1. ปรับปรุงใบสั่งสินค้า]
 ARC2[2] = [2. ยกเลิกใบสั่งสินค้าทั้งใบ]
 ARC2[3] = [3. ยกเลิกสินค้าในใบสั่งสินค้า]
 ARC2[4] = [4. เพิ่มเงินเดือนพนักงาน]
 ARC2[5] = [5. ปรับปรุงแฟ้มต่าง ๆ]
 ARC2[6] = [6. กลับเมนูหลัก]
 ARS3 = 9
 ARC3 = ARRAY(ARS3)

```

ARC3[1] = [1. พิมพ์สรุปยอดขายของสินค้า ตามเดือน]
ARC3[2] = [2. พิมพ์สรุปยอดขายของสินค้า ตามผู้ผลิตสินค้า]
ARC3[3] = [3. พิมพ์สรุปยอดขายของสินค้า ตามผู้ผลิตสินค้า และเดือน]
ARC3[4] = [4. พิมพ์สรุปยอดขายของสินค้าแต่ละประเภท ตามเดือน]
ARC3[5] = [5. พิมพ์สรุปยอดขายของใบสั่งสินค้า ตามวันที่]
ARC3[6] = [6. พิมพ์สรุปยอดขายของใบสั่งสินค้า ตามรหัสพนักงานขาย]
ARC3[7] = [7. พิมพ์สรุปยอดขายสินค้า ตามประเภทสินค้า]
ARC3[8] = [8. พิมพ์รายการขายสินค้าแต่ละรายการ ตามวันที่]
ARC3[9] = [9. กลับเมนูหลัก]
ARS4 = 8
ARC4 = ARRAY(ARS4)
ARC4[1] = [1. ซื้อสินค้าเพิ่ม เมื่อสินค้าถึงจุดสั่งซื้อ]
ARC4[2] = [2. ซื้อสินค้ารายการใหม่ มาจำหน่าย]
ARC4[3] = [3. พิมพ์รายการสินค้าที่ ถึงจุดสั่งซื้อ เรียงตามรหัสสินค้า]
ARC4[4] = [4. พิมพ์รายการสินค้าที่ ถึงจุดสั่งซื้อ ตามผู้ผลิต และยังไม่ได้สั่งซื้อเพิ่ม]
ARC4[5] = [5. พิมพ์รายการสินค้าที่ เหลือไม่ถึง 10 เปอร์เซ็นต์ จะถึงจุดสั่งซื้อ]
ARC4[6] = [6. พิมพ์รายการสินค้าที่ กำลังสั่งซื้อเพิ่มปัจจุบัน]
ARC4[7] = [7. พิมพ์รายการสินค้าที่ ขายดี จากมากไปน้อย ตามช่วงวันที่]
ARC4[8] = [8. กลับเมนูหลัก]
ARS5 = 8
ARC5 = ARRAY(ARS5)
ARC5[1] = [1. พิมพ์สรุปค่าขนส่งให้บริษัทขนส่ง ตามเดือน]
ARC5[2] = [2. พิมพ์สถิติการใช้บริษัทขนส่งของลูกค้าแต่ละราย ตามช่วงวันที่]
ARC5[3] = [3. พิมพ์จดหมายขอโทษ ที่ส่งสินค้าไม่ทันกำหนด โดยระบุเลขที่ใบสั่งสินค้า]
ARC5[4] = [4. พิมพ์รายชื่อลูกค้า ที่พนักงานแต่ละคนรับผิดชอบ ตามช่วงวันที่]
ARC5[5] = [5. พิมพ์เลขที่ใบสั่งสินค้า ที่ส่งไม่ทันกำหนด ตามเดือน]
ARC5[6] = [6. พิมพ์เลขที่ใบสั่งสินค้า ที่มีกำหนดส่งไม่เกิน 10 วันจากวันนี้]
ARC5[7] = [7. พิมพ์ข้อมูลจากแฟ้มอย่างง่าย]
ARC5[8] = [8. กลับเมนูหลัก]
OPT = 1 // กำหนดตัวเลือกเริ่มต้น
TOPLMT = OPT // กำหนดบรรทัดบนสุดของเมนูปัจจุบัน
BOTTOMLMT = ARMAIN // กำหนดบรรทัดล่างสุดของเมนูปัจจุบัน
ALLSUB = ARMAIN // จำนวนเมนูย่อยทั้งหมด

```

```

DO WRITEMENU // พิมพ์เมนูครั้งแรก
DO WHILE LASTKEY() != 27
  @ 1,5 SAY TIME() // แสดงเวลาตลอดเวลา
  INKEY(0.2) // หยุดรับแป้นพิมพ์ทุก 0.2 วินาที
  IF LASTKEY() != 0
    // กรณีกดปุ่มใด ๆ จะมาตรวจสอบในส่วนนี้
    DO CASE
      CASE LASTKEY() = 27 // กด ESC
        IF OPT <= ALLSUB
          DO QUITPROC
        ELSE
          OPT = (OPT - MOD(OPT,10))/10
        ENDIF
      CASE LASTKEY() = 4 // กดขวา
        IF OPT <= ALLSUB // ตัวเลือกในเมนูหลัก
          DO PRESSDOWN
        ELSE
          IF (OPT-MOD(OPT,10))/10 = ALLSUB // ซ้ำมไปเมนูทางขวาหรือเริ่มใหม่
            OPT = 11
          ELSE
            OPT = (OPT-MOD(OPT,10))+11
          ENDIF
        ENDIF
      CASE LASTKEY() = 19 // กดซ้าย
        IF OPT < 10 // ตัวเลือกในเมนูหลัก
          DO PRESSUP
        ELSE
          IF (OPT-MOD(OPT,10))/10 = 1 // ซ้ำมไปเมนูทางซ้ายหรือสุดท้าย
            OPT = ALLSUB*10+1
          ELSE
            OPT = (OPT-MOD(OPT,10))-9
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF

```

```

CASE LASTKEY() = 24          // กดลง
  DO PRESSDOWN
CASE LASTKEY() = 5          // กดขึ้น
  DO PRESSUP
// กรณีกด ENTER จะมาเลือกประมวลผลในส่วนนี้
CASE LASTKEY() = 13         // กดเอ็นเทอร์ (ENTER)
  DO CASE
    CASE OPT >= 1 .AND. OPT <= ARMAIN -1
      OPT = OPT * 10 + 1
    CASE OPT = ARMAIN
      @ 20,5 SAY "BYE BYE BYE"
      DO QUITPROC
    CASE OPT=10+ARS1 .OR. OPT=20+ARS2 .OR. ;
      OPT=30+ARS3 .OR. OPT=40+ARS4 .OR. ;
      OPT=50+ARS5          // กลับเมนูหลัก
      OPT = (OPT - MOD(OPT,10))/10
    OTHERWISE              // กระทำตัวเลือก
      ATTAGX = LTRIM(STR(MOD(OPT,10),1))
      ATSUBX = LTRIM(STR((OPT - MOD(OPT,10))/10,1))
      SET CURSOR ON
      CLS
      DO &("S0" + ATSUBX + "0" + ATTAGX)
      SET CURSOR OFF
  ENDCASE
ENDCASE
// ตั้งค่า TOPLMT และ BOTTOMLMT ใหม่
DO CASE
  CASE OPT <10
    TOPLMT = 1 ; BOTTOMLMT = ARMAIN
  CASE OPT >= 11 .AND. OPT <= 10+ARS1
    TOPLMT = 11 ; BOTTOMLMT = 10+ARS1
  CASE OPT >= 21 .AND. OPT <= 20+ARS2
    TOPLMT = 21 ; BOTTOMLMT = 20+ARS2

```



```

CASE OPT >= 31 .AND. OPT<=30+ARS3
  TOPLMT = 31 ; BOTTOMLMT = 30+ARS3
CASE OPT >= 41 .AND. OPT<=40+ARS4
  TOPLMT = 41 ; BOTTOMLMT = 40+ARS4
CASE OPT >= 51 .AND. OPT<=50+ARS5
  TOPLMT = 51 ; BOTTOMLMT = 50+ARS5
CASE OPT = 61
  TOPLMT = 1; BOTTOMLMT = ARMAIN; OPT = 6
ENDCASE
DO WRITEMENU // เขียนเมนูใหม่
KEYBOARD CHR(0)
INKEY() // เพื่อตั้งค่าของ LASTKEY() ใหม่
ENDIF
ENDDO
RETURN
: ตัวอย่างที่ 6.2
& ผลลัพธ์
// ใช้ประกอบโปรแกรมเมนูหลัก โดยกระทำเมื่อกดปุ่มลง หรือขึ้น ในการเลือกตัวเลือก
// โดยมีเป้าหมายคือปรับปรุงค่าของตัวเลือกปัจจุบันให้เพื่อขึ้นหรือลดลง
// กระทำเมื่อกดปุ่มลงในเมนู
PROCEDURE PRESSDOWN
  IF OPT = BOTTOMLMT // ถ้าถึงตัวล่างสุดแล้วยังกดลง
    OPT = TOPLMT // ให้ OPT ไปที่ตัวแรก คือ TOPLMT (Top limit)
  ELSE
    OPT = OPT + 1
  ENDIF
RETURN
// กระทำเมื่อกดปุ่มขึ้นในเมนู
PROCEDURE PRESSUP
  IF OPT = TOPLMT // ถ้าถึงตัวแรกแล้วยังกดขึ้น
    OPT = BOTTOMLMT // ให้ OPT ไปยังตัวสุดท้าย
  ELSE
    OPT = OPT - 1
  ENDIF
RETURN

```

: ตัวอย่างที่ 6.3

& _พลัฟธ์

// ใช้ประกอบโปรแกรมหลัก เพื่อทำการเขียนจอเมนูให้ถูกต้องโดยใช้ค่าตัวเลือกปัจจุบัน

// การเขียนเมนูของโปรแกรมนี จะเขียนใหม่ทุกครั้งที่มีการกดปุ่มใด ๆ

// โปรแกรมย่อยนี้จึงใช้สำหรับเขียนหน้าเมนูใหม่เมื่อมีการกดแป้นพิมพ์

PROCEDURE WRITEMENU

SET COLOR TO W/B

CLS ; SETPOS(2,0)

XCOL = ARRAY(ARMAIN)

FOR J = 1 TO ARMAIN

// พิมพ์เมนูหลัก

XCOL[J] = COL() + 1

// เก็บหลักของเมนูย่อยแต่ละเมนู

@ 2,XCOL[J] SAY ARCM[J]

NEXT

ATTAG = MOD(OPT,10)

// หางของตัวเลือก

ATSUB = (OPT - ATTAG)/10

// เมนูย่อยที่

IF ATSUB > 0

I = 4

// ระบุเลขที่บรรทัดเริ่มต้น

FOR J = I TO I - 1 + &("ARS"+LTRIM(STR(ATSUB,1)))

@ J,XCOL[ATSUB] SAY &("ARC"+LTRIM(STR(ATSUB,1)))[J-I+1]

NEXT

TEMPAR = &("ARC"+LTRIM(STR(ATSUB,1)))

ENDIF

SET COLOR TO B/W

IF OPT > 10

// สั่งเลือกตัวเลือกย่อย

@ I-1+ATTAG,XCOL[ATSUB] SAY TEMPAR[ATTAG]

// พิมพ์ตัวเลือกปัจจุบัน

SET COLOR TO W/N

@ 2,XCOL[ATSUB] SAY ARCM[ATSUB]

// พิมพ์หัวของเมนูย่อย

ELSE

@ 2,XCOL[ATTAG] SAY ARCM[ATTAG]

// พิมพ์หัวของเมนูย่อยในขณะที่เลือกหัว

ENDIF

SET COLOR TO W/B

RETURN

: ตัวอย่างที่ 6.4

& ผลลัพธ์

// ใช้แทนฟังก์ชัน ALERT เพราะทำให้ใช้ตัวเลือกได้มากกว่า 4 ตัวเลือก และแสดง
 // ตัวเลือกในลักษณะ PULLDOWN CHOICE แล้วสามารถระบุสีได้อย่างอิสระ
 // โดยไม่ทำให้สถานะเดิมของสีเปลี่ยนไป เพราะมีการจัดเก็บไว้แล้ว

FUNCTION ALERTNEW

PARAMETER HEADING,CHOICEAR

CNTAR = 0

AEVAL(CHOICEAR,{|ELE|CNTAR:=CNTAR+1})

OLDCOLOR = SETCOLOR()

SAVE SCREEN TO SCR

CLS

SET COLOR TO W/BR

@ 10,20,12+CNTAR,60 BOX "++|++| "

@ 11,21 SAY PADDC(HEADING,38)

FOR I = 1 TO CNTAR

@ 11+I,30 PROMPT PADR(LTRIM(STR(I))+ ". "+CHOICEAR[I],30)

NEXT

MENU TO VALNEW

SETCOLOR(OLDCOLOR)

RESTORE SCREEN FROM SCR

RETURN (VALNEW)

: ตัวอย่างที่ 6.5**& ผลลัพธ์**

// ตัวอย่างนี้มี 2 ฟังก์ชันคือ ฟังก์ชันเพื่อเลือกปลายทางของผลลัพธ์

// และฟังก์ชันตามย้ำการปรับปรุงข้อมูล เมื่อมีการแก้ไขข้อมูล

// ให้เสียงประกอบการเลือกทุกครั้ง

// ใช้ถามว่าจะพิมพ์ผลลัพธ์ออกทางใด

FUNCTION ASKPRT()

ASKPRT = ALERTNEW("ผลลัพธ์ของรายงาน",{ 'จอภาพ', 'เครื่องพิมพ์'})

IF ASKPRT = 2

TONE(500,5)

ELSE

TONE(800,8)

ENDIF

RETURN (ASKPRT)

// ใช้ถามว่าจะยอมรับการปรับปรุงข้อมูลหรือไม่

FUNCTION ASKUPD()

ASKUPD = ALERTNEW("การปรับปรุงข้อมูล",{'ยอมรับการปรับปรุง','ยกเลิกการปรับปรุง'})

IF ASKPRT = 2

TONE(500,5)

ELSE

TONE(800,8)

ENDIF

RETURN (ASKUPD)

: ตัวอย่างที่ 6.6

& พลลัพท์

// พิมพ์ข้อมูลของผู้พัฒนาโปรแกรมทางจอภาพ หากกดปุ่มฟังก์ชัน F1

// เพราะระบุใน MENUALL ว่า SET KEY 28 TO HELPFORM

PROCEDURE HELPFORM

SAVE SCREEN TO SCR

CLS

? " วันที่ :";DTOC(DATE())

? " เวลา :";TIME()

? "====="

? "โปรแกรมนี้ถูกพัฒนาโดย. อ.บุรินทร์ รุจจนพันธุ์"

? "เริ่มพัฒนาเมื่อ 8 กุมภาพันธ์ 2545"

? "โปรแกรมนี้เป็นรุ่นที่ 2.200202"

? "หากสนใจ หรือต้องการตัวโปรแกรม ติดต่อขอรับฟรีได้ที่"

? "วิทยาลัยโยนก ลำปาง 52000"

? "TEL.(054)265170 EXT.210 FAX.(054)265184"

INKEY(5)

RESTORE SCREEN FROM SCR

RETURN

: ตัวอย่างที่ 6.7

& แพลตฟอร์ม

```
// ฟังก์ชันนี้ช่วยให้การใส่ค่าในตัวแปรบางตัวแปรถูกต้อง เพราะจะแสดง
// ค่าที่เหมาะสมกับฟิลด์นั้นมาให้เลือก โดยการอ่านค่าจากแฟ้มมาแสดงให้ดู
// แต่จะต้องระบุให้ชัดเจนว่าจะนำฟิลด์ใดมาใช้บ้าง
// โปรแกรมนี้จึงแสดงข้อมูลที่เป็นไปได้สำหรับตัวแปรที่กำลังรอรับค่า
```

FUNCTION HELPREADVAR

```
OLDALIAS = ALIAS()           // เก็บพื้นที่ทำงานเดิม
SHOWAR = {}
DO CASE
  CASE LEFT(READVAR(),4) = "_PRO" // เพราะอาจเป็น _PRO[1] ได้
    IF OLDALIAS != "PRO"
      USE PRO NEW           // ถ้าเปิดแล้วต้องไม่เปิดอีก
    ENDIF
    GO TOP
    WHILE !EOF()
      AADD(SHOWAR,STR(PRO)+"|" +PRONAME)
      SKIP
    END
    IND := ACHOICE(2,61,5,78,SHOWAR)
    IF IND > 0              // ป้องกันการใช้นุ้ม ESC
      IF RIGHT(READVAR(),1) = "]"
        _PRO[J] = VAL(LEFT(SHOWAR[IND],4))
      ELSE
        _PRO = VAL(LEFT(SHOWAR[IND],4))
      ENDIF
    ENDIF
  IF OLDALIAS != "PRO"     // เพราะอนุญาตให้เรียก F2 2 ครั้ง
    CLOSE                 // ในจุดที่ไม่ได้เปิดแฟ้มนี้
  ENDIF
CASE READVAR() = "_ORDER"
  SAVE SCREEN TO SCR
  IF OLDALIAS != "ORDER"
    USE ORDER NEW        // ถ้าเปิดแล้วต้องไม่เปิดอีก
```

```

ENDIF
GO TOP
WHILE !EOF()
  AADD(SHOWAR,STR(ORDER))
  SKIP
END
@ 2,69,24,79 BOX "++|++|"
IND := ACHOICE(3,70,23,78,SHOWAR)
IF IND > 0 // ป้องกันการใช้ปุ่ม ESC
  _ORDER = VAL(SHOWAR[IND])
ENDIF
IF OLDALIAS != "ORDER" // เพราะอนุญาตให้เรียก F2 2 ครั้ง
  CLOSE // ในจุดที่ไม่ได้เปิดแฟ้มนี้
ENDIF
RESTORE SCREEN FROM SCR
OTHERWISE
  X = ALERTNEW("ขอโทษ 'ไม่ได้เตรียมส่วนสนับสนุน สำหรับจุดนี้' ,{"OK."})
ENDCASE
IF LEN(OLDALIAS) > 0
  SELE &OLDALIAS // กลับสู่พื้นที่ทำงานเดิม
ENDIF
RETURN
: ตัวอย่างที่ 6.8
& _พลัฟธ์
// สอบถามการเลิกการทำงาน หากเลิกการทำงานจะออกเสียง 3 เสียง
PROCEDURE QUITPROC
  Q = ALERTNEW("สอบถามการเลิกการทำงาน",{"เลิกการทำงาน","ยังไม่เลิกงาน"})
  IF Q != 2
    FOR I = 1 TO 3
      TONE(50 * I,5-I)
    NEXT
    QUIT
  ENDIF
RETURN

```

6.4.2 โปรแกรมต่าง ๆ ใน SALE1.PRG

โปรแกรมใน SALE1.PRG นี้มีโปรแกรมย่อยภายในทั้งหมด 3 โปรแกรม โดยรวมโปรแกรมที่เกี่ยวกับระบบงานขาย เช่น โปรแกรมขายด้วยตัวแปรชุดเดียว หลายชุด หรืออาเรย์ ทั้ง 3 โปรแกรมนี้เป็นโปรแกรมขายทั้งหมด แต่แตกต่างกันที่หลักการเขียนโปรแกรม เพราะโปรแกรมขายด้วยตัวแปรชุดเดียวจะมุ่งปรับปรุงข้อมูลโดยไม่มีการค้านวนยอดต่าง ๆ ให้ชัดเจน ส่วนโปรแกรมขายด้วยตัวแปรหลายชุดจะมีตัวแปรมารับผลการคำนวณต่าง ๆ และโปรแกรมขายด้วยตัวแปรอาเรย์จะทำการเก็บข้อมูลลงอาเรย์ สามารถพิมพ์ใบเสร็จโดยอ่านข้อมูลจากตัวแปรอาเรย์มาพิมพ์ได้

ระบบงานขายเป็นระบบที่สำคัญยิ่ง เพราะระบบนี้ถูกพัฒนาขึ้นเพื่อเก็บประวัติการขาย และโปรแกรมในระบบนี้จะบันทึกข้อมูลการขายเข้าไปเก็บในแฟ้มสั่งซื้อ ซึ่งเป็นข้อมูลหลักของระบบ และทำให้นำข้อมูลไปปรับปรุงนำไปใช้ และประมวลผลในงานของระบบอื่นได้ต่อไป

ในหัวข้อนี้จะแสดงตัวอย่างโปรแกรมต่าง ๆ ที่มีในโปรแกรม SALE1.PRG ทั้งหมด 3 โปรแกรม โดยเริ่มจากตัวอย่างที่ 6.9 ถึง ตัวอย่างที่ 6.11 (R0101 - R0103)

: ตัวอย่างที่ 6.9

& แพลตฟอร์ม

// โปรแกรมรับข้อมูลการขาย โดยใช้ตัวแปรชุดเดียว คือชุดที่เกี่ยวข้องกับการปรับปรุง

// หน้าทีของโปรแกรมจะบันทึกข้อมูลในแฟ้มใบสั่งซื้อสินค้า และใบสั่งซื้อสินค้าละเอียด

// แล้วไปลดยอดสินค้าในแฟ้มสินค้า ทำให้ตรวจสอบสินค้าคงคลังในภายหลังได้

// โดยโปรแกรมจะรับข้อมูลที่ละรายการ พิมพ์ผลการคำนวณของรายการ แล้วบันทึกข้อมูล

PROCEDURE S0101

```
_ORDER := _CUST := _EMPL := _SHIPPER := _FREIGHT := 0
```

```
_SHIPADDR = SPACE(40)
```

```
_DATEORD = DATE()
```

```
_DATESHIP = DATE() + 7
```

```
_DATEREQ = CTOD("")
```

```
// =====
```

```
USE ORDER
```

```
@ 2,5 SAY "ป้อนเลขที่ใบสั่งซื้อสินค้า : " GET _ORDER PICT "99999"
```

```
@ 3,5 SAY "รหัสลูกค้า : " GET _CUST PICT "9999"
```

```
@ 4,5 SAY "รหัสพนักงาน : " GET _EMPL PICT "9999"
```

```
@ 5,5 SAY "สถานที่จัดส่ง : " GET _SHIPADDR
```

```
@ 6,5 SAY "วันที่สั่ง : " GET _DATEORD
```

```
@ 7,5 SAY "วันที่ต้องการ : " GET _DATEREQ
```

```
@ 8,5 SAY "วันที่ส่ง : " GET _DATESHIP
```

```

@ 9,5 SAY "รหัสบริษัทรับขนส่ง : " GET _SHIPPER PICT "99"
@ 10,5 SAY "ค่าขนส่ง : " GET _FREIGHT PICT "9999.99"
READ
APPEND BLANK // เก็บลงแฟ้มสั่งสินค้า (ORDER.DBF)
REPLACE ORDER WITH _ORDER,CUST WITH _CUST,EMPL WITH _EMPL,;
    SHIPADDR WITH _SHIPADDR,DATEORD WITH _DATEORD,;
    DATEREQ WITH _DATEREQ,DATESHIP WITH _DATESHIP,;
    SHIPPER WITH _SHIPPER,FREIGHT WITH _FREIGHT
_SUM = 0
LN = 11
DO WHILE LASTKEY() != 27
    USE ORDERD
    _PRO := _OPRICE := _OQUANT := _ODISC := 0
    @ LN,5 SAY "สินค้า" GET _PRO PICT "9999"
    @ LN,18 SAY "ราคาขาย" GET _OPRICE PICT "9999.99"
    @ LN,35 SAY "ปริมาณ" GET _OQUANT PICT "9999"
    @ LN,52 SAY "ส่วนลด" GET _ODISC PICT "999.99"
    READ
    IF _PRO != 0
        _SUM = _SUM + (_OPRICE * _OQUANT) - _ODISC
        @ LN, 68 SAY _SUM PICT "9,999,999.99"
        APPEND BLANK // เก็บลงแฟ้มสั่งสินค้าละเอียด (ORDERD.DBF)
        REPLACE ORDER WITH _ORDER,PRO WITH _PRO,OPRICE WITH _OPRICE ,;
            OQUANT WITH _OQUANT,ODISC WITH _ODISC
        // ส่วนที่ใช้ลดปริมาณในคลังสินค้า
        USE PRO INDEX PRO1
        SEEK _PRO
        IF FOUND()
            REPLACE PROREST WITH PROREST - _OQUANT
        ENDIF
    ELSE
        @ LN+1,68 SAY _SUM PICT "9,999,999.99"
        KEYBOARD CHR(27)

```



```

INKEY()
ENDIF
LN++
ENDDO
INKEY(3)
RETURN

```

: ตัวอย่างที่ 6.10

& แพลตฟอร์ม

```

// โปรแกรมรับข้อมูลการขาย โดยใช้ตัวแปรหลายชุด เพื่อคำนวณในหลายหน้าที่
// หน้าที่หลักของโปรแกรมคือ บันทึกข้อมูลในแฟ้มใบสั่งซื้อสินค้า และใบสั่งซื้อสินค้าละเอียด
// แล้วไปลดยอดสินค้าในแฟ้มสินค้า ทำให้ตรวจสอบสินค้าคงคลังในภายหลังได้
// โดยโปรแกรมจะรับข้อมูลที่ละรายการ ตรวจสอบจำนวนในคลังสินค้า แล้วปรับปรุงข้อมูล
// สุดท้ายแสดงผลรวมเงินของการขายสินค้าแต่ละรายการ ในบรรทัดล่างสุด
// แล้วสอบถามความต้องการพิมพ์ใบเสร็จ ทางเครื่องพิมพ์หรือจอภาพ

```

PROCEDURE S0102

```

_ORDER := _CUST := _EMPL := _SHIPPER := _FREIGHT := 0
_SHIPADDR = SPACE(40) ; _DATEORD = DATE()
_DATESHIP = DATE() + 7 ;
_DATEREQ = CTOD("")
// =====
USE ORDER
@ 2,5 SAY "เลขที่ใบสั่งซื้อสินค้า:" GET _ORDER PICT "99999"
@ 3,5 SAY " รหัสลูกค้า:" GET _CUST PICT "9999"
@ 4,5 SAY " รหัสพนักงาน:" GET _EMPL PICT "9999"
@ 5,5 SAY " สถานที่จัดส่ง:" GET _SHIPADDR
@ 6,5 SAY " วันที่สั่ง:" GET _DATEORD
@ 7,5 SAY " วันที่ต้องการ:" GET _DATEREQ
@ 8,5 SAY " วันที่ส่ง:" GET _DATESHIP
@ 9,5 SAY " รหัสบริษัทรับขน:" GET _SHIPPER PICT "99"
@ 10,5 SAY " ค่าขนส่ง:" GET _FREIGHT PICT "9999.99"
READ
APPEND BLANK // เก็บลงแฟ้มสั่งซื้อสินค้า (ORDER.DBF)
REPLACE ORDER WITH _ORDER,CUST WITH _CUST,EMPL WITH _EMPL,;
SHIPADDR WITH _SHIPADDR,DATEORD WITH _DATEORD,;

```

```

    DATEREQ WITH _DATEREQ,DATESHIP WITH _DATESHIP,;
    SHIPPER WITH _SHIPPER,FREIGHT WITH _FREIGHT
    _SUM = 0
    @ 11,10 SAY "รหัสสินค้า      ราคา      ปริมาณ      ส่วนลด      รวมเงิน"
    LN = 12
    DO WHILE LASTKEY() != 27
        _PRO := _OPRICE := _OQUANT := _ODISC := 0
        USE PRO INDEX PRO1
        WHILE .T. // ป้อนรหัสสินค้าจนกว่าจะพบ และมีพอขายให้ลูกค้า
            @ LN,10 GET _PRO PICT "9999"
            @ LN,35 GET _OPRICE PICT "9999.99"
            @ LN,45 GET _OQUANT PICT "9999"
            @ LN,55 GET _ODISC PICT "999.99"
            READ
            SEEK _PRO
            IF FOUND()
                @ LN,18 SAY LEFT(PRONAME,15)
                IF PROREST >= _OQUANT
                    EXIT
                ELSE
                    @ LN,18 SAY "ปริมาณมีไม่พอขาย"
                ENDIF
            ELSE
                @ LN,18 SAY "ไม่พบสินค้า"
            ENDIF
            IF _PRO = 0
                EXIT
            ENDIF
        END
        IF _PRO != 0
            _SUM = _SUM + (_OPRICE * _OQUANT) - _ODISC
            USE ORDERD
            APPEND BLANK // เก็บลงแฟ้มสั่งสินค้าละเอียด (ORDERD.DBF)
        
```

```

REPLACE ORDER WITH _ORDER,PRO WITH _PRO,OPRICE WITH _OPRICE ;
  OQUANT WITH _OQUANT,ODISC WITH _ODISC
USE PRO INDEX PRO1           // ส่วนที่ใช้ลดปริมาณในคลังสินค้า
  SEEK _PRO
  IF FOUND()
    REPLACE PROREST WITH PROREST - _OQUANT
  ENDIF
@ LN,65 SAY STR((_OPRICE * _OQUANT) - _ODISC)
ELSE
@ LN,1 CLEAR TO LN,78
@ LN,65 SAY _SUM PICT "9,999,999.99"
INKEY(3)
KEYBOARD CHR(27)
INKEY()
ENDIF
LN++
ENDDO
// การสั่งพิมพ์ใบเสร็จ
IF _SUM > 0
  GASKPRT = ASKPRT()
  IF GASKPRT = 2
    SET DEVICE TO PRINTER
  ENDIF
  CLS
  SELE 1 ; USE ORDERD
  SET FILTER TO ORDER = _ORDER
  GO TOP
  @ 2,10 SAY "ใบสั่งซื้อเลขที่" + STR(_ORDER)
  @ 3,10 SAY "===== "
  SELE 2 ; USE PRO INDEX PRO1 NEW
  LN = 4 ; _SUM = 0
  DO WHILE !EOF()           // เพิ่มหลักในรูปนี้คือ เพิ่มใบสั่งซื้อละเอียด
    SELE 1 ; _PRO = PRO

```

```

SELE 2 ; SEEK_PRO           // เพิ่มสินค้า
  IF FOUND()
    @ LN,10 SAY PRO
    @ LN,20 SAY LEFT(PRONAME,18)
  ENDIF
SELE 1                       // เพิ่มใบสั่งซื้อละเอียด
  @ LN,36 SAY OPRICE
  @ LN,44 SAY OQUANT
  @ LN,52 SAY ODISC
  @ LN,60 SAY (OPRICE * OQUANT) - ODISC
  _SUM = _SUM + (OPRICE * OQUANT) - ODISC
  LN++
SKIP
ENDDO
@ LN,60 SAY _SUM
IF GASKPRT = 2
  EJECT
ENDIF
SET DEVICE TO SCREEN
WAIT " กดปุ่มใด ๆ เพื่อทำงานต่อ"
CLOSE ALL
ENDIF
// สอบถามการป้อนใบสั่งซื้อสินค้ารายการใหม่
ASKLOOP = ALERTNEW("สอบถามการป้อนใบสั่งซื้อรายการใหม่";
  {"ป้อนใบสั่งซื้อใหม่","เลิกงานสั่งซื้อ"})
IF ASKLOOP = 1
  CLS
  DO S0102
ENDIF
RETURN
: ตัวอย่างที่ 6.11
& _ผลลัพธ์
// โปรแกรมรับข้อมูลการขาย โดยใช้ตัวแปรของเรา
// หน้าที่หลักของโปรแกรมคือ บันทึกข้อมูลใบเพิ่มใบสั่งซื้อสินค้า และใบสั่งซื้อสินค้าละเอียด

```

```
// แล้วไปลดยอดสินค้าในแฟ้มสินค้า ทำให้ตรวจสอบสินค้าคงคลังในภายหลังได้
// โดยโปรแกรมจะรับข้อมูลที่ละรายการ ตรวจสอบจำนวนในคลังสินค้า แล้วปรับปรุงข้อมูล
// สุดท้ายแสดงผลรวมเงินของการขายสินค้าแต่ละรายการ ในบรรทัดล่างสุด
// แล้วสอบถามความต้องการพิมพ์ใบเสร็จ ทางเครื่องพิมพ์หรือจอภาพ
// ในการพิมพ์ข้อมูลจะอ่านข้อมูลจากอาเรย์ที่เก็บไว้มาพิมพ์ แทนการอ่านจากแฟ้มโดยตรง
```

PROCEDURE S0103

ASKLOOP = 1

DO WHILE ASKLOOP = 1

CLS

_ORDER := _CUST := _EMPL := _SHIPPER := _FREIGHT := 0

_SHIPADDR = SPACE(40)

_DATEORD = DATE()

_DATESHIP = DATE() + 7

_DATEREQ = CTOD("")

// =====

USE ORDER

@ 2,5 SAY "เลขที่ใบสั่งสินค้า : " GET _ORDER PICT "99999"

@ 3,5 SAY " รหัสลูกค้า : " GET _CUST PICT "9999"

@ 4,5 SAY " รหัสพนักงาน : " GET _EMPL PICT "9999"

@ 5,5 SAY " สถานที่จัดส่ง : " GET _SHIPADDR

@ 6,5 SAY " วันที่สั่ง : " GET _DATEORD

@ 7,5 SAY " วันที่ต้องการ : " GET _DATEREQ

@ 8,5 SAY " วันที่ส่ง : " GET _DATESHIP

@ 9,5 SAY " รหัสบริษัทรับขน : " GET _SHIPPER PICT "99"

@ 10,5 SAY " ค่าขนส่ง : " GET _FREIGHT PICT "9999.99"

READ

IF UPDATED()

APPEND BLANK

// เก็บลงแฟ้มสั่งสินค้า (ORDER.DBF)

REPLACE ORDER WITH _ORDER,CUST WITH _CUST,EMPL WITH _EMPL,;

SHIPADDR WITH _SHIPADDR,DATEORD WITH _DATEORD,;

DATEREQ WITH _DATEREQ,DATESHIP WITH _DATESHIP,;

SHIPPER WITH _SHIPPER,FREIGHT WITH _FREIGHT

ELSE

```

RETURN
ENDIF
_SUM = 0
LN = 12 ; J = 1
_PRO := ARRAY(15)
_OPRICE := ARRAY(15)
_OQUANT := ARRAY(15)
_ODISC := ARRAY(15)
_MONEY := ARRAY(15)
@ 11,10 SAY "รหัสสินค้า      ราคา      ปริมาณ      ส่วนลด      รวมเงิน"
DO WHILE LASTKEY() != 27
  USE PRO INDEX PRO1
  _PRO[J] = 1
  DO WHILE _PRO[J] != 0
    _PRO[J] := _OPRICE[J] := _OQUANT[J] := _ODISC[J] := 0
    @ LN,10 GET _PRO[J] PICT "9999"
    @ LN,35 GET _OPRICE[J] PICT "9999.99"
    @ LN,45 GET _OQUANT[J] PICT "9999"
    @ LN,55 GET _ODISC[J] PICT "999.99"
    READ
    SEEK _PRO[J]
    IF FOUND()
      @ LN,18 SAY LEFT(PRONAME,15)
      IF PROREST >= _OQUANT[J]           // ปรับยอดในคลังสินค้า
        REPLACE PROREST WITH PROREST - _OQUANT[J]
        _MONEY[J] = _OPRICE[J]*_OQUANT[J]-_ODISC[J]
        @ LN,65 SAY STR(_MONEY[J])
      EXIT
    ENDIF
  ENDIF
ENDDO
IF _PRO[J] != 0
  _SUM = _SUM + _MONEY[J]

```

```

USE ORDERD
APPEND BLANK           // เก็บลงเพิ่มสิ่งสินค้าละเอียด (ORDERD.DBF)
REPLACE ORDER WITH _ORDER;
    PRO WITH _PRO[J];
    OPRICE WITH _OPRICE[J];
    OQUANT WITH _OQUANT[J];
    ODISC WITH _ODISC[J]
ELSE
    @ LN,1 CLEAR TO LN,78
    @ LN,65 SAY _SUM PICT "9,999,999.99"
    INKEY(3) ; KEYBOARD CHR(27) ; INKEY()
ENDIF
LN++ ; J++
ENDDO
// การสั่งพิมพ์ใบเสร็จ
IF _SUM > 0
    ALLPRO = J - 2
    GASKPRT = ASKPRT()
    IF GASKPRT = 2
        SET DEVICE TO PRINTER
    ENDIF
    CLS
    @ 2,10 SAY "ใบสั่งซื้อเลขที่" + STR(_ORDER)
    @ 3,10 SAY "======"
    USE PRO INDEX PRO1
    LN = 4
    FOR I = 1 TO ALLPRO
        SEEK _PRO[I] // เพิ่มสินค้า
        IF FOUND()
            @ LN,10 SAY PRO
            @ LN,20 SAY LEFT(PRONAME,18)
        ENDIF
        @ LN,36 SAY _OPRICE[I]
    
```

```

@ LN,44 SAY _OQUANT[I]
@ LN,52 SAY _ODISC[I]
@ LN,60 SAY _MONEY[I]
LN++
NEXT
@ LN,60 SAY _SUM
IF GASKPRT = 2 ; EJECT ; ENDIF
SET DEVICE TO SCREEN
WAIT "กดปุ่มใด ๆ เพื่อทำงานต่อ"
CLOSE ALL
ENDIF
ASKLOOP = ALERTNEW("สอบถามการป้อนใบสั่งซื้อรายการใหม่";
    {"ป้อนใบสั่งซื้อใหม่","เลิกงานสั่งซื้อ"})
ENDDO
RETURN

```

6.4.3 โปรแกรมต่าง ๆ ใน SALE2.PRG

โปรแกรมใน SALE2.PRG นี้มีโปรแกรมย่อยภายในทั้งหมด 5 โปรแกรม โดยรวมโปรแกรมที่เกี่ยวข้องกับระบบปรับปรุง เช่น โปรแกรมปรับปรุงใบสั่งซื้อสินค้า โปรแกรมเพิ่มเงินเดือนให้พนักงาน โปรแกรมยกเลิกใบสั่งซื้อสินค้า หรือการปรับปรุงเพิ่มต่าง ๆ เป็นต้น

ระบบปรับปรุงเป็นระบบที่ช่วยแก้ปัญหาต่าง ๆ เช่น ป้อนข้อมูลการสั่งซื้อผิด หรือลูกค้ายกเลิกใบสั่งซื้อรวมถึงการเพิ่มเงินเดือนให้พนักงาน และสามารถแก้ไขข้อมูลในแฟ้มทุกแฟ้มได้อย่างง่าย ๆ

ในหัวข้อนี้จะแสดงตัวอย่างโปรแกรมต่าง ๆ ที่มีในโปรแกรม SALE2.PRG ทั้งหมด 5 โปรแกรม โดยเริ่มจากตัวอย่างที่ 6.12 ถึง ตัวอย่างที่ 6.16 (R0201 - R0205)

: ตัวอย่างที่ 6.12

& ผลลัพธ์

```

// ปรับปรุงใบสั่งซื้อสินค้า โดยรับรหัสใบสั่ง แล้วนำข้อมูลในใบสั่งซื้อสินค้ามาให้แก้ไข
// เมื่อแก้ไขข้อมูลในใบสั่งซื้อสินค้าแล้ว จะนำข้อมูลในใบสั่งซื้อสินค้าละเอียดเก็บลงอาเรย์
// แล้วนำข้อมูลในอาเรย์มาให้ปรับปรุง เมื่อปรับปรุงแล้วจะเขียนทับลงไปในเรคอร์ดเดิม

```

PROCEDURE S0201

```

_ORDER := 0
DO WHILE LASTKEY() != 27
@ 1,5 SAY "เลขที่ใบสั่งซื้อ : " GET _ORDER PICT "99999"
READ

```



```

USE ORDER
LOCATE FOR ORDER = _ORDER
IF FOUND()
  @ 2,5 SAY "รหัสลูกค้า : " GET CUST
  @ 3,5 SAY "รหัสพนักงาน : " GET EMPL
  @ 4,5 SAY "สถานที่จัดส่ง : " GET SHIPADDR
  @ 5,5 SAY "วันที่สั่งซื้อสินค้า : " GET DATEORD
  @ 6,5 SAY "วันที่ต้องการสินค้า : " GET DATEREQ
  @ 7,5 SAY "วันที่จัดส่งสินค้า : " GET DATESHIP
  @ 8,5 SAY "บริษัทจัดส่ง : " GET SHIPPER
  @ 9,5 SAY "ค่าขนส่ง : " GET FREIGHT
READ
CLOSE // 필ด์จะได้รับ การปรับปรุงทันที โดยไม่ต้อง REPLACE
SELE 1 ; USE ORDERD
  SET FILTER TO ORDER = _ORDER
  COUNT TO CNTALL FOR ORDER = _ORDER
  GO TOP
  _PRO := ARRAY(15)
  _OPRICE := ARRAY(15)
  _OQUANT := ARRAY(15)
  _ODISC := ARRAY(15)
  FOR I := 1 TO CNTALL // เก็บค่าลงอาเรย์
    _PRO[I] = PRO
    _OPRICE[I] = OPRICE
    _OQUANT[I] = OQUANT
    _ODISC[I] = ODISC
  SKIP
NEXT
FOR J := 1 TO CNTALL // รับค่าปรับปรุง
  @ J+10, 10 GET _PRO[J]
  @ J+10, 40 GET _OPRICE[J]
  @ J+10, 50 GET _OQUANT[J]
  @ J+10, 57 GET _ODISC[J]

```

```

SELE 2 ; USE PRO INDEX PRO1
SEEK _PRO[J]
IF FOUND()
  @ J+10,18 SAY LEFT(PRONAME,20)
  @ J+10,65 SAY _OPRICE[J] * _OQUANT[J] - _ODISC[J]
ENDIF
NEXT
READ
IF UPDATED() // ตรวจสอบว่าปรับปรุงหรือไม่
  SELE 1
  GO TOP
  FOR I := 1 TO CNTALL
    REPLACE PRO WITH _PRO[I],OPRICE WITH _OPRICE[I];
    OQUANT WITH _OQUANT[I],ODISC WITH _ODISC[I]
  SKIP
NEXT
ENDIF
CLOSE ALL
EXIT // ถ้าปรับปรุงเสร็จแล้ว จะเลิกโปรแกรมนี้ทันที
ENDIF
ENDDO
RETURN

```

: ตัวอย่างที่ 6.13

& _พลัฟร์

// ยกเลิกใบสั่งซื้อสินค้า โดยรับเลขที่ใบสั่งซื้อสินค้าที่ต้องการลบ แล้วค้นหาเรคอร์ดที่ตรง
// กับใบเพิ่มใบสั่งซื้อ และใบสั่งซื้อละเอียด
// เมื่อพบจะลงเรคอร์ดจากทั้ง 2 เพิ่มที่มีเลขที่ใบสั่งซื้อตรงกับที่ต้องการ

```

PROCEDURE S0202
  _ORDER = 0
  USE ORDER
  DO WHILE .T.
    @ 5,5 SAY "เลขที่ใบสั่งซื้อที่ต้องการลบ : " GET _ORDER PICT "99999"
    READ
    LOCATE FOR _ORDER = ORDER
  
```

```

IF FOUND()                                // เห็นเพียงเรคอร์ดเดียวในตารางของ DBEDIT
  DBEDIT(5,10,23,70,{"ORDER","CUST","EMPL"})
  IF ASKUPD() = 1
    DELETE
    PACK
    USE ORDERD
    DELETE ALL FOR ORDER = _ORDER
    PACK
  ENDIF
  EXIT
ENDIF
ENDDO
RETURN
: ตัวอย่างที่ 6.14
& _พลัฟธ์
// ยกเลิกสินค้าในใบสั่งสินค้าบางรายการ โดยรับเลขที่ใบสั่งสินค้า
// แล้วเลือกเรคอร์ดที่เกี่ยวข้องจากแฟ้มใบสั่งสินค้าละเอียดมาให้ปรับปรุงโดยง่าย
// โดยการแก้ไขข้อมูลใช้ฟังก์ชัน BROWSE หลังจาก que เลือกข้อมูลที่ตรงกับเลขที่ใบสั่ง
PROCEDURE S0203
  _ORDER = 0
  USE ORDERD
  DO WHILE .T.
    @ 5,5 SAY "เลขที่ใบสั่งซื้อที่ต้องการลบสินค้า : " GET _ORDER PICT "99999"
    READ
    LOCATE FOR _ORDER = ORDER
    IF FOUND()
      SET FILTER TO ORDER = _ORDER
      BROWSE(5,10,20,70)
      PACK
      EXIT
    ENDIF
  ENDDO
RETURN

```

: ตัวอย่างที่ 6.15**& _พลัฟธ์**

// ปรับปรุงเงินเดือนพนักงานทุกคน โดยป้อนเปอร์เซ็นต์ของการปรับปรุง

// แล้วจึงใช้คำสั่งปรับปรุงข้อมูลพร้อมกันให้พนักงานทุกคน

PROCEDURE S0204

_ADDING = 0

@ 5,5 SAY "จำนวนเปอร์เซ็นต์ที่เพิ่มเงินเดือนให้พนักงาน : " GET _ADDING

READ

IF ASKUPD() = 1

USE EMPL

REPLACE ALL EMPLSAL WITH EMPLSAL + (EMPLSAL * _ADDING / 100)

CLOSE

ENDIF

RETURN

: ตัวอย่างที่ 6.16**& _พลัฟธ์**

// ปรับปรุงแฟ้มต่าง ๆ อย่างง่าย โดยเลือกแฟ้มจากตัวเลือก แล้วเปิดแฟ้ม

// จากนั้นใช้ฟังก์ชัน BROWSE ช่วยให้การปรับปรุงข้อมูลง่ายขึ้นมาก

PROCEDURE S0205

OMENU = ALERTNEW("เลือกปรับปรุงแฟ้มตามต้องการ";

{ "เพิ่มสิ่งสินค้า", "เพิ่มสิ่งสินค้าละเอียด", "เพิ่มสินค้า";

"เพิ่มพนักงาน", "เพิ่มลูกค้า", "เพิ่มบริษัทรับส่งของ";

"เพิ่มประเภทสินค้า", "เพิ่มผู้ผลิต", "กลับเมนู"})

DO CASE

CASE OMENU = 1 ; USE ORDER

CASE OMENU = 2 ; USE ORDERD

CASE OMENU = 3 ; USE PRO

CASE OMENU = 4 ; USE EMPL

CASE OMENU = 5 ; USE CUST

CASE OMENU = 6 ; USE SHIPPER

CASE OMENU = 7 ; USE CAT

CASE OMENU = 8 ; USE SUP

ENDCASE

IF OMENU >= 1 .AND. OMENU <= 8

```

BROWSE(5,5,22,75)
PACK
ENDIF
CLOSE ALL
RETURN

```

6.4.4 โปรแกรมต่าง ๆ ใน SALE3.PRG

โปรแกรมใน SALE3.PRG นี้มีโปรแกรมย่อยภายในทั้งหมด 8 โปรแกรม โดยรวมโปรแกรมที่เกี่ยวข้องกับระบบรายงาน เช่น โปรแกรมพิมพ์รายงานยอดขายของสินค้า ตามเงื่อนไขต่าง ๆ โปรแกรมพิมพ์รายงานใบสั่งซื้อสินค้าตามเงื่อนไขต่าง ๆ และโปรแกรมสรุปใบสั่งซื้อของแต่ละวัน

ระบบรายงานเป็นระบบที่ช่วยทำให้เห็นคุณค่า และผลงานจากระบบงานขาย ทำให้ผู้บริหารบริหารสินค้าในคลัง วางแผนงานหรือนโยบายทางการตลาดได้อย่างถูกต้อง เพราะรายงานจะแสดงยอดขาย แสดงการรับบริการของลูกค้า เห็นยอดจำหน่ายของสินค้าแต่ละประเภท จัดการกับสินค้าที่ขายไม่ได้ หรือสินค้าที่ขายดี ให้ขายดียิ่ง ๆ ขึ้น เพื่อให้บริษัทเจริญรุ่งเรือง และเติบโตต่อไป

ในหัวข้อนี้จะแสดงตัวอย่างโปรแกรมต่าง ๆ ที่มีในโปรแกรม SALE3.PRG ทั้งหมด 8 โปรแกรม โดยเริ่มจากตัวอย่างที่ 6.17 ถึง ตัวอย่างที่ 6.24 (R0301 - R0308)

: ตัวอย่างที่ 6.17

& walph

```

// พิมพ์สรุปยอดขายของสินค้า โดยระบุเดือนที่สั่งซื้อสินค้า
// รายงานแสดงรหัสสินค้า และจำนวนเงิน ที่ได้ทั้งหมด ตามเดือน

```

```
PROCEDURE S0301
```

```
USE ORDERD
```

```
SORT ON PRO,ORDER TO TEMPF
```

```
USE TEMPF
```

```
COPY TO ORDERD
```

```
_MONTH = 0
```

```
@ 5,5 SAY "เดือนที่ต้องการ : " GET _MONTH PICT "99"
```

```
READ
```

```
SELE 1 ; USE ORDER
```

```
SELE 2 ; USE ORDERD
```

```
_PROO = 0
```

```
_TQUANT = 0
```

```

CHKLINE = 1
DO WHILE !EOF()
  SELE 2 // เพิ่มใบสั่งสินค้าละเอียด
  _ORDER = ORDER
  SELE 1 // เพิ่มใบสั่งสินค้า
  _FOUNDM = 0
  LOCATE FOR ORDER = _ORDER
  IF FOUND()
    IF MONTH(DATEORD) = _MONTH // ตรวจสอบเดือนที่ต้องการ
      _FOUNDM = 1
    ENDIF
  ENDIF
  SELE 2 // เพิ่มใบสั่งสินค้า
  IF _FOUNDM = 1
    IF PRO != _PROO // ถ้าต่างกลุ่ม
      ? PRO
      _PROO = PRO
      CHKLINE = 0
    ENDIF
    _TQUANT = _TQUANT + OQUANT
  ENDIF
  SKIP
  IF (_PROO != PRO .OR. EOF()) .AND. CHKLINE = 0
    ?? _TQUANT // ถ้าเปลี่ยนกลุ่ม
    _TQUANT = 0
    CHKLINE = 1 // แสดงการพิมพ์บรรทัดแต่ละบรรทัดสมบูรณ์
  ENDIF
ENDDO
CLOSE ALL
INKEY(3)
RETURN

```

: ตัวอย่างที่ 6.18**& _พลัฟธ์**

// พิมพ์สรุปยอดขายของสินค้า โดยระบุผู้ผลิตสินค้า

// รายงานแสดงรหัสสินค้า ชื่อสินค้า ปริมาณขาย และจำนวนเงินของสินค้า ตามผู้ผลิต

PROCEDURE S0302

_SUP = 0

@ 5,5 SAY "ผู้ผลิตสินค้า : " GET _SUP PICT "999"

READ

SELE 1 ; USE PRO INDEX PRO1

SELE 2 ; USE ORDERD

INDEX ON PRO TO ORDERD2

SET RELATION TO PRO INTO PRO

_PROO = 0

_TQUANT = 0

_TMONEY = 0

DO WHILE !EOF()

IF PRO->SUP = _SUP

IF _PROO != PRO // ถ้าต่างกลุ่ม

? PRO,PRO->PRONAME

_PROO = PRO

ENDIF

_TQUANT = _TQUANT + OQUANT

_TMONEY = _TMONEY + (OPRICE * OQUANT - ODISC)

SKIP

IF _PROO != PRO .OR. EOF()

?? _TQUANT,_TMONEY // ถ้าเปลี่ยนกลุ่ม

_TQUANT := _TMONEY := 0

ENDIF // ตรวจสอบกลุ่มของสินค้า

ELSE // ถ้าไม่ใช่ผู้ผลิตที่ต้องการ

SKIP

ENDIF // ตรวจสอบผู้ผลิต

ENDDO

CLOSE ALL

INKEY(3)

RETURN

: ตัวอย่างที่ 6.19

& _พลัพ์ธ

// พิมพ์สรุปยอดขายของสินค้า โดยระบุผู้ผลิตสินค้า และเดือนที่สั่ง
 // รายงานแสดงรหัสสินค้า ชื่อสินค้า และปริมาณ ตามผู้ผลิตและเดือน
 // และแสดงให้เห็นการเชื่อมแฟ้มสินค้า ใบสั่งสินค้า และใบสั่งสินค้าละเอียด

PROCEDURE S0303

_SUP := _MONTH := 0

@ 5,5 SAY "เดือนที่ต้องการ : " GET _MONTH PICT "99"

@ 6,5 SAY "ผู้ผลิตสินค้า : " GET _SUP PICT "999"

READ

SELE 1 ; USE PRO INDEX PRO1

SELE 2 ; USE ORDER

SELE 3 ; USE ORDERD

INDEX ON PRO TO ORDERD2

SET RELATION TO PRO INTO PRO

_PROO = 0

_TQUANT = 0

CHKLINE = 1

DO WHILE !EOF()

_ORDER = ORDER

SELE 2 // แฟ้มใบสั่งสินค้า

_FOUNDM = 0

LOCATE FOR _ORDER = ORDER .AND. MONTH(DATEORD) = _MONTH

IF FOUND()

_FOUNDM = 1

ENDIF

SELE 3 // แฟ้มใบสั่งสินค้า

IF _FOUNDM = 1 .AND. PRO->SUP = _SUP

IF PRO != _PROO // ถ้าต่างกลุ่ม

? PRO,PRO->PRONAME

_PROO = PRO

CHKLINE = 0

ENDIF


```

    _TQUANT = _TQUANT + OQUANT
ENDIF
SKIP
IF (_PROO != PRO .OR. EOF()) .AND. CHKLINE = 0
    ?? _TQUANT                // ถ้าเปลี่ยนกลุ่ม
    _TQUANT = 0
    CHKLINE = 1                // แสดงการพิมพ์บรรทัดแต่ละบรรทัดสมบูรณ์
ENDIF
ENDDO
CLOSE ALL ; INKEY(3)
RETURN

```

: ตัวอย่างที่ 6.20

& ผลลัพธ์

```

// พิมพ์สรุปยอดขายของประเภทสินค้า โดยระบุเดือนที่สั่ง
// รายงานแสดงรหัสใบสั่ง รหัสสินค้า ชื่อสินค้า และจำนวนเงินในใบสั่ง ตามประเภท
// แล้วสรุปยอดขายของสินค้าแต่ละประเภท
// และแสดงให้เห็นการเชื่อมแฟ้มสินค้า ใบสั่งสินค้า และใบสั่งสินค้าละเอียด

```

PROCEDURE S0304

```

    _MONTH := 0
    @ 5,5 SAY "เดือนที่ต้องการ : " GET _MONTH PICT "99"
    READ
    SELE 1 ; USE PRO INDEX PRO1
    SELE 2 ; USE ORDER
    SELE 3 ; USE ORDERD INDEX ORDERD2
    SET RELATION TO PRO INTO PRO
    SELE 4 ; USE CAT
    _CAT = 0
    _TQUANT = 0
    DO WHILE !EOF()
        _CAT = CAT
        ? CAT,CATNAME
        SELE 3                // แฟ้มใบสั่งสินค้าละเอียด
        SET FILTER TO PRO->CAT = _CAT // จับกลุ่มให้ประเภทสินค้า
        GO TOP
    
```

```

_TQUANT = 0
DO WHILE !EOF()
  _ORDER = ORDER
  SELE 2                      // เพิ่มใบสั่งสินค้า
  LOCATE FOR ORDER = _ORDER .AND. MONTH(DATEORD) = _MONTH
  IF FOUND()
    SELE 3
    _TQUANT = _TQUANT + OQUANT
    ? ORDER,PRO,PRO->PRONAME,OQUANT
  ENDIF
  SELE 3
  SKIP
ENDDO
SELE 4                      // เพิ่มประเภทสินค้า
? "   จำนวน = ",_TQUANT
? "===== "
SKIP
ENDDO
CLOSE ALL
INKEY(3)
RETURN

```

: ตัวอย่างที่ 6.21

& ผลลัพธ์

```

// พิมพ์สรุปยอดขายใบสั่งสินค้าแต่ละใบ โดยระบุวันที่
// รายงานแสดงลำดับ รหัสใบสั่ง และยอดขายของใบสั่งสินค้า ตามวันที่
// โดยการหายยอดขายจะใช้ประโยชน์จากตัวแปรอาเรย์มาเก็บข้อมูล
// แล้วจึงนำข้อมูลจากอาเรย์มาพิมพ์อีกครั้งหนึ่ง

```

```

PROCEDURE S0305
  SELE 1 ; USE ORDERD
  SELE 2 ; USE ORDER
  DO WHILE LASTKEY() != 27
    _DATEORD = CTOD("")
    @ 5,5 SAY "วันที่ที่มีการสั่งสินค้า : " GET _DATEORD
  
```

```

READ
SELE 2
COUNT TO CNTORDER FOR DATEORD = _DATEORD
IF CNTORDER > 0
  @ 6,5 SAY "จำนวนใบสั่งซื้อ : " + STR(CNTORDER)
  SET FILTER TO DATEORD = _DATEORD
  GO TOP
  ARORDER = {}
  DO WHILE !EOF()
    AADD(ARORDER,ORDER)
    SKIP
  ENDDO
  ARMONEY = ARRAY(CNTORDER)
  AFILL(ARMONEY,0)
  SELE 1          // เริ่มหาจำนวนเงินของใบสั่งแต่ละใบ ในแฟ้มใบสั่งสินค้าละเอียด
  FOR I = 1 TO CNTORDER
    SET FILTER TO ORDER = ARORDER[I]
    GO TOP
    _MONEY = 0
    DO WHILE ! EOF()
      _MONEY = _MONEY + OPRICE * OQUANT - ODISC
      SKIP
    ENDDO
    ARMONEY[I] = _MONEY
  NEXT
  @ 8,5 SAY "ลำดับ รหัสใบสั่งสินค้า จำนวนเงิน"
  FOR I = 1 TO CNTORDER
    @ 8+I,5 SAY STR(I,4)+STR(ARORDER[I],6)+STR(ARMONEY[I],9)
  NEXT
ELSE
  @ 8,5 SAY "NOT FOUND"
ENDIF
INKEY(0)

```

```

@ 8,5 CLEAR TO 20,70
ENDDO
RETURN
: ตัวอย่างที่ 6.22
& _พลัฟธ์
// พิมพ์สรุปยอดขายใบสั่งสินค้าแต่ละใบ โดยระบุรหัสพนักงานขาย
// รายงานแสดงลำดับ รหัสใบสั่ง และยอดขายของใบสั่งสินค้า ตามรหัสพนักงานขาย
// ในการป้อนรหัสพนักงานขาย ต้องป้อนจนกว่าจะมีรหัสนั้นในแฟ้มจริง ๆ จึงจะยอมรับ
// โดยการหายยอดขายจะใช้ประโยชน์จากตัวแปรอาเรย์มาเก็บข้อมูล
// แล้วจึงนำข้อมูลจากอาเรย์มาพิมพ์อีกครั้งหนึ่ง ซึ่งเลือกปลายทางของผลลัพธ์ได้

```

PROCEDURE S0306

```

SELE 1 ; USE EMPL
SELE 2 ; USE ORDERD
SELE 3 ; USE ORDER
DO WHILE LASTKEY() != 27
  _EMPL = 0
  SELE 1
  DO WHILE LASTKEY() != 27
    @ 5,5 SAY "รหัสพนักงาน : " GET _EMPL PICT "9999"
    READ
    LOCATE FOR EMPL = _EMPL
    IF FOUND()
      @ 7,5 SAY EMPLNAME
      EXIT
    ENDIF
  ENDDO
  SELE 3
  COUNT TO CNTEMPL FOR EMPL = _EMPL
  IF CNTEMPL > 0
    GASKPRT = ASKPRT() // สอบถามการนำออกเครื่องพิมพ์
    IF GASKPRT = 2
      SET DEVICE TO PRINTER
      @ 5,5 SAY "รหัสพนักงาน : "+STR(_EMPL)
    ENDIF
  ENDIF

```

```

@ 6,5 SAY "จำนวนใบสั่งซื้อ : " + STR(CNTEMPL)
SET FILTER TO EMPL = _EMPL
GO TOP
ARORDER = {}
DO WHILE !EOF()
  AADD(ARORDER,ORDER)
  SKIP
ENDDO
ARMONEY = ARRAY(CNTEMPL)
AFILL(ARMONEY,0)
_TMONEY = 0
SELE 2                                     // เริ่มหาจำนวนเงินของใบสั่งแต่ละใบ ในแฟ้มใบสั่งสินค้าละเอียด
FOR I = 1 TO CNTEMPL
  SET FILTER TO ORDER = ARORDER[I]
  GO TOP
  _MONEY = 0
  DO WHILE ! EOF()
    _MONEY = _MONEY + OPRICE * OQUANT - ODISC
    SKIP
  ENDDO
  ARMONEY[I] = _MONEY
  _TMONEY = _TMONEY + _MONEY
NEXT
@ 8,5 SAY "ลำดับ รหัสใบสั่งสินค้า จำนวนเงิน"
FOR I = 1 TO CNTEMPL
  @ 8+I,5 SAY STR(I,4)+STR(ARORDER[I],6)+STR(ARMONEY[I],9)
NEXT
@ 9+CNTEMPL,5 SAY "รวมเงิน : "+STR(_TMONEY)
IF GASKPRT = 2
  EJECT
ENDIF
SET DEVICE TO SCREEN
ELSE

```

```

    @ 8,5 SAY "NOT FOUND"
ENDIF
INKEY(0)
@ 7,5 CLEAR TO 20,70
ENDDO
RETURN
: ตัวอย่างที่ 6.23
& ผลลัพธ์
// พิมพ์สรุปยอดขายสินค้า โดยระบุประเภท
// รายงานแสดงยอดขายของสินค้าแต่ละรายการในแฟ้มไปสั่งสินค้าละเอียด ตามประเภท
// โดยแสดงลำดับ เลขที่ไปสั่งซื้อ ชื่อสินค้า และจำนวนเงิน บรรทัดสุดท้ายแสดงยอดรวม
// สามารถเลือกปลายทางของผลลัพธ์ได้
PROCEDURE S0307
    _CAT := 0
    @ 5,5 SAY "ประเภทสินค้า : " GET _CAT PICT "99"
    READ
    USE PRO INDEX PRO1
    USE ORDERD NEW
    SET RELATION TO PRO INTO PRO
    SET FILTER TO PRO->CAT = _CAT
    COUNT TO CNTCAT
    GO TOP
    IF CNTCAT > 0
        GASKPRT = ASKPRT()
        IF GASKPRT = 2
            SET PRINT ON
        ENDIF
        SETPOS(7,5); ? _CAT,":",CNTCAT," RECORDS"
        I = 1 ; _TMONEY = 0
        DO WHILE !EOF()
            _MONEY = OPRICE * OQUANT - ODISC
            _TMONEY = _TMONEY + _MONEY
            SETPOS(I+7,5); ? I,ORDER,LEFT(PRO->PRONAME,20),_MONEY
            SKIP
    
```

```

I++
ENDDO
SETPOS(I+7,5); ? "เป็นเงินทั้งหมด = ",_TMONEY
IF GASKPRT = 2
  EJECT
  SET PRINT OFF
ENDIF
ELSE
  @ 6,5 SAY "NOT FOUND"
ENDIF
INKEY(5)
@ 6,5 CLEAR TO 25,75
CLOSE ALL
RETURN
: ตัวอย่างที่ 6.24
& _ผลลัพธ์
// พิมพ์รายการขายแต่ละรายการ โดยระบุวันที่
// รายงานแสดงลำดับ รหัสสินค้า ราคา ปริมาณ และส่วนลด ตามวันที่
// และได้ผลลัพธ์ทางจอภาพเท่านั้น
PROCEDURE S0308
  _DATEORD := CTOD("")
  @ 5,5 SAY "วันที่ลูกค้าสั่ง : " GET _DATEORD
  READ
  SELE 1 ; USE ORDERD
  INDEX ON STR(ORDER)+STR(PRO) TO ORDERD1
  SELE 2 ; USE ORDER
  SET FILTER TO DATEORD = _DATEORD
  COUNT TO CNTORD
  IF CNTORD > 0
    GO TOP ; SETPOS(7,1)
    DO WHILE !EOF()
      _ORDER = ORDER
      SELE 1
      SET FILTER TO ORDER = _ORDER

```

```
? "*" ,_ORDER
GO TOP ; I = 1
DO WHILE !EOF()
  ? I,PRO,OPRICE,OQUANT,ODISC
  SKIP ; I++
ENDDO
? "===== "
SELE 2 ; SKIP
ENDDO
ELSE
  @ 8,5 SAY "NOT FOUND"
ENDIF
INKEY(3) ; CLOSE ALL
RETURN
```


6.4.5 โปรแกรมต่าง ๆ ใน SALE4.PRG

โปรแกรมใน SALE4.PRG นี้มีโปรแกรมย่อยภายในทั้งหมด 7 โปรแกรม โดยรวมโปรแกรมที่เกี่ยวกับระบบสินค้าคงคลัง เช่น โปรแกรมพิมพ์รายงานสินค้าที่ถึงจุดสั่งซื้อ โปรแกรมเพิ่มจำนวนสินค้า โปรแกรมเพิ่มรายการสินค้าใหม่ในคลัง โปรแกรมรายงานเกี่ยวกับจำนวนสินค้าแบบต่าง ๆ

ระบบสินค้าคงคลังเป็นระบบที่ช่วยทำให้บริษัทมีสินค้าจำหน่ายตลอดเวลาเมื่อลูกค้าถาม และไม่มีสินค้าในคลังมากเกินไป เพราะการสั่งซื้อสินค้ามาก ๆ ทำให้เงินทุนจมอยู่ในคลัง ซึ่งเป็นการบริหารเงินที่ไม่ดีนัก ดังนั้นในระบบนี้จึงช่วยผู้บริหารใจการตัดสินใจซื้อสินค้าเพิ่มเมื่อใด เท่าใด และจากใคร

ในหัวข้อนี้จะแสดงตัวอย่างโปรแกรมต่าง ๆ ที่มีในโปรแกรม SALE4.PRG ทั้งหมด 7 โปรแกรม โดยเริ่มจากตัวอย่างที่ 6.24 ถึง ตัวอย่างที่ 6.30 (R0401 - R0407)

: ตัวอย่างที่ 6.24

& _พลัฟร์

// เพิ่มยอดสินค้าเข้ามาในคลังสินค้า ของสินค้าแต่ละรายการ

// มักกระทำเมื่อสินค้าในคลังหมด แล้วซื้อเพิ่ม จึงต้องเพิ่มจำนวนสินค้าในแฟ้ม

// โดยเพิ่มยอดสินค้าได้ที่ละรายการ หลายรายการ จนกว่าจะกดปุ่ม ESC

PROCEDURE S0401

```
USE PRO INDEX PRO1
```

```
LN = 5
```

```
DO WHILE LASTKEY() != 27
```

```
  _PRO := _PROREST := 0
```

```
  @ LN,10 SAY "รหัสสินค้า : " GET _PRO PICT "9999" ; READ
```

```
  SEEK _PRO
```

```
  IF FOUND()
```

```
    @ LN,30 SAY LEFT(PRONAME,15)
```

```
    @ LN,50 SAY "เพิ่มจำนวนสินค้าอีก : " GET _PROREST PICT "999"
```

```
    READ
```

```
    IF ASKUPD() = 1
```

```
      REPLACE PROREST WITH PROREST + _PROREST,PROONORD WITH 0
```

```
    ENDIF
```

```
    LN++
```

```
  ELSE
```

```
    @ LN,50 SAY "NOT FOUND"
```

```
  ENDIF
```

```
ENDDO
```

```
RETURN
```

: ตัวอย่างที่ 6.25**& _พลัฟธ์**

// เพิ่มข้อมูลสินค้าใหม่เข้าไปในแฟ้มสินค้า เมื่อมีสินค้าชนิดใหม่ที่ไม่เคยมีจำหน่าย

// เมื่อรอรับข้อมูลจะแสดงรหัสสินค้ารายการใหม่มาให้อัตโนมัติ

// โดยเพิ่มรายการสินค้าได้ที่ละรายการ หลายรายการจนกว่าจะกดปุ่ม ESC

PROCEDURE S0402

```
USE PRO INDEX PRO1
```

```
_PRO = 1
```

```
DO WHILE _PRO != 0 .AND. LASTKEY() !=27
```

```
GO BOTTOM
```

```
_PRO = PRO + 1
```

```
_SUP = 0 ; _CAT = 0
```

```
_PRONAME = SPACE(40)
```

```
_PROUNIT = SPACE(10)
```

```
_PROPRICE := _PROREST := _PROONORD := _REORDER := 0
```

```
@ 10,5 SAY "รหัสสินค้า" GET _PRO PICT "9999"
```

```
@ 11,5 SAY "รหัสผู้ผลิต" GET _SUP PICT "999"
```

```
@ 12,5 SAY "รหัสประเภท" GET _CAT PICT "99"
```

```
@ 13,5 SAY "ชื่อสินค้า" GET _PRONAME
```

```
@ 14,5 SAY "ชื่อหน่วย" GET _PROUNIT
```

```
@ 15,5 SAY "ราคาทุน" GET _PROPRICE PICT "9999.99"
```

```
@ 16,5 SAY "จำนวนที่เหลือ" GET _PROREST PICT "9999"
```

```
@ 17,5 SAY "จำนวนที่กำลังสั่งเพิ่ม" GET _PROONORD PICT "9999"
```

```
@ 18,5 SAY "จุดสั่งซื้อ" GET _REORDER PICT "999"
```

```
READ
```

```
IF UPDATED() .AND. ASKUPD() = 1
```

```
APPEND BLANK
```

```
REPLACE PRO WITH _PRO,SUP WITH _SUP,CAT WITH _CAT,;
```

```
PRONAME WITH _PRONAME,PROUNIT WITH _PROUNIT,;
```

```
PROPRICE WITH _PROPRICE,PROREST WITH _PROREST,;
```

```
PROONORD WITH _PROONORD
```

```
ENDIF
```

```
ENDDO
```

```
RETURN
```

: ตัวอย่างที่ 6.26**& แพลตฟอร์ม**

// พิมพ์รายการสินค้าที่ ถึงจุดสั่งซื้อ เรียงตามรหัสสินค้า

// รายงานแสดงลำดับ รหัสสินค้า ชื่อ จำนวนเหลือ จำนวนสั่งเพิ่ม และจุดสั่งซื้อ

PROCEDURE S0403

USE PRO INDEX PRO1

REINDEX

COUNT TO CNTREORDER FOR PROREST < REORDER

@ 5,5 SAY "จำนวนสินค้าที่ถึงจุดสั่งซื้อ = " + STR(CNTREORDER)

IF CNTREORDER > 0

CLS ; GO TOP ; I = 1

DO WHILE !EOF()

IF PROREST < REORDER

@ ROW()+1,2 SAY I

@ ROW(),15 SAY PRO

@ ROW(),25 SAY PRONAME

@ ROW(),35 SAY PROREST

@ ROW(),45 SAY PROONORD

@ ROW(),55 SAY REORDER

I++

ENDIF

SKIP

ENDDO

ELSE

@ 6,5 SAY "ยังไม่มีสินค้าชนิดใดถึงจุดสั่งซื้อ"

ENDIF

INKEY(3)

CLOSE ALL

RETURN

: ตัวอย่างที่ 6.27**& _พลัฟธ์**

// พิมพ์รายการสินค้าที่ ถึงจุดสั่งซื้อตามผู้ผลิต และยังไม่ได้สั่งเพิ่ม

// รายงานแสดงลำดับ รหัสสินค้า ชื่อ จำนวนเหลือ จำนวนสั่งเพิ่ม จุดสั่งซื้อ

// รายงานหยุดให้ดูหน้าละ 20 รายการ

PROCEDURE S0404

_SUP = 0

@ 5,5 SAY "รหัสผู้ผลิต : " GET _SUP PICT "999"

READ

USE PRO INDEX PRO1

REINDEX

CLS ; I:= 0 ; J:= 0

DO WHILE !EOF()

IF REORDER > PROREST .AND. SUP = _SUP .AND. PROONORD = 0

I++ ; J++

@ J,2 SAY I

@ J,15 SAY PRO

@ J,25 SAY PRONAME

@ J,35 SAY PROREST

@ J,45 SAY PROONORD

@ J,55 SAY REORDER

IF MOD(I,20) = 0

INKEY(3)

J:= 0

CLS

ENDIF

ENDIF

SKIP

ENDDO

INKEY(3)

CLOSE ALL

RETURN

: ตัวอย่างที่ 6.28**& ผลลัพธ์**

// พิมพ์รายการสินค้าที่เหลือไม่ถึง 10 เปอร์เซ็นต์จะถึงจุดสั่งซื้อ

// รายงานแสดงลำดับ รหัสสินค้า ชื่อ จำนวนเหลือ จำนวนสั่งเพิ่ม จุดสั่งซื้อ

// รายงานหยุดให้ดูหน้าละ 20 รายการ

// สามารถเลือกปลายทางของผลลัพธ์ได้

PROCEDURE S0405

USE PRO INDEX PRO1

REINDEX

COUNT TO CNTREORDER FOR PROREST < REORDER * 1.1

@ 5,5 SAY "จำนวนสินค้าใกล้ ถึงจุดสั่งซื้อ =" + STR(CNTREORDER)

IF CNTREORDER > 0

INKEY(1); CLS

GASKPRT = ASKPRT()

IF GASKPRT = 2 .AND. ISPRINTER() // ฟังก์ชันตรวจการมีอยู่ของเครื่องพิมพ์

SET DEVICE TO PRINTER

ENDIF

SET FILTER TO PROREST < REORDER * 1.1

GO TOP

J = 3

FOR I = 1 TO CNTREORDER

@ J,2 SAY STR(I) + "|"

@ J,15 SAY STR(PRO) + "|"

@ J,25 SAY LEFT(PRONAME,8) + "|"

@ J,35 SAY STR(PROREST) + "|"

@ J,45 SAY STR(PROONORD) + "|"

@ J,55 SAY STR(REORDER) + "|"

SKIP

J++

IF J > 22

INKEY(0); CLS

J = 3

IF GASKPRT = 2 // สำหรับเครื่องพิมพ์

EJECT

```

    ENDIF
  ENDIF
NEXT
IF GASKPRT = 2
  EJECT ; SET DEVICE TO SCREEN
ENDIF
ELSE
  @ 6,5 SAY "ยังไม่มีสินค้าชนิดใดใกล้ ถึงจุดสั่งซื้อ"
ENDIF
INKEY(3)
CLOSE ALL
RETURN

```

: ตัวอย่างที่ 6.29

& ผลลัพธ์

```

// พิมพ์รายการสินค้าที่กำลังสั่งซื้อเพิ่มปัจจุบัน โดยเรียงตามชื่อสินค้า
// รายงานเลือกเรคคอร์ดที่ จำนวนสั่งเพิ่มมากกว่า 0
// รายงานแสดงลำดับ รหัสสินค้า จำนวนเหลือ จำนวนสั่งเพิ่ม จุดสั่งซื้อ

```

PROCEDURE S0406

```

USE PRO
SORT ON PRONAME TO TEMPF
USE TEMPF
COPY TO PRO
USE PRO
I = 1
DO WHILE !EOF()
  IF PROONORD > 0
    ? I,PRO,PROREST,PROONORD,REORDER
    I++
  ENDIF
  SKIP
ENDDO
INKEY(3)
CLOSE
RETURN

```

: ตัวอย่างที่ 6.30**& แพลตฟอร์ม**

// พิมพ์รายการสินค้าที่ขายดีจากมากไปน้อย ตามช่วงวันที่
 // โดยระบุช่วงวันเริ่มต้น และวันสิ้นสุด ของวันสั่งสินค้า
 // รายงานแสดงเลขที่ใบสั่ง ปริมาณขาย และวันที่สั่ง
 // โดยข้อมูลถูกจัดเรียงตามปริมาณขาย จากมากไปน้อย

PROCEDURE S0407

USE ORDERD

SORT ON OQUANT/D TO TEMPF

USE TEMPF

COPY TO ORDERD

SELE 1 ; USE PRO INDEX PRO1

SELE 2 ; USE ORDER

INDEX ON ORDER TO ORDER1

SELE 3 ; USE ORDERD

SET RELATION TO ORDER INTO ORDER,PRO INTO PRO

_DATEORD1 := _DATEORD2 := CTOD("")

@ 5,5 SAY "วันที่สั่งสินค้าเริ่มต้น = " GET _DATEORD1

@ 6,5 SAY "วันที่สั่งสินค้าสิ้นสุด = " GET _DATEORD2

READ

SET FILTER TO ORDER->DATEORD >= _DATEORD1 .AND. ;

ORDER->DATEORD <= _DATEORD2

GO TOP

_PROO = 0

DO WHILE !EOF()

IF PRO != _PROO

? "*" ,PRO,LEFT(PRO->PRONAME,10)

_PROO = PRO

ENDIF

? ORDER,OQUANT,ORDER->DATEORD

SKIP

ENDDO

INKEY(3) ; CLOSE ALL

RETURN

6.4.6 โปรแกรมต่าง ๆ ใน SALE5.PRG

โปรแกรมใน SALE5.PRG นี้มีโปรแกรมน้อยภายในทั้งหมด 7 โปรแกรม โดยรวมโปรแกรมที่เกี่ยวกับระบบอื่น เช่น โปรแกรมพิมพ์สรุปค่าขนส่งให้บริษัทขนส่ง โปรแกรมพิมพ์จดหมายขอโทษที่ส่งสินค้าล่าช้ากว่าวันที่ลูกค้าต้องการสินค้า โปรแกรมพิมพ์ข้อมูลจากแฟ้มอย่างง่าย ๆ โปรแกรมพิมพ์รายชื่อลูกค้าที่พนักงานรับผิดชอบ

ระบบอื่น เป็นระบบที่รวมงานต่าง ๆ ที่ไม่เข้ากับระบบย่อยอื่น ๆ ซึ่งเป็นงานที่หลายหลาย หากต้องการรวมงานในระบบนี้เข้ากับระบบรายงานก็ย่อมทำได้ เพราะโปรแกรมเหล่านี้เน้นการจัดทำรายงาน แต่ผู้เขียนไม่รวมเข้าไปเพราะงานในระบบรายงานมีมาก จึงแยกโปรแกรมบางอย่างมาสร้างระบบใหม่จนได้ระบบอื่นขึ้นมา

ในหัวข้อนี้จะแสดงตัวอย่างโปรแกรมต่าง ๆ ที่มีในโปรแกรม SALE5.PRG ทั้งหมด 7 โปรแกรม โดยเริ่มจากตัวอย่างที่ 6.31 ถึง ตัวอย่างที่ 6.37 (R0501 - R0507)

: ตัวอย่างที่ 6.31

& _พลัฟธ์

// พิมพ์สรุปค่าขนส่งให้บริษัทขนส่ง โดยระบุเดือน

// รายงานแสดงรหัสบริษัทขนส่ง ชื่อบริษัทขนส่ง และจำนวนเงินจากการขนส่ง

PROCEDURE S0501

_MONTH = 0

@ 5,5 SAY "เดือนที่ต้องการ : " GET _MONTH PICT "99"

READ

SELE 1 ; USE SHIPPER

INDEX ON SHIPPER TO SHIPPER1

SELE 2 ; USE ORDER

INDEX ON SHIPPER TO ORDER2

SET RELATION TO SHIPPER INTO SHIPPER

SET FILTER TO MONTH(ORDER) = _MONTH

GO TOP

_SHIPPERO = 0 ; _TFREIGHT = 0

DO WHILE !EOF()

IF SHIPPER != _SHIPPERO

? "=====

? SHIPPER,SHIPPER->SHIPNAME

_SHIPPERO = SHIPPER

ENDIF

? ORDER,DATEORD,FREIGHT

_TFREIGHT = _TFREIGHT + FREIGHT


```

SKIP
IF EOF() .OR. _SHIPPERO != SHIPPER
  ? "*" , _TFREIGHT
  _TFREIGHT = 0
ENDIF
ENDDO
INKEY(3)
CLOSE ALL
RETURN

```

: ตัวอย่างที่ 6.32

& ผลลัพธ์

```

// พิมพ์สถิติการใช้บริการบริษัทขนส่งของลูกค้าแต่ละราย โดยระบุช่วงวันที่
// แยกกลุ่มข้อมูลเป็น 2 ระดับ คือแยกรหัสลูกค้า แล้วจึงแยกบริษัทขนส่ง
// รายงานแสดงจำนวนเงินที่ลูกค้าแต่ละราย ใช้บริการแต่ละบริษัทเท่าใด
// สุดท้ายของลูกค้าแต่ละราย แสดงจำนวนเงินที่ลูกค้าต้องจ่ายรวมเท่าใด

```

```

PROCEDURE S0502
  _DATEORD1 := _DATEORD2 := CTOD("")
  @ 5,5 SAY "วันที่สั่งซื้อสินค้าเริ่มต้น = " GET _DATEORD1
  @ 6,5 SAY "วันที่สั่งซื้อสินค้าสิ้นสุด = " GET _DATEORD2
  READ
  USE ORDER
  INDEX ON STR(CUST)+STR(SHIPPER) TO ORDER3
  SET FILTER TO ORDER->DATEORD >= _DATEORD1 .AND. ;
    ORDER->DATEORD <= _DATEORD2
  GO TOP
  _CUSTO := _SHIPPERO := 0
  _TFREIGHTC := _TFREIGHTS := 0
  DO WHILE !EOF()
    IF _CUSTO != CUST
      ? "*****",CUST
      _CUSTO = CUST
      _SHIPPERO = 0
    ENDIF
    IF _SHIPPERO != SHIPPER

```

```

?" **,"SHIPPER
  _SHIPPERO = SHIPPER
ENDIF
_TFREIGHTC = _TFREIGHTC + FREIGHT
_TFREIGHTS = _TFREIGHTS + FREIGHT
SKIP
IF EOF() .OR. _SHIPPERO != SHIPPER
  ?? _TFREIGHTS
  _TFREIGHTS := 0
ENDIF
IF EOF() .OR. _CUSTO != CUST
  ?" TOTAL = ",_TFREIGHTC
  _TFREIGHTC := 0
ENDIF
ENDDO
INKEY(3)
CLOSE ALL
RETURN
: ตัวอย่างที่ 6.33
& _ผลลัพธ์
// พิมพ์จดหมายขอโทษที่ส่งสินค้าไม่ทันกำหนด โดยระบุเลขที่ใบส่งสินค้า
// สามารถเลือกปลายทางของผลลัพธ์ได้
PROCEDURE S0503
  _ORDER = 0
  @ 4,5 SAY "ระบุเลขที่ใบส่งสินค้า : " GET _ORDER PICT "99999"
  READ
  IF LASTKEY() != 27
    GASKPRT = ASKPRT()
    IF GASKPRT = 2
      SET DEVICE TO PRINTER
    ENDIF
    @ 5,5 SAY "เรียน ท่านผู้มีอุปการะคุณ"
    @ 6,5 SAY "ตามที่ท่านได้สั่งซื้อสินค้ากับบริษัทของเรา"
    @ 7,5 SAY "โดยมีรายการตามใบสั่งซื้อสินค้าเลขที่ "+STR(_ORDER)

```

```

@ 8,5 SAY "แล้วทางบริษัทของเราได้ส่งสินค้าให้ท่านช้ากว่ากำหนด"
@ 9,5 SAY "กระผมในฐานะประธานกรรมการของบริษัท"
@ 10,5 SAY "ใคร่ขออภัยเป็นอย่างสูง และได้ชำระค่าปรับมาพร้อมกันนี้"
@ 11,5 SAY "ท้ายนี้ กระผมขอให้คำมั่นว่าเราจะปรับปรุงระบบจัดส่ง"
@ 12,5 SAY "ให้เร็วทันกำหนดเวลาที่ท่านต้องการทุกครั้งไป"
@ 13,5 SAY " จึงเรียนมาเพื่อกราบขออภัยมา ณ ที่นี้"
@ 14,5 SAY "           ประธานบริษัทขายส่ง จำกัด"
IF GASKPRT = 2
  EJECT
  SET DEVICE TO SCREEN
ENDIF
INKEY(3)
ENDIF
RETURN

```

: ตัวอย่างที่ 6.34

& _พลัฟธ์

```

// พิมพ์ชื่อลูกค้าที่พนักงานแต่ละคนรับผิดชอบ โดยระบุช่วงวันที่
// รายงานแยกกลุ่มข้อมูลตามรหัสพนักงาน
// รายงานแสดงรหัสและชื่อพนักงาน รหัสลูกค้า ชื่อลูกค้า เลขที่ใบสั่ง วันที่สั่ง

```

PROCEDURE S0504

```

  _DATEORD1 := _DATEORD2 := CTOD("")
@ 5,5 SAY "วันที่สั่งสินค้าเริ่มต้น = " GET _DATEORD1
@ 6,5 SAY "วันที่สั่งสินค้าสิ้นสุด = " GET _DATEORD2
READ
IF LASTKEY() != 27
  SELE 1 ; USE EMPL INDEX EMPL1
  SELE 2 ; USE CUST
    INDEX ON CUST TO CUST1
  SELE 3 ; USE ORDER
    INDEX ON EMPL TO ORDER4
  SET RELATION TO CUST INTO CUST,EMPL INTO EMPL
  SET FILTER TO DATEORD >= _DATEORD1 .AND. ;
    DATEORD <= _DATEORD2
GO TOP

```

```

_EMPLO = 0
DO WHILE !EOF()
  IF EMPL != _EMPLO
    ? "*",EMPL,EMPL->EMPLNAME
    _EMPLO = EMPL
  ENDIF
  ? "-",CUST,CUST->CUSTNAME,ORDER,DATEORD
  SKIP
ENDDO
INKEY(3)
CLOSE ALL
ENDIF
RETURN
: ตัวอย่างที่ 6.35
& _พลัฟธ์
// พิมพ์เลขที่ใบสั่งสินค้าที่ส่งไม่ทันกำหนด โดยระบุเดือน
// เลือกเดือนที่ตรงกับวันส่ง วันขนส่ง หรือวันที่ลูกค้าต้องการ
// รายงานแสดงข้อมูลทุกฟิลด์จากแฟ้มใบสั่งสินค้า
PROCEDURE S0505
  _MONTH := 0
  @ 5,5 SAY "เดือนที่ต้องการรายงาน = " GET _MONTH PICT "99"
  READ
  IF LASTKEY() != 27
    USE ORDER
    SET FILTER TO (MONTH(DATEORD) = _MONTH .OR. ;
      MONTH(DATEREQ) = _MONTH .OR. ;
      MONTH(DATESHIP) = _MONTH) .AND. ;
      DATESHIP > DATEREQ
    COUNT TO CNTREC
    IF CNTREC > 0
      GO TOP
      DO WHILE !EOF()
        ? ORDER,CUST,EMPL,DATEORD,DATEREQ,DATESHIP,SHIPPER,FREIGHT
        SKIP

```

```

ENDDO
ELSE
  @ 6,5 SAY "ไม่พบข้อมูลที่ถูกสั่งในเดือนนี้เลย"
ENDIF
INKEY(3)
CLOSE ALL
ENDIF
RETURN
: ตัวอย่างที่ 6.36
& _พลัพ์
// พิมพ์เลขที่ใบสั่งสินค้าที่มีกำหนดส่งไม่เกิน 10 วันนับจากวันนี้
// รายงานแสดงข้อมูลทุกฟิลด์ จากแฟ้มใบสั่งซื้อ
PROCEDURE S0506
  USE ORDER
  SET FILTER TO DATESHIP < DATE() + 10 .AND. ;
    DATESHIP > DATE() // วันที่ส่งต้องยังไม่ถึง
  COUNT TO CNTREC
  IF CNTREC > 0
    GO TOP
    DO WHILE !EOF()
      ? ORDER,CUST,EMPL,DATEORD,DATEREQ,DATESHIP,SHIPPER,FREIGHT
      SKIP
    ENDDO
  ELSE
    ? "ไม่มีรายการที่ต้องจัดส่งใน 10 วันข้างหน้า"
  ENDIF
  INKEY(3)
  CLOSE ALL
  RETURN
: ตัวอย่างที่ 6.37
& _พลัพ์
// พิมพ์ข้อมูลจากแฟ้มตามที่ต้องการ โดยเลือกแฟ้มจากตัวเลือก แล้วเปิดแฟ้ม
// นำข้อมูลในแฟ้มมาพิมพ์ โดยใช้ฟังก์ชันอ่านชื่อฟิลด์อัตโนมัติ
// สามารถเลือกปลายทางของผลลัพธ์ได้

```

PROCEDURE S0507

```

OMENU = ALERTNEW("เลือกปรับปรุงแฟ้มตามต้องการ";
  {"เพิ่มสิ่งสินค้า","เพิ่มสิ่งสินค้าละเอียด","เพิ่มสินค้า";
  "เพิ่มพนักงาน","เพิ่มลูกค้า","เพิ่มบริษัทรับส่งของ";
  "เพิ่มประเภทสินค้า","เพิ่มผู้ผลิต","กลับเมนู"})
IF MOD(OMENU,9) != 0
  GASKPRT = ASKPRT()
  IF !ISPRINTER()
    GASKPRT = 1
    @ 2,5 SAY "บอกให้ทราบว่า ... เครื่องพิมพ์ปิดอยู่"
  ENDIF
  MAXLINE = 20
  IF GASKPRT = 2
    SET DEVICE TO PRINTER
    MAXLINE = 35
    ? "รายงานผลทางเครื่องพิมพ์ ... โปรดรอสักครู่"
  ENDIF
  DO CASE
    CASE OMENU = 1 ; USE ORDER
    CASE OMENU = 2 ; USE ORDERD
    CASE OMENU = 3 ; USE PRO
    CASE OMENU = 4 ; USE EMPL
    CASE OMENU = 5 ; USE CUST
    CASE OMENU = 6 ; USE SHIPPER
    CASE OMENU = 7 ; USE CAT
    CASE OMENU = 8 ; USE SUP
  ENDCASE
  ARFLDNAME = ARRAY(AFIELDS()) // สร้างอาร์เรย์เท่าจำนวนฟิลด์ของแฟ้ม
  ARFLDTYPE = ARRAY(AFIELDS()) // สร้างอาร์เรย์เท่าจำนวนฟิลด์ของแฟ้ม
  ARFLDWIDTH= ARRAY(AFIELDS()) // สร้างอาร์เรย์เท่าจำนวนฟิลด์ของแฟ้ม
  AFIELDS(ARFLDNAME,ARFLDTYPE,ARFLDWIDTH) // เก็บชื่อฟิลด์
  LN := 1 ; NUM = 1
  DO WHILE !EOF()
    @ LN+2,2 SAY LTRIM(STR(NUM,3)) + " : "
    FOR I = 1 TO AFIELDS()
      IF GASKPRT = 2

```

```
@ LN+2,PCOL() SAY &(ARFLDNAME[I])
@ LN+2,PCOL() SAY "|"
ELSE
@ LN+2,COL() SAY &(ARFLDNAME[I])
@ LN+2,COL() SAY "|"
ENDIF
NEXT
IF MOD(LN,MAXLINE) = 0 .AND. LN != 0
IF GASKPRT = 2
EJECT
ENDIF
?"กดปุ่มใด ๆ เพื่อพิมพ์หน้าต่อไป"
LN = 0 ; INKEY(3) ; CLS
ENDIF
LN++ ; NUM++
SKIP
ENDDO
IF GASKPRT = 2
EJECT
SET DEVICE TO SCREEN
ENDIF
INKEY(5)
ENDIF
CLOSE ALL
RETURN
```