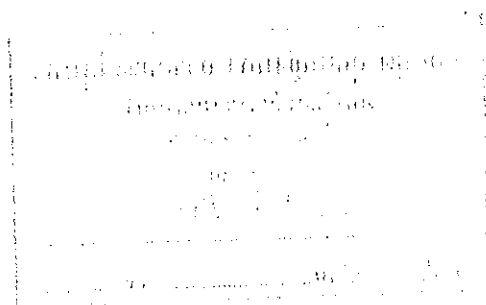




การสร้างตัวแปลภาษาสอบถามฐานข้อมูล ERAO  
An Implementation of ERAO Query Language Interpreter



สุภาภรณ์ กานต์สมเกียรติ  
Supaporn Karnsomkiat

วิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์  
มหาวิทยาลัยสงขลานครินทร์  
Master of Science Thesis in Computer Science  
Prince of Songkla University

2538

๒


เลขที่	QA ๒๑๖.๙.๐๖ ๘๙๐ ๒๕๓๘ ๐๒
ยวดยาน	๖๖๖๙

(1)

ชื่อวิทยานิพนธ์      การสร้างตัวแปลภาษาสอบถามฐานข้อมูล ERAO  
ผู้เขียน              นางสาวสุภาภรณ์ กานต์สมเกียรติ  
สาขาวิชา              วิทยาการคอมพิวเตอร์

คณะกรรมการที่ปรึกษา

คณะกรรมการสอบ

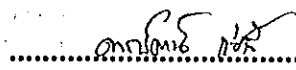
  
..... ประธานกรรมการ

  
..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.อัชฌา อีร์เชษฐมงคล)

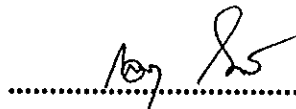
(ผู้ช่วยศาสตราจารย์ ดร.อัชฌา อีร์เชษฐมงคล)

  
..... กรรมการ

  
..... กรรมการ

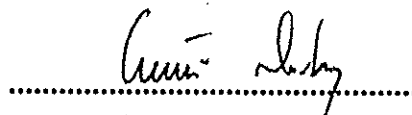
(อาจารย์ ดารารัตน์ แซ่ลี)

(อาจารย์ ดารารัตน์ แซ่ลี)

  
..... กรรมการ  
(อาจารย์ เลขา ไชยสร)

  
..... กรรมการ  
(ผู้ช่วยศาสตราจารย์ เสียง คูบุรัตน์)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้บัณฑิตวิทยานิพนธ์  
ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชา  
วิทยาการคอมพิวเตอร์

  
.....

(ดร. ไพรัตน์ สงวนไทร)

คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์      การสร้างตัวแปลภาษาสอบถามฐานข้อมูล ERAO  
ผู้เขียน                นางสาวสุภาภรณ์ กานต์สมเกียรติ  
สาขาวิชา              วิทยาการคอมพิวเตอร์  
ปีการศึกษา            2537

### บทคัดย่อ

ภาษาสอบถามฐานข้อมูลเชิงสัมพันธ์ถูกจัดแบ่งออกเป็น 2 แบบ : ภาษาแคลคูลัสสัมพันธ์ และภาษาพีชคณิตสัมพันธ์ ภาษาแคลคูลัสสัมพันธ์เป็นภาษาที่ใช้เพียงแต่บ่งบอกลักษณะของข้อมูลที่ต้องการโดยไม่ต้องบอกลำดับขั้นตอนของการได้มา ในขณะที่ภาษาพีชคณิตสัมพันธ์เป็นภาษาที่ใช้จะต้องบ่งบอกลำดับขั้นตอนของการดำเนินงานเพื่อให้ได้มาซึ่งข้อมูลที่ต้องการ ดังนั้นแต่ละข้อความสอบถามฐานข้อมูลด้วยภาษาแคลคูลัสสัมพันธ์จะมีชุดข้อความสั่งซึ่งประกอบด้วยหนึ่งข้อความสั่งขึ้นไปในภาษาพีชคณิตสัมพันธ์ที่สมนัยกันเสมอ

รายงานวิทยานิพนธ์นี้ดำเนินงานออกแบบรูปแบบข้อความสั่งในรูปแถวคำสั่ง และสร้างตัวแปลภาษาสอบถามฐานข้อมูลพีชคณิตเชิงสัมพันธ์ เพื่อศึกษาความสัมพันธ์ระหว่างแต่ละข้อความสอบถามของภาษาแคลคูลัสสัมพันธ์กับชุดของข้อความสั่งซึ่งสมนัยกันในภาษาพีชคณิตสัมพันธ์ และเพื่อให้การดำเนินงานประมวลผลตามรูปแบบของแต่ละข้อความสอบถามของภาษาแคลคูลัสสัมพันธ์ใด ๆ ไม่ต้องพัฒนาโปรแกรมขึ้นมาเพื่อประมวลผลเฉพาะข้อความสอบถามนั้น ๆ แต่สามารถเรียกใช้โปรแกรมการประมวลผลของชุดข้อความสั่งที่สมนัยกันในภาษาพีชคณิตสัมพันธ์ได้

**Thesis Title**      An Implementation of ERAO Query Language Interpreter  
**Author**             Miss Supaporn Kamsomkiat  
**Major Program**    Computer Science  
**Academic Year**    1994

### Abstract

There are two categories of relational database query language : the relational calculus and the relational algebra. In relational calculus the user merely writes an expression to specify the desired data, whereas in relational algebra the user must give a sequence of operations. Thus, there is always a correspondence between a query expression in relational calculus and a sequence of operations in relational algebra.

This report presents the design of command lines which represent the relational algebra operations that can be used to study and implement the correspondence between any relational calculus query and a sequence of relational algebra operations. An relational algebra interpreter is also implemented.

## กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงได้ด้วยดี ด้วยความช่วยเหลือและสนับสนุนจากบุคคลหลายฝ่ายซึ่งผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูงมา ณ. โอกาสนี้ คือ

ผู้ช่วยศาสตราจารย์ ดร. อัชฌา อีระเชษฐมงคล อาจารย์ที่ปรึกษา ที่กรุณาให้คำปรึกษา ข้อเสนอแนะทางวิชาการและแก้ไขปัญหิต่าง ๆ ตลอดจนตรวจทานวิทยานิพนธ์ให้แก่ผู้วิจัยมาโดยตลอด

อาจารย์ดารารัตน์ แซ่ลี อาจารย์ที่ปรึกษาร่วม ที่ได้กรุณาให้คำปรึกษาตรวจทานวิทยานิพนธ์ให้แก่ผู้วิจัย

คณะกรรมการสอบวิทยานิพนธ์ทุกท่าน ที่กรุณาช่วยตรวจและแก้ไขวิทยานิพนธ์มูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสาร ซึ่งได้สนับสนุนทุนสำหรับการศึกษาและทำวิจัย

เจ้าหน้าที่ภาควิชาคณิตศาสตร์ทุกท่านที่ให้ความช่วยเหลือและอำนวยความสะดวกในการทำวิทยานิพนธ์

คุณพ่อ คุณแม่ พี่ชาย พี่สาว และเพื่อน ๆ ซึ่งได้สนับสนุนช่วยเหลือและให้กำลังใจในการทำวิทยานิพนธ์มาโดยตลอด

สุภาภรณ์ กานต์สมเกียรติ

## สารบัญ

	หน้า
บทคัดย่อ	(3)
Abstract	(4)
กิตติกรรมประกาศ	(5)
สารบัญ	(6)
รายการตาราง	(8)
รายการรูป	(9)
บทที่	
1 บทนำ	1
1.1 วัตถุประสงค์	1
1.2 ขอบเขตการดำเนินงาน	2
1.3 ขั้นตอนและระยะเวลาการดำเนินงาน	2
1.4 สถานที่ และเครื่องมือที่ใช้	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	4
1.6 สรุปเนื้อหาในวิทยานิพนธ์	4
2 ภาษาสอบถามฐานข้อมูลพีชคณิตสัมพันธ์	5
2.1 ภาษาสอบถามฐานข้อมูลเชิงสัมพันธ์	7
2.2 ภาษาพีชคณิตสัมพันธ์	8
2.3 ความสัมพันธ์ระหว่างภาษาแคลคูลัสสัมพันธ์ และภาษาพีชคณิตสัมพันธ์	16
2.4 ERAO	24
3 โครงสร้างระบบ Mrdbms	27
3.1 องค์ประกอบของระบบ Mrdbms	27
3.2 การปรับเปลี่ยน และเพิ่มเติมระบบ	35
3.3 โครงสร้างระบบ Mrdbms ใหม่	36

## สารบัญ (ต่อ)

บทที่	หน้า
4 ตัวแปลภาษา ERAO	38
4.1 องค์ประกอบของตัวแปลภาษา ERAO	38
4.2 โครงสร้างระบบของตัวแปลภาษา ERAO	60
4.3 การตรวจสอบความผิดพลาด	73
4.4 การดำเนินงานของทั้งระบบ	76
4.5 การดำเนินงานของคำสั่งปฏิบัติการใน ERAO	87
5 สรุป และข้อเสนอแนะ	132
5.1 สรุปผลการวิจัย	132
5.2 อุปสรรคและปัญหา	133
5.3 ข้อเสนอแนะ	133
เอกสารอ้างอิง	134
ภาคผนวก ก แผนภาพกระแสข้อมูล	136
ภาคผนวก ข แผนภาพวากยสัมพันธ์ของตัวแปลภาษา ERAO	138
ภาคผนวก ค เอกสารสำหรับผู้ไ้	148
ประวัติผู้เขียน	168

รายการตาราง

ตารางที่	หน้า
4.1 ข้อความแสดงความผิดพลาดในขณะที่ตรวจสอบความหมาย	74



## รายการรูป

รูปที่	หน้า
2.1 ตัวอย่างตารางข้อมูลของฐานข้อมูลเชิงสัมพันธ์	6
2.2 ตารางข้อมูล Supplier1 และตารางข้อมูล Supplier2	8
2.3 ตารางข้อมูล SP1 และตารางข้อมูล SP2	9
2.4 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ union ระหว่าง ตารางข้อมูล Supplier1 และตารางข้อมูล Supplier2	10
2.5 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ intersect ระหว่าง ตารางข้อมูล Supplier1 และตารางข้อมูล Supplier2	10
2.6 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ minus ระหว่าง ตารางข้อมูล Supplier1 และตารางข้อมูล Supplier2	11
2.7 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ cross ระหว่าง ตารางข้อมูล S และตารางข้อมูล SP	12
2.8 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ select ตารางข้อมูล S	13
2.9 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ project ตารางข้อมูล S	14
2.10 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ divide ระหว่าง ตารางข้อมูล S และตารางข้อมูล SP	14
2.11 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ join ระหว่าง ตารางข้อมูล S และตารางข้อมูล SP	15
3.1 แผนภาพโออาร์ของปทานุกรมข้อมูล และโครงสร้างรีเลชันที่สมนัยกัน	28
3.2 โครงสร้างข้อมูลของปทานุกรมข้อมูลขณะที่อยู่ในหน่วยความจำหลัก	32
3.3 แผนภูมิตะบบ Mrdbms	33
3.4 การจัดเก็บสารบบแฟ้มข้อมูลของ Mrdbms	33
3.5 แผนภูมิตะบบ RAO	37
3.6 แผนภูมิตะบบ ERAO	37
4.1 ข้อมูลเข้า/ออก และผลลัพธ์ที่ได้จากการใช้โปรแกรม Lex และ yylex()	60
4.2 ข้อมูลเข้า/ออก และผลลัพธ์ที่ได้จากการใช้โปรแกรม Yacc และ yyparse()	65
4.3 แผนภาพกระแสข้อมูลของระบบ Mrdbms	78
4.4 แผนภาพกระแสข้อมูลของกระบวนการ Query	79
4.5 แผนภาพกระแสข้อมูลของกระบวนการ Build Data Dictionary	80
4.6 แผนภาพกระแสข้อมูลของกระบวนการ Query Process	81

## รายการรูป (ต่อ)

รูปที่	หน้า
4.7 แผนภาพกระแสข้อมูลของกระบวนการ Editor Command	82
4.8 แผนภาพกระแสข้อมูลของกระบวนการ Query Command	89
4.9 แผนภาพกระแสข้อมูลของกระบวนการ R_operation	90
4.10 แผนภาพกระแสข้อมูลของกระบวนการ S_operation	91
4.11 แผนภาพกระแสข้อมูลของกระบวนการ F_operation	92
4.12 แผนภาพกระแสข้อมูลของกระบวนการ Intersect_Stmt	93
4.13 แผนภาพกระแสข้อมูลของกระบวนการ Union_Stmt	94
4.14 แผนภาพกระแสข้อมูลของกระบวนการ Minus_Stmt	95
4.15 แผนภาพกระแสข้อมูลของกระบวนการ Cross_Stmt	96
4.16 แผนภาพกระแสข้อมูลของกระบวนการ Select_Stmt	97
4.17 แผนภาพกระแสข้อมูลของกระบวนการ Project_Stmt	98
4.18 แผนภาพกระแสข้อมูลของกระบวนการ Join_Stmt	99
4.19 แผนภาพกระแสข้อมูลของกระบวนการ Divide_Stmt	100
4.20 แผนภาพกระแสข้อมูลของกระบวนการ Sequence_Stmt	101
4.21 แผนภาพกระแสข้อมูลของกระบวนการ Unique_Stmt	102
4.22 แผนภาพกระแสข้อมูลของกระบวนการ Allfunction_Stmt	103
4.23 แผนภาพกระแสข้อมูลของกระบวนการ Groupfunction_Stmt	104
4.24 ผังงานของโปรแกรมย่อย Union	120
4.25 ผังงานของโปรแกรมย่อย Intersect	121
4.26 ผังงานของโปรแกรมย่อย Minus	122
4.27 ผังงานของโปรแกรมย่อย Cross	123
4.28 ผังงานของโปรแกรมย่อย Select	124
4.29 ผังงานของโปรแกรมย่อย Project	125
4.30 ผังงานของโปรแกรมย่อย Join	126
4.31 ผังงานของโปรแกรมย่อย Divide	127
4.32 ผังงานของโปรแกรมย่อย Sequence	128
4.33 ผังงานของโปรแกรมย่อย Unique	129
4.34 ผังงานของโปรแกรมย่อย All_Function	130
4.35 ผังงานของโปรแกรมย่อย Group_Function	131

## รายการรูป (ต่อ)

รูปที่	หน้า
ค.1 ส่วนประกอบหลักของจอภาพ	149
ค.2 ส่วนรายการหลัก Help	150
ค.3 ส่วนรายการหลัก Database	151
ค.4 ส่วนรายการหลัก Relation	152
ค.5 ส่วนรายการหลัก Tuple	153
ค.6 ส่วนรายการหลัก Query	154
ค.7 ตัวอย่างฐานข้อมูล	156

## บทที่ 1

### บทนำ

งานวิทยานิพนธ์นี้เป็นการศึกษา ออกแบบรูปแบบแถวคำสั่ง และสร้างตัวแปลภาษาสอบถามฐานข้อมูลเชิงสัมพันธ์แบบพีชคณิต หรือที่เรียกย่อๆ ว่าภาษาสอบถามฐานข้อมูลพีชคณิตสัมพันธ์ (Relational Algebra Query Language) ในลักษณะที่เรียกว่า Interpreter โดยกลุ่มคำสั่งถูกนำเข้าไปในรูปแถวของคำสั่ง (Line Command) หลังจากการตรวจสอบความถูกต้องทางวากยสัมพันธ์ของภาษาแล้วจะแปลความหมาย ดำเนินงาน และแสดงผลลัพธ์ที่ได้ทันที ตัวแปลภาษาที่สร้างขึ้นจะครอบคลุมเฉพาะส่วนของภาษาสอบถาม (Query Language) เท่านั้นจะไม่รวมถึงส่วนของภาษาที่ใช้กำหนดโครงสร้างฐานข้อมูล และส่วนของการดำเนินงานพื้นฐานกับข้อมูลในฐานข้อมูล

การดำเนินงานในวิทยานิพนธ์นี้ได้ศึกษาคำสั่งปฏิบัติการและโปรแกรมดำเนินงานของทั้ง 8 คำสั่งมาตรฐานที่ถูกพัฒนาในงานวิทยานิพนธ์ของคุณทวีศักดิ์ [ทวีศักดิ์ (2534)] ซึ่งเรียกโปรแกรมการดำเนินงานของคำสั่งปฏิบัติการทั้ง 8 นี้ว่า “RAO” (Relational Algebra Operation) และศึกษาโปรแกรมการดำเนินงานในงานวิทยานิพนธ์ของคุณเพียรพร [เพียรพร (2534)] ซึ่งเรียกว่า “Mrdbms” (Mini-Relational DataBase Management System) เพื่อนำมาใช้ร่วมกับงานที่ทำ

จากการศึกษา RAO พบว่าคำสั่งปฏิบัติการมาตรฐานทั้ง 8 คำสั่งยังไม่เพียงพอในการดำเนินงานสอบถามข้อมูลได้ครอบคลุมเพียงพอเท่าที่ควรจะเป็น จึงได้เพิ่มเติมคำสั่งปฏิบัติการเพื่อให้สามารถดำเนินงานสอบถามข้อมูลได้ครอบคลุมรายละเอียดมากขึ้น และเรียกโปรแกรมใหม่นี้ว่า “ERAO” (Extended Relational Algebra Operation)

#### 1.1 วัตถุประสงค์

1. เพื่อออกแบบวากยสัมพันธ์ของภาษาสอบถามฐานข้อมูลพีชคณิตสัมพันธ์ในลักษณะที่แต่ละข้อความสั่งการดำเนินงานอยู่ในรูปแถวคำสั่งซึ่งประกอบด้วยตัวดำเนินงาน (Operator) และตัวข้อมูลที่ใช้ในการดำเนินงาน (Operand)
2. เพื่อศึกษา และสร้างตัวแปลภาษาสอบถามฐานข้อมูลพีชคณิตสัมพันธ์ที่ออกแบบ
3. เพื่อโอนย้ายโปรแกรมในงานวิจัยเดิมในส่วนการกำหนดโครงสร้างฐานข้อมูล และการดำเนินงานพื้นฐานกับฐานข้อมูลบนเครื่องคอมพิวเตอร์ VAX-11/785 มาทำงานบนเครื่องคอมพิวเตอร์ SUN SPARC IPX และเชื่อมโยงให้ทำงานร่วมกันได้กับโปรแกรมที่ถูกพัฒนาในงานวิจัยนี้

## 1.2 ขอบเขตการดำเนินงาน

1. ออกแบบวากยสัมพันธ์สำหรับแต่ละแถวคำสั่งของภาษาสอบถามฐานข้อมูลพีชคณิตสัมพันธ์
2. ออกแบบ และพัฒนาโปรแกรมแปลภาษาที่สามารถใช้ปทานุกรมข้อมูล และปฏิบัติการต่าง ๆ ร่วมกับงานวิจัยเดิมบางส่วนได้ โดยการเชื่อมโยงโปรแกรมตัวแปลภาษาที่พัฒนากับโปรแกรมของงานวิทยานิพนธ์ที่มีมาก่อนหน้านี้
3. ออกแบบ และพัฒนาให้ผลลัพธ์ที่ได้จากการดำเนินงานแสดงในรูปตารางข้อมูลทางจอภาพ

## 1.3 ขั้นตอนและระยะเวลาการดำเนินงาน

### ขั้นตอนการดำเนินงาน

1. ศึกษาแนวความคิดทั่วไปของวากยสัมพันธ์ และความหมายของภาษาสอบถามฐานข้อมูลเชิงสัมพันธ์ทั้งแบบแคลคูลัส และแบบพีชคณิต
2. ศึกษา และวิเคราะห์งานวิจัยเดิมเกี่ยวกับ โครงสร้างปทานุกรมข้อมูล โครงสร้างข้อมูล โครงสร้างสารบบแฟ้มข้อมูล และวิธีจัดเก็บแฟ้มข้อมูลลงบนสื่อบันทึกข้อมูล
3. ศึกษา และวิเคราะห์กระบวนการดำเนินงานต่าง ๆ ของระบบงานเดิม
4. ศึกษาการใช้โปรแกรมมอรรถประโยชน์ Lex, Yacc และ Dbx
5. นำระบบเดิมบางส่วนมาแก้ไขเพื่อให้สามารถปฏิบัติงานได้บนเครื่องคอมพิวเตอร์ SUN SPARC IPX
6. ออกแบบวากยสัมพันธ์ และวิเคราะห์ความหมายของแต่ละแถวคำสั่งในภาษาสอบถามฐานข้อมูลพีชคณิตสัมพันธ์ที่จะพัฒนา
7. พัฒนาโปรแกรมแปลภาษา โดยใช้ภาษาซี ร่วมกับโปรแกรมมอรรถประโยชน์ Lex, Yacc และ Dbx
8. ทดสอบ และแก้ไขโปรแกรมเพื่อให้โปรแกรมทำงานได้สมบูรณ์ที่สุด
9. จัดทำเอกสาร (Documentation) และรายงานวิทยานิพนธ์

### ระยะเวลาการดำเนินงาน

ใช้เวลาทั้งสิ้นประมาณ 1 ปี 4 เดือน ดังรายละเอียดตามตารางต่อไปนี้

เดือน,ปี ขั้นตอนที่	2535				2536												
	กย.	ตค.	พย.	ธค.	มค.	กพ.	มีค.	เมย.	พค.	มิย.	กค.	สค.	กย.	ตค.	พย.	ธค.	
1	■																
2		■															
3			■														
4				■													
5				■	■												
6						■											
7							■	■	■	■	■	■	■	■	■	■	■
8													■	■	■	■	■
9																■	■

#### 1.4 สถานที่ และเครื่องมือที่ใช้

##### สถานที่

ห้องปฏิบัติการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์ มหาวิทยาลัยสงขลานครินทร์  
วิทยาเขตหาดใหญ่

##### เครื่องมือที่ใช้

1. เครื่องคอมพิวเตอร์ SUN SPARC IPX ซึ่งเป็นสถานีงาน (Workstation) ที่มีหน่วยความจำหลัก 16 MB ใช้ระบบปฏิบัติการ SunOS รุ่น 4.1.1
2. ตัวแปลภาษาซี (Classic C)
3. โปรแกรมมอรรถประโยชน์ตรวจจับกลุ่มคำ (Token) ชื่อ Lex
4. โปรแกรมมอรรถประโยชน์ช่วยการตรวจจាកยสัมพันธ์ชื่อ Yacc
5. โปรแกรมมอรรถประโยชน์ช่วยการตรวจสอบ และแก้ไขข้อผิดพลาดของโปรแกรมขณะทำงาน ภายใต้ระบบปฏิบัติการ SunOS รุ่น 4.1.1 ชื่อ Dbx

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำโปรแกรมที่พัฒนานี้ไปใช้ ประกอบการเรียนการสอนในเนื้อหาบางส่วน ของวิชาการระบบจัดการฐานข้อมูล (Database Management System) โดยทั่วไปได้
2. ผู้วิจัยมีความเข้าใจในการดำเนินงาน และการพัฒนาระบบจัดการฐานข้อมูลมากขึ้น
3. ได้โปรแกรมตัวแปลภาษาสอบถามฐานข้อมูลพีชคณิตสัมพันธ์ ซึ่งเป็นส่วนหนึ่งของ ระบบจัดการฐานข้อมูล

### 1.6 สรุปเนื้อหาในวิทยานิพนธ์

บทที่ 1 กล่าวถึงความเป็นมาของปัญหา วัตถุประสงค์ ขอบเขต ขั้นตอน และระยะเวลาในการทำวิทยานิพนธ์ และประโยชน์ที่คาดว่าจะได้รับ

บทที่ 2 กล่าวถึงทฤษฎีเบื้องต้น และที่มาของงานวิทยานิพนธ์นี้ พร้อมทั้งโปรแกรม อรรถประโยชน์ต่าง ๆ ที่เกี่ยวข้อง

บทที่ 3 กล่าวถึงส่วนที่เกี่ยวข้องกับงานวิทยานิพนธ์ที่ได้ดำเนินมาก่อนหน้านี้ โครงสร้างระบบโดยรวมของงานวิทยานิพนธ์

บทที่ 4 กล่าวถึงรายละเอียดการพัฒนาตัวแปลภาษาสอบถามฐานข้อมูลพีชคณิต สัมพันธ์ ERAO

บทที่ 5 จะเป็นการสรุปผล ปัญหา และข้อเสนอแนะต่าง ๆ

## บทที่ 2

### ภาษาสอบถามฐานข้อมูลที่ชนิดสัมพันธ์

ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เป็นฐานข้อมูลที่ข้อมูลถูกมองจากผู้  
ใช้ทั้งรูปแบบการจัดเก็บข้อมูลและการดำเนินงานเพื่อใช้ข้อมูล อยู่ในรูปตารางข้อมูล (Table)  
2 มิติที่มีความสัมพันธ์กันจำนวนมากซึ่งเรียกว่ารีเลชัน (Relation) ตารางข้อมูลเหล่านี้ก็คือ  
เซต (Set) ในทางคณิตศาสตร์นั่นเอง โดยสมาชิกแต่ละตัวในเซต ถูกนำมาจัดใหม่ให้เป็น  
ข้อมูลแต่ละแถวในตารางข้อมูลเรียกว่าทูปเปล (Tuple) หรือแถวข้อมูล (Row) ส่วนสละตมภ์  
(Column) ของตารางข้อมูลถูกเรียกว่าแอตทริบิว (Attribute) และค่าข้อมูล (Attribute  
Value) ที่เป็นไปได้ของแต่ละสละตมภ์ก็คือค่าของเซตที่เรียกว่าโดเมน (Domain) ของ  
แอตทริบิวนั้นๆ รูปที่ 2.1 เป็นตัวอย่างตารางข้อมูลของฐานข้อมูลเชิงสัมพันธ์จากหนังสือ  
An Introduction to Database System ของ C.J. Date [Date (1989)] โดยตารางข้อมูล  
เหล่านี้เมื่อพูดถึงเฉพาะโครงสร้างไม่รวมข้อมูลภายในตารางจะเขียนดังนี้

S(S#,SNAME,STATUS,CITY)

P(P#,PNAME,COLOR,WEIGHT,CITY)

SP(S#,P#,QTY)

โดย S เป็นตารางข้อมูลซึ่งแต่ละแถวข้อมูลแทนรายละเอียดของผู้ส่งสินค้าแต่ละราย อัน  
ประกอบด้วย รหัสผู้ส่งสินค้า (S#) ชื่อผู้ส่งสินค้า (SNAME) สถานะผู้ส่งสินค้า  
(STATUS) และ เมืองที่ตั้งของผู้ส่งสินค้า (CITY) โดยมี S# เป็นคีย์หลัก

P เป็นตารางข้อมูลซึ่งแต่ละแถวข้อมูลแทนรายละเอียดของชิ้นส่วนสินค้าแต่ละชนิด  
อันประกอบด้วย รหัสชิ้นส่วนสินค้า (P#) ชื่อชิ้นส่วนสินค้า (PNAME) สีชิ้น  
ส่วนสินค้า (COLOR) น้ำหนักชิ้นส่วนสินค้า (WEIGHT) และเมืองที่ผลิตชิ้น  
ส่วนสินค้า (CITY) โดยมี P# เป็นคีย์หลัก

SP เป็นตารางข้อมูลซึ่งแต่ละแถวข้อมูลแทนรายละเอียดการส่งชิ้นส่วนสินค้าแต่ละชนิด  
ของผู้ส่งสินค้าแต่ละรายอันประกอบด้วย รหัสผู้ส่งสินค้า (S#) รหัสชิ้นส่วนสินค้า  
(P#) และปริมาณชิ้นส่วนสินค้าที่ถูกส่ง (QTY) แต่ละแถวข้อมูลให้สารสนเทศว่า  
“ชิ้นส่วนสินค้าหนึ่งๆ (P#) ถูกส่งโดยผู้ส่งสินค้าใด (S#) ด้วยปริมาณเท่าไร  
(QTY)” โดยมี S# และ P# ผสมกันเป็นคีย์หลักรวม (Compound Key)



S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

SP

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

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

รูปที่ 2.1 ตัวอย่างตารางข้อมูลของฐานข้อมูลเชิงสัมพันธ์

## 2.1 ภาษาสอบถามฐานข้อมูลเชิงสัมพันธ์

ภาษาสอบถามฐานข้อมูลเชิงสัมพันธ์ซึ่งใช้กับข้อมูลซึ่งมีรูปแบบการจัดเก็บข้อมูลในรูป ตารางข้อมูล มีอยู่ 2 แบบ คือ

- ภาษาแคลคูลัสสัมพันธ์ (Relational Calculus Language)
- ภาษาพีชคณิตสัมพันธ์ (Relational Algebra Language)

การสอบถามข้อมูลหรือสารสนเทศ ด้วยภาษาแคลคูลัสสัมพันธ์มีลักษณะการสอบถามเป็นการบอกคุณลักษณะของข้อมูลที่ต้องการ เนื่องจากตารางข้อมูลคือเซตในทางคณิตศาสตร์ ดังนั้นอาจกล่าวได้ว่าภาษาสอบถามแคลคูลัสสัมพันธ์ก็คือภาษาซึ่งมีลักษณะการสอบถามเป็นการบอกคุณลักษณะของสมาชิกภายในเซตที่ต้องการนั่นเอง เช่น จากตารางข้อมูล S ในรูปที่ 2.1 ถ้าต้องการทราบค่า S# ที่มีภูมิลำเนาอยู่ที่เมือง Paris อาจเขียนเป็นภาษาสอบถามแคลคูลัสสัมพันธ์ได้ดังนี้

S.S# WHERE S.CITY = 'Paris'

ผลลัพธ์ที่ได้จากการสอบถามข้างต้นคือ ค่า S# จากแถวข้อมูลในตารางข้อมูล S ซึ่งมีค่า CITY เท่ากับ 'Paris' ซึ่งก็คือ S2 และ S3 ของ 2 แถวข้อมูลต่อไปนี้

S#	SNAME	STATUS	CITY
S2	Jones	10	Paris
S3	Blake	30	Paris

การสอบถามข้อมูลหรือสารสนเทศ ด้วยภาษาพีชคณิตสัมพันธ์มีลักษณะการสอบถามเป็นการบอกขั้นตอนการดำเนินงานกับข้อมูลในฐานข้อมูลโดยตรง ด้วยการบอกคำสั่งปฏิบัติการเป็นลำดับเพื่อให้ได้มาซึ่งข้อมูลสารสนเทศที่ต้องการ คำสั่งปฏิบัติการที่ใช้ทั่วไปทั้งหมดถูกแบ่งออกได้เป็น 2 กลุ่ม กลุ่มแรกเป็นคำสั่งปฏิบัติการมาตรฐานที่ใช้กับเซตในทางคณิตศาสตร์ (Set Operations) ส่วนกลุ่มที่สองเป็นคำสั่งปฏิบัติการพิเศษ (Special Operations) เพิ่มขึ้นนอกเหนือจากคำสั่งปฏิบัติการมาตรฐานที่ใช้กับเซต [Date (1989)]

## 2.2 ภาษาพีชคณิตสัมพันธ์

คำสั่งปฏิบัติการในภาษาพีชคณิตสัมพันธ์โดยทั่ว ๆ ไปประกอบด้วย 8 คำสั่ง โดยสามารถแยกคำสั่งเหล่านี้ได้เป็น 2 กลุ่มคือ

- กลุ่มคำสั่งปฏิบัติการกับเซต ประกอบด้วยคำสั่ง Union, Intersect, Minus (Differ) และ Cross (Cross Product)
- กลุ่มคำสั่งพิเศษ ประกอบด้วยคำสั่ง Select, Project, Join และ Divide

โดยคำสั่งปฏิบัติการ Union, Intersect และ Minus จะต้องดำเนินงานกับสองตารางข้อมูลที่มีคุณสมบัติ union-compatible กล่าวคือ ตารางข้อมูลทั้งสองจะต้องมีจำนวนแอตทริบิวต์เท่ากัน และแอตทริบิวต์ในตำแหน่งที่สมนัยกัน จะต้องใช้ค่าจากโดเมนเดียวกัน รูปที่ 2.2 เป็นตัวอย่างของสองตารางข้อมูล Supplier1 และ Supplier2 ซึ่งมีคุณสมบัติ union-compatible กัน (แถวข้อมูลในตารางทั้งสองเป็นแถวข้อมูลส่วนหนึ่งในตารางข้อมูล S)

จำนวนของแอตทริบิวต์ในแต่ละตารางข้อมูล เรียกว่าดีกรี (Degree) ของตารางข้อมูลนั้น ตารางข้อมูลซึ่งมีค่าดีกรีเท่ากับ 1 จะเรียกว่าตารางข้อมูลยูนารี (Unary Relation) ถ้าดีกรีเท่ากับ 2 จะเรียกว่าตารางข้อมูลไบนารี (Binary Relation) ตารางข้อมูลมีดีกรีเท่ากับ 3 จะเรียกว่าตารางข้อมูลเทอร์นารี (Ternary Relation) กรณีที่ตารางข้อมูลมีดีกรีเท่ากับ N เมื่อ N เป็นเลขจำนวนเต็มบวกใด ๆ จะเรียกว่าตารางข้อมูลเอ็น-เอรี (N-ary Relation) รูปที่ 2.3 เป็นตัวอย่างของสองตารางข้อมูล SP1 และ SP2 ซึ่งมีดีกรีเป็น 2 และ 1 ตามลำดับ

การสอบถามข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ของระบบจัดการฐานข้อมูลที่มีขายตามท้องตลาดจะใช้แต่ภาษาแคลคูลัสสัมพันธ์ ไม่นิยมใช้ภาษาพีชคณิตสัมพันธ์เนื่องจากการใช้ภาษาพีชคณิตสัมพันธ์สอบถามฐานข้อมูลกระทำได้ยากกว่า สำหรับผู้ใช้ทั่ว ๆ ไปที่ไม่ค่อยมีความรู้เรื่องคณิตศาสตร์ และคอมพิวเตอร์มากนัก ดังนั้นภาษาพีชคณิตสัมพันธ์จึงไม่มีวากยสัมพันธ์มาตรฐานสำหรับให้ใช้ทั่วไป นอกจากที่ออกแบบเพื่อใช้กันในสถานศึกษาต่าง ๆ เช่น IRA (Instructional Relational Algebra) ถูกออกแบบและใช้เพื่อการศึกษาในมหาวิทยาลัยโมนาช ประเทศออสเตรเลีย [Ken (1983)]

Supplier1

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S3	Blake	30	Paris

Supplier2

S#	SNAME	STATUS	CITY
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

รูปที่ 2.2 ตารางข้อมูล Supplier1 และตารางข้อมูล Supplier2

SP1		SP2
S#	P#	P#
S1	P1	P1
S1	P2	P2
S1	P3	P3
S1	P4	P4
S1	P5	P5
S1	P6	P6
S2	P1	
S2	P2	
S3	P2	
S4	P2	
S4	P4	
S4	P5	

รูปที่ 2.3 ตารางข้อมูล SP1 และตารางข้อมูล SP2

ต่อไปนี้เป็นความหมายของแต่ละคำสั่งปฏิบัติการพีชคณิตสัมพันธ์มาตรฐานทั้งแปดคำสั่ง [Date (1989)] และตัวอย่างการสอบถามข้อมูลจากตารางข้อมูลในรูปที่ 2.1 รูปที่ 2.2 และรูปที่ 2.3 ด้วยคำสั่งปฏิบัติการ IRA พร้อมผลลัพธ์ที่ได้

**Union** เป็นคำสั่งปฏิบัติการในการรวมแถวข้อมูลของสองตารางข้อมูลที่มีคุณสมบัติ union-compatible ผลลัพธ์ที่ได้ก็จะเป็นตารางข้อมูลที่มีคุณสมบัติ union-compatible กับตารางข้อมูลทั้งสองที่นำมาดำเนินงานโดยใช้คำสั่ง union โดยในการรวมแถวข้อมูลที่ซ้ำกันจะนำมาใส่ในตารางข้อมูลผลลัพธ์เพียงแถวเดียว

**ตัวอย่าง** ต้องการรายละเอียดของผู้ส่งสินค้าทุกคนในตารางข้อมูล Supplier1 หรือในตารางข้อมูล Supplier2

แถวคำสั่ง IRA ที่สมนัยกัน :

RESULT = Supplier1 UNION Supplier2

ผลลัพธ์ที่ได้คือตารางข้อมูลในรูปที่ 2.4

## RESULT

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

รูปที่ 2.4 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ union ระหว่างตารางข้อมูล Supplier1 และตารางข้อมูล Supplier2

**Intersect** เป็นคำสั่งปฏิบัติการในการเลือกแถวข้อมูลที่เหมือนกันจากสองตารางข้อมูลที่มีคุณสมบัติ union-compatible ผลลัพธ์ที่ได้จะเป็นตารางข้อมูลที่มีคุณสมบัติ union-compatible กับตารางข้อมูลทั้งสองที่ถูกนำมาดำเนินงานกัน

**ตัวอย่าง** ต้องการรายละเอียดของผู้ส่งสินค้าที่ปรากฏอยู่ทั้งในตารางข้อมูล Supplier1 และในตารางข้อมูล Supplier2

แถวคำสั่ง IRA ที่สมนัยกัน :

RESULT = Supplier1 INTERSECT Supplier2

ผลลัพธ์ที่ได้คือตารางข้อมูลในรูปที่ 2.5

## RESULT

S#	SNAME	STATUS	CITY
S3	Blake	30	Paris

รูปที่ 2.5 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ intersect ระหว่างตารางข้อมูล Supplier1 และตารางข้อมูล Supplier2

**Minus** เป็นคำสั่งปฏิบัติการในการเลือกแถวข้อมูลจากตารางข้อมูลอันที่หนึ่งที่มีคุณสมบัติ union-compatible กับตารางข้อมูลอันที่สอง โดยจะเลือกเฉพาะแถวข้อมูลที่ปรากฏในตารางข้อมูลอันที่หนึ่งแต่ไม่ปรากฏในตารางข้อมูลอันที่สอง ผลลัพธ์ที่ได้ก็จะเป็นตารางข้อมูลที่มีคุณสมบัติ union-compatible กับตารางข้อมูลทั้งสองที่ถูนำมาดำเนินงานกัน

**ตัวอย่าง** ต้องการรายละเอียดของผู้ส่งสินค้าซึ่งปรากฏอยู่ในตารางข้อมูล Supplier1 แต่ไม่ปรากฏในตารางข้อมูล Supplier2

แถวคำสั่ง IRA ที่สมนัยกัน :

RESULT = Supplier1 MINUS Supplier2

ผลลัพธ์ที่ได้คือตารางข้อมูลในรูปที่ 2.6

RESULT

S#	SNAME	STATUS	CITY
S1	Smith	20	London

รูปที่ 2.6 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ minus ระหว่างตารางข้อมูล Supplier1 และตารางข้อมูล Supplier2

**Cross** เป็นคำสั่งปฏิบัติการในการจับคู่แถวข้อมูลทุกคู่ที่เป็นไปได้ระหว่าง 2 ตารางข้อมูลที่ถูกดำเนินงาน ผลลัพธ์ของตารางข้อมูลที่ได้จะประกอบด้วยแถวข้อมูลซึ่งเกิดจากการนำแถวข้อมูลของตารางข้อมูลทีหนึ่งต่อท้ายด้วยแถวข้อมูลของตารางข้อมูลที่สองที่นำมาจับคู่กัน

**ตัวอย่าง** ใช้คำสั่งปฏิบัติการ Cross (Cross Product) ดำเนินงานกับตารางข้อมูล S และตารางข้อมูล SP

แถวคำสั่ง IRA ที่สมนัยกัน :

RESULT = S CROSS SP

ผลลัพธ์ที่ได้คือตารางข้อมูลในรูปที่ 2.7

## RESULT

S#	SNAME	STATUS	CITY	S#	P#	QTY
S1	Smith	20	London	S1	P1	300
S1	Smith	20	London	S1	P2	200
S1	Smith	20	London	S1	P3	400
S1	Smith	20	London	S1	P4	200
S1	Smith	20	London	S1	P5	100
S1	Smith	20	London	S1	P6	100
S1	Smith	20	London	S2	P1	300
S1	Smith	20	London	S2	P2	400
S1	Smith	20	London	S3	P2	200
S1	Smith	20	London	S4	P2	200
S1	Smith	20	London	S4	P4	300
S1	Smith	20	London	S4	P5	400
S2	Jones	10	Paris	S1	P1	300
S2	Jones	10	Paris	S1	P2	200
S2	Jones	10	Paris	S1	P3	400
S2	Jones	10	Paris	S1	P4	200
S2	Jones	10	Paris	S1	P5	100
S2	Jones	10	Paris	S1	P6	100
S2	Jones	10	Paris	S2	P1	300
S2	Jones	10	Paris	S2	P2	400
S2	Jones	10	Paris	S3	P2	200
S2	Jones	10	Paris	S4	P2	200
S2	Jones	10	Paris	S4	P4	300
S2	Jones	10	Paris	S4	P5	400
S3	Blake	30	Paris	S1	P1	300
S3	Blake	30	Paris	S1	P2	200
S3	Blake	30	Paris	S1	P3	400
S3	Blake	30	Paris	S1	P4	200
S3	Blake	30	Paris	S1	P5	100
S3	Blake	30	Paris	S1	P6	100

## RESULT (ต่อ)

S#	SNAME	STATUS	CITY	S#	P#	QTY
S3	Blake	30	Paris	S2	P1	300
S3	Blake	30	Paris	S2	P2	400
S3	Blake	30	Paris	S3	P2	200
S3	Blake	30	Paris	S4	P2	200
S3	Blake	30	Paris	S4	P4	300
S3	Blake	30	Paris	S4	P5	400
S4	Clark	20	Londor	S1	P1	300
S4	Clark	20	Londor	S1	P2	200
S4	Clark	20	Londor	S1	P3	400
S4	Clark	20	Londor	S1	P4	200
S4	Clark	20	Londor	S1	P5	100
S4	Clark	20	Londor	S1	P6	100
S4	Clark	20	Londor	S2	P1	300
S4	Clark	20	Londor	S2	P2	400
S4	Clark	20	Londor	S3	P2	200
S4	Clark	20	Londor	S4	P2	200
S4	Clark	20	Londor	S4	P4	300
S4	Clark	20	Londor	S4	P5	400
S4	Clark	20	Londor	S4	P6	100
S5	Adams	30	Athens	S1	P1	300
S5	Adams	30	Athens	S1	P2	200
S5	Adams	30	Athens	S1	P3	400
S5	Adams	30	Athens	S1	P4	200
S5	Adams	30	Athens	S1	P5	100
S5	Adams	30	Athens	S1	P6	100
S5	Adams	30	Athens	S2	P1	300
S5	Adams	30	Athens	S2	P2	400
S5	Adams	30	Athens	S3	P2	200
S5	Adams	30	Athens	S4	P2	200
S5	Adams	30	Athens	S4	P4	300
S5	Adams	30	Athens	S4	P5	400

รูปที่ 2.7 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ cross ระหว่างตารางข้อมูล S และตารางข้อมูล SP

**Select** (บางครั้งอาจเรียกอีกอย่างว่า “restrict” เพื่อป้องกันความสับสนกับคำสั่งวน (Keyword) “select” ของภาษา SQL) เป็นคำสั่งปฏิบัติการในการเลือกกลุ่มของแถวข้อมูลจากตารางข้อมูลที่ป้อน โดยใช้ค่าแอตทริบิวของตารางข้อมูลเป็นตัวกำหนดเงื่อนไขในการเลือกแถวข้อมูลที่ต้องการ หรือกล่าวอีกนัยหนึ่งคือแถวข้อมูลจะถูกเลือกโดย select ถ้าค่าของแอตทริบิวที่บอกในแถวคำสั่งของแถวข้อมูลนั้นอยู่ในเกณฑ์ของเงื่อนไขการเลือก

**ตัวอย่าง** ต้องการรายละเอียดของผู้ส่งสินค้าที่อยู่ในเมือง Paris จากตารางข้อมูล S

แถวคำสั่ง IRA ที่สมนัยกัน :

RESULT = SELECT S[CITY="Paris"]

ผลลัพธ์ที่ได้คือตารางข้อมูลในรูปที่ 2.8

RESULT

S#	SNAME	STATUS	CITY
S2	Jones	10	Paris
S3	Blake	30	Paris

รูปที่ 2.8 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ select ตารางข้อมูล S

**Project** เป็นคำสั่งปฏิบัติการในการสกัดข้อมูลของบางแอตทริบิวจากตารางข้อมูลที่ต้องการ

**ตัวอย่าง** ต้องการรายชื่อของผู้ส่งสินค้า (SNAME) ทั้งหมดจากตารางข้อมูล S

แถวคำสั่ง IRA ที่สมนัยกัน :

RESULT = PROJECT S[SNAME]

ผลลัพธ์ที่ได้คือตารางข้อมูลในรูปที่ 2.9



## RESULT

SNAME
Smith
Jones
Blake
Clark
Adams

รูปที่ 2.9 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ project ตารางข้อมูล S

**Divide** เป็นคำสั่งปฏิบัติการสำหรับดำเนินงานกับสองตารางข้อมูล ในรูปง่ายที่สุดตารางข้อมูลอันที่หนึ่งจะเป็นตารางข้อมูลไบนารี (มีสองแอดทริบิว) ทำหน้าที่เป็นตารางข้อมูลตัวตั้ง (Dividend) และตารางข้อมูลอันที่สองเป็นตารางข้อมูลยูนารี (มีหนึ่งแอดทริบิว) ทำหน้าที่เป็นตารางข้อมูลตัวหาร (Divisor) โดยแอดทริบิวของตารางข้อมูลตัวหารจะต้องใช้โดเมนเดียวกันกับหนึ่งแอดทริบิวในตารางข้อมูลตัวตั้ง ผลลัพธ์ที่ได้จะเป็นตารางข้อมูลซึ่งประกอบด้วยค่าของอีกแอดทริบิวหนึ่งของตารางข้อมูลตัวตั้งที่จับคู่กับค่าของแอดทริบิวที่ใช้โดเมนร่วมกับแอดทริบิวของตารางข้อมูลตัวหารทุกค่าของแอดทริบิวในตารางข้อมูลตัวหาร

**ตัวอย่าง** ใช้คำสั่งปฏิบัติการ divide ดำเนินงานกับตารางข้อมูลตัวตั้ง SP1 และตารางข้อมูลตัวหาร SP2 ในรูปที่ 2.3

แถวคำสั่ง IRA ที่สมนัยกัน :

RESULT = DIVIDE SP1[P#]SP2

ผลลัพธ์ที่ได้คือตารางข้อมูลในรูปที่ 2.10

## RESULT

S#
S1

รูปที่ 2.10 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ divide ระหว่างตารางข้อมูล SP1 และตารางข้อมูล SP2

**Join** เป็นคำสั่งปฏิบัติการจับคู่แถวข้อมูลของสองตารางข้อมูลทำนองเดียวกับคำสั่งปฏิบัติการ Cross แต่สองตารางข้อมูลที่ถูกดำเนินการด้วยจะต้องมีแอตทริบิวต์ซึ่งใช้โดเมน (Domain) ร่วมกัน และการจับคู่ของแถวข้อมูลทั้งสองจะต้องเป็นไปตามเงื่อนไขการเปรียบเทียบค่า ( = (เท่ากัน) , < (น้อยกว่า) , ≤ (น้อยกว่าหรือเท่ากับ) , > (มากกว่า) , ≥ (มากกว่าหรือเท่ากับ)) ของแอตทริบิวต์ที่ใช้โดเมนร่วมกัน

**ตัวอย่าง** ใช้คำสั่งปฏิบัติการ join ดำเนินงานกับตารางข้อมูล S และตารางข้อมูล SP ด้วยเงื่อนไข S.S# (ค่า S# ของตารางข้อมูล S) เท่ากับ SP.S# (ค่า S# ของตารางข้อมูล SP)

แถวคำสั่ง IRA ที่สมนัยกัน :

RESULT = JOIN S[S#=S#]SP หรือ

RESULT = JOIN S[S#]SP

ผลลัพธ์ที่ได้คือตารางข้อมูลในรูปที่ 2.11

RESULT

S#	SNAME	STATUS	CITY	S#	P#	QTY
S1	Smith	20	London	S1	P1	300
S1	Smith	20	London	S1	P2	200
S1	Smith	20	London	S1	P3	400
S1	Smith	20	London	S1	P4	200
S1	Smith	20	London	S1	P5	100
S1	Smith	20	London	S1	P6	100
S2	Jones	10	Paris	S2	P1	300
S2	Jones	10	Paris	S2	P2	400
S3	Blake	30	Paris	S3	P2	200
S4	Clark	20	London	S4	P2	200
S4	Clark	20	London	S4	P4	300
S4	Clark	20	London	S4	P5	400

รูปที่ 2.11 ตารางข้อมูลผลลัพธ์ซึ่งได้จากการใช้คำสั่งปฏิบัติการ join ระหว่างตารางข้อมูล S และตารางข้อมูล SP

### 2.3 ความสัมพันธ์ระหว่างภาษาแคลคูลัสสัมพันธ์ และภาษาพีชคณิตสัมพันธ์

ความสัมพันธ์ระหว่างภาษาแคลคูลัสสัมพันธ์ และภาษาพีชคณิตสัมพันธ์นั้นมีลักษณะคล้ายกับความสัมพันธ์ระหว่างภาษาระดับสูง และภาษาระดับต่ำในการเขียนโปรแกรมคอมพิวเตอร์ทั่วไป คือภาษาระดับสูงหนึ่งข้อความสั่ง (Statement) จะถูกแปลงให้เป็นภาษาระดับต่ำได้หลายคำสั่ง นั่นคือภาษาแคลคูลัสสัมพันธ์หนึ่งข้อความสั่งจะสามารถแทนได้ด้วยหลายคำสั่งปฏิบัติการของภาษาพีชคณิตสัมพันธ์

ตัวอย่างต่อไปนี้ เป็นการศึกษาค้นคว้าความสัมพันธ์ระหว่างข้อความสั่งในภาษาแคลคูลัสสัมพันธ์และชุดของคำสั่งในภาษาพีชคณิตสัมพันธ์ซึ่งสมมูลกัน โดยข้อความสั่งในภาษาแคลคูลัสสำหรับสอบถามฐานข้อมูลจะใช้ SQL (Structure Query Language) ซึ่งเป็นภาษาสอบถามฐานข้อมูลเชิงสัมพันธ์ที่กำลังเป็นที่นิยมใช้กันมาก โดย SQL ที่ใช้จะใช้วากยสัมพันธ์ของระบบจัดการฐานข้อมูลโอราเคิล (ORACLE) และคำสั่งปฏิบัติการของภาษาพีชคณิตสัมพันธ์จะใช้ IRA และฐานข้อมูลที่ใช้ คือฐานข้อมูลในรูปแบบที่ 2.1

#### ตัวอย่าง 2.1

คำถาม Get full details of all suppliers in Paris.

SQL

```
SELECT *
FROM S
WHERE CITY = 'Paris';
```

IRA

```
T1 = SELECT S[CITY = "Paris"]
DISPLAY T1
```

ผลลัพธ์

T1

S#	SNAME	STATUS	CITY
S2	Jones	10	Paris
S3	Blake	30	Paris

ตัวอย่าง 2.2

คำถาม Get part numbers for all parts.

SQL

```
SELECT P#
FROM P;
```

ผลลัพธ์

T1
P#
P1
P2
P3
P4
P5
P6

IRA

```
T1 = PROJECT P[P#]
DISPLAY T1
```

ตัวอย่าง 2.3

คำถาม Get supplier numbers and names for suppliers in Paris.

SQL

```
SELECT S#,SNAME
FROM S
WHERE CITY = 'Paris';
```

IRA

```
T1 = SELECT S[CITY = "Paris"]
T2 = PROJECT T1[S#,SNAME]
DISPLAY T2
```

ผลลัพธ์

T2	
S#	SNAME
S2	Jones
S3	Blake

**ตัวอย่าง 2.4**

**คำถาม** Get full details of all suppliers who are located in London or having status greater than 20.

**SQL**

```
SELECT *
FROM S
WHERE CITY = 'London' OR STATUS > 20;
```

**IRA**

```
T1 = SELECT S[CITY = "London"]
T2 = SELECT S[STATUS > 20]
T3 = T1 UNION T2
DISPLAY T3
```

**ผลลัพธ์**

T3

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

**ตัวอย่าง 2.5**

**คำถาม** Get part names for red parts with weight greater than 5 kilograms.

**SQL**

```
SELECT PNAME
FROM P
WHERE COLOR = 'Red' AND WEIGHT > 5;
```

IRA

```

T1 = SELECT P[COLOR = "Red"]
T2 = SELECT P[WEIGHT > 5]
T3 = T1 INTERSECT T2
T4 = PROJECT T3[PNAME]
DISPLAY T4

```

ผลลัพธ์

T4
PNAME
Nut
Screw
Cog

**ตัวอย่าง 2.6**

คำถาม Get full details of all parts but not part P1 or P2.

SQL

```

SELECT *
FROM P
WHERE P# NOT IN ('P1','P2');

```

IRA

```

T1 = SELECT P[P# = "P1"]
T2 = SELECT P[P# = "P2"]
T3 = T1 UNION T2
T4 = P MINUS T3
DISPLAY T4

```

ผลลัพธ์

T4	P#	PNAME	COLOR	WEIGHT	CITY
	P3	Screw	Blue	17	Rome
	P4	Screw	Red	14	London
	P5	Cam	Blue	12	Paris
	P6	Cog	Red	19	London

ฝ่ายหอสมุด  
คุณหญิงหลง อรรถกระวีสุนทร

**ตัวอย่าง 2.7**

คำถาม Get supplier names for suppliers who supply part P1.

SQL

```
SELECT SNAME
FROM S,SP
WHERE S.S# = SP.S#
AND P# = 'P1';
```

IRA

```
T1 = SELECT SP[P# = "P1"]
T2 = JOIN S[S#]T1
T3 = PROJECT T2[SNAME]
DISPLAY T3
```

ผลลัพธ์

T3	
SNAME	
Smith	
Jones	

**ตัวอย่าง 2.8**

คำถาม Get supplier numbers and names for suppliers who supply green part.

SQL

```
SELECT S.S#,SNAME
FROM S,SP,P
WHERE S.S# = SP.S#
AND SP.P# = P.P#
AND P.COLOR = 'Green';
```

IRA

```
T1 = SELECT P[COLOR = "Green"]
T2 = JOIN SP[P#]T1
T3 = PROJECT T2[S#]
T4 = JOIN S[S#]T3
T5 = PROJECT T4[S#,SNAME]
DISPLAY T5
```

ผลลัพธ์

T5	
S#	SNAME
S1	Smith
S2	Jones
S3	Blake
S4	Clark

**ตัวอย่าง 2.9**

**คำถาม** Get supplier names for suppliers who supply both part P1 and P2.

**SQL**

```

SELECT SNAME
FROM S
WHERE S# IN
      (SELECT S#
       FROM SP
       GROUP BY S#
        HAVING P# CONTAINS ('P1','P2'));

```

**IRA**

```

T1 = SELECT SP[P# = "P1"]
T2 = SELECT SP[P# = "P2"]
T3 = PROJECT T1[S#]
T4 = PROJECT T2[S#]
T5 = T3 INTERSECT T4
T6 = JOIN S[S#]T5
T7 = PROJECT T6[SNAME]
DISPLAY T7

```

**ผลลัพธ์**

T7

SNAME
Smith
Jones



ตัวอย่าง 2.10

คำถาม Get supplier names for suppliers who supply at least one part supplied by supplier S2.

## SQL

```

SELECT SNAME
FROM S
WHERE S# != 'S2'
AND S# IN
    (SELECT S#
     FROM SP
     WHERE P# IN
        (SELECT P#
         FROM SP
         WHERE S# = 'S2'));

```

## IRA

```

T1 = SELECT SP[S# = "S2"]
T2 = PROJECT T1[P#]
T3 = JOIN SP[P#]T2
T4 = SELECT T3[S# != "S2"]
T5 = PROJECT T4[S#]
T6 = JOIN S[S#]T5
T7 = PROJECT T6[SNAME]
DISPLAY T7

```

ผลลัพธ์

T7

SNAME
Smith
Blake
Clark

**ตัวอย่าง 2.11**

**คำถาม** Get supplier names for suppliers who supply at least all parts supplied by supplier S2.

**SQL**

```

SELECT SNAME
FROM S
WHERE S# != 'S2'
AND S# IN
  (SELECT S#
   FROM SP
   GROUP BY S#
   HAVING P# CONTAINS
     (SELECT P#
      FROM SP
      WHERE S# = 'S2'));

```

**IRA**

```

T1 = SELECT SP[S# = "S2"]
T2 = PROJECT T1[P#]
T3 = SELECT SP[S# <> "S2"]
T4 = PROJECT T3[S#,P#]
T5 = DIVIDE T4[P#]T2
T6 = JOIN S[S#]T5
T7 = PROJECT T6[SNAME]
DISPLAY T7

```

**ผลลัพธ์**

T7

SNAME
Smith

## 2.4 ERAO

เนื่องจากงานวิทยานิพนธ์นี้เป็นงานซึ่งเกี่ยวข้องกับงานวิทยานิพนธ์ทางสาขาวิทยาการคอมพิวเตอร์ที่มีมาก่อนหน้านี้ 2 งานคือ งานวิทยานิพนธ์ชิ้นที่ 1 เรื่อง “โปรแกรมส่วนการกำหนดโครงสร้างฐานข้อมูลและการดำเนินงานพื้นฐาน” ของคุณเพียรพร พรหมโชติ [เพียรพร (2534)] และงานวิทยานิพนธ์ชิ้นที่ 2 เรื่อง “การพัฒนาระบบจัดการฐานข้อมูลเชิงสัมพันธ์ส่วนภาษาฐานข้อมูล” ของคุณทวีศักดิ์ ศิริฐานนท์ [ทวีศักดิ์ (2534)] โดยโปรแกรมซึ่งถูกพัฒนาในงานวิทยานิพนธ์ทั้งสองต่างใช้ระบบปฏิบัติการ Ultrix-32 บนเครื่องมินิคอมพิวเตอร์ VAX 11/785 ของศูนย์คอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ และเครื่อง VAX 11/785 กำลังจะถูกเลิกใช้เร็ว ๆ นี้ เนื่องจากค่อนข้างเก่าและถูกใช้มานาน งานวิทยานิพนธ์นี้จึงเลือกใช้เครื่องคอมพิวเตอร์ SUN SPARC IPX ของภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ แทน

จากการศึกษาจากงานวิทยานิพนธ์ชิ้นที่ 2 เกี่ยวกับการดำเนินงานกับฐานข้อมูลด้วยคำสั่งปฏิบัติการของภาษาฐานข้อมูลแบบพีชคณิตสัมพันธ์ พบว่าการให้ผู้ใช้เลือกใช้คำสั่งจากรายการเลือกของเมนูมีข้อดีคือ

- ผู้ใช้ใช้งานได้ง่าย ไม่ต้องจดจำรูปแบบของแต่ละคำสั่งปฏิบัติการ และ
- การพัฒนาโปรแกรมทำได้ง่าย เพราะไม่ต้องสร้างตัวแปลภาษาสำหรับตรวจสอบไวยากรณ์ หรือวากยสัมพันธ์ของภาษาของรูปแบบคำสั่ง

แต่ถ้าต้องการศึกษาวากยสัมพันธ์และความหมายของแต่ละคำสั่งปฏิบัติการ และศึกษาชุดคำสั่งปฏิบัติการที่สมนัยกับแต่ละข้อความสั่งของภาษาสอบถามฐานข้อมูลแบบแคลคูลัสสัมพันธ์ มีความจำเป็นต้องมีรูปแบบของแต่ละคำสั่งปฏิบัติของภาษาสอบถามพีชคณิตสัมพันธ์ไว้ใช้

งานวิทยานิพนธ์นี้จึงได้ศึกษารูปแบบและความหมายของคำสั่งปฏิบัติการมาตรฐานที่ C.J. Date เสนอ [Date (1989)] พร้อมทั้งศึกษาคำสั่งเพิ่มเติมที่จำเป็นเพื่อให้สามารถเขียนชุดคำสั่งที่สมนัยกับทุกข้อความสั่งแบบแคลคูลัสสัมพันธ์ที่เป็นไปได้ พร้อมทั้งออกแบบรูปแบบคำสั่ง และพัฒนาโปรแกรมตัวแปลภาษาด้วย

เนื่องจากการดำเนินงานของคำสั่งปฏิบัติการมาตรฐานทั้งแปดของภาษาพีชคณิตสัมพันธ์ได้ถูกพัฒนามาแล้วโดยคุณทวีศักดิ์ [ทวีศักดิ์ (2534)] ในรูปแบบเมนูโดยให้ชื่อโปรแกรมที่พัฒนาว่า “RAO” (Relational Algebra Operation) ในงานวิทยานิพนธ์นี้จะยังคงใช้บางโปรแกรมการดำเนินงานของคำสั่งปฏิบัติการมาตรฐานทั้งแปดของคุณทวีศักดิ์ และเพิ่มเติมการดำเนินงานของคำสั่งปฏิบัติการใหม่ เพื่อให้ครอบคลุมการสอบถามที่เป็นไปได้มากที่สุดเท่าที่จะทำได้ โปรแกรมการดำเนินงานใหม่นี้จึงขอเรียกว่า “ERAO” (Extended Relational Algebra Operation) และประกอบด้วยคำสั่งปฏิบัติการซึ่งถูกแบ่งออกเป็น 3 กลุ่มดังนี้

1) กลุ่มคำสั่งปฏิบัติการกับเซต ประกอบด้วย

- union
- intersect
- minus
- cross

2) กลุ่มคำสั่งปฏิบัติการพิเศษ ประกอบด้วย

- select
- project
- join
- divide
- sequence
- unique

3) กลุ่มคำสั่งปฏิบัติการในรูปฟังก์ชัน ประกอบด้วย

- MAX
- MIN
- SUM
- AVG
- VAR
- STD
- CNT

ในการดำเนินงานพัฒนาตัวแปลภาษา ERAO ได้นำโปรแกรมรรถประโยชน์ที่มีอยู่บนเครื่องคอมพิวเตอร์ SUN SPARC IPX ภายใต้ระบบปฏิบัติการ SunOS มาช่วยเพื่ออำนวยความสะดวก และความเร็ว ดังนี้

“Dbx” เป็นโปรแกรมรรถประโยชน์ซึ่งใช้ในการตรวจสอบข้อผิดพลาดจากการดำเนินงานของโปรแกรม เนื่องจากการพัฒนาโปรแกรมที่มีขนาดใหญ่บ่อยครั้งที่ผู้พัฒนาไม่ทราบว่าเกิดข้อผิดพลาดจากที่ใดของโปรแกรม ผู้พัฒนาจะเสียเวลามากหากต้องตรวจสอบโปรแกรมตั้งแต่ต้นจนจบซึ่งจะทำให้ใช้เวลามากในการพัฒนาโปรแกรม ปัญหาเหล่านี้จะลดลงเมื่อใช้โปรแกรมตรวจสอบข้อผิดพลาด (Debugger) ช่วย ในระบบปฏิบัติการ SunOS จะมีโปรแกรมตรวจสอบข้อผิดพลาดชื่อ “Dbx” ซึ่งจะช่วยให้ผู้พัฒนาโปรแกรมสามารถตรวจสอบค้นหา และแก้ไขข้อผิดพลาดต่างๆ ในการทำงานของโปรแกรมเป้าหมาย (Object Program) ทั้งนี้มีข้อจำกัดว่า โปรแกรมเป้าหมายที่จะนำมาใช้งานกับ Dbx จะต้องเป็นชุดคำสั่งที่ถูกแปลด้วยตัวแปลภาษาซี (C Compiler) โดยการใช้เงื่อนไข “-g” ในการเรียกใช้โปรแกรมตัวแปลภาษาซี (cc)

```
cc filename.c -g
```

โดยที่ filename.c เป็นชื่อแฟ้มข้อมูลซึ่งบรรจุโปรแกรมภาษาซี

“Lex” เป็นโปรแกรมรรถประโยชน์ซึ่งใช้ในการผลิตโปรแกรมภาษาซีที่เรียกกันว่าสแกนเนอร์ (Scanner) สำหรับตรวจจับแบบรูปเหมือน (Pattern Mathing) ของภาษา

“Yacc” เป็นโปรแกรมรรถประโยชน์ซึ่งใช้ในการตรวจสอบ และตีความวากยสัมพันธ์ของภาษา โดยเรียกใช้สแกนเนอร์ที่ได้จากการใช้โปรแกรม Lex มาตรวจจับแบบรูปเหมือนของแต่ละข้อความสั่งของภาษา จากนั้นจะตรวจสอบวากยสัมพันธ์ และตีความสำหรับการดำเนินงานต่อไป

### บทที่ 3

#### โครงสร้างระบบ Mrdbms

เนื่องจากต้องการให้ตัวแปลภาษาสอบถามฐานข้อมูลพีชคณิตสัมพันธ์ที่พัฒนาในงานวิทยานิพนธ์นี้ สามารถทำงานร่วมกับงานวิทยานิพนธ์ที่ได้ดำเนินการมาก่อนหน้านี้ [เพียรพร (2534)] ดังนั้นในงานวิทยานิพนธ์นี้จึงใช้องค์ประกอบสำคัญของระบบงานที่ได้ดำเนินการมาก่อนมาปรับเปลี่ยนเพิ่มเติม

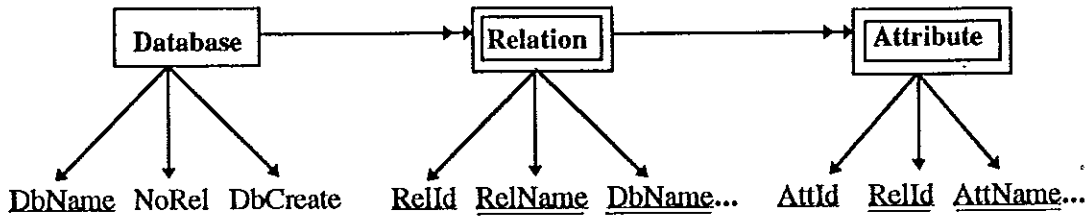
##### 3.1 องค์ประกอบของระบบ Mrdbms

จากงานวิทยานิพนธ์ที่ได้ดำเนินการมาก่อนหน้านี้ [เพียรพร (2534)] เพื่อความเข้าใจ และสามารถใช้โครงสร้างบางส่วนจากระบบเดิมได้ผู้วิจัยจึงจำเป็นต้องศึกษาโครงสร้างระบบเดิม พร้อมรายละเอียดต่าง ๆ

##### ปทานุกรมข้อมูล

ปทานุกรมข้อมูลเปรียบเสมือนฐานข้อมูลของระบบจัดการฐานข้อมูล เนื่องจากรายละเอียดต่าง ๆ ของข้อมูลที่ระบบจัดการฐานข้อมูลใช้ในการดำเนินงานต่าง ๆ จะถูกจัดเก็บในปทานุกรมข้อมูลนี้ และรูปแบบของข้อมูลในปทานุกรมข้อมูลจะเป็นรูปแบบเดียวกับรูปแบบข้อมูลที่ระบบจัดการฐานข้อมูลใช้ ซึ่งในที่นี้รูปแบบข้อมูลในฐานข้อมูลก็คือตารางข้อมูล หรือรีเลชัน (Relation) นั่นเอง รูปที่ 3.1 แสดงข้อมูล และความสัมพันธ์ระหว่างข้อมูลในปทานุกรมด้วยรูปแผนภาพโอ-อาร์ (Object-Relationship Diagram) [Theerachetmongkol and Montgomery (1980)] และโครงสร้างรีเลชันที่สมนัยกัน โดยความหมายหลักของข้อมูลในปทานุกรมมีว่า

- แต่ละฐานข้อมูลประกอบด้วยรีเลชันจำนวนมากได้ในขณะที่รีเลชันหนึ่ง ๆ ต้องอยู่ในฐานข้อมูลใดฐานข้อมูลเดียวเท่านั้น
- แต่ละรีเลชันประกอบด้วยแอตทริบิวต์จำนวนมากได้ ในขณะที่แต่ละแอตทริบิวต์ต้องอยู่ในรีเลชันเดียวเท่านั้น



โครงสร้างรีเลชันที่สมนัยกัน :

DataBase(DbName, NoRel, DbCreate)

Relation(RelId, RelName, DbName, ...)

Attribute(AttId, RelId, AttName, ...)

รูปที่ 3.1 แผนภาพโออาร์ของปทานุกรมข้อมูล และโครงสร้างรีเลชันที่สมนัยกัน

โครงสร้างแฟ้มข้อมูลของปทานุกรมข้อมูล

จากปทานุกรมข้อมูลในรูปที่ 3.1 ในแต่ละสี่เหลี่ยมผืนผ้าหรือหนึ่งรีเลชันจะถูกแทนด้วยหนึ่งแฟ้มข้อมูล โดยรายละเอียดของแต่ละแฟ้มข้อมูลมีดังนี้

Database คือแฟ้มข้อมูลสำหรับเก็บรายละเอียดข้อมูลของฐานข้อมูลทั้งหมดที่ถูกสร้างในระบบขณะเวลาหนึ่ง ๆ และหนึ่งเรคอร์ด (Record) แทนข้อมูลของหนึ่งฐานข้อมูล รายละเอียดของโครงสร้างแฟ้มข้อมูลมีดังนี้

ชื่อฟิลด์	ชนิด	รูปแบบ	ขนาด	คำอธิบาย
DbName	P	C	15	ชื่อของฐานข้อมูล
NoRel	N	I	2	จำนวนตารางข้อมูลที่มีในฐานข้อมูล
DbCreate	N	D	8	วันที่ที่มีการสร้างฐานข้อมูล

**Relation** คือแฟ้มข้อมูลสำหรับเก็บรายละเอียดของทุกตารางข้อมูลของแต่ละชื่อฐานข้อมูลที่ปรากฏในแฟ้มข้อมูล Database มีรายละเอียดดังนี้

ชื่อฟิลด์	ชนิด	รูปแบบ	ขนาด	คำอธิบาย
RelId	P	I	3	รหัสตารางข้อมูล
RelName	S	C	15	ชื่อตารางข้อมูล
DbName	S	C	15	ชื่อของฐานข้อมูลที่เป็นเจ้าของตารางข้อมูลนี้
NoAtt	N	I	2	จำนวนแอตทริบิวต์ที่มีในตารางข้อมูล
NoTup	N	I	5	จำนวนทูเปิล
TupLngth	N	I	3	ขนาดทูเปิล (ต้องไม่เกิน 256 อักขระ)
RelCreate	N	D	8	วันที่ที่สร้างตารางข้อมูล
RelLstUpdate	N	D	8	วันที่ที่มีการปรับปรุงข้อมูลในตารางข้อมูลครั้งล่าสุด
RelStatus	N	C	1	สถานะของตารางข้อมูล [ I : Independent *D : Dependent *A : Association ]
RelType	N	C	1	ชนิดของตารางข้อมูล [ B : Base Relation K : Keep Relation T : Temp Relation ]



Attribute คือแฟ้มข้อมูลสำหรับเก็บรายละเอียดของแอตทริบิวต์ที่มีในแต่ละตารางข้อมูลในฐานข้อมูลหนึ่งๆ มีรายละเอียดดังนี้

ชื่อฟิลด์	ชนิด	รูปแบบ	ขนาด	คำอธิบาย
AttrId	S	I	2	รหัสแอตทริบิวต์
AttName	P	C	15	ชื่อแอตทริบิวต์
RelId	N	I	3	รหัสตารางข้อมูลที่เป็นเจ้าของ
AttFstPos	N	I	3	ตำแหน่งที่เริ่มต้นของข้อมูลในเรคอร์ด
AttLstPos	N	I	3	ตำแหน่งที่สุดท้ายของข้อมูลในเรคอร์ด
AttLngth	N	I	3	ขนาดของข้อมูลในเรคอร์ด
AttDomPos	N	I	1	โดเมนของแอตทริบิวต์
AttStatus	N	C	1	สถานะของแอตทริบิวต์

[P : Primary Key

\*S : Secondary Key

N : Non Key ]

#### หมายเหตุ

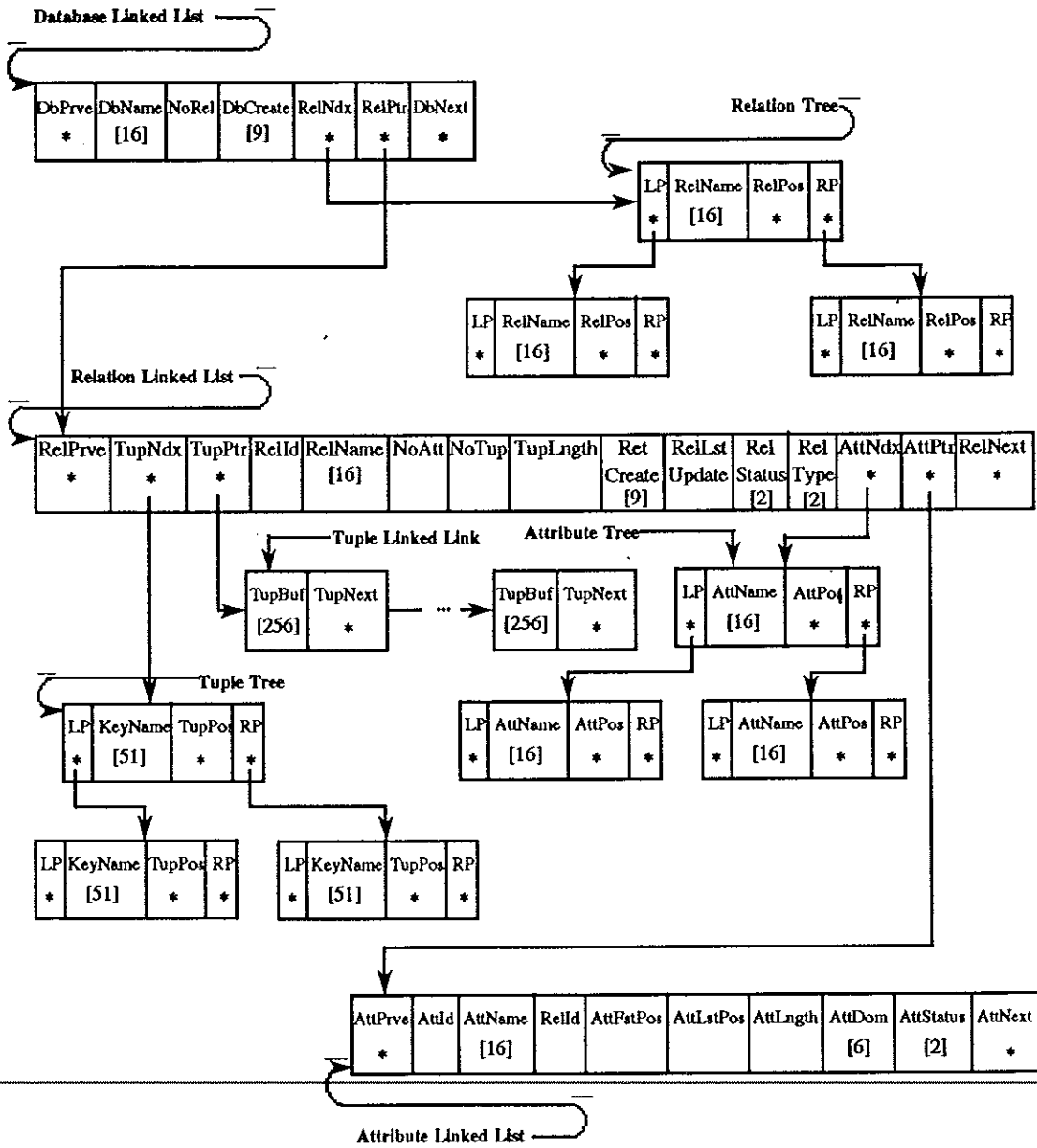
- ชนิด คือข้อมูลบอกชนิดหรือประเภทของแอตทริบิวต์ว่าเป็น Primary Key (P) Secondary Key (S) หรือ Non Key (N)
- รูปแบบ คือประเภทของข้อมูลว่าเป็นตัวอักษร (C) เลขจำนวนเต็ม (I) หรือวันที่ (D)
- ขนาด คือจำนวนอักขระ หรือตัวเลขที่มากที่สุดที่สามารถแทนข้อมูลชิ้นหนึ่ง ๆ
- เครื่องหมาย "\*" หมายถึง ยังไม่มีการพัฒนาในส่วนนี้

#### โครงสร้างข้อมูล

โครงสร้างข้อมูล (Data Structure) ของระบบ เป็นแบบรายการโยง (Linked List) และรูปต้นไม้แบบค้นทวิภาค (Binary Search Tree) ดังแสดงในรูปที่ 3.2 โดยมีรายละเอียดดังนี้

- 1) รายการโยงของฐานข้อมูล (Database Linked List) เป็นแบบหลายตัวโยง (Multilink) สร้างจากข้อมูลหนึ่งเรคอร์ดที่ปรากฏในแฟ้มข้อมูล system.dbs แต่ละโหนดจะแทนรายละเอียดของหนึ่งฐานข้อมูลในระบบ
- 2) รายการโยงของตารางข้อมูล (Relation Linked List) เป็นแบบหลายตัวโยงสร้างจากข้อมูลหนึ่งเรคอร์ดที่ปรากฏในแฟ้มข้อมูล system.rel แต่ละโหนดของรายการใช้แทนรายละเอียดของหนึ่งตารางข้อมูลในฐานข้อมูล

- 3) รายการโยงของแอตทริบิวต์ (Attribute Linked List) เป็นแบบรายการโยงคู่ (Doubly Linked List) สร้างจากข้อมูลหนึ่งเรคอร์ดในแฟ้มข้อมูล <relationN>.att แต่ละโหนดของรายการใช้แทนรายละเอียดของหนึ่งแอตทริบิวต์ของตารางข้อมูล
- 4) รายการโยงของทิวเปิล (Tuple Linked List) เป็นแบบรายการโยงเดี่ยว (Singly Linked List) สร้างจากข้อมูลหนึ่งเรคอร์ดในแฟ้มข้อมูล <relationN> ซึ่งเป็นแฟ้มข้อมูลที่ใช้สำหรับเก็บข้อมูลจริง ๆ ของหนึ่งตารางข้อมูล แต่ละโหนดของรายการใช้แทนข้อมูลหนึ่งแถวข้อมูล
- 5) ต้นไม้ของตารางข้อมูล (Relation Tree) เป็นต้นไม้แบบคันทวิภาคเป็นส่วนหนึ่งของรายการโยงของฐานข้อมูล แต่ละโหนดจะเป็นตัวแทนของหนึ่งตารางข้อมูลใช้ในการค้นหาตารางข้อมูลต่าง ๆ ในหนึ่งฐานข้อมูล
- 6) ต้นไม้ของแอตทริบิวต์ (Attribute Tree) เป็นต้นไม้แบบคันทวิภาคเป็นส่วนหนึ่งของรายการโยงของแอตทริบิวต์ ซึ่งแต่ละโหนดจะเป็นตัวแทนของหนึ่งแอตทริบิวต์ใช้ในการค้นหาแอตทริบิวต์ต่าง ๆ ในหนึ่งตารางข้อมูล
- 7) ต้นไม้ของทิวเปิล (Tuple Tree) เป็นต้นไม้แบบคันทวิภาคเป็นส่วนหนึ่งของรายการโยงของทิวเปิล ซึ่งแต่ละโหนดจะเป็นตัวแทนของหนึ่งทิวเปิลใช้ในการค้นหาข้อมูลจริง ๆ ของหนึ่งตารางข้อมูล

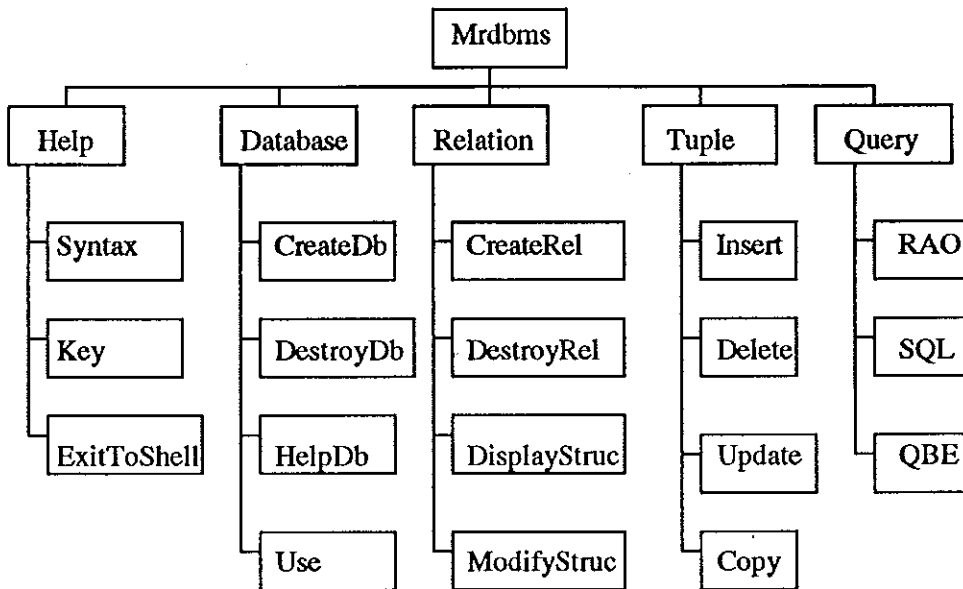


รูปที่ 3.2 โครงสร้างข้อมูลของปทานุกรมข้อมูลและที่อยู่ในหน่วยความจำหลัก

หมายเหตุ \* หมายถึง ตัวแปรชนิด Pointer

## แผนภูมิระบบ Mrdbms

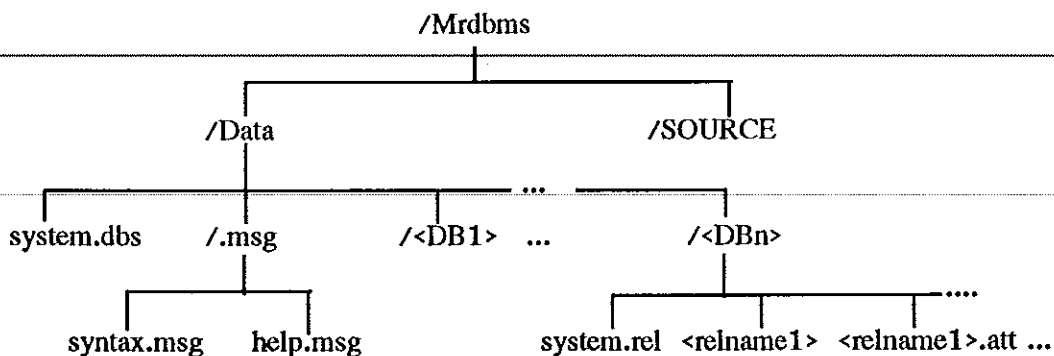
แผนภูมิระบบ (System Chart) ของระบบงานเดิม ซึ่งชื่อว่า "Mrdbms" ในรูปที่ 3.3 แสดงให้เห็นการดำเนินงานในส่วนต่างๆ



รูปที่ 3.3 แผนภูมิระบบ Mrdbms

## สารบบแฟ้มข้อมูล

สารบบแฟ้มข้อมูล (File Directory) สำหรับการจัดเก็บแฟ้มข้อมูลทั้งหมดของระบบถูกออกแบบโครงสร้างเป็นแผนผังต้นไม้ (Tree Diagram) โดยได้เปลี่ยนโหนดรากจากเดิมมาเป็นอันใหม่ซึ่งจะใช้ดำเนินงานบนเครื่องคอมพิวเตอร์ SUN SPARC IPX รูปที่ 3.4 แผนผังต้นไม้แสดงการจัดเก็บสารบบแฟ้มข้อมูลของ Mrdbms



รูปที่ 3.4 การจัดเก็บสารบบแฟ้มข้อมูลของ Mrdbms

สารบบ `"/Mrdbms/SOURCE"` ใช้เก็บโปรแกรมที่พัฒนาทั้งหมด

สารบบ `"/Mrdbms/Data"` ใช้เก็บแฟ้มข้อมูลซึ่งเป็นข้อมูลที่เกี่ยวข้องกับการทำงานทั้งหมดของระบบ

- แฟ้มข้อมูล `"system.dbs"` ใช้เก็บรายละเอียดของแต่ละฐานข้อมูลที่ถูกสร้างขึ้นในระบบ
- สารบบ `"usr1/g3424903/THESIS/Data/.msg"` ใช้เก็บแฟ้มข้อมูลเกี่ยวกับคำแนะนำในการใช้คำสั่งปฏิบัติการต่างๆ
- สารบบ `"usr1/g3424903/THESIS/Data/<DBn>"` โดย  $n=1,2,\dots$  ใช้เก็บแฟ้มข้อมูลบอกรายละเอียดโครงสร้างของแต่ละฐานข้อมูลพร้อมข้อมูลที่ถูกสร้างขึ้นโดยแต่ละสารบบ ประกอบด้วยแฟ้มข้อมูลต่างๆ ของหนึ่งฐานข้อมูล คือ
  - . แฟ้มข้อมูล `"system.rel"` ใช้เก็บรายละเอียดของทุกตารางข้อมูลในฐานข้อมูลนั้นๆ
  - . `<relnameN>` โดย  $N=1,2,\dots$  เป็นชื่อแฟ้มข้อมูลเก็บข้อมูลจริงของแต่ละตารางข้อมูลในฐานข้อมูล
  - . `<relname N>.att` โดย  $N=1,2,\dots$  เป็นชื่อแฟ้มข้อมูลที่จัดเก็บรายละเอียดของแอตทริบิวของแต่ละตารางข้อมูลในฐานข้อมูล

### 3.2 การปรับเปลี่ยน และเพิ่มเติมระบบ

ดังได้กล่าวมาข้างต้นว่างานวิทยานิพนธ์นี้เกี่ยวข้องกับงานวิทยานิพนธ์ที่ได้ดำเนินการมาก่อนหน้านี้ 2 งาน โดยได้ทำการย้ายโปรแกรมของงานเดิมจากเครื่องคอมพิวเตอร์ VAX 11/785 ซึ่งใช้ระบบปฏิบัติการ Ultrix-32 มาติดตั้งบนเครื่องคอมพิวเตอร์ SUN SPARC IPX ซึ่งใช้ระบบปฏิบัติการ SunOS แก้ไขปรับปรุงโปรแกรมในงานวิจัยเดิมบางส่วนเพื่อให้ทำงานบนเครื่อง SUN SPARC IPX ได้ และเพิ่มโปรแกรมการทำงานบางส่วนเพื่อให้งานมีความสมบูรณ์มากขึ้น

การปรับเปลี่ยนงานในวิทยานิพนธ์ชิ้นที่ 1 [เพียรพร (2534)] ได้นำโปรแกรมของงานส่วนนี้ซึ่งเป็นส่วนของการใช้ภาษาสำหรับกำหนดโครงสร้างและรายละเอียดของฐานข้อมูล (DDL-Data Definition Language) และใช้ภาษาสำหรับดำเนินการพื้นฐาน (เพิ่มข้อมูลเข้า ลบข้อมูลออก และเปลี่ยนแปลงค่าข้อมูล) กับฐานข้อมูล (DML-Data Manipulation Language) โดยตัดรายละเอียดในส่วนการเลือกใช้ภาษาคำถามในการดำเนินงานเพื่อให้ได้มาซึ่งสารสนเทศจากข้อมูลที่มีในฐานข้อมูล ซึ่งเป็นรายการย่อย "RAO" , "SQL" และ "QBE" ออก ทั้งนี้เนื่องจากข้อจำกัดของความแตกต่างของโปรแกรมมอรรถประโยชน์ระหว่างเครื่องคอมพิวเตอร์ VAX 11/785 และเครื่องคอมพิวเตอร์ SUN SPARC IPX รูปแบบของข้อความสั่งถูกออกแบบและพัฒนาอยู่ในรายการเลือกจากเมนู [เพียรพร (2534)] เพื่อให้ความสะดวกแก่ผู้ใช้และหลีกเลี่ยงการเขียนโปรแกรมสำหรับแปลข้อความสั่งต่างๆ ในการพัฒนาโปรแกรมรายการเลือกจากเมนูดังกล่าวผู้พัฒนาใช้โปรแกรมมอรรถประโยชน์ ncourse ซึ่งเป็นโปรแกรมอำนวยความสะดวกในการพัฒนาโปรแกรมเกี่ยวกับการจัดหน้าจอแบบต่างๆ และการรับข้อมูลจากแป้นพิมพ์เพื่อแสดงบนหน้าจอ ระบบปฏิบัติการ SUN SPARC IPX ไม่มี ncourse มีแต่ curse ซึ่งเป็นโปรแกรมมอรรถประโยชน์ซึ่งทำงานคล้าย ncourse แต่มีขีดความสามารถต่ำกว่า ดังนั้นในการย้ายโปรแกรมเก่ามาติดตั้งใหม่บน SUN SPARC IPX จึงต้องมีการปรับเปลี่ยน และเพิ่มเติม เพื่อให้โปรแกรมของงานเดิมยังสามารถทำงานได้เช่นเดิมทุกประการ และตัดโปรแกรมบางส่วนของงานเดิมที่ใช้ไม่ได้บน SUN SPARC IPX ออก โปรแกรมที่ถูกพัฒนาเพิ่มเติมคืองานส่วนรับข้อมูลจากแป้นพิมพ์และส่วนการแสดงข้อความและผลลัพธ์บนหน้าจอ

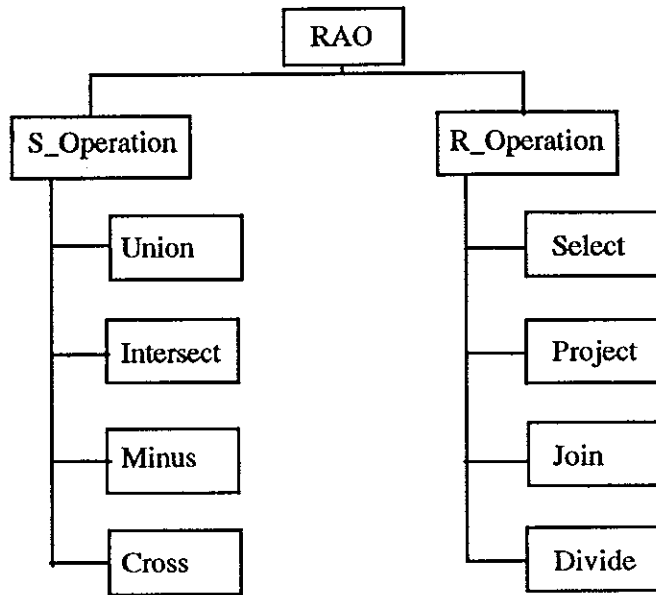
การปรับเปลี่ยนงานในวิทยานิพนธ์ชิ้นที่ 2 [ทวีศักดิ์ (2534)] ได้นำโปรแกรมของงานส่วนนี้ซึ่งเป็นส่วนของระบบจัดการฐานข้อมูลส่วนภาษารฐานข้อมูล (QRL-Query/Retrieval Language) โดยรูปแบบของข้อความสั่งถูกออกแบบและพัฒนาในรูปรายการเลือกจากเมนู ในส่วนของงานวิทยานิพนธ์นี้ผู้วิจัยได้นำเฉพาะโปรแกรมย่อยในการดำเนินงานพื้นฐานสำหรับ 8 คำสั่ง อันประกอบด้วยคำสั่ง Union, Intersect, Minus, Cross Product, Select, Project, Join และ Divide โดยการเรียกใช้โปรแกรมย่อยเหล่านี้ได้เปลี่ยนแปลงการรับข้อมูลจากเดิมซึ่งรับมาจากแป้นพิมพ์มาเป็นการรับค่า และตัวแปรเหล่านั้นจากแถวคำสั่ง โดยใช้โปรแกรมมอรรถประโยชน์ Lex และ Yacc ช่วย

นอกจากนี้โปรแกรมย่อยที่นำมาใช้เหล่านั้นจะถูกแก้ไขในบางส่วนเนื่องจากตัวแปลภาษา C ที่ใช้ในงานวิจัยเดิมคือ Standard C ซึ่งในระบบปฏิบัติการ SunOS ไม่มี ในระบบนี้มีแต่ Classic C ซึ่งมีขีดความสามารถต่ำกว่า ตัวอย่างความแตกต่างเช่นการใช้งานกับตัวแปรพอยน์เตอร์ (Pointer) และการขอใช้เนื้อที่ในหน่วยความจำ เป็นต้น

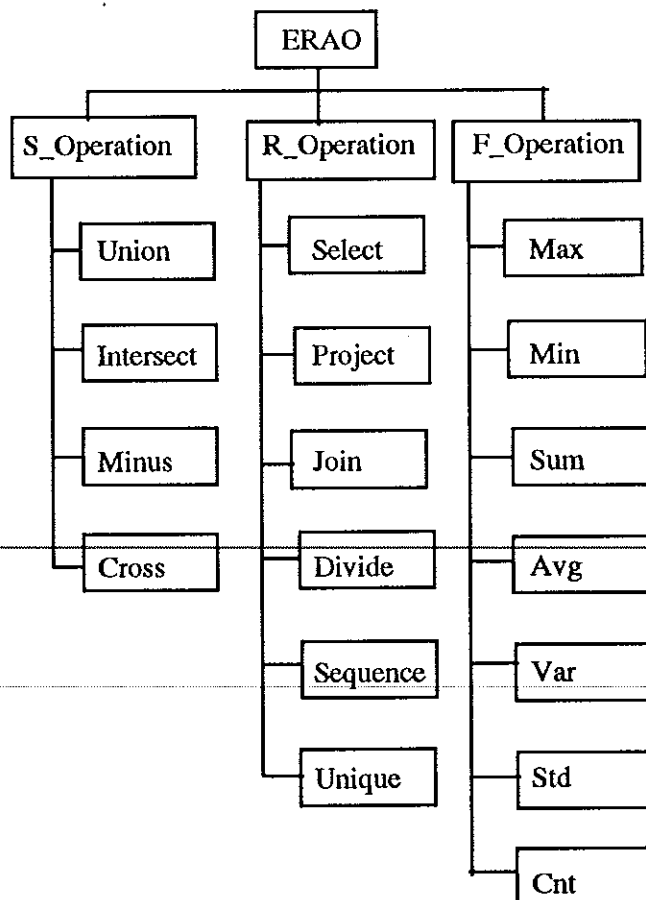
### 3.3 โครงสร้างระบบ Mrdbms ใหม่

โครงสร้างระบบภายในงานวิทยานิพนธ์นี้จะประกอบด้วยส่วนหลักๆ 2 ส่วน คือ ส่วนที่ 1 การกำหนดโครงสร้างฐานข้อมูลและการดำเนินงานพื้นฐาน ส่วนที่ 2 ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ส่วนภาษาฐานข้อมูลเช่นเดียวกับระบบ Mrdbms เดิม

ส่วนการกำหนดโครงสร้างฐานข้อมูลและการดำเนินงานพื้นฐานใช้โปรแกรมจากงานวิทยานิพนธ์ที่ได้ดำเนินการมาก่อนหน้านี้ [เพียรพร (2534)] ส่วนระบบจัดการฐานข้อมูลเชิงสัมพันธ์ส่วนภาษาฐานข้อมูล ได้นำโปรแกรมจากวิทยานิพนธ์ที่ได้ดำเนินการมาก่อนหน้านี้ [ทวีศักดิ์ (2534)] ที่ชื่อว่า "RAO" บางส่วนมาแก้ไขปรับปรุง พร้อมทั้งได้ออกแบบเพิ่มเติมเพื่อให้มีความสมบูรณ์มากขึ้น โดยโปรแกรมในส่วนของ RAO จะถูกแทนที่ด้วยโปรแกรม ERAO ของงานวิทยานิพนธ์นี้ แผนภูมিরะบบงานของ RAO และ ERAO แสดงดังรูปที่ 3.5 และรูปที่ 3.6 ตามลำดับ



รูปที่ 3.5 แผนภูมิระบบ RAO



รูปที่ 3.6 แผนภูมิระบบ ERAO



## บทที่ 4

### ตัวแปลภาษา ERAO

ตัวแปลภาษา ERAO ในงานวิทยานิพนธ์นี้ถูกพัฒนาโดยใช้ตัวแปลภาษาคลาสสิกซี (Classic C) ซึ่งดำเนินงานบนเครื่องคอมพิวเตอร์ SUN SPARC IPX โดยใช้โปรแกรมอรรถประโยชน์ Lex ช่วยในการวิเคราะห์คำศัพท์ และโปรแกรมอรรถประโยชน์ Yacc ช่วยในการตรวจสอบวากยสัมพันธ์ และตีความ

#### 4.1 องค์ประกอบของตัวแปลภาษา ERAO

ตัวแปลภาษา ERAO ประกอบด้วยองค์ประกอบหลักคือเซตของอักขระที่ใช้ คำสงวน ตัวแปร คำคงที่ และการปฏิบัติการเชิงสัมพันธ์ โดยมีรายละเอียดดังนี้

##### เซตของอักขระ (Character Sets)

เซตของอักขระใน ERAO ประกอบด้วย

- 1) ตัวอักษร ประกอบด้วยตัวอักษร 52 ตัว คือ อักษรตัวใหญ่ A-Z และอักษรตัวเล็ก a-z
- 2) ตัวเลข ประกอบด้วยตัวเลข 10 จำนวน คือ 0-9
- 3) อักขระพิเศษ ประกอบด้วยอักขระทั้งหมด 11 ตัว คือ " > < = ! [ ] / . \_ และช่องว่าง

##### คำสงวน (Reserved Word)

คำสงวนใน ERAO จะถือว่าอักษรตัวใหญ่ หรือตัวเล็กไม่ต่างกัน คำสงวนทั้งหมดมีดังนี้

---

AVG CNT CROSS DIVIDE INTERSECT JOIN MAX  
MIN MINUS PROJECT SELECT SEQUENCE STD  
SUM UNION UNIQUE VAR

---

##### ตัวแปร (Variable)

กฎการตั้งชื่อตัวแปรของ ERAO คือ ต้องขึ้นต้นด้วยอักษร และตามด้วยตัวอักษร หรือตัวเลข หรืออักขระพิเศษ “\_” (Under Score) ก็ได้ โดยชื่อตัวแปรจะต้องไม่ตรงกับคำสงวน ตัวอย่างของการตั้งชื่อตัวแปร เช่น SNAME, PNUM หรือ S1 เป็นต้น

## ค่าคงที่ (Constant)

ค่าคงที่มี 2 ประเภท คือ

- 1) ค่าคงที่ชนิดเลขจำนวนเต็ม (Integer Constant) เป็นค่าคงที่ที่เป็นเลขจำนวน จะมีเพียงตัวเลขเท่านั้น ซึ่งจะต้องขึ้นต้นด้วยตัวเลขจำนวนนับ 1-9 และตามด้วยตัวเลข 0-9
- 2) ค่าคงที่ชนิดอักขระ (Character Constant) จะได้แก่สายอักขระ (String) ใดๆ ตั้งแต่หนึ่งอักขระขึ้นไป ซึ่งทุกอักขระในสายอักขระจะต้องเป็นอักขระที่อยู่ในเซตของอักขระ

## การดำเนินงานเชิงเปรียบเทียบ (Comparison Operation)

การปฏิบัติการเชิงเปรียบเทียบซึ่งใช้ในบางคำสั่งปฏิบัติการพีชคณิตสัมพันธ์ของ ERAO ประกอบด้วยการเปรียบเทียบค่าสองค่า โดยมีเครื่องหมายเปรียบเทียบ (Comparison Operator) คั่นระหว่างค่าสองค่า เครื่องหมายเปรียบเทียบที่ใช้มีดังนี้

">"	หมายถึง	มากกว่า
"<"	หมายถึง	น้อยกว่า
">="	หมายถึง	มากกว่า หรือเท่ากับ
"<="	หมายถึง	น้อยกว่า หรือเท่ากับ
"="	หมายถึง	เท่ากับ
"!="	หมายถึง	ไม่เท่ากับ
"<>"	หมายถึง	ไม่เท่ากับ

## รูปแบบและความหมายของคำสั่งปฏิบัติการใน ERAO

รูปแบบในการสอบถามฐานข้อมูลด้วยแต่ละคำสั่งปฏิบัติการใน ERAO ได้ถูกออกแบบให้อยู่ในรูปแบบคำสั่งคล้ายรูปแบบของภาษาแอสเซมบลี กล่าวคือหนึ่งแถวคำสั่งใช้สำหรับดำเนินงานหนึ่งอย่าง โดยแต่ละแถวคำสั่งประกอบด้วยคำสั่งปฏิบัติการ ตารางข้อมูลสำหรับการใช้ในการดำเนินงาน และตารางข้อมูลผลลัพธ์ดังรูปแบบต่อไปนี้

<result> = <operator> <operand>

หรือ

<result> = <operator> <operand1> <operand2>

โดย <operator> คือคำสั่งปฏิบัติการใน 3 กลุ่มข้างต้น

<operand> <operand1> และ <operand2> คือตารางข้อมูลสำหรับใช้เป็นข้อมูลในการดำเนินงาน

<result> คือตารางข้อมูลสำหรับเก็บผลลัพธ์จากการดำเนินงานของคำสั่งปฏิบัติการกับข้อมูล ซึ่งอยู่ทางขวามือของเครื่องหมาย "="

ความหมายของสัญลักษณ์ซึ่งใช้ในการเขียนรูปแบบคำสั่งปฏิบัติการ

- [ ] หมายถึงข้อความภายในวงเล็บนี้จะเลือกใช้หรือไม่ใช้ก็ได้ เวลาใช้งานไม่ต้องพิมพ์วงเล็บนี้
- < > หมายถึงข้อความภายในเครื่องหมายนี้เป็นส่วนย่อยของคำสั่งปฏิบัติการ เช่นตัวแปรหรือค่าคงที่ซึ่งผู้ใช้เป็นผู้กำหนดสำหรับใช้แทนค่าข้อมูลจริงลงไป เวลาใช้งานไม่ต้องพิมพ์เครื่องหมายนี้
- ... หมายถึงยังมีต่อ ละไว้ในฐานที่เข้าใจ
- ' ' หมายถึงข้อความภายในเครื่องหมายนี้จะต้องเป็นสัญลักษณ์ หรือค่าเช่นเดียวกับข้อความนั้น
- คำที่เป็นอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ทั้งคำจะหมายถึงคำสงวน ภายในคำสั่งปฏิบัติการต้องปรากฏคำเหล่านั้น

กลุ่มคำสั่งปฏิบัติการกับเซต

รูปแบบ <result> '=' UNION <table1> <table2>  
 <result> '=' INTERSECT <table1> <table2>  
 <result> '=' MINUS <table1> <table2>  
 <result> '=' CROSS <table1> <table2>

ตารางข้อมูลสองตารางซึ่งนำมาดำเนินการกับคำสั่งปฏิบัติการกับเซตจะต้องมีคุณสมบัติ union-compatible และตารางข้อมูลผลลัพธ์ที่ได้จากคำสั่งปฏิบัติการเหล่านี้จะมีคุณสมบัติ union-compatible กับตารางข้อมูลทั้งสองด้วย

ความหมาย union	เป็นคำสั่งปฏิบัติการในการรวมแถวข้อมูลของตารางข้อมูล <table1> และ <table2> มาไว้ในตารางข้อมูล <result> แถวข้อมูลใดที่ซ้ำกันจากตารางข้อมูลทั้งสองจะถูกนำมาเพียงแถวเดียว
intersect	เป็นคำสั่งปฏิบัติการในการเลือกแถวข้อมูลที่เหมือนกันจากตารางข้อมูล <table1> และ <table2> มาไว้ในตารางข้อมูล <result>
minus	เป็นคำสั่งปฏิบัติการในการเลือกแถวข้อมูลที่อยู่ในตารางข้อมูล <table1> แต่ไม่อยู่ในตารางข้อมูล <table2> มาไว้ในตารางข้อมูล <result>
cross	เป็นคำสั่งปฏิบัติการในการจับคู่แถวข้อมูลทุกคู่ที่เป็นไปได้ของตารางข้อมูล <table1> และ <table2> มาไว้ในตารางข้อมูล <result> โดยแต่ละแถวข้อมูลในตารางข้อมูล <table1> จะถูกต่อท้ายด้วยแถวข้อมูลในตารางข้อมูล <table2>

**ตัวอย่าง** `ResultT = union temp1 temp2`  
 เป็นการรวมแถวข้อมูลในตารางข้อมูล temp1 และ temp2 ไว้ในตารางข้อมูล ResultT โดยแถวข้อมูลที่ซ้ำกัน (มีอยู่ทั้งในตารางข้อมูล temp1 และ temp2) จะนำมาเก็บในตารางข้อมูล ResultT เพียงแถวเดียว

### กลุ่มคำสั่งปฏิบัติการพิเศษ

**รูปแบบ** `<result> '=' SELECT <table> '[' <attribute> <operator> <constant> ']'`

**ความหมาย** select เป็นคำสั่งปฏิบัติการในการเลือกแถวข้อมูลจากตารางข้อมูล <table> โดยใช้ค่าของแอตทริบิวต์ <attribute> เป็นเกณฑ์ในการเลือกแถวข้อมูล โดย <operator> คือเครื่องหมายแทนการเปรียบเทียบค่า <constant> คือค่าคงที่ซึ่งอาจเป็นชนิดเลขจำนวน หรือค่าคงที่ชนิดอักขระ

**ตัวอย่าง** `ResultT = select temp [NAME=Suda]` หรือ  
`ResultT = select temp [NAME="Suda"]`  
 เป็นการเลือกแถวข้อมูลจากตารางข้อมูล temp ซึ่งมีค่าข้อมูลของแอตทริบิวต์ NAME เท่ากับ Suda

**รูปแบบ** `<result> '=' PROJECT <table> '[' <attribute1> [<attribute2>...] ']'`

**ความหมาย** project เป็นคำสั่งปฏิบัติการในการสกัดข้อมูลของแอตทริบิวต์ <attributeN> โดยที่ N=1,2,3,... จากตารางข้อมูล <table> อย่างน้อยต้องมีหนึ่งแอตทริบิวต์ กรณีที่มีมากกว่าหนึ่งแอตทริบิวต์แยกแอตทริบิวต์เหล่านั้นด้วยช่องว่างอย่างน้อยหนึ่งช่องว่าง

**ตัวอย่าง** `ResultT = project temp [NAME ADD]`  
 เป็นการสกัดข้อมูลเฉพาะแอตทริบิวต์ NAME และ ADD จากตารางข้อมูล temp

**รูปแบบ** <result> '=' JOIN <table1> <table2> '[' <attribute> ']' หรือ  
 <result> '=' JOIN <table1> <table2> '[' <attribute1> <operator>  
 <attribute2>·']'

**ความหมาย** join เป็นคำสั่งปฏิบัติการในการจับคู่แถวข้อมูลของสองตารางข้อมูลทำนองเดียวกับคำสั่งปฏิบัติการ Cross ต่างกันที่แถวข้อมูลที่จะนำมาจับคู่กันเป็นแถวข้อมูลใหม่จะต้องมีค่าของแอตทริบิวต์ <attribute> <attribute1> และ <attribute2> เป็นไปตามเงื่อนไขที่กำหนด ในรูปแบบแรกตารางข้อมูล <table1> และ <table2> ต้องมีแอตทริบิวต์ <attribute> และเงื่อนไขการเปรียบเทียบค่าคือ เท่ากับ ในรูปแบบที่สองตารางข้อมูล <table1> ต้องมีแอตทริบิวต์ <attribute1> และตารางข้อมูล <table2> ต้องมีแอตทริบิวต์ <attribute2> ซึ่งใช้โดเมนร่วมกับ โดย <operator> คือเครื่องหมายแทนการเปรียบเทียบค่า

**ตัวอย่าง** Result{T = join temp1 temp2 [Number]  
 เป็นการจับคู่แถวข้อมูลจากตารางข้อมูล temp1 และ temp2 โดยต่อท้ายแถวข้อมูลจากตารางข้อมูล temp1 ด้วยแถวข้อมูลจากตารางข้อมูล temp2 แถวข้อมูลที่จะนำมาจับคู่กันจากทั้งสองตารางจะต้องมีค่าข้อมูลของแอตทริบิวต์ Number เท่ากัน

**รูปแบบ** <result> '=' DIVIDE <table1> <table2>

**ความหมาย** divide เป็นคำสั่งปฏิบัติการสำหรับดำเนินงานกับสองตารางข้อมูล โดยตารางข้อมูล <table1> จะต้องเป็นตารางข้อมูลไบนารีทำหน้าที่เป็นตัวตั้ง และตารางข้อมูล <table2> ต้องเป็นตารางข้อมูลยูนิารีทำหน้าที่เป็นตารางข้อมูลตัวหาร แอตทริบิวต์ของตารางข้อมูล <table2> ต้องใช้โดเมนเดียวกันกับหนึ่งแอตทริบิวต์ในตารางข้อมูล <table1> ผลลัพธ์ที่ได้จะเป็นตารางข้อมูลยูนิารีซึ่งประกอบด้วยแอตทริบิวต์หนึ่งของตารางข้อมูล <table1> ที่ไม่ได้ใช้โดเมนเดียวกับตารางข้อมูล <table2> และมีแต่ละค่าแอตทริบิวต์ที่ใช้โดเมนเดียวกับตารางข้อมูล <table2> จับคู่กับทุกค่าของแอตทริบิวต์ในตารางข้อมูล <table2>

**ตัวอย่าง** ResultT = divide temp1 temp2

เป็นการดำเนินงานระหว่างตารางข้อมูล temp1 ซึ่งเป็นตารางข้อมูลไบนารี และ ตารางข้อมูล temp2 ซึ่งเป็นตารางข้อมูลยูนิารี ภายในตารางข้อมูล temp1 จะต้องมีแอตทริบิวหนึ่งใช้โดเมนร่วมกับตารางข้อมูล temp2 ตารางข้อมูลผลลัพธ์ ResultT จะประกอบด้วยแอตทริบิวเพียงแอตทริบิวเดียวจากตารางข้อมูล temp1 ที่ไม่ใช่โดเมนร่วมกับตารางข้อมูล temp2 ซึ่งแต่ละค่าของแอตทริบิวที่ใช้โดเมนร่วมกับตารางข้อมูล temp1 จับคู่กับทุกค่าของแอตทริบิวในตารางข้อมูล temp2

**รูปแบบ** <result> '=' SEQUENCE <table> '[' <attribute1> [<attribute2>...] ']'

**ความหมาย** sequence เป็นคำสั่งปฏิบัติการในการลำดับแถวข้อมูลใหม่โดยใช้แอตทริบิว <attributeN> โดยที่ N=1,2,3,... เป็นตัวกำหนดการจัดลำดับแถวข้อมูล การจัดลำดับใหม่จะถูกกำหนดโดยแอตทริบิวตัวแรกและตัวถัดๆ ไปตามลำดับ อย่างน้อยจะต้องมีหนึ่งแอตทริบิวที่ใช้ในการจัดลำดับ กรณีที่มีมากกว่าหนึ่งแอตทริบิวจะแยกแอตทริบิวเหล่านั้นด้วยช่องว่างอย่างน้อยหนึ่งช่อง

**ตัวอย่าง** ResultT = sequence temp [Number Name]

เป็นการจัดลำดับแถวข้อมูลในตารางข้อมูล temp ใหม่ ตารางข้อมูลผลลัพธ์ ResultT จะประกอบด้วยจำนวนของแถวข้อมูล และจำนวนแอตทริบิวเท่ากับ ตารางข้อมูล temp ตารางข้อมูลผลลัพธ์จะเรียงลำดับแถวข้อมูลใหม่ตามแอตทริบิว Number และ Name ตามลำดับ

**รูปแบบ** <result> '=' UNIQUE <table>

**ความหมาย** unique เป็นคำสั่งปฏิบัติการในการกำจัดแถวข้อมูลซ้ำกันในตารางข้อมูล <table>

**ตัวอย่าง** ResultT = unique temp

เป็นการกำจัดแถวข้อมูล ตารางข้อมูลผลลัพธ์ ResultT จะได้แถวข้อมูลที่ไม่ซ้ำกันจากตารางข้อมูล temp จำนวนแถวข้อมูลของตารางข้อมูลผลลัพธ์จะเท่ากับหรือน้อยกว่าตารางข้อมูล temp เสมอ และโครงสร้างของตารางข้อมูลผลลัพธ์ ResultT เหมือนกับตารางข้อมูล temp

**กลุ่มคำสั่งปฏิบัติการในรูปฟังก์ชัน**

**รูปแบบ** <result> '=' <case of function> <table> '[' <attribute> ']' หรือ  
<result> '=' <case of function> <table> '[' <attribute1> '/'  
<attribute2> [<attribute3>...] ']'

**ความหมาย** เป็นคำสั่งปฏิบัติการเพื่อหาฟังก์ชันมี 2 รูปแบบ รูปแบบแรกจะเป็นการหาฟังก์ชันจากแอตทริบิวต์ <attribute> ในทุกๆ แถวข้อมูลของตารางข้อมูล <table> ตารางข้อมูลผลลัพธ์ <result> จะประกอบด้วยแอตทริบิวต์เพียงแอตทริบิวต์เดียวคือแอตทริบิวต์ที่ได้จากการหาฟังก์ชัน รูปแบบที่สองจะเป็นการหาฟังก์ชันจากแอตทริบิวต์ <attribute1> สำหรับแต่ละกลุ่มของแถวข้อมูลของตารางข้อมูล <table> แถวข้อมูลจะถูกแยกกลุ่มตามแอตทริบิวต์ <attributeN> โดยที่ N=2,3,... ตารางข้อมูลผลลัพธ์ <result> จะประกอบด้วยแอตทริบิวต์ <attributeN> โดยที่ N=2,3,... และแอตทริบิวต์ที่ได้จากการหาฟังก์ชัน คำสั่งปฏิบัติการเพื่อหาฟังก์ชัน CNT (Count) สามารถแทน <attribute> ในรูปแบบแรก และ <attribute1> ในรูปแบบที่สองด้วยเครื่องหมาย "\*" เพื่อแสดงการนับโดยไม่เจาะจงแอตทริบิวต์

<case of function> สามารถแทนได้ด้วยคำสั่งเพื่อบ่งบอกถึงฟังก์ชันที่ต้องการคำนวณมีดังนี้ MAX, MIN, SUM, AVG, VAR, STD และ CNT

โดยที่

- MAX เป็นการหาค่ามากที่สุด
- MIN เป็นการหาค่าน้อยที่สุด
- SUM เป็นการคำนวณค่าผลรวม
- AVG เป็นการคำนวณค่าเฉลี่ย
- VAR เป็นการคำนวณค่าความแปรปรวน
- STD เป็นการคำนวณค่าส่วนเบี่ยงเบนมาตรฐาน
- CNT เป็นการนับจำนวนชุดของข้อมูล

**ตัวอย่าง** ResultT = sum temp[Mark]

เป็นการหาผลรวมของค่าแอดทริบิว Mark จากทุกแถวข้อมูลในตารางข้อมูล temp ตารางข้อมูลผลลัพธ์ ResultT ประกอบด้วยแอดทริบิวเพียงแอดทริบิวเดียวชื่อ \_SUM

ResultT = cnt temp[\*]

เป็นการนับจำนวนแถวข้อมูลจากทุกแถวข้อมูลของตารางข้อมูล temp ตารางข้อมูลผลลัพธ์ ResultT จะประกอบด้วยแอดทริบิวชื่อ \_CNT

ResultT = max temp2[Mark / Number]

เป็นการหาค่าของแอดทริบิว Mark ที่มากที่สุดในแต่ละกลุ่มของแถวข้อมูลจากตารางข้อมูล temp แถวข้อมูลแต่ละกลุ่มจะมีค่าของแอดทริบิว Number เท่ากัน ตารางข้อมูลผลลัพธ์ ResultT จะประกอบด้วยแอดทริบิวชื่อ Number และ \_MAX



วากยสัมพันธ์ของตัวแปลภาษา ERAO อธิบายโดยใช้แผนภาพวากยสัมพันธ์ (Syntax Diagram) ดังรายละเอียดแสดงในภาคผนวก ค ตัวอย่างการใช้คำสั่งปฏิบัติการ ERAO ต่อไปนี้เป็นตัวอย่างแสดงการสอบถามที่สมนัยกันระหว่างการใช้ภาษา SQL และ ERAO ฐานข้อมูลที่ใช้คือฐานข้อมูลในรูปที่ 2.1

#### ตัวอย่าง 4.1

คำถาม Get full details of all suppliers in Paris.

SQL

```
SELECT *
FROM S
WHERE CITY = 'Paris';
```

ERAO

```
R1 = SELECT S[CITY = "Paris"]
```

ผลลัพธ์

R1

S#	SNAME	STATUS	CITY
S2	Jones	10	Paris
S3	Blake	30	Paris

#### ตัวอย่าง 4.2

คำถาม Get part numbers for all parts.

SQL

```
SELECT P#
FROM P;
```

ผลลัพธ์

R1

P#
P1
P2
P3
P4
P5
P6

ERAO

```
R1 = PROJECT P[P#]
```

ตัวอย่าง 4.3

คำถาม Get supplier numbers and names for suppliers in Paris.

SQL

```
SELECT S#,SNAME
FROM S
WHERE CITY = 'Paris';
```

ผลลัพธ์

S#	SNAME
S2	Jones
S3	Blake

ERAO

```
R1 = SELECT S[CITY = "Paris"]
R2 = PROJECT R1[S# SNAME]
```

ตัวอย่าง 4.4

คำถาม Get full details of all suppliers who are located in London or having status greater than 20.

SQL

```
SELECT *
FROM S
WHERE CITY = 'London' OR STATUS > 20;
```

ERAO

```
R1 = SELECT S[CITY = "London"]
R2 = SELECT S[STATUS > 20]
R3 = UNION R1 R2
```

ผลลัพธ์

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

ตัวอย่าง 4.5

คำถาม Get part names for red parts with weight greater than 5 kilograms.

SQL

```
SELECT PNAME
FROM P
WHERE COLOR = 'Red'
      AND WEIGHT > 5;
```

ผลลัพธ์

R4

PNAME
Nut
Screw
Cog

ERAO

```
R1 = SELECT P[COLOR = "Red"]
R2 = SELECT P[WEIGHT > 5]
R3 = INTERSECT R1 R2
R4 = PROJECT R3[PNAME]
```

ตัวอย่าง 4.6

คำถาม Get full details of all parts but not part P1 or P2.

SQL

```
SELECT *
FROM P
WHERE P# NOT IN ('P1','P2');
```

ERAO

```
R1 = SELECT P[P# = "P1"]
R2 = SELECT P[P# = "P2"]
R3 = UNION R1 R2
R4 = MINUS P R3
```

ผลลัพธ์

R4

P#	PNAME	COLOR	WEIGHT	CITY
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

**ตัวอย่าง 4.7**

**คำถาม** Get supplier names for suppliers who supply part P1.

**SQL**

```
SELECT SNAME
FROM S,SP
WHERE S.S# = SP.S#
AND P# = 'P1';
```

**ERAO**

```
R1 = SELECT SP[P# = "P1"]
R2 = JOIN S R1[S#]
R3 = PROJECT R2[SNAME]
```

**ผลลัพธ์**

R3	
SNAME	
Smith	
Jones	

**ตัวอย่าง 4.8**

**คำถาม** Get supplier numbers and names for suppliers who supply green part.

**SQL**

```
SELECT S.S#,SNAME
FROM S,SP,P
WHERE S.S# = SP.S#
AND SP.P# = P.P#
AND P.COLOR = 'Green';
```

**ERAO**

```
R1 = SELECT P[COLOR = "Green"]
R2 = JOIN SP R1[P#]
R3 = PROJECT R2[S#]
R4 = JOIN S R3[S#]
R5 = PROJECT R4[S# SNAME]
```

**ผลลัพธ์**

R5	
S#	SNAME
S1	Smith
S2	Jones
S3	Blake
S4	Clark

**ตัวอย่าง 4.9**

คำถาม Get supplier names for suppliers who supply both part P1 and P2.

SQL

```

SELECT SNAME
FROM S
WHERE S# IN
      (SELECT S#
       FROM SP
       GROUP BY S#
       HAVING P# CONTAINS ('P1','P2'));

```

ERAO

```

R1 = SELECT SP[P# = "P1"]
R2 = SELECT SP[P# = "P2"]
R3 = PROJECT R1[S#]
R4 = PROJECT R2[S#]
R5 = INTERSECT R3 R4
R6 = JOIN S R5[S#]
R7 = PROJECT R6[SNAME]

```

ผลลัพธ์

R7

SNAME
Smith
Jones

**ตัวอย่าง 4.10**

**คำถาม** Get supplier names for suppliers who supply at least one part supplied by supplier S2.

**SQL**

```

SELECT SNAME
FROM S
WHERE S# != 'S2'
AND S# IN
    (SELECT S#
     FROM SP
     WHERE P# IN
        (SELECT P#
         FROM SP
         WHERE S# = 'S2'));

```

**ERAO**

```

R1 = SELECT SP[S# = "S2"]
R2 = PROJECT R1[P#]
R3 = JOIN SP R2[P#]
R4 = SELECT R3[S# <> "S2"]
R5 = PROJECT R4[S#]
R6 = UNIQUE R5
R7 = JOIN S R6[S#]
R8 = PROJECT R7[SNAME]

```

**ผลลัพธ์**

R8

SNAME
Smith
Blake
Clark

## ตัวอย่าง 4.11

คำถาม Get supplier names for suppliers who supply at least all parts supplied by supplier S2.

SQL

```

SELECT SNAME
FROM S
WHERE S# != 'S2'
AND S# IN
  (SELECT S#
   FROM SP
   GROUP BY S#
   HAVING P# CONTAINS
     (SELECT P#
      FROM SP
      WHERE S# = 'S2'));

```

ERAO

```

R1 = SELECT SP[S# = "S2"]
R2 = PROJECT R1[P#]
R3 = SELECT SP[S# <> "S2"]
R4 = PROJECT R3[S# P#]
R5 = DIVIDE R4 R2
R6 = JOIN S R5[S#]
R7 = PROJECT R6[SNAME]

```

ผลลัพธ์

R7

SNAME
Smith

## ตัวอย่าง 4.12

คำถาม Get full details of all suppliers who are currently supplying some parts.

SQL

```
SELECT *
FROM S
WHERE S# IN
      (SELECT S#
       FROM SP);
```

ERAO

```
R1 = PROJECT SP[S#]
R2 = UNIQUE R1
R3 = JOIN S R2[S#]
R4 = PROJECT R3[S# SNAME STATUS CITY]
```

ผลลัพธ์

R4

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London



ตัวอย่าง 4.13

คำถาม Get supplier names for suppliers who do not supply part P1.

SQL

```

SELECT DISTINCT SNAME
FROM S,SP
WHERE S.S# = SP.S#
AND S.S# NOT IN
      (SELECT S#
       FROM SP
       WHERE P# = 'P1');

```

ERAO

```

R1 = SELECT SP[P# = "P1"]
R2 = PROJECT R1[S#]
R3 = PROJECT SP[S#]
R4 = UNIQUE R3
R5 = MINUS R4 R2
R6 = JOIN S R5[S#]
R7 = PROJECT R6[SNAME]

```

ผลลัพธ์

R7	
SNAME	
Blake	
Clark	

**ตัวอย่าง 4.14**

**คำถาม** For part supplied, get part numbers and names of all suppliers supplying the part.

**SQL**

```
SELECT P#,SNAME
FROM S,SP
WHERE S.S#=SP.S#
ORDER BY P#;
```

**ERAO**

```
R1 = SEQUENCE SP[P#]
R2 = JOIN R1 S[S#]
R3 = PROJECT R2[P# SNAME]
```

**ผลลัพธ์**

R3

P#	SNAME
P1	Smith
P1	Jones
P2	Smith
P2	Jones
P2	Blake
P2	Clark
P3	Smith
P4	Smith
P4	Clark
P5	Smith
P5	Clark
P6	Smith

ตัวอย่าง 4.15

คำถาม Get supplier names for suppliers who supply exactly the same set of parts supplied by supplier S2.

SQL

```
SELECT SNAME FROM S
WHERE S# != 'S2'
AND S# IN
  (SELECT S#
   FROM SP
   GROUP BY S#
    HAVING P# =
      (SELECT P#
       FROM SP
       WHERE S# = 'S2'));
```

ผลลัพธ์ (SQL)

no row selected

ERAO

```
R1 = SELECT SP[S# = "S2"]
R2 = CNT R1[*]
R3 = PROJECT R1[P#]
R4 = SELECT SP[S# != "S2"]
R5 = PROJECT R4[S# P#]
R6 = DIVIDE R5 R3
R7 = JOIN R6 SP[S#]
R8 = CNT R7[P#/S#]
R9 = DIVIDE R8 R2
R10 = JOIN R9 S[S#]
R11 = PROJECT R10[SNAME]
```

ผลลัพธ์ (ERAO)

Empty relation

## ตัวอย่าง 4.16

คำถาม count the number of parts available.

SQL

```
SELECT COUNT(DISTINCT P#)
FROM SP;
```

ผลลัพธ์ (SQL)

$$\frac{\text{COUNT(DISTINCT P\#)}}{6}$$

ERAO

```
R1 = PROJECT SP[P#]
R2 = UNIQUE R1
R3 = CNT R2[*]
```

ผลลัพธ์ (ERAO)

R3
_CNT
6

## ตัวอย่าง 4.17

คำถาม sum the weight for parts in London.

SQL

```
SELECT SUM(WEIGHT)
FROM P
WHERE CITY = 'London';
```

ผลลัพธ์ (SQL)

$$\frac{\text{SUM(WEIGHT)}}{45}$$

ERAO

```
R1 = SELECT P[CITY = "London"]
R2 = SUM R1[WEIGHT]
```

ผลลัพธ์ (ERAO)

R2
_SUM
45

**ตัวอย่าง 4.18**

คำถาม Get supplier numbers and their total quantities currently being supplied.

SQL

```
SELECT S#,SUM(QTY)
FROM SP
GROUP BY S#;
```

ผลลัพธ์ (SQL)

S#	SUM(QTY)
S1	1300
S2	700
S3	200
S4	900

ERAO

R1 = SUM SP[QTY/S#]

ผลลัพธ์ (ERAO)

R1

S#	_SUM
S1	1300
S2	700
S3	200
S4	900

## ตัวอย่าง 4.19

คำถาม Get part number and their average quantities currently being supplied.

## SQL

```
SELECT P#,AVG(QTY)
FROM SP
GROUP BY P#;
```

## ผลลัพธ์(SQL)

P#	AVG(QTY)
P1	300
P2	250
P3	400
P4	250
P5	250
P6	100

## ERAO

R1 = SEQUENCE SP[P#]

R2 = AVG R1[QTY/P#]

## ผลลัพธ์(ERAO)

R2

P#	_AVG
P1	300
P2	250
P3	400
P4	250
P5	250
P6	100

## 4.2 โครงสร้างระบบของตัวแปลภาษา ERAO

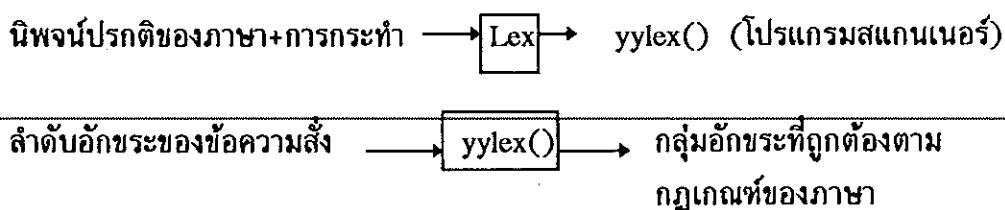
ตัวแปลภาษา ERAO ของงานวิทยานิพนธ์นี้เป็นตัวแปลภาษาแบบ Interpreter ผู้วิจัยได้ดำเนินการตามขั้นตอนทั่วไปเพื่อสร้างตัวแปลภาษาซึ่งมี 3 ขั้นตอนดังนี้

- ขั้นตอนการแยกและวิเคราะห์ศัพท์
- ขั้นตอนการวิเคราะห์วากยสัมพันธ์
- ขั้นตอนการวิเคราะห์ความหมายและตีความ

### ขั้นตอนการแยกและวิเคราะห์ศัพท์

โดยทั่วไปขั้นตอนการวิเคราะห์ศัพท์จะเริ่มจากการทำงานของโปรแกรมที่เรียกว่า สแกนเนอร์ (Scanner) ทำหน้าที่อ่านข้อความสั่งของภาษาทีละอักขระ รวมกลุ่มของอักขระตามรูปแบบที่กำหนดในไวยากรณ์ของภาษาโดยแต่ละกลุ่มของอักขระนี้เรียกว่า โทเคน (Token) และถ้ามีข้อผิดพลาดเกิดขึ้นจะแสดงข้อความบอกข้อผิดพลาดด้วย

ในการสร้างตัวแปลภาษา ERAO นี้ได้ใช้โปรแกรมอรรถประโยชน์ชื่อ Lex ช่วยในการสร้างสแกนเนอร์ โดยข้อมูลเข้าของ Lex จะคือชุดอักขระที่ใช้ คำสงวน สัญลักษณ์ต่าง ๆ และกฎเกณฑ์ของการตั้งชื่อตัวแปรต่าง ๆ ในรูปนิพจน์ปรกติ (Regular Expression) และการกระทำ (Action) ตามรูปแบบที่ใช้ในโปรแกรม Lex รายละเอียดสามารถศึกษาได้จากหนังสือ Compilers Principles, Techniques, and Tools [Alfred V. Aho, Ravi Sethi and Jeffrey D. Ullman (1986)] ในการเรียกใช้สแกนเนอร์จะเรียกผ่านโปรแกรมย่อยชื่อ yylex() รูปที่ 4.1 แสดงข้อมูลเข้า/ออก และผลลัพธ์ที่ได้จากการใช้โปรแกรม Lex และ yylex()



รูปที่ 4.1 ข้อมูลเข้า/ออก และผลลัพธ์ที่ได้จากการใช้โปรแกรม Lex และ yylex()

รูปแบบของข้อมูลเข้าของ Lex ของตัวแปลภาษา ERAO จะเป็นดังนี้คือ

```
%{
/* Include the symbol table */
include "ira_extm.h"
}%
/* Definition for text replacement of regular expression */
alpha      [a-zA-Z_]
alphanum   [a-zA-Z0-9_]
cnum       [1-9]
digit      [0-9]
%%

/* rules part */
\"          return  QUTATION;

">"       return  GT;

"<"       return  LT;

">="     return  GE;

"<="     return  LE;

"="       return  EQ;



---


"<>"     return  NE;



---


"!="      return  NE;

"\n"     return  ENTER;
```



```
{cnum}{digit}*      return CONSTANT;
```

```
{alpha}{alphanum}*  return lookup(IDENT);
```

```
"]"                return RBCK;
```

```
"["                return LBCK;
```

```
"/"                return SLA;
```

```
[\t]+              ;  
.  
return yytext[0];
```

```
%%
```

```
/* user subroutine part */
```

```
static struct keyword {
```

```
    char *name;
```

```
    int  token_yylex;
```

```
} keytable[]={
```

```
    "avg",      AVG,
```

```
    "close",    CLOSE,
```

```
    "cnt",      CNT,
```

```
    "cross",    CROSS,
```

```
    "delete",   DELETE,
```

```
    "display",  DISPLAY,
```

```
    "dispstruc", DISPSTRUC,
```

```
    "divide",   DIVIDE,
```

```
    "function", FUNCTION,
```

```
    "help",     HELP,
```

```
    "intersect", INTERSECT,
```

```
    "join",     JOIN,
```

```
    "keep",     KEEP,
```

```

"listdb",    LISTDB,
"listrel",  LISTREL,
"max",      MAX,
"min",      MIN,
"minus",    MINUS,
"open",     OPEN,
"project",  PROJECT,
"quit",     QUIT,
"select",   SELECT,
"sequence", SEQUENCE,
"std",      STD,
"sum",      SUM,
"union",    UNION,
"unique",   UNIQUE,
"var",      VAR,
    0, 0    );

```

```
/*=====*/
```

```

char *lower(n) char *n;
{
    static char name[20];
    register char *p;
    p = (char*)name;
    while (*n){
        isupper(*n) ? (*p++ = *n++ + 'a' - 'A') : (*p++ = *n++);
    }
    *p = '\0';
    return(char *)name;
}

```

```

lookup(t) int t;
{
    struct keyword *p;
    register int type=0,num=0;
    p = keytable;
    while (p->name)
        if (!strcmp(lower(yytext),p->name))
            return p->token_yylex;
    else
        p++;
    return t; }

```

### ขั้นตอนการวิเคราะห์วากยสัมพันธ์

ขั้นตอนการวิเคราะห์วากยสัมพันธ์เป็นการทำงานของโปรแกรมที่เรียกว่าพาสเซอร์ (Parser) เพื่อใช้วิเคราะห์วากยสัมพันธ์ (Syntax Analysis) ของข้อความสิ่งที่ใช้โดยนำโทเคนที่ได้จากสแกนเนอร์มาตรวจสอบว่ามีการเรียงถูกต้องตามรูปแบบที่กำหนดไว้ในไวยากรณ์ของภาษาหรือไม่ ขั้นตอนการวิเคราะห์ศัพท์จะทำงานร่วมกับขั้นตอนการวิเคราะห์วากยสัมพันธ์ กล่าวคือสแกนเนอร์จะถูกเรียกใช้จากพาสเซอร์ทุกครั้งที่พาสเซอร์ต้องการใช้โทเคนตัวถัดไป

ในการสร้างตัวแปลภาษา ERAO นี้ได้ใช้โปรแกรมมอดปรโยชน์ชื่อ Yacc ช่วยในการสร้างพาสเซอร์ การเรียกใช้พาสเซอร์จะเรียกผ่านโปรแกรมย่อยชื่อ yyparse() โดยข้อมูลเข้าของ Yacc เป็นไวยากรณ์ไม่พึ่งบริบท (Context Free Grammar) และลักษณะเฉพาะ (Attribute) รวมกันเรียกว่า ไวยากรณ์กรณีลักษณะเฉพาะ (Attributed Grammar) ไวยากรณ์ไม่พึ่งบริบทของแต่ละข้อความสั่ง ซึ่งใช้ในโปรแกรม Yacc นี้จะเขียนอยู่ในรูปกฎ (Rule) แต่ละกฎประกอบด้วย 2 ส่วน แยกทั้งสองส่วนด้วยเครื่องหมาย ":" ส่วนแรกหน้าเครื่องหมาย ":" คือสัญลักษณ์ไม่สิ้นสุด (Nonterminal Symbol) ส่วนหลังเครื่องหมาย ":" คือสัญลักษณ์ไม่สิ้นสุดหรือสัญลักษณ์สิ้นสุด (Terminal Symbol) หรือสายอักขระว่าง (Empty String) รายละเอียดสามารถศึกษาได้จากหนังสือ Compilers Principles, Techniques, and Tools [Alfred V. Aho, Ravi Sethi and Jeffrey D. Ullman (1986)]

รูปที่ 4.2 แสดงให้เห็นข้อมูลเข้า/ออก และผลลัพธ์ที่ได้จากการใช้โปรแกรม Yacc และ yyparse()

ไวยากรณ์ไม่พึงบริบท+ลักษณะเฉพาะ → **Yacc** → yyparse() (โปรแกรมพาชเชอร์)

ชุดข้อความสั่งที่ต้องการวิเคราะห์วากยสัมพันธ์ → **yyparse()** → ยอมรับ หรือปฏิเสธ

รูปที่ 4.2 ข้อมูลเข้า/ออก และผลลัพธ์ที่ได้จากการใช้โปรแกรม Yacc และ yyparse()

รูปแบบของข้อมูลเข้าของ Yacc ของตัวแปลภาษา ERAO จะเป็นดังนี้คือ

```
%{
/* Include the symbol */
#include "ira_extm.h"

#define token(x) x
%}

/* The definition of a terminal symbol name or keyword */
%token AVG CLOSE CONSTANT CNT CROSS DELETE
%token DISPLAY DISPSTRUC DIVIDE ENTER HELP FUNCTION
%token IDENT INTERSECT KEEP JOIN LISTREL LISTDB MAX
%token MIN MINUS OPEN PROJECT QUIT SELECT SEQUENCE
%token STD SLA SUM UNION UNIQUE VAR

/* operator token */
%token EQ GT LT GE LE NE LBCK RBCK QUTATION

%%
```

/\* rules part \*/

```

Line      :
           | Line LineCommand set010;
LineCommand : enter
           | QueryCommand
           | SetCommand
           | error enter set020;
QueryCommand : set040 set050 ResultName EQ
              QueryStmt set042;
QueryStmt   : S_Operation
           | R_Operation
           | F_Operation;
S_Operation : ProjectStmt
           | SelectStmt
           | JoinStmt
           | DivideStmt
           | SequenceStmt
           | UniqueStmt;
ProjectStmt : PROJECT set055 RelName set056
            LBCK AttList RBCK enter set065;
AttList     : set060 AttName set061 set062
            AttributeList;
AttributeList :
           | AttList;
SelectStmt   : SELECT set055 RelName set056
            LBCK SelectExpr RBCK enter set081;
SelectExpr   : set060 AttName set061 set077 Option set078
            set079 Value;
JoinStmt     : JOIN set055 RelName set056
            set057 RelName set058 LBCK
            JoinOp RBCK enter
            set098;
JoinOp      : set060 AttName set061 JoinOpAgain;

```

JoinOpAgain : set092  
| set077 Option set094 AttName set095;  
DivideStmt : DIVIDE set055 RelName set056 set057  
RelName set058 enter set100;  
SequenceStmt : SEQUENCE set055 RelName set056 LBCK  
AttList RBCK enter set110;  
UniqueStmt : UNIQUE set055 RelName set056 enter  
set105;  
R\_Operation : INTERSECT R\_Get\_Rname set150  
| UNION R\_Get\_Rname set152  
| MINUS R\_Get\_Rname set154  
| CROSS R\_Get\_Rname set156;  
R\_Get\_Rname : set055 RelName set056 set057  
RelName set058  
F\_Operation : set290 CaseF\_Op set055 RelName  
set056 F\_Parameter;  
CaseF\_Op : MAX set300  
| MIN set301  
| SUM set302  
| AVG set303  
| VAR set304  
| STD set305  
| CNT set306;  
F\_Parameter : LBCK set210 AttName set095 RBCK set200  
| LBCK set210 AttName set095 SLA AttList  
RBCK set215;

---

```
SetCommand : QUIT set320 enter
| DISPLAY DisplayOp set322 enter
| DISPSTRUC set055 RelName set056
  set324 enter
| HELP HelpCase enter
| OPEN DbName set328 enter
| CLOSE DbName set330 enter
| LISTREL enter set332
| LISTDB enter set334
| KEEP set055 RelName set056
  enter set336
| DELETE set055 RelName set056
  enter set338;
```

```
DisplayOp : set055 RelName set056;
```

```
HelpCase : set326
| HelpCommand set327;
```

```
HelpCommand : QUIT
| DISPLAY
| DISPSTRUC
| HELP
| OPEN
| CLOSE
| LISTREL
| LISTDB
| KEEP
| DELETE
```

---

| INTERSECT  
 | UNION  
 | MINUS  
 | CROSS  
 | SELECT  
 | PROJECT  
 | JOIN  
 | DIVIDE  
 | SEQUENCE  
 | UNIQUE  
 | FUNCTION  
 | error;  
 DbName : IDENT;  
 RelName : IDENT;  
 AttName : IDENT;  
 Value : IDENT set080  
 | QUTATION IDENT set080 QUTATION  
 | CONSTANT set080  
 | error;  
 ResultName : IDENT set053;  
 enter : ENTER;  
 Option : EQ  
 | NE  
 | GT  
 | GE  


---

 | LT  
 | LE;



### ขั้นตอนการวิเคราะห์ความหมายและตีความ

ขั้นตอนการวิเคราะห์ความหมายและตีความเป็นส่วนที่ผู้วิจัยเขียนขึ้นโดยใช้ภาษาซี จะแทรกอยู่ในส่วนข้อมูลเข้าของ Yacc ดังนั้นขั้นตอนนี้จะกระทำไปพร้อม ๆ กับขั้นตอน วิเคราะห์วากยสัมพันธ์ ในขณะที่โปรแกรมพาสเซอร์ดำเนินการตรวจสอบการจัดเรียงของ โทกเคนซึ่งได้จากสแกนเนอร์ พาสเซอร์จะดำเนินการขั้นตอนการวิเคราะห์ความหมายและ ตีความไปพร้อม ๆ กันเพื่อจัดเตรียมค่าต่าง ๆ และดำเนินการตามคำสั่งปฏิบัติการที่วิเคราะห์ งานวิทยานิพนธ์นี้ผู้วิจัยได้ใช้สัญลักษณ์แอกชั่น (Action Symbol) อยู่ในรูป setddd (เมื่อ ddd เป็นเลขจำนวนเต็ม) แทนการดำเนินงานต่าง ๆ ในรูปแอกชั่นรูทีน (Action Routine) โดยสัญลักษณ์แอกชั่น และแอกชั่นรูทีนทั้งหมดในตัวแปลภาษา ERAO มีรายละเอียดดังต่อไปนี้

#### สัญลักษณ์แอกชั่น

#### การดำเนินงาน

set010	แสดงข้อความเพื่อแสดงสถานะพร้อมรับคำสั่งปฏิบัติการ และเซตตัว กำหนดซึ่งระบุความผิดพลาดให้เท่ากับ 0
set020	เมื่อเกิดความผิดพลาดขึ้นในขั้นตอนการวิเคราะห์วากยสัมพันธ์จะ ดำเนินการเพื่อให้ตัวแปลภาษา ERAO สามารถดำเนินงานต่อได้ พร้อมทั้งสกัดโทเคนหลังจากเกิดความผิดพลาดทั้งจนกระทั่งพบ โทกเคนที่เป็นสัญลักษณ์สิ้นสุดของคำสั่งปฏิบัติการ
set040	กำหนดค่าเริ่มต้น Headatt ซึ่งเป็นตัวชี้ที่ใช้ชี้โหนดเริ่มต้นของรายการ โยงแอดทริบิวให้ชี้ไปสู่ว่างเปล่า (Null)
set042	คืนหน่วยความจำที่ใช้สำหรับรายการโยงแอดทริบิวให้กับระบบซึ่งมี Headatt เป็นตัวชี้โหนดเริ่มต้นของรายการโยง
set050	กำหนดค่าเริ่มต้นเป็นอักษรว่างเปล่าให้กับตัวแปรซึ่งเตรียมไว้สำหรับ เก็บชื่อตารางข้อมูลผลลัพธ์
set053	สำเนาโทกเคนตัวล่าสุดให้กับตัวแปรชื่อตารางข้อมูลผลลัพธ์
set055	กำหนดค่าเริ่มต้นเป็นอักษรว่างเปล่าให้กับตัวแปรซึ่งเตรียมไว้สำหรับ เก็บชื่อตารางข้อมูลชั่วคราวที่ 1
set056	สำเนาโทกเคนตัวล่าสุดให้กับตัวแปรชื่อตารางข้อมูลชั่วคราวที่ 1

## สัญลักษณ์แยกชั้น

## การดำเนินงาน

- set057 กำหนดค่าเริ่มต้นเป็นอักขระว่างเปล่าให้กับตัวแปรซึ่งเตรียมไว้สำหรับเก็บชื่อตารางข้อมูลชั่วคราวที่ 2
- set058 สำเนาโทกเคนตัวล่าสุดให้กับตัวแปรชื่อตารางข้อมูลชั่วคราวที่ 2
- set060 กำหนดค่าเริ่มต้นเป็นอักขระว่างเปล่าให้กับตัวแปรซึ่งเตรียมไว้สำหรับเก็บชื่อแอตทริบิวชั่วคราวที่ 1
- set061 สำเนาโทกเคนตัวล่าสุดให้กับตัวแปรชื่อแอตทริบิวชั่วคราวที่ 1
- set062 เพิ่มชื่อแอตทริบิวชั่วคราวที่ 1 ในรายการโยงแอตทริบิวซึ่งมี Headatt เป็นตัวชี้โหนดเริ่มต้น
- set065 กรณีที่มีการเรียกใช้ฐานข้อมูลเรียบร้อยแล้วจะดำเนินการคำสั่งปฏิบัติการ Project
- set077 กำหนดค่าเริ่มต้นเป็นอักขระว่างเปล่าให้กับตัวแปรซึ่งเตรียมไว้สำหรับโอเปอเรเตอร์เชิงเปรียบเทียบ
- set078 สำเนาโทกเคนตัวล่าสุดให้กับตัวแปรโอเปอเรเตอร์เชิงเปรียบเทียบ
- set079 กำหนดค่าเริ่มต้นเป็นอักขระว่างเปล่าให้กับตัวแปรซึ่งเตรียมไว้สำหรับเก็บค่าที่ใช้เป็นเกณฑ์ในการเลือกแถวข้อมูลในคำสั่งปฏิบัติการ Select
- set080 สำเนาโทกเคนตัวล่าสุดให้กับตัวแปรที่เก็บค่าที่ใช้เป็นเกณฑ์ในการเลือกแถวข้อมูลในคำสั่งปฏิบัติการ Select
- set081 กรณีที่มีการเรียกใช้ฐานข้อมูลเรียบร้อยแล้วจะดำเนินการคำสั่งปฏิบัติการ Select
- set092
- กำหนดค่าเริ่มต้นเป็นอักขระว่างเปล่าให้กับตัวแปรซึ่งเตรียมไว้สำหรับเก็บชื่อแอตทริบิวชั่วคราวที่ 2
  - สำเนาค่าในตัวแปรชื่อแอตทริบิวชั่วคราวที่ 1 ให้กับตัวแปรชื่อแอตทริบิวชั่วคราวที่ 2
- 
- กำหนดค่าเริ่มต้นเป็นอักขระว่างเปล่าให้กับตัวแปรซึ่งเตรียมไว้สำหรับโอเปอเรเตอร์เชิงเปรียบเทียบ
  - กำหนดให้ตัวแปรโอเปอเรเตอร์เชิงเปรียบเทียบมีค่าเท่ากับ "="
- set094
- สำเนาโทกเคนตัวล่าสุดให้กับตัวแปรโอเปอเรเตอร์เชิงเปรียบเทียบ
  - กำหนดค่าเริ่มต้นเป็นอักขระว่างเปล่าให้กับตัวแปรซึ่งเตรียมไว้สำหรับเก็บชื่อแอตทริบิวชั่วคราวที่ 2
- set095 สำเนาโทกเคนตัวล่าสุดให้กับตัวแปรชื่อแอตทริบิวชั่วคราวที่ 2
- set098 กรณีที่มีการเรียกใช้ฐานข้อมูลเรียบร้อยแล้วจะดำเนินการคำสั่งปฏิบัติการ Join

## สัญลักษณ์แอกชั่น

## การดำเนินงาน

set100	กรณีที่มีการเรียกใช้ฐานข้อมูลเรียงร้อยจะดำเนินการคำสั่งปฏิบัติการ Divide
set105	กรณีที่มีการเรียกใช้ฐานข้อมูลเรียงร้อยจะดำเนินการคำสั่งปฏิบัติการ Unique
set110	กรณีที่มีการเรียกใช้ฐานข้อมูลเรียงร้อยจะดำเนินการคำสั่งปฏิบัติการ Sequence
set150	กรณีที่มีการเรียกใช้ฐานข้อมูลเรียงร้อยจะดำเนินการคำสั่งปฏิบัติการ Intersection
set152	กรณีที่มีการเรียกใช้ฐานข้อมูลเรียงร้อยจะดำเนินการคำสั่งปฏิบัติการ Union
set154	กรณีที่มีการเรียกใช้ฐานข้อมูลเรียงร้อยจะดำเนินการคำสั่งปฏิบัติการ Minus
set156	กรณีที่มีการเรียกใช้ฐานข้อมูลเรียงร้อยจะดำเนินการคำสั่งปฏิบัติการ Cross
set200	กรณีที่มีการเรียกใช้ฐานข้อมูลเรียงร้อยจะดำเนินการคำสั่งปฏิบัติการ ในรูปฟังก์ชันจากทุกแถวของตารางข้อมูล
set210	กำหนดค่าเริ่มต้นเป็นอักขระว่างเปล่าให้กับตัวแปรซึ่งเตรียมไว้ สำหรับเก็บชื่อแอตทริบิวชั่วคราวที่ 2
set215	กรณีที่มีการเรียกใช้ฐานข้อมูลเรียงร้อยจะดำเนินการคำสั่งปฏิบัติการ ในรูปฟังก์ชันสำหรับแต่ละกลุ่มย่อยของแถวของตารางข้อมูล
set290	กำหนดให้ฟังก์ชันที่ต้องการคำนวณเท่ากับ 0
set300	กำหนดให้ฟังก์ชันที่ต้องการคำนวณเท่ากับ MAX
set301	กำหนดให้ฟังก์ชันที่ต้องการคำนวณเท่ากับ MIN
set302	กำหนดให้ฟังก์ชันที่ต้องการคำนวณเท่ากับ SUM
set303	กำหนดให้ฟังก์ชันที่ต้องการคำนวณเท่ากับ AVG
set304	กำหนดให้ฟังก์ชันที่ต้องการคำนวณเท่ากับ VAR
set305	กำหนดให้ฟังก์ชันที่ต้องการคำนวณเท่ากับ STD
set306	กำหนดให้ฟังก์ชันที่ต้องการคำนวณเท่ากับ CNT
set320	ดำเนินการคำสั่ง Quit
set322	ดำเนินการคำสั่ง Display
set324	ดำเนินการคำสั่ง DispStruc
set326	ดำเนินการคำสั่ง Help

**สัญลักษณ์แอกชั่น****การดำเนินงาน**

set327	ดำเนินการคำสั่ง HelpCommand ซึ่งเป็นคำสั่งแสดงคำแนะนำเพียงบางส่วนที่ระบุ
set328	ดำเนินการคำสั่ง Quit
set330	ดำเนินการคำสั่ง Close
set332	ดำเนินการคำสั่ง ListRel
set334	ดำเนินการคำสั่ง ListDb

**4.3 การตรวจสอบความผิดพลาด**

การตรวจสอบความผิดพลาดของตัวแปลภาษา ERAO เกิดขึ้นได้ 2 รูปแบบคือ ความผิดพลาดในการตรวจสอบรูปแบบของภาษา และความผิดพลาดในขณะตรวจสอบความหมาย

**ความผิดพลาดในการตรวจสอบรูปแบบของภาษา (Syntax Error)**

ในการดำเนินงานกับแต่ละข้อความสั่งเมื่อสแกนเนอร์อ่านแต่ละอักขระมาเรียงกันเป็นโทกเคนจะมีการตรวจสอบรูปแบบของโทกเคนว่าถูกต้องตามไวยากรณ์ที่กำหนดไว้หรือไม่ กรณีที่โทกเคนนั้นถูกต้องตรงกับรูปแบบของไวยากรณ์ สแกนเนอร์จะส่งโทกเคนนั้นให้กับพาสเซอร์ดำเนินการต่อไป กรณีที่โทกเคนนั้นไม่ถูกต้องตามรูปแบบของไวยากรณ์จะมีการแสดงข้อความบอกความผิดพลาด ความผิดพลาดในการตรวจสอบรูปแบบของภาษาอาจเกิดขึ้นเมื่อพาสเซอร์นำโทกเคนมาเรียงกัน แล้วพบว่าไม่ถูกต้องตามรูปแบบที่กำหนดไว้ในไวยากรณ์ของภาษา ตัวอย่างความผิดพลาดในรูปแบบนี้เช่น

```
R1 = selcct s[SNUM = S1]
```

---

จากตัวอย่างข้างต้นจะพบว่ามีการพิมพ์คำว่า "selcct" ผิด ที่ถูกต้องคือ "select"

ดังนั้นจะมีข้อความแสดงให้เห็นความผิดพลาดคือ

---

```
[Error] near "selcct" : syntax error
```

### ความผิดพลาดในขณะที่ตรวจสอบความหมาย (Semantic Error)

เมื่อพาสเซอร์ได้ตรวจสอบการเรียงโทกเคนถูกต้องตามรูปแบบของไวยากรณ์จะดำเนินการตามคำสั่งปฏิบัติการที่วิเคราะห์ได้ ในขั้นตอนการดำเนินงานนี้ต้องเข้าถึงข้อมูล แต่ข้อมูลเหล่านั้นอาจไม่ถูกต้องตามข้อกำหนด หรือข้อมูลนั้นไม่มี จึงไม่สามารถดำเนินการกับคำสั่งปฏิบัติการได้ ดังนั้นได้ออกแบบการแสดงความผิดพลาดในรูปแบบนี้โดยแสดงข้อความเพื่อแสดงสาเหตุที่เกิดขึ้น ข้อความแสดงความผิดพลาดนี้ได้รวบรวมดังตารางที่ 4.1 ตัวอย่างความผิดพลาดในขณะที่ตรวจสอบความหมายเช่น

```
R2 = select s[PNUM=S1]
```

จากตัวอย่างในกรณีนี้ตารางที่มีชื่อว่า "s" ไม่มีแอตทริบิวต์ชื่อ "PNUM" จะมีข้อความแสดงให้เห็นความผิดพลาดคือ

Error : Attribute does not exist

ข้อความ	สาเหตุ
Error : Attribute domain not compatible	แอตทริบิวต์ 2 แอตทริบิวต์ที่นำมาพิจารณาพบว่ามีโดเมนต่างชนิดกัน
Error : Attribute length not compatible	แอตทริบิวต์ 2 แอตทริบิวต์ที่นำมาพิจารณาพบว่ามีควมยาวไม่เท่ากัน
Error : Attribute status not compatible	แอตทริบิวต์ 2 แอตทริบิวต์ที่นำมาพิจารณาพบว่ามีสถานะต่างกัน
Error : Data type not compatible	ไม่สามารถดำเนินการได้เนื่องจากมีความผิดพลาดเกี่ยวกับแอตทริบิวต์อย่างใดอย่างหนึ่ง
Error : Database not open	ไม่มีฐานข้อมูลใดถูกเรียกใช้
Error : Do not binary relation	จำนวนแอตทริบิวต์ของตารางข้อมูลไม่เท่ากับ 2 (ไม่เป็น Binary relations)
Error : Do not unary relation	จำนวนแอตทริบิวต์ของตารางข้อมูลไม่เท่ากับ 1 (ไม่เป็น Unary relations)
Error : Duplicate relation's name	ตารางข้อมูล 2 ตารางซึ่งจะนำมาปฏิบัติการเป็นตารางข้อมูลเดียวกัน

ตารางที่ 4.1 ข้อความแสดงความผิดพลาดในขณะที่ตรวจสอบความหมาย

ข้อความ	ความหมาย
Error : Key type not compatible	สถานะของแอตทริบิวต์ที่นำมาพิจารณาไม่ สถานะไม่ถูกต้อง
Error : Opcode join condition error	โอเปอเรเตอร์เชิงเปรียบเทียบสำหรับการ join ไม่ถูกต้อง
Error : Relation already exist	ตารางข้อมูลนี้มีอยู่ในฐานข้อมูลเรียบร้อยแล้ว
Error : Relation's name found in database	พบตารางข้อมูลนี้ในฐานข้อมูล
Error : Relation's name not found	ไม่พบตารางข้อมูลที่ต้องการในฐานข้อมูล
Error : Structure of two relations are not compatible	จำนวนแอตทริบิวต์ในตารางข้อมูลที่นำมา พิจารณาของ 2 ตารางมีจำนวนไม่เท่ากัน
Error : The second relation is null	ตารางข้อมูลที่ 2 ที่นำมาดำเนินการเป็นตาราง ข้อมูลที่ไม่มีข้อมูล
Error : Attribute does not exist	ไม่พบแอตทริบิวต์ที่ต้องการในตารางข้อมูล
Error : You can not create a relation which already exist	ไม่สามารถสร้างตารางข้อมูลที่มีอยู่แล้ว
Error : Database SSS open already	ฐานข้อมูลชื่อ SSS ถูกเรียกใช้เรียบร้อยแล้ว
Error : Database SSS does not open	ฐานข้อมูล SSS ยังไม่ได้ถูกเรียกใช้
Error : Attribute's name not found	ไม่พบชื่อแอตทริบิวต์ที่ต้องการ
Error : No case function	ไม่มีการดำเนินการสำหรับฟังก์ชันนี้
Error : Wrong Data Type	ไม่สามารถนำชนิดของแอตทริบิวต์ไปดำเนินการ ฟังก์ชันได้
Error : Open more than one database	มีการเรียกใช้ฐานข้อมูลมากกว่าหนึ่ง
Error : Attribute does not exist. Attribute-->XXX	ไม่พบแอตทริบิวต์ XXX ในตารางข้อมูลที่กำลัง ดำเนินการ
Error : Attribute does not exist. Attribute-->XXX on Relation-->YYY	ไม่พบแอตทริบิวต์ XXX ในตารางข้อมูล YYY
Error : Relation's name not found. Relation-->XXX	ไม่พบตารางข้อมูล XXX ในฐานข้อมูลที่กำลัง เรียกใช้
Error : Relation already exist. You can not create a relation which already exist	ตารางข้อมูลผลลัพธ์ที่ต้องการมีอยู่แล้วใน ฐานข้อมูลที่กำลังเรียกใช้

ตารางที่ 4.1 ข้อความแสดงความผิดพลาดในขณะที่ตรวจสอบความหมาย (ต่อ)

#### 4.4 การดำเนินงานของทั้งระบบ

การดำเนินงานของระบบได้นำโปรแกรมที่ถูกพัฒนาไว้บนเครื่องคอมพิวเตอร์ VAX 11/785 เรียบร้อยแล้วของคุณเพียรพร [เพียรพร (2534)] มาปรับเปลี่ยนเพื่อให้ทำงานได้บนเครื่องคอมพิวเตอร์ SUN SPARC IPX และได้ตัดส่วนของการสอบถามข้อมูลออก ทั้งนี้เนื่องจากโปรแกรมในส่วนการสอบถามนี้จะต้องปรับเปลี่ยนรายละเอียดต่างๆ ค่อนข้างมาก และเพื่อนำการดำเนินงานของคำสั่งปฏิบัติการ ERAO จากงานวิจัยนี้มาดำเนินงานแทน รายละเอียดการดำเนินงานของระบบแทนด้วยแผนภาพกระแสข้อมูล (DFD : Data Flow Diagram) ซึ่งมีกฎเกณฑ์การใช้แสดงดังภาคผนวก ก.

#### แผนภาพกระแสข้อมูล

การดำเนินงานของระบบเรียกใช้โดยพิมพ์ "Mrdbms" เมื่อการดำเนินงานของระบบถูกเรียกใช้จะมีตัวแปรกลุ่มหนึ่งที่ถูกกำหนดให้รู้จักตลอดการดำเนินงาน ตัวแปรเหล่านี้คือ

```

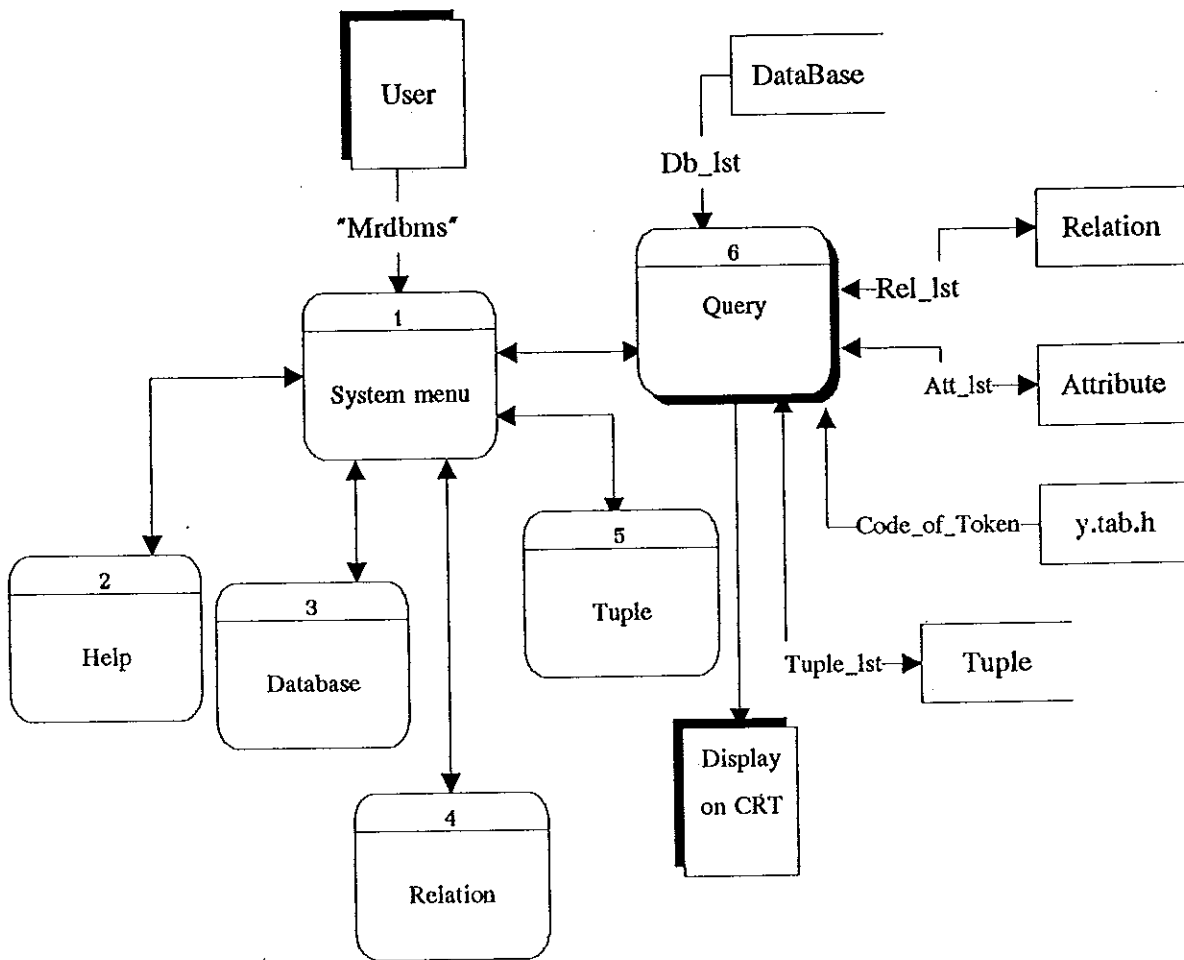
int    USEDDB = OFF;      /* flag for check database is used */
int    USEREL = OFF;     /* flag for check relation is used */
int    NoFun;            /* flag for check type of function */
int    Err;              /* flag for check error command */
char   Gdate[9];        /* date */
DBNODE *usedb;          /* Pointer to database that used */
DBNODE *Headdb;        /* Database pointer point to header */
RELNODE *userel;       /* Pointer to relation that used */
ATTLIST *Headatt;     /* Pointer to head temporary att list */
ATTLIST *useatt;      /* Pointer to temporary att list */

```

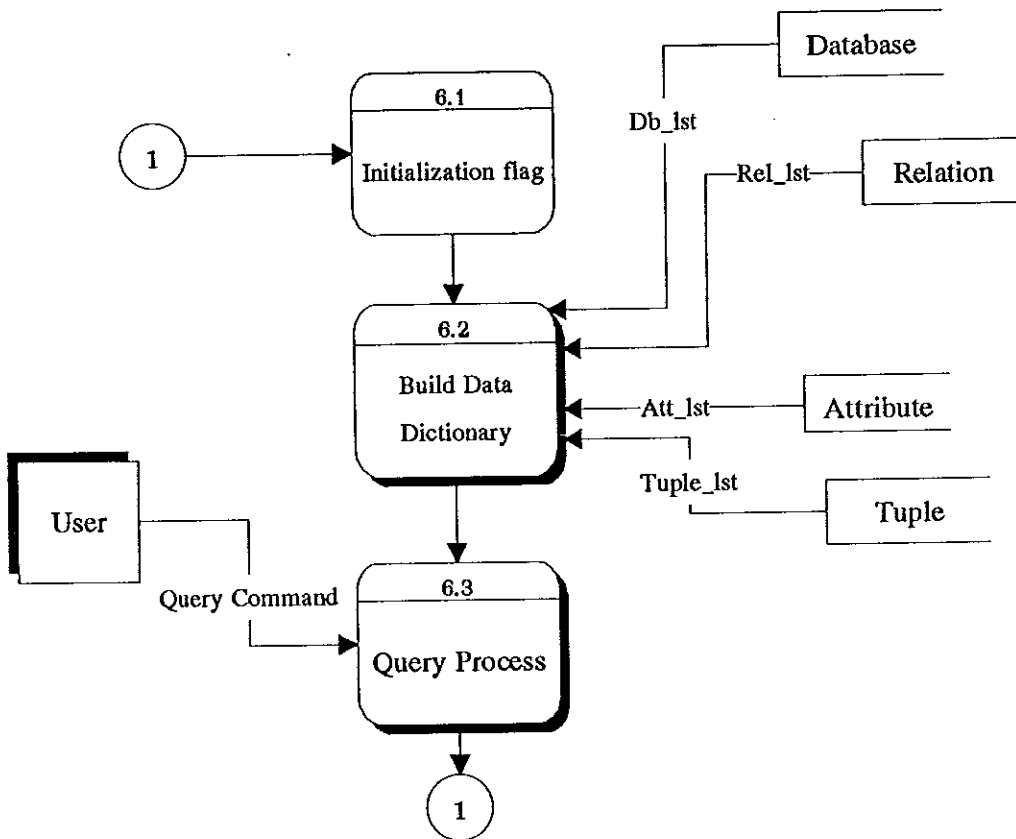
ข้อมูลต่าง ๆ ที่มีการส่งผ่านภายในแผนภาพกระแสข้อมูลมีรายละเอียดดังนี้

“Mrdbms”	เป็นคำสั่งเพื่อเรียกใช้การดำเนินการของระบบ Mrdbms
Db_lst	รายละเอียดข้อมูลของฐานข้อมูล จากแฟ้มข้อมูล system.dbs
Rel_lst	รายละเอียดข้อมูลของตารางข้อมูล จากแฟ้มข้อมูล system.rel
Att_lst	รายละเอียดข้อมูลของแอตทริบิว จากแฟ้มข้อมูล <relnameN>.att โดยที่ N=1,2,3,...
Tuple_lst	รายละเอียดข้อมูลของข้อมูลจริง ๆ จากแฟ้มข้อมูล <relnameN> โดยที่ N=1,2,3,...
Code_of_Token	รหัสของโทเคน จากแฟ้มข้อมูล y.tab.h
Query Command	ข้อความสั่งการดำเนินงานในรูปแบบคำสั่ง
DbName	ตัวแปรที่ใช้แทนชื่อของฐานข้อมูลซึ่งรับจากแถวคำสั่งปฏิบัติการ
RelName	ตัวแปรที่ใช้แทนชื่อของตารางข้อมูลซึ่งรับจากแถวคำสั่งปฏิบัติการ

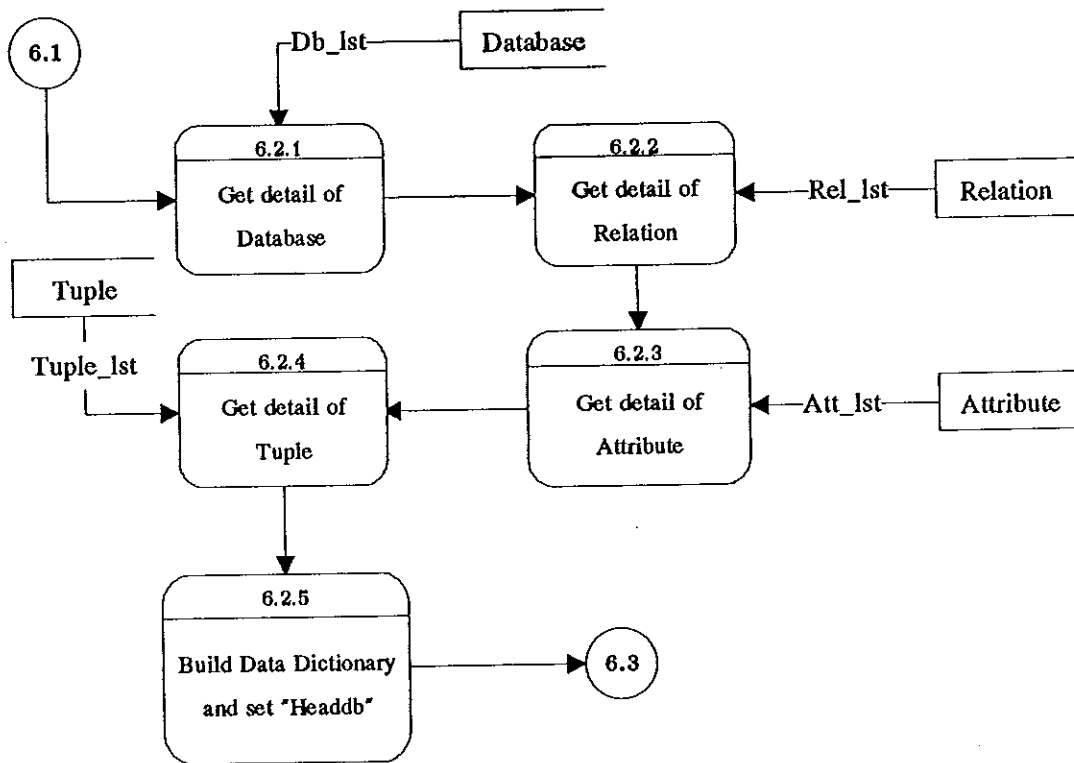




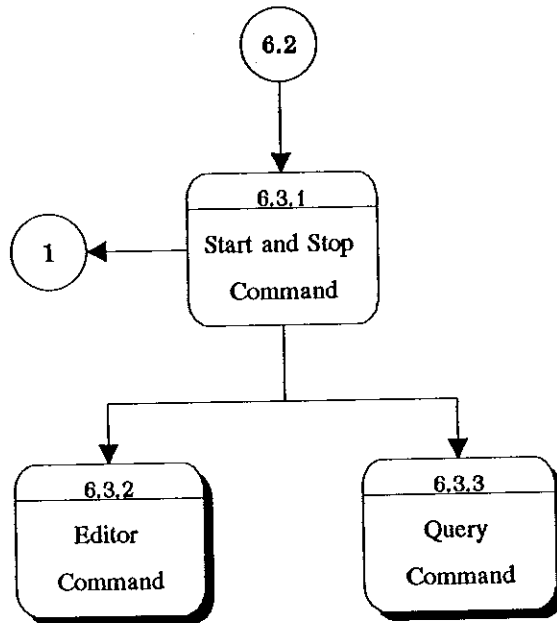
รูปที่ 4.3 แผนภาพกระแสข้อมูลของระบบ Mrdbms



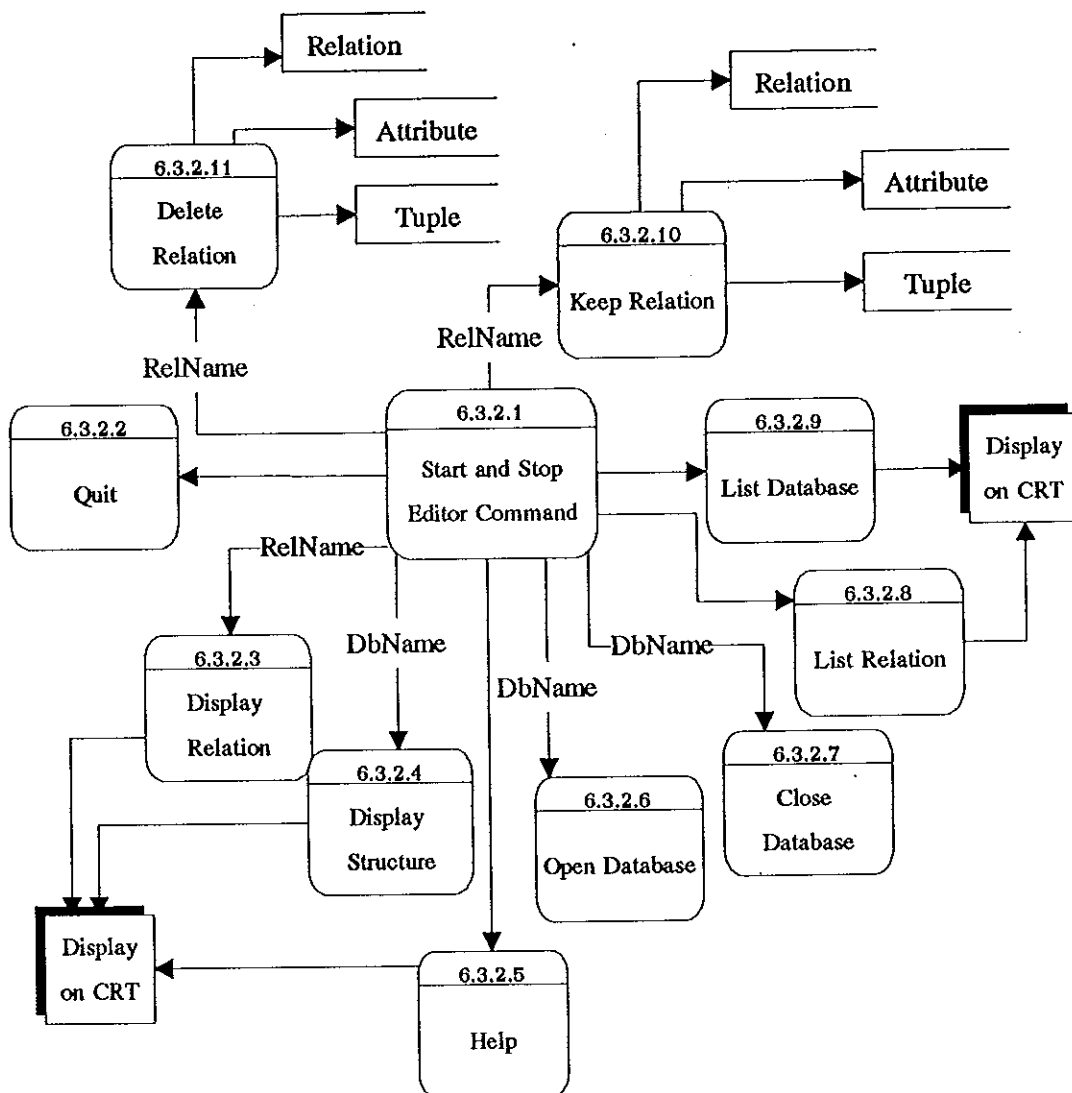
รูปที่ 4.4 แผนภาพกระแสข้อมูลของกระบวนการ Query



รูปที่ 4.5 แผนภาพกระแสข้อมูลของกระบวนการ Build Data Dictionary



รูปที่ 4.6 แผนภาพกระแสข้อมูลของกระบวนการ Query Process



รูปที่ 4.7 แผนภาพกระแสข้อมูลของกระบวนการ Editor Command

### คำอธิบายการดำเนินงานแผนภาพกระแสข้อมูล

#### จากรูปที่ 4.3

- กระบวนการที่ 1 เป็นการดำเนินงานในส่วนการแสดง และควบคุมการเลือกรายการหลักของระบบงานเดิม เป็นการเลือกรายการจากเมนูในลักษณะ Pull Down Menu
- กระบวนการที่ 2-5 เป็นการดำเนินงานในส่วนต่างๆ ของระบบงานเดิม ผู้สนใจสามารถดูรายละเอียดเพิ่มเติมได้จากงานวิทยานิพนธ์ที่ได้ดำเนินการมาก่อนหน้านี้ [เพียรพร (2534)]
- กระบวนการที่ 6 เป็นส่วนการดำเนินงานเรียกใช้ส่วนการสอบถาม

#### จากรูปที่ 4.4

##### กระบวนการที่ 6.1

ชื่อกระบวนการ Initialization flag

คำอธิบาย เป็นการกำหนดค่าเริ่มต้นให้กับตัวแปร

##### กระบวนการที่ 6.2

ชื่อกระบวนการ Build Data Dictionary

คำอธิบาย เป็นการสร้างปทานุกรมข้อมูลก่อนการดำเนินงานส่วนการสอบถาม

##### กระบวนการที่ 6.3

ชื่อกระบวนการ Query Process

คำอธิบาย เป็นกระบวนการดำเนินงานส่วนการสอบถาม

#### จากรูปที่ 4.5

##### กระบวนการที่ 6.2.1

ชื่อกระบวนการ Get detail of Database

คำอธิบาย เป็นการนำชื่อและรายละเอียดเกี่ยวกับฐานข้อมูลทั้งหมดที่อยู่ในแฟ้มข้อมูล system.dbs และ Dbn โดย n=1,2,3,... มาเก็บในความจำหลัก

##### กระบวนการที่ 6.2.2

ชื่อกระบวนการ Get detail of Relation

คำอธิบาย เป็นการนำชื่อและรายละเอียดเกี่ยวกับตารางข้อมูลทั้งหมดที่อยู่ในแฟ้มข้อมูล system.rel มาเก็บในความจำหลัก

### กระบวนการที่ 6.2.3

ชื่อกระบวนการ Get detail of Attribute

คำอธิบาย เป็นการนำชื่อและรายละเอียดเกี่ยวกับแอตทริบิวต์ทั้งหมดที่อยู่ในแฟ้มข้อมูล  
reNameN.att โดย N=1,2,3,... มาเก็บในความจำหลัก

### กระบวนการที่ 6.2.4

ชื่อกระบวนการ Get detail of Tuple

คำอธิบาย เป็นการนำรายละเอียดของข้อมูลจริงของแต่ละตารางข้อมูลที่อยู่ในแฟ้ม  
ข้อมูล reNameN โดย N=1,2,3,... มาเก็บในความจำหลัก

### กระบวนการที่ 6.2.5

ชื่อกระบวนการ Build Data Dictionary and set "Headdb"

คำอธิบาย เป็นการนำข้อมูลที่ได้จากกระบวนการที่ 6.2.1 ถึง 6.2.4 มาสร้างปทานุกรม  
ข้อมูลในหน่วยความจำหลักพร้อมกำหนดตัวชี้ Headdb ชี้ไปยังจุดเริ่มต้น  
ของปทานุกรมข้อมูลนั้น

## จากรูปที่ 4.6

### กระบวนการที่ 6.3.1

ชื่อกระบวนการ Start and Stop Command

คำอธิบาย เป็นการควบคุมการเริ่มต้นและสิ้นสุดการดำเนินงานในแต่ละกระบวนการ

### กระบวนการที่ 6.3.2

ชื่อกระบวนการ Editor Command

คำอธิบาย เป็นการดำเนินงานเพื่ออำนวยความสะดวกต่าง ๆ ในการใช้คำสั่งปฏิบัติการ  
ERAO

### กระบวนการที่ 6.3.3

ชื่อกระบวนการ Query Command

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการ ERAO คำสั่งปฏิบัติการเหล่านี้จะทำให้  
ให้ได้ตารางข้อมูลผลลัพธ์

## จากรูปที่ 4.7

กระบวนการที่ 6.3.2.1

ชื่อกระบวนการ Start and Stop Editor Command

คำอธิบาย เป็นการควบคุมกระบวนการประมวลผลสำหรับคำสั่งเพื่ออำนวยความสะดวกในการใช้คำสั่งปฏิบัติการ ERAO

กระบวนการที่ 6.3.2.2

ชื่อกระบวนการ Quit

คำอธิบาย เป็นการยกเลิกการทำงานในส่วนคำสั่งปฏิบัติการ ERAO ออกไปสู่ระบบงานเดิม

กระบวนการที่ 6.3.2.3

ชื่อกระบวนการ Display Relation

คำอธิบาย เป็นการดำเนินงานแสดงรายละเอียดข้อมูลจริงของตารางข้อมูลที่ใช้ระบุ

กระบวนการที่ 6.3.2.4

ชื่อกระบวนการ Display Structure

คำอธิบาย เป็นการดำเนินงานแสดงโครงสร้างข้อมูลของตารางข้อมูลที่ใช้ระบุ

กระบวนการที่ 6.3.2.5

ชื่อกระบวนการ Help

คำอธิบาย เป็นกระบวนการให้ความช่วยเหลือ หรือคำแนะนำในการใช้ และดูรูปแบบคำสั่งปฏิบัติการ ERAO

กระบวนการที่ 6.3.2.6

ชื่อกระบวนการ Open Database

คำอธิบาย เป็นการเรียกใช้ฐานข้อมูลตามที่ผู้ใช้ระบุ

กระบวนการที่ 6.3.2.7

ชื่อกระบวนการ Close Database

คำอธิบาย เป็นการยกเลิกการเรียกใช้ฐานข้อมูลปัจจุบันที่กำลังใช้อยู่



กระบวนการที่ 6.3.2.8

ชื่อกระบวนการ List Relation

คำอธิบาย เป็นการดำเนินงานแสดงชื่อตารางข้อมูลทั้งหมดในฐานข้อมูลที่เรียกใช้

กระบวนการที่ 6.3.2.9

ชื่อกระบวนการ List Database

คำอธิบาย เป็นการดำเนินงานแสดงชื่อฐานข้อมูลทั้งหมดที่มีอยู่ในระบบ

กระบวนการที่ 6.3.2.10

ชื่อกระบวนการ Keep Relation

คำอธิบาย เป็นการดำเนินงานบรรจุตารางข้อมูลที่ได้จากคำสั่งปฏิบัติการ ERAO ลงใน  
แฟ้มข้อมูลต่าง ๆ

กระบวนการที่ 6.3.2.11

ชื่อกระบวนการ Delete Relation

คำอธิบาย เป็นการดำเนินงานกำจัดตารางข้อมูลออกจากแฟ้มข้อมูลต่าง ๆ

#### 4.5 การดำเนินงานของคำสั่งปฏิบัติการใน ERAO

จากการดำเนินงานของระบบเมื่อต้องการสอบถามจะเป็นการดำเนินงานในส่วนของคำสั่งปฏิบัติการ ERAO โปรแกรมในส่วนการสอบถามได้นำบางส่วนมาจากโปรแกรมที่ได้พัฒนาเรียบร้อยแล้ว [ทวีศักดิ์ (2534)] และพัฒนาเพิ่มเติมเพื่อให้การดำเนินงานถูกต้องตามที่ได้ออกแบบ

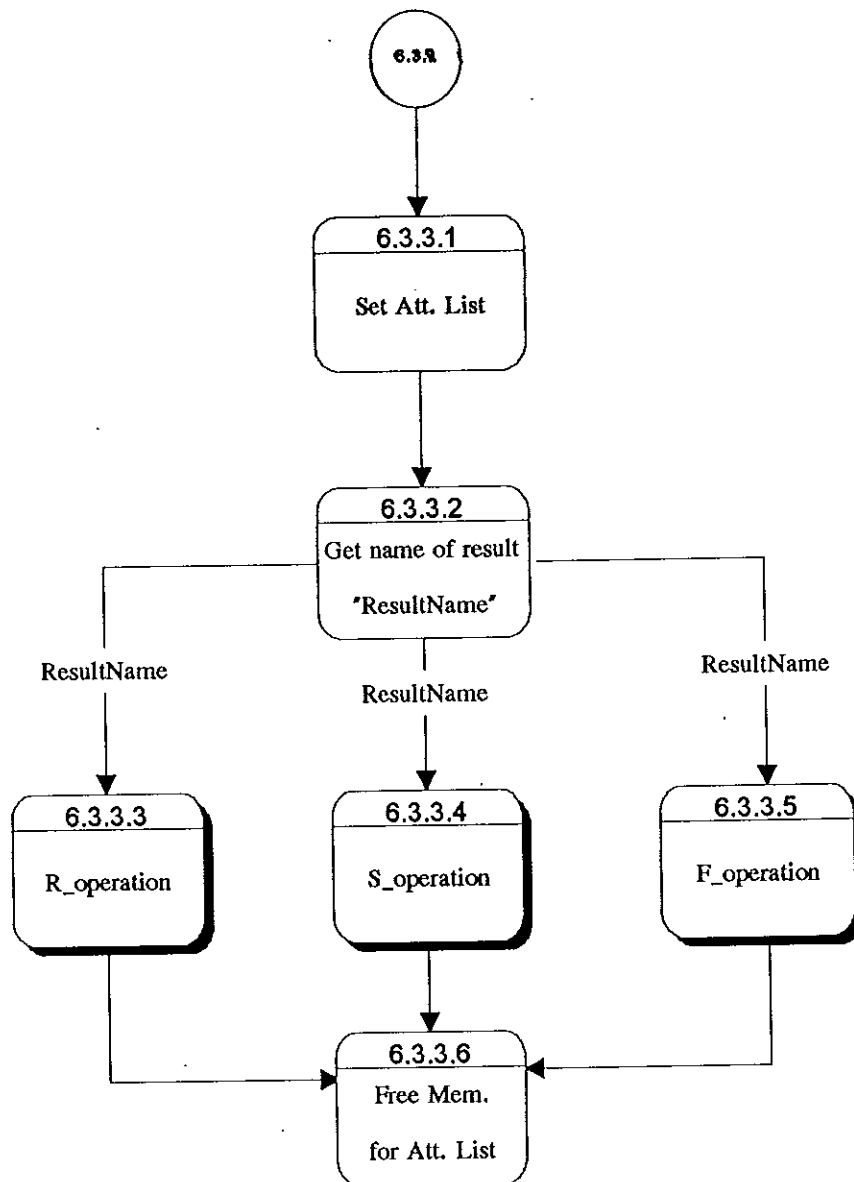
โปรแกรมที่ได้พัฒนาเรียบร้อยแล้วนำมาเพียงโปรแกรมย่อยบางโปรแกรม ซึ่งดำเนินการหลักๆ ในคำสั่งปฏิบัติการมาตรฐาน 8 คำสั่ง ก่อนการเรียกใช้โปรแกรมย่อยที่นำมานี้ได้พัฒนาในส่วนการเตรียมค่าต่างๆ เพื่อใช้ในโปรแกรมย่อยเหล่านั้น ภายในโปรแกรมย่อยทั้ง 8 ได้นำมาแก้ไขปรับปรุง ดังนี้

- โปรแกรมย่อย Union โปรแกรมย่อย Intersect โปรแกรมย่อย Minus และโปรแกรมย่อย Cross Product ถูกนำมาใช้โดยปรับปรุงเพียงเล็กน้อย
- โปรแกรมย่อย Select ผู้ทำวิจัยพบว่าในงานวิจัยเดิมกรณีที่มีการเปรียบเทียบจะทำการเปรียบเทียบโดยใช้ค่า ASCII ซึ่งกรณีที่ค่าที่นำมาเปรียบเทียบนั้นเป็นเลขจำนวน พบว่าในบางกรณีคำตอบที่ได้จะผิดจากความเป็นจริง เช่นการเปรียบเทียบค่า  $1300 > 700$  จะได้คำตอบเป็นเท็จ ดังนั้นผู้วิจัยจึงได้พัฒนาเพิ่มเติมในส่วนนี้
- โปรแกรมย่อย Project เปลี่ยนแปลงการรับแอดทริบิวเพื่อใช้ในการสกัดข้อมูล
- โปรแกรมย่อย Join ได้ปรับปรุงการสร้างเงื่อนไข และแอดทริบิวที่ใช้
- โปรแกรมย่อย Divide ตัดขั้นตอนการตรวจสอบคุณสมบัติของแอดทริบิวบางอย่างที่ไม่จำเป็นออก

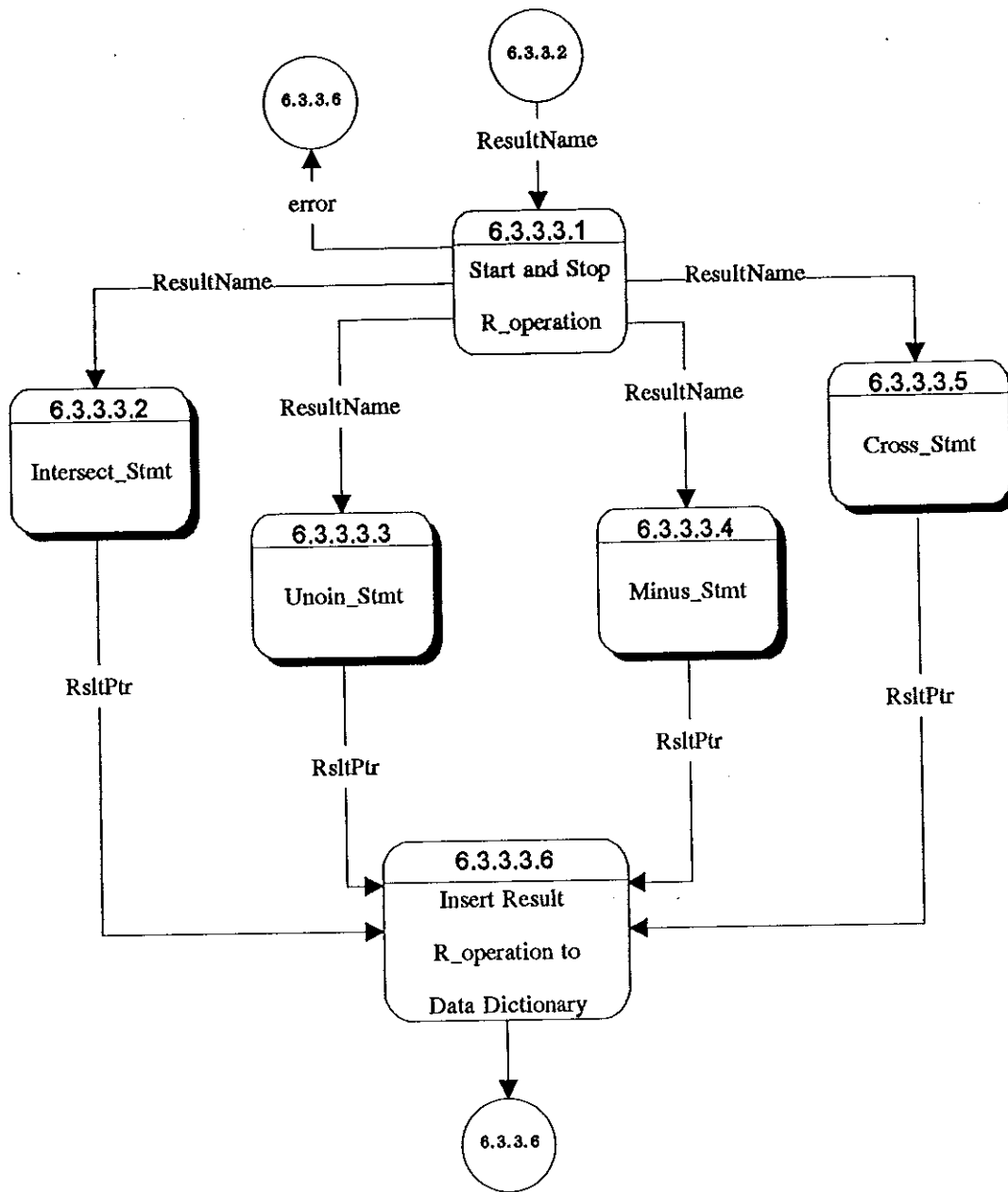
## แผนภาพกระแสข้อมูล

รายละเอียดการดำเนินงานของแต่ละคำสั่งปฏิบัติการใน ERAO จะแทนด้วยแผนภาพกระแสข้อมูล โดยภายในแผนภาพกระแสข้อมูล กระบวนการซึ่งมีเครื่องหมาย “\*” แสดงภายในชื่อกระบวนการ จะหมายถึงกระบวนการนั้น ๆ จะมีการอธิบายขั้นตอนการดำเนินการอย่างละเอียดภายในกระบวนการด้วยผังงาน (Flow Chart) ควบคู่กันไปด้วยข้อมูลต่าง ๆ ที่มีการส่งผ่านภายในแผนภาพกระแสข้อมูลมีรายละเอียดดังนี้

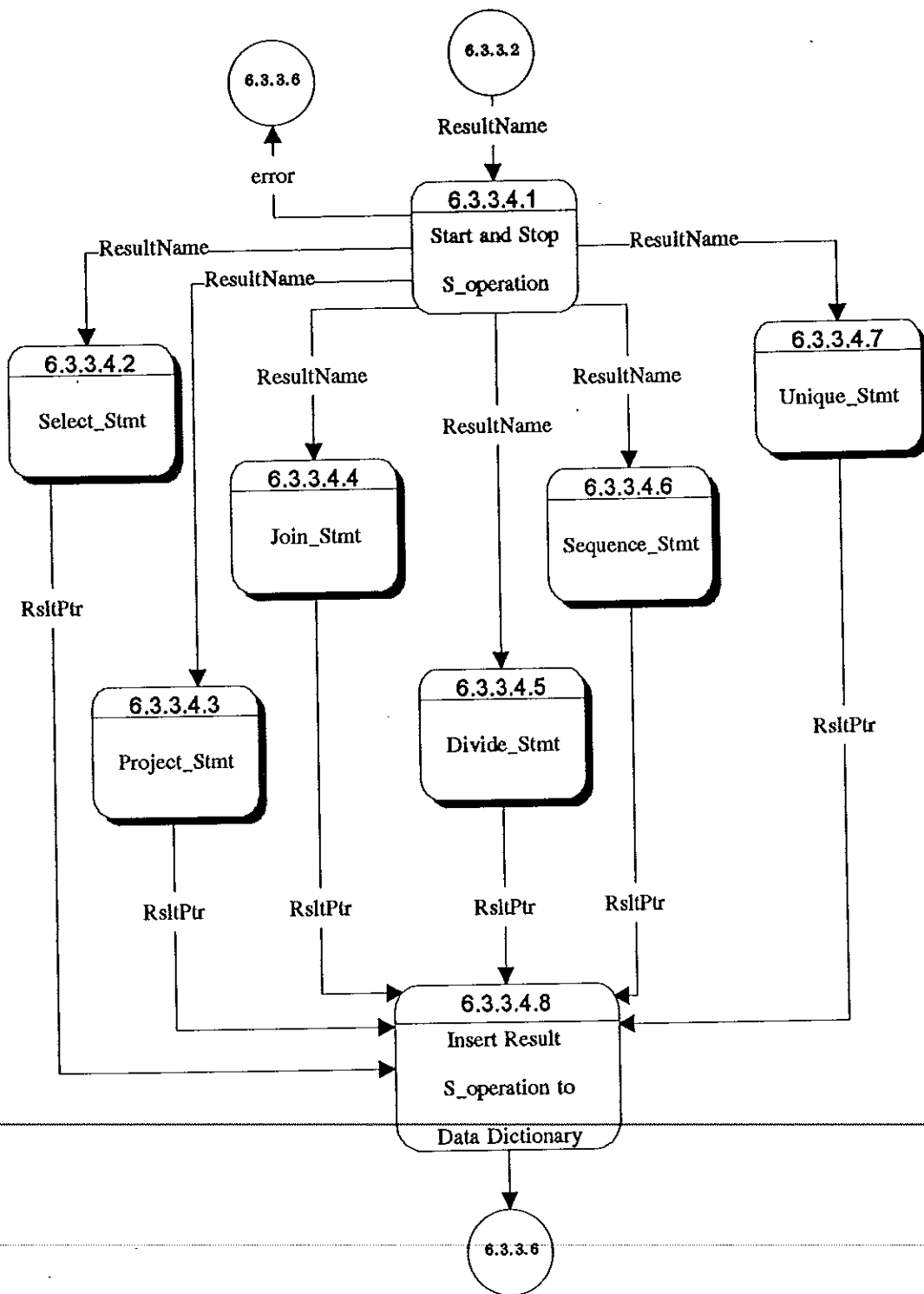
RelsuitName	ตัวแปรที่ใช้แทนชื่อของตารางข้อมูลผลลัพธ์ซึ่งรับจาก แถวคำสั่งปฏิบัติการ
RsltPtr	ตัวแปรชนิด Pointer ซึ่งชี้ไปยังโหนดของตารางข้อมูลผลลัพธ์
Table, Table1, Table2	ตัวแปรที่ใช้แทนชื่อของตารางข้อมูลซึ่งรับจากแถวคำสั่งปฏิบัติการ เพื่อใช้ตารางข้อมูลนี้ในการดำเนินการคำสั่งปฏิบัติการต่าง ๆ
Cnd	เครื่องหมายการเปรียบเทียบซึ่งรับจากแถวคำสั่งปฏิบัติการ
Attr, Attr1, Attr2, Attr_name	ตัวแปรที่ใช้แทนชื่อของแอตทริบิวต์ซึ่งรับจากแถวคำสั่ง ปฏิบัติการ
Attr_lst	กลุ่มตัวแปรที่ใช้แทนชื่อของแอตทริบิวต์ซึ่งรับจากแถวคำสั่งปฏิบัติการ
Constant	ค่าคงที่ซึ่งรับจากแถวคำสั่งปฏิบัติการ
CaseFun	ตัวแปรซึ่งรับจากแถวคำสั่งปฏิบัติการใช้บ่งบอกถึงฟังก์ชันที่จะหา



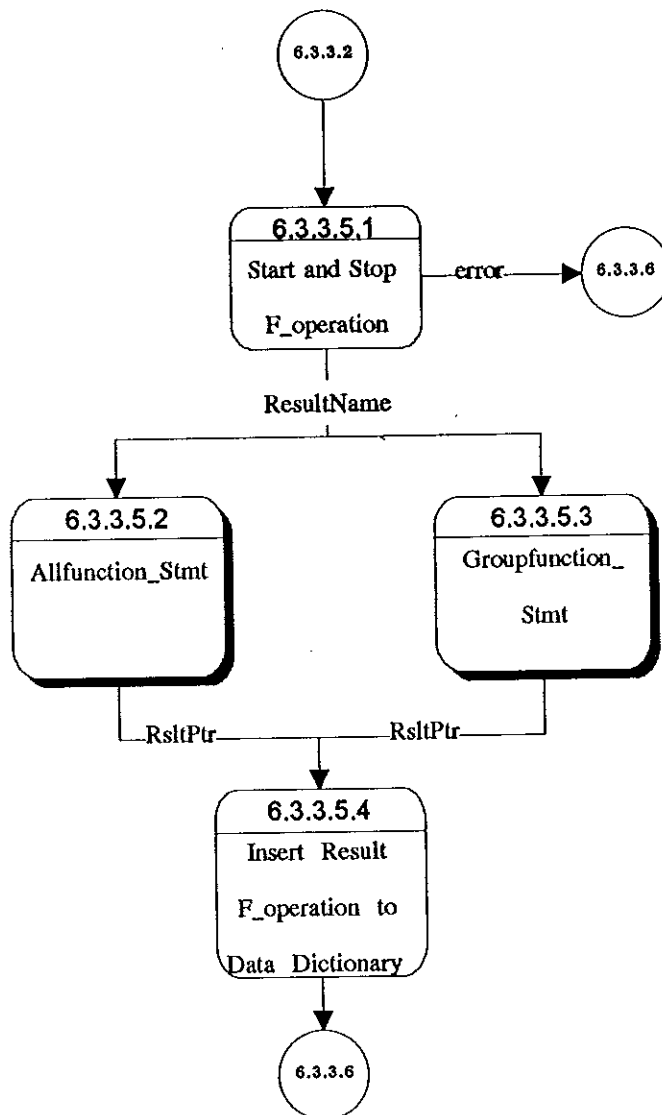
รูปที่ 4.8 แผนภาพกระแสข้อมูลของกระบวนการ Query Command



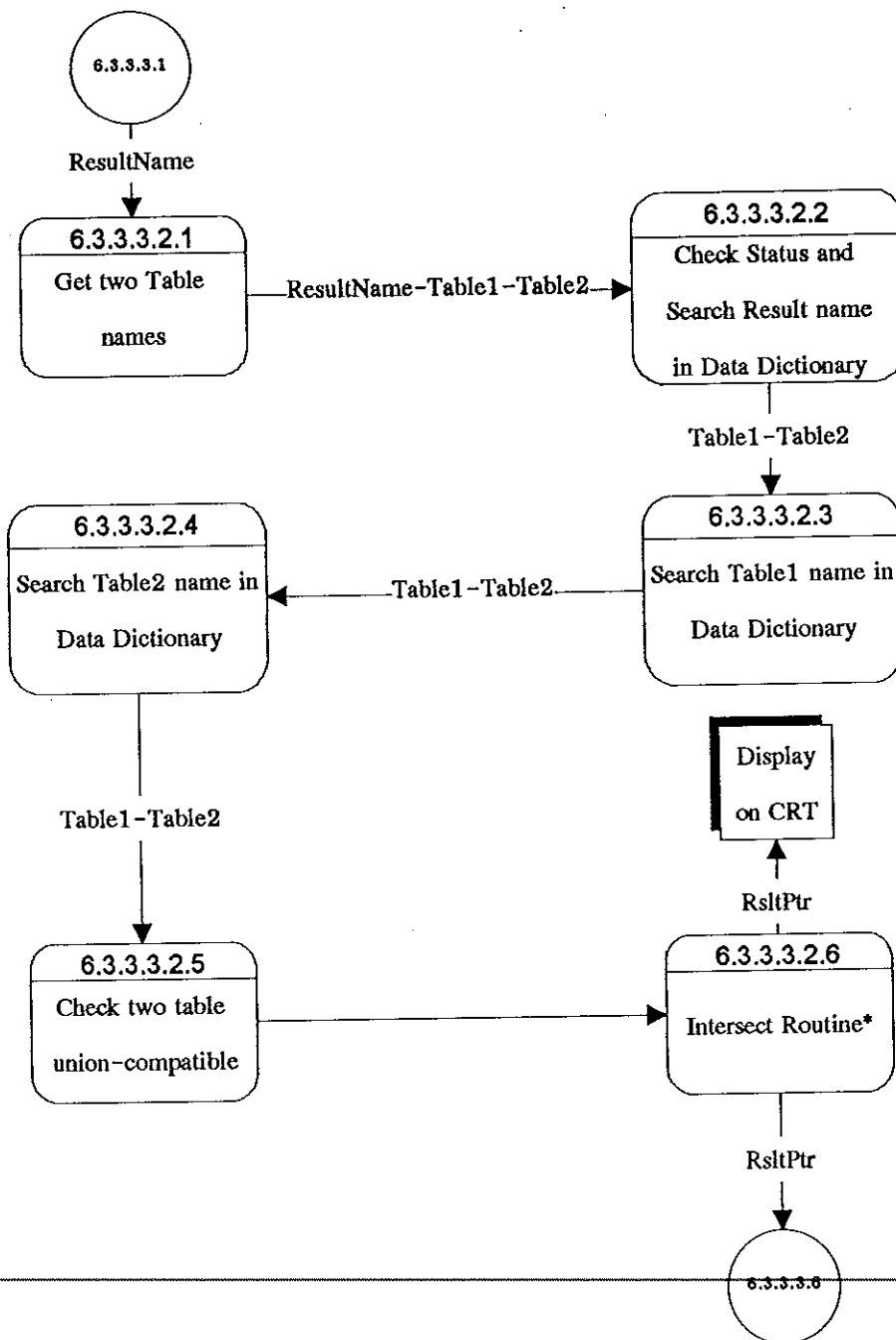
รูปที่ 4.9 แผนภาพกระแสข้อมูลของกระบวนการ R\_operation



รูปที่ 4.10 แผนภาพกระแสข้อมูลของกระบวนการ S\_operation

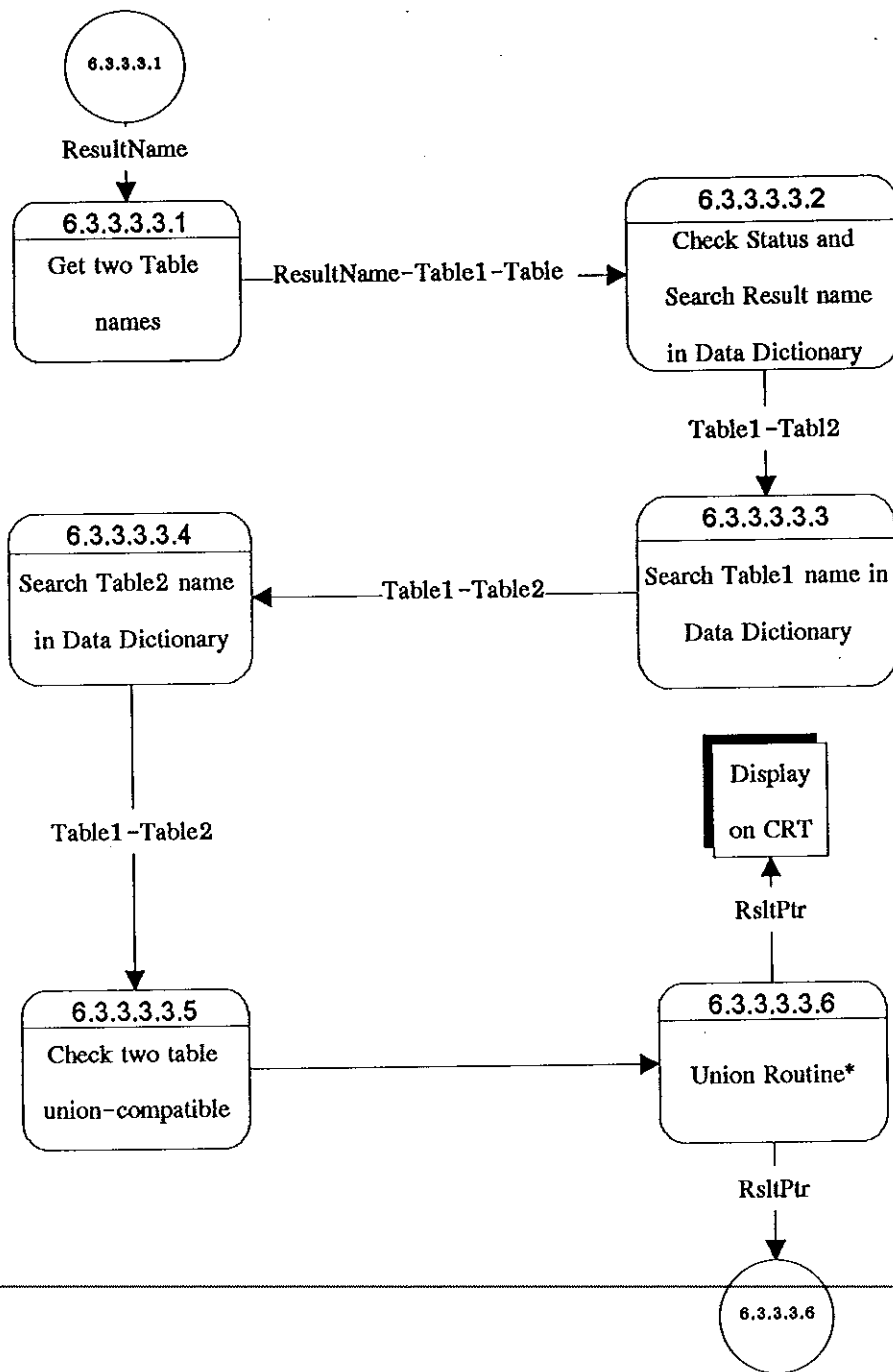


รูปที่ 4.11 แผนภาพกระแสข้อมูลของกระบวนการ F\_operation

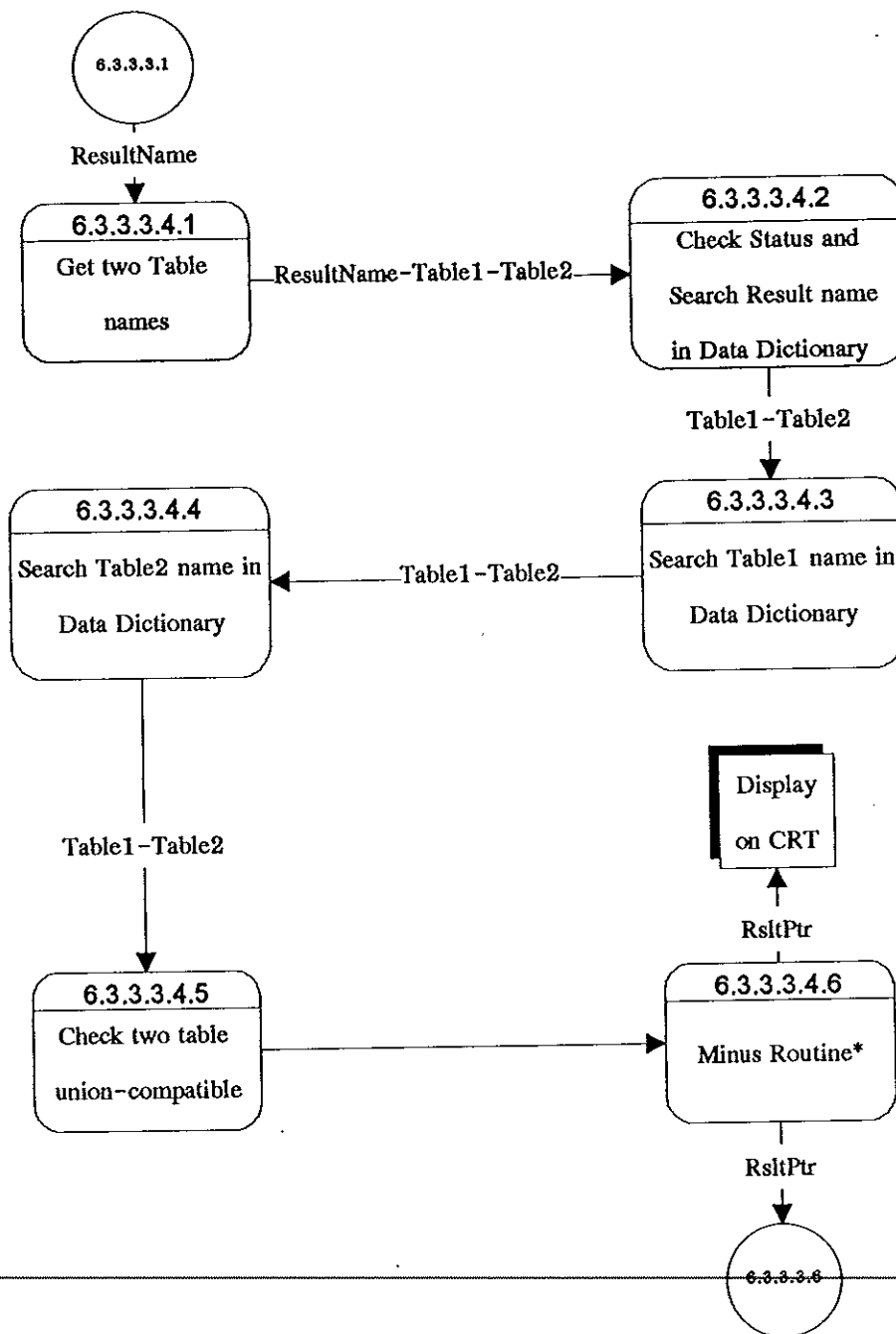


รูปที่ 4.12 แผนภาพกระแสข้อมูลของกระบวนการ Intersect\_Stmt

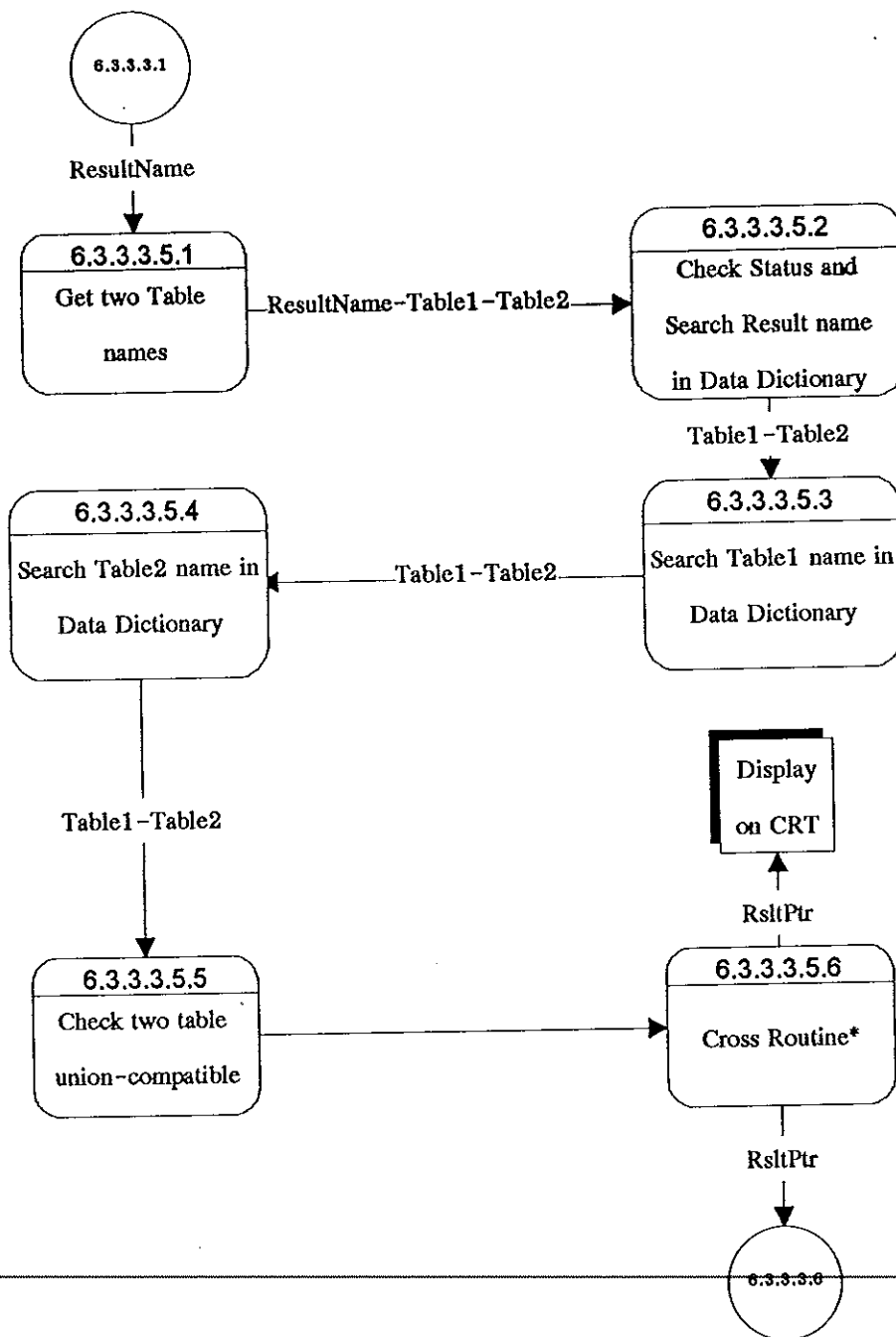




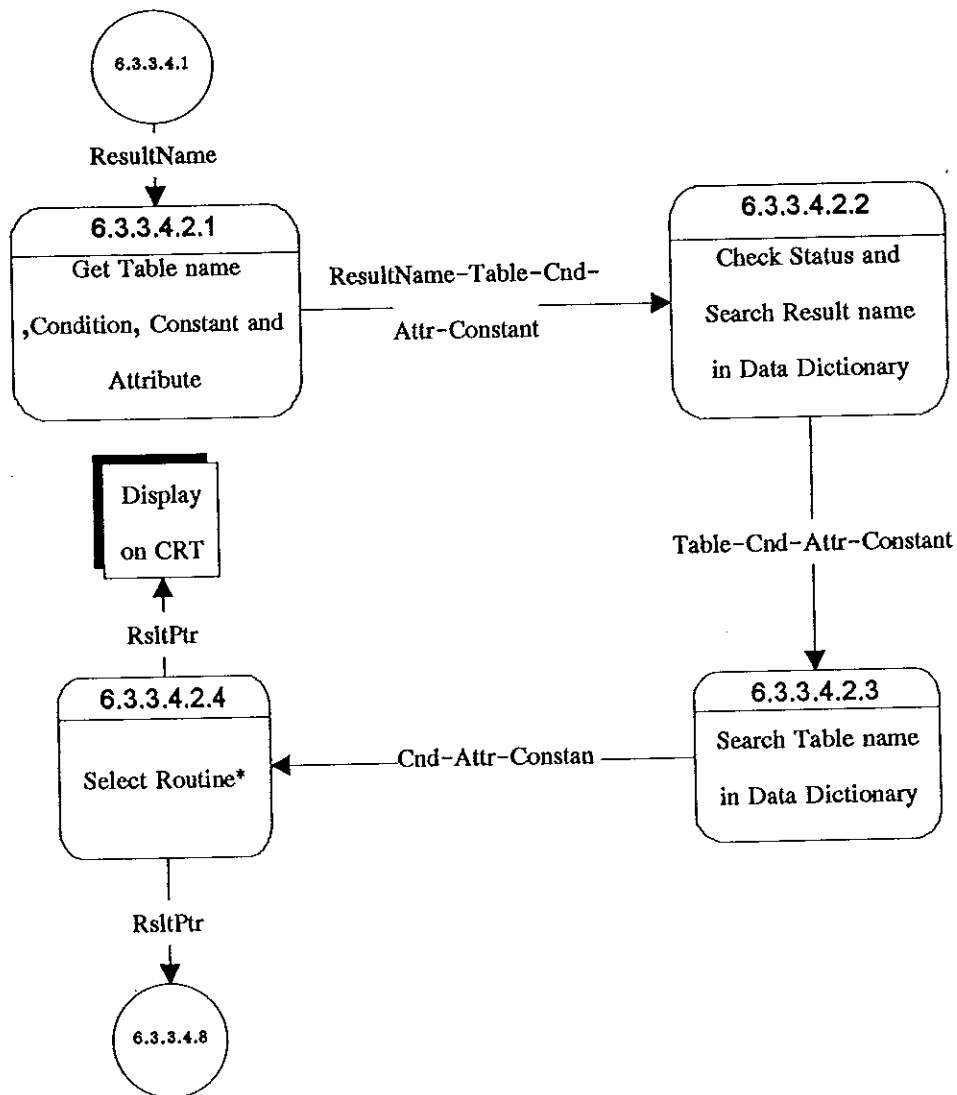
รูปที่ 4.13 แผนภาพกระแสข้อมูลของกระบวนการ Union Stmt



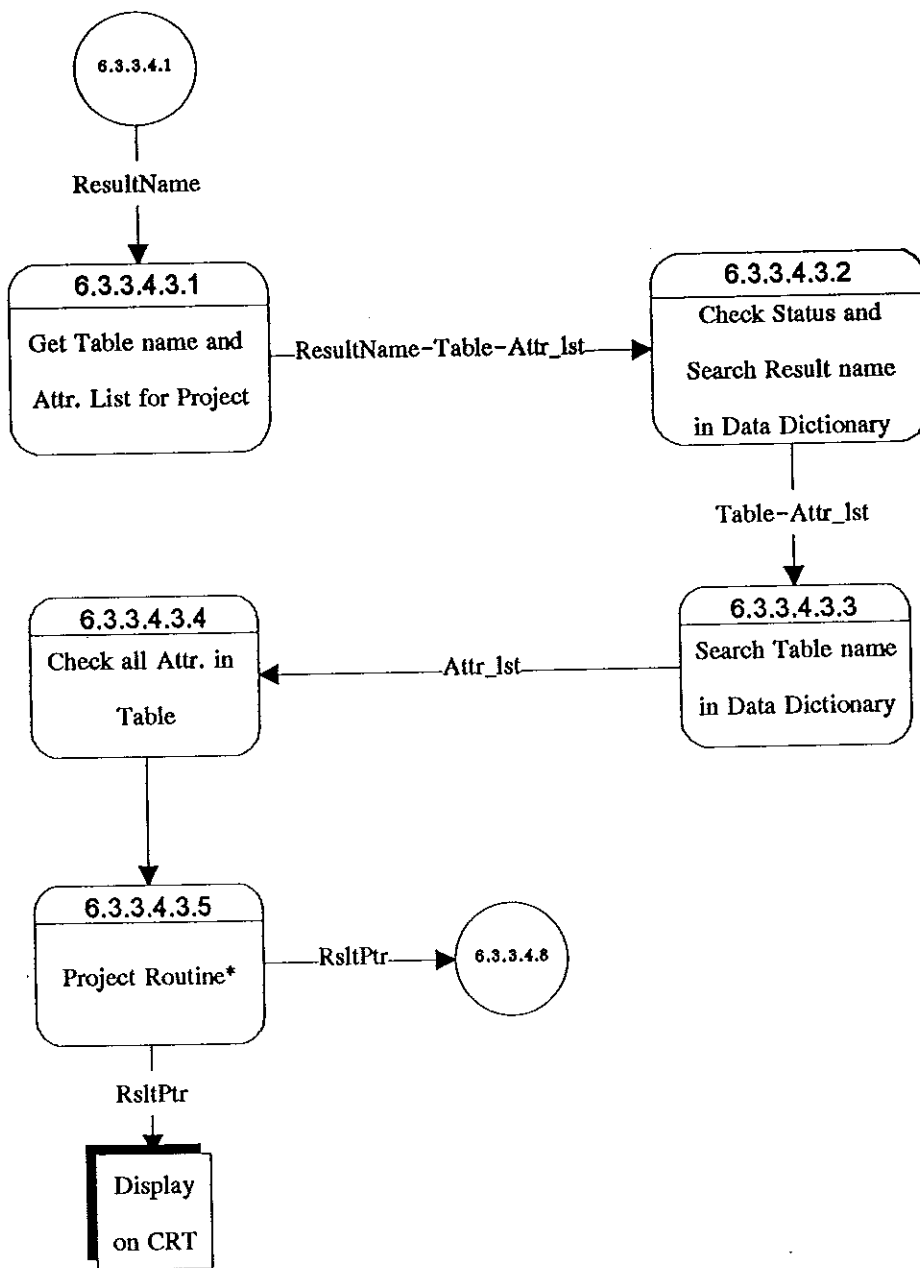
รูปที่ 4.14 แผนภาพกระแสข้อมูลของกระบวนการ Minus\_Stmt



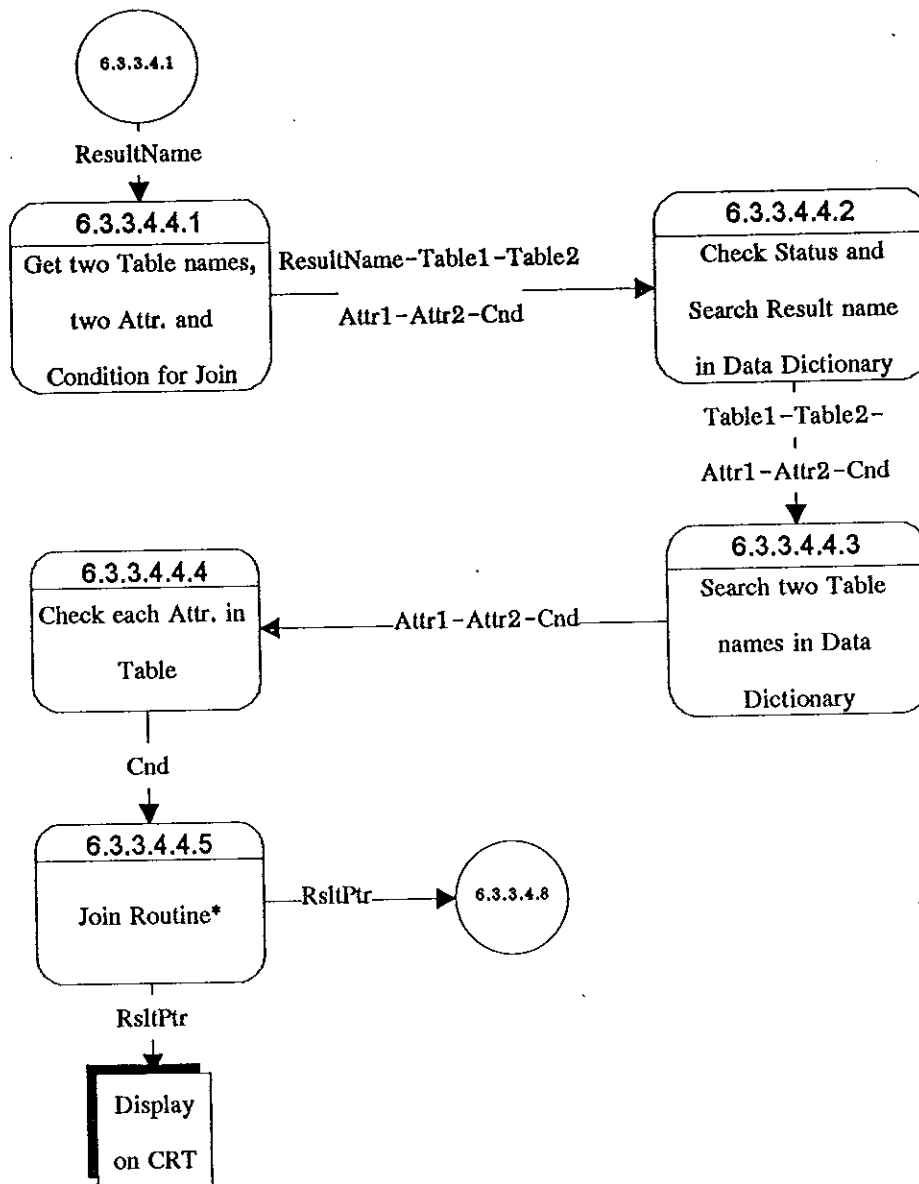
รูปที่ 4.15 แผนภาพกระแสข้อมูลของกระบวนการ Cross\_Stmt



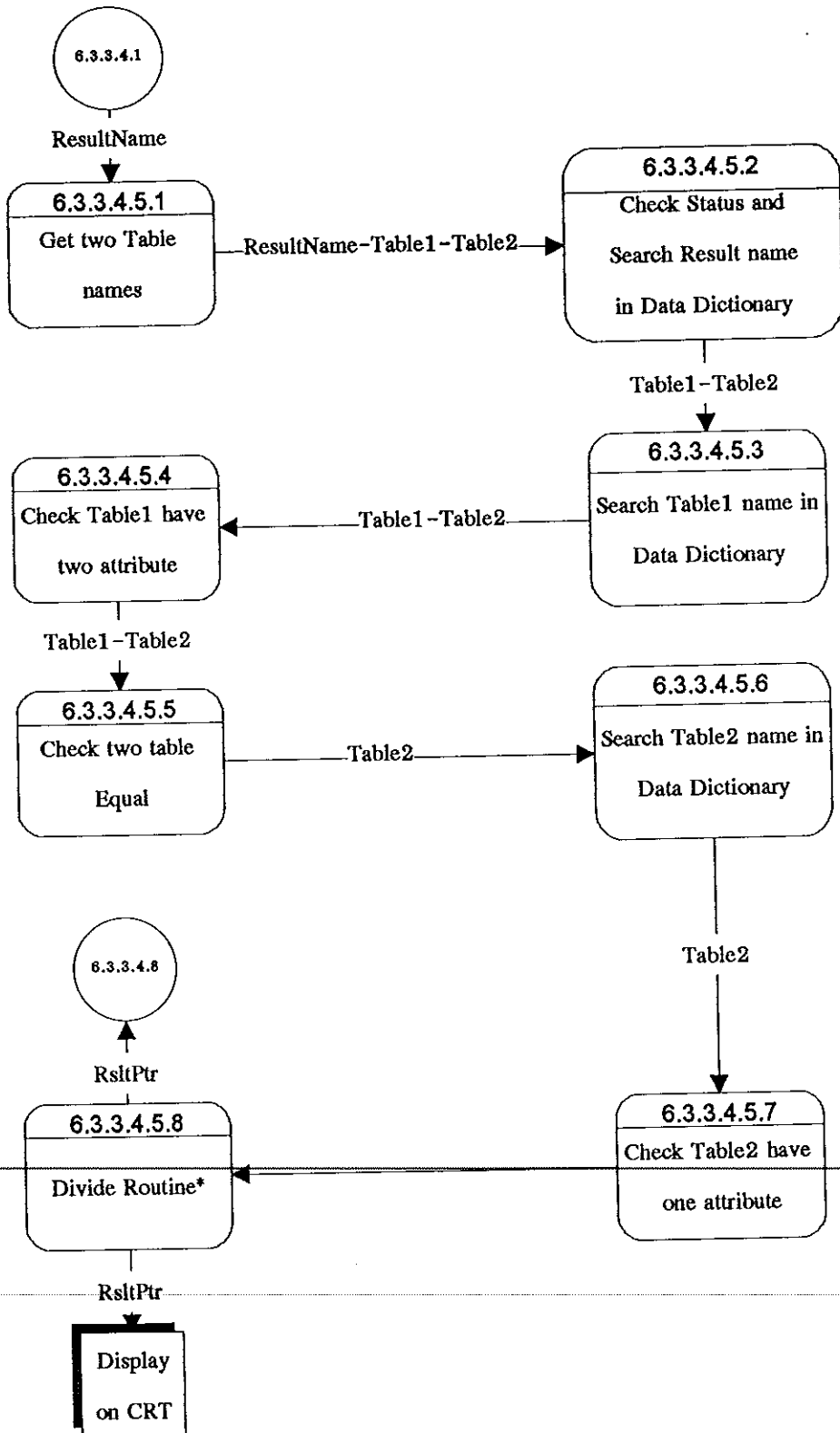
รูปที่ 4.16 แผนภาพกระแสข้อมูลของกระบวนการ Select Stmt



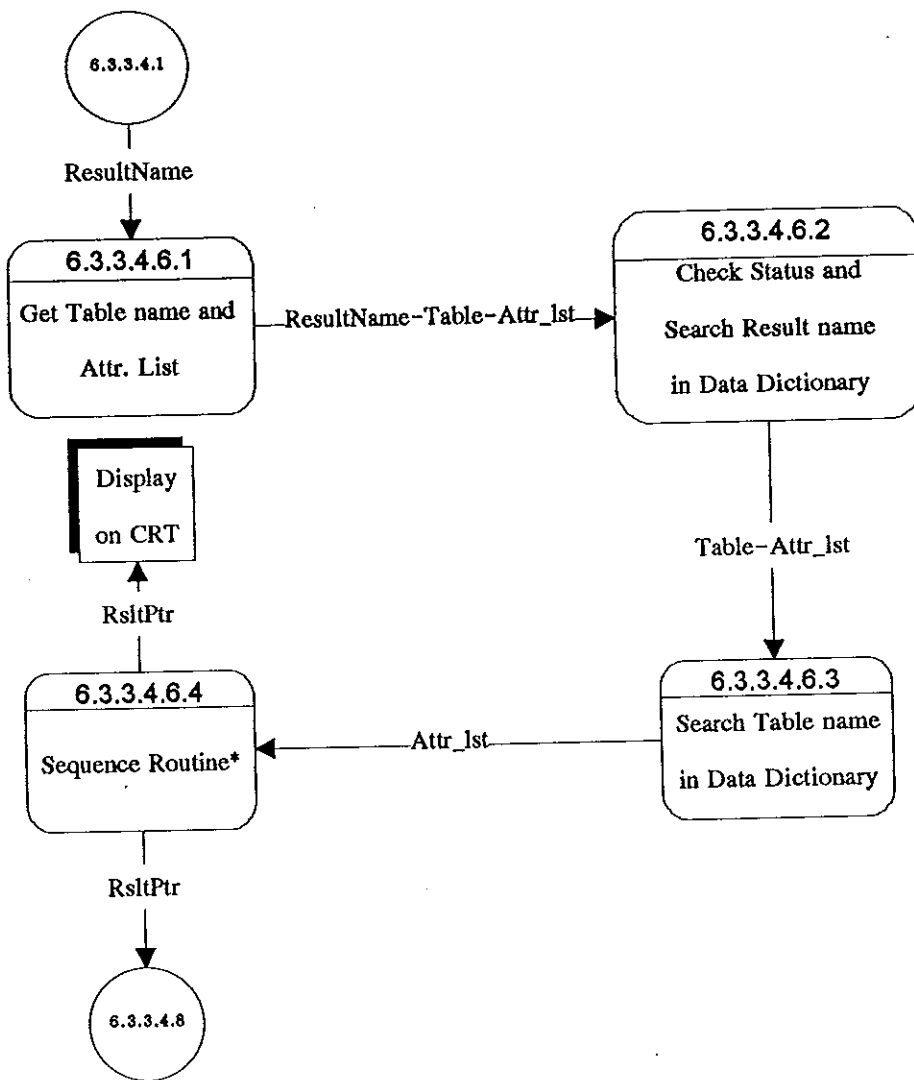
รูปที่ 4.17 แผนภาพกระแสข้อมูลของกระบวนการ Project Stmt



รูปที่ 4.18 แผนภาพกระแสข้อมูลของกระบวนการ Join\_Stmt

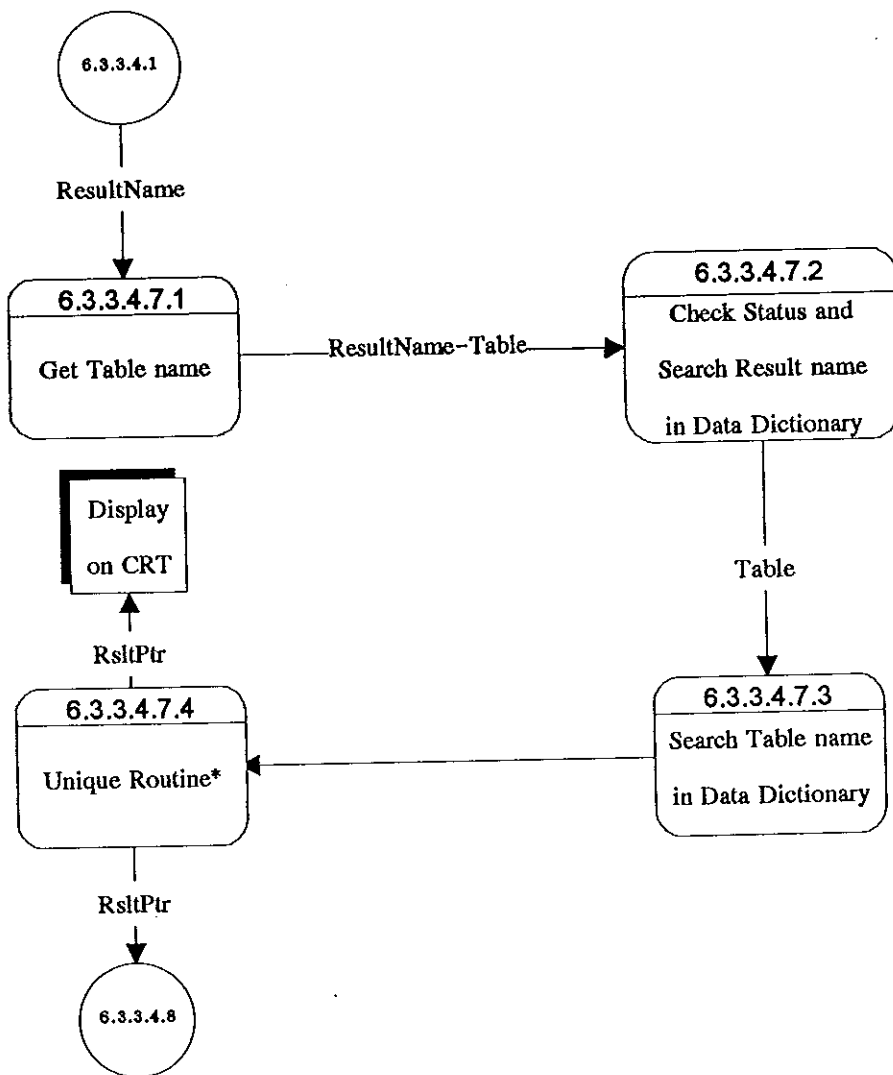


รูปที่ 4.19 แผนภาพกระแสข้อมูลของกระบวนการ Divide\_Stmt

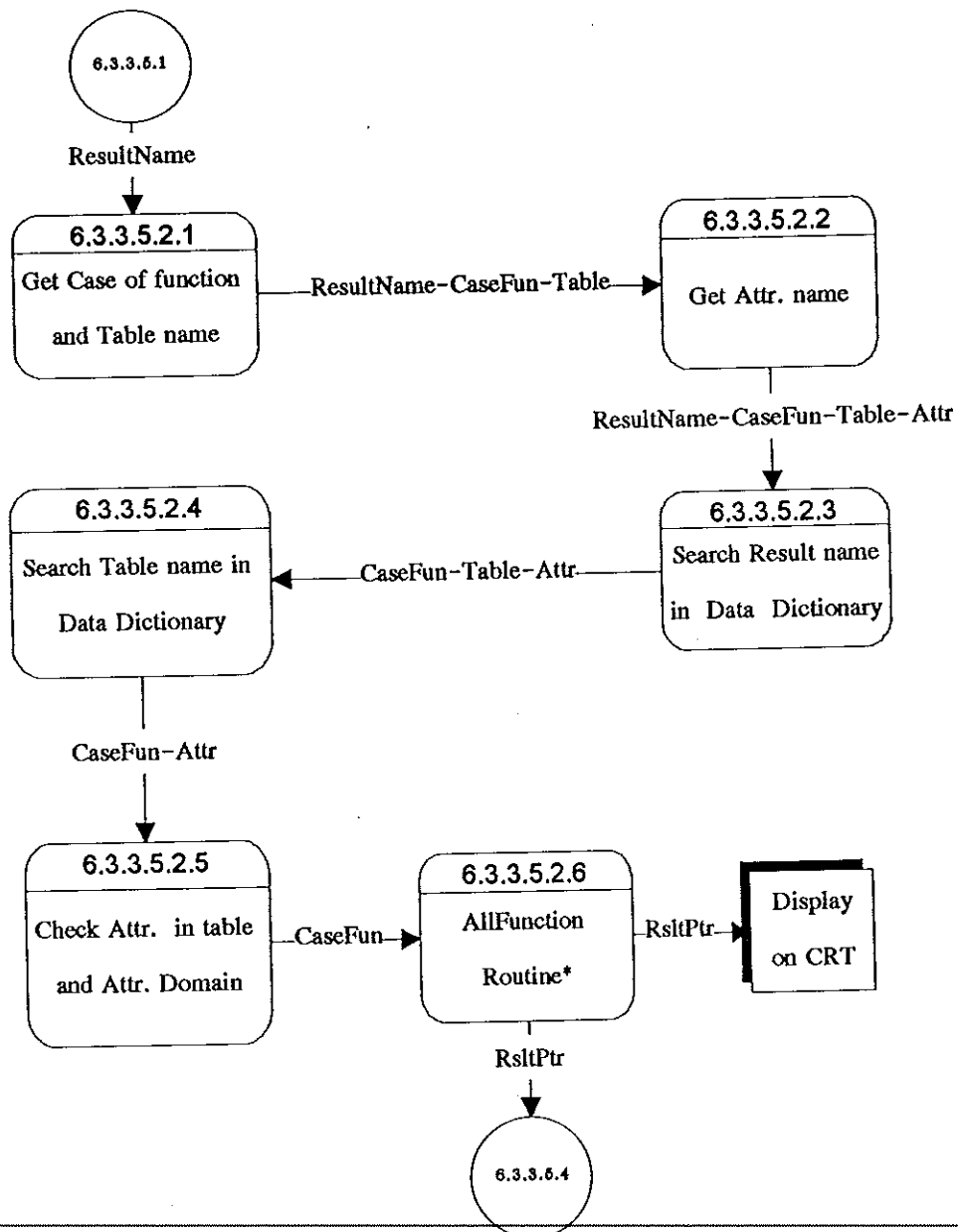


รูปที่ 4.20 แผนภาพกระแสข้อมูลของกระบวนการ Sequence Stmt

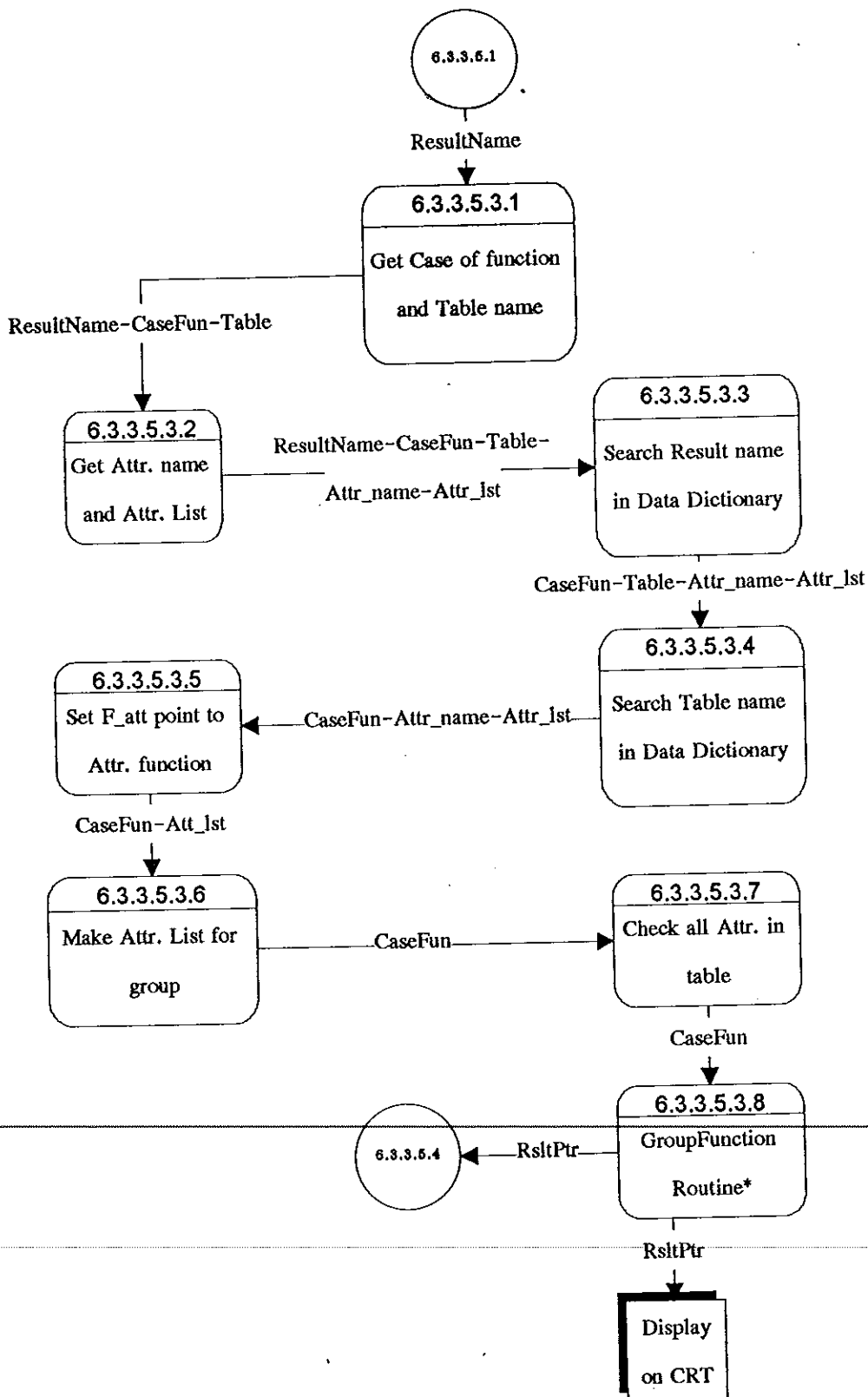




รูปที่ 4.21 แผนภาพกระแสข้อมูลของกระบวนการ Unique Stmt



รูปที่ 4.22 แผนภาพกระแสข้อมูลของกระบวนการ Allfunction\_Stmt



รูปที่ 4.23 แผนภาพกระแสข้อมูลของกระบวนการ Groupfunction Stmt

## คำอธิบายการดำเนินงานแผนภาพกระแสข้อมูล

### จากรูปที่ 4.8

กระบวนการที่ 6.3.3.1

ชื่อกระบวนการ Set Att.-List

คำอธิบาย เป็นการจัดเตรียมตัวชี้สำหรับชี้รายการโยงแอตทริบิวชั่วคราว

กระบวนการที่ 6.3.3.2

ชื่อกระบวนการ Get name of result "ResultName"

คำอธิบาย เป็นการรับชื่อตารางข้อมูลผลลัพธ์

กระบวนการที่ 6.3.3.3

ชื่อกระบวนการ R\_operation

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการ ERAO ในกลุ่มคำสั่งปฏิบัติการกับเซต

กระบวนการที่ 6.3.3.4

ชื่อกระบวนการ S\_operation

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการ ERAO ในกลุ่มคำสั่งปฏิบัติการพิเศษ

กระบวนการที่ 6.3.3.5

ชื่อกระบวนการ F\_operation

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการ ERAO ในกลุ่มคำสั่งปฏิบัติการในรูป ฟังก์ชัน

กระบวนการที่ 6.3.3.6

ชื่อกระบวนการ Free Mem. for Att. List

คำอธิบาย เป็นการคืนหน่วยความจำสำหรับรายการโยงแอตทริบิวชั่วคราวให้กับระบบ

## จากรูปที่ 4.9

กระบวนการที่ 6.3.3.3.1

ชื่อกระบวนการ Start and Stop R\_operation

คำอธิบาย เป็นการควบคุมการเริ่มต้น และสิ้นสุดการดำเนินงานในแต่ละแถวคำสั่งของกลุ่มคำสั่งปฏิบัติการกับเซต

กระบวนการที่ 6.3.3.3.2

ชื่อกระบวนการ Intersect\_Stmt

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการ Intersect

กระบวนการที่ 6.3.3.3.3

ชื่อกระบวนการ Union\_Stmt

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการ Union

กระบวนการที่ 6.3.3.3.4

ชื่อกระบวนการ Minus\_Stmt

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการ Minus

กระบวนการที่ 6.3.3.3.5

ชื่อกระบวนการ Cross\_Stmt

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการ Cross

กระบวนการที่ 6.3.3.3.6

ชื่อกระบวนการ Insert Result R\_operation to Data Dictionary

คำอธิบาย เป็นการดำเนินงานบรรจุตารางข้อมูลผลลัพธ์ที่ได้จากการดำเนินการคำสั่งปฏิบัติการในกลุ่มคำสั่งปฏิบัติการกับเซตเพิ่มในปทานุกรมข้อมูล

**จากรูปที่ 4.10**

กระบวนการที่ 6.3.3.4.1

ชื่อกระบวนการ Start and Stop S\_operation

คำอธิบาย เป็นการควบคุมการเริ่มต้น และสิ้นสุดการดำเนินงานในแต่ละแถวคำสั่งของกลุ่มคำสั่งปฏิบัติการพิเศษ

กระบวนการที่ 6.3.3.4.2

ชื่อกระบวนการ Select\_Stmt

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการ Select

กระบวนการที่ 6.3.3.4.3

ชื่อกระบวนการ Project\_Stmt

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการ Project

กระบวนการที่ 6.3.3.4.4

ชื่อกระบวนการ Join\_Stmt

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการ Join

กระบวนการที่ 6.3.3.4.5

ชื่อกระบวนการ Divide\_Stmt

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการ Divide

กระบวนการที่ 6.3.3.4.6

ชื่อกระบวนการ Sequence\_Stmt

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการ Sequence

กระบวนการที่ 6.3.3.4.7

ชื่อกระบวนการ Unique\_Stmt

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการ Unique

กระบวนการที่ 6.3.3.4.8

ชื่อกระบวนการ Insert Result S\_operation to Data Dictionary

คำอธิบาย เป็นการดำเนินงานบรรจุตารางข้อมูลผลลัพธ์ที่ได้จากการดำเนินคำสั่งปฏิบัติการในกลุ่มคำสั่งปฏิบัติการพิเศษเพิ่มในปทานุกรมข้อมูล

จากรูปที่ 4.11

กระบวนการที่ 6.3.3.5.1

ชื่อกระบวนการ Start and Stop F\_operation

คำอธิบาย เป็นการควบคุมการเริ่มต้น และสิ้นสุดการดำเนินงานในแต่ละแถวคำสั่งของกลุ่มคำสั่งปฏิบัติการในรูปฟังก์ชัน

กระบวนการที่ 6.3.3.5.2

ชื่อกระบวนการ Allfunction Stmt

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการในรูปฟังก์ชันจากทุกแถวข้อมูลของตารางข้อมูล

กระบวนการที่ 6.3.3.5.3

ชื่อกระบวนการ Groupfunction Stmt

คำอธิบาย เป็นการดำเนินงานคำสั่งปฏิบัติการในรูปฟังก์ชันสำหรับแต่ละกลุ่มย่อยของแถวข้อมูลของตารางข้อมูล

กระบวนการที่ 6.3.3.5.4

ชื่อกระบวนการ Insert Result F\_operation to Data Dictionary

คำอธิบาย เป็นการดำเนินงานบรรจุตารางข้อมูลผลลัพธ์ที่ได้จากการดำเนินคำสั่งปฏิบัติการในรูปฟังก์ชันเพิ่มในปทานุกรมข้อมูล

## จากรูปที่ 4.12

กระบวนการที่ 6.3.3.3.2.1

ชื่อกระบวนการ Get two Table names

คำอธิบาย เป็นการรับชื่อตารางข้อมูล 2 ตาราง เพื่อนำไปใช้ในการดำเนินคำสั่งปฏิบัติการ Intersect

กระบวนการที่ 6.3.3.3.2.2

ชื่อกระบวนการ Check Status and Search Result name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบว่ามีการเรียกใช้ฐานข้อมูลหรือไม่ และตรวจสอบชื่อตารางข้อมูลผลลัพธ์ว่ามีอยู่แล้วในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.3.2.3

ชื่อกระบวนการ Search Table1 name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลที่ 1 ว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.3.2.4

ชื่อกระบวนการ Search Table2 name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลที่ 2 ว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.3.2.5

ชื่อกระบวนการ Check two table union-compatible

คำอธิบาย เป็นการตรวจสอบว่าตารางข้อมูลทั้งสองมีคุณสมบัติ union-compatible หรือไม่

กระบวนการที่ 6.3.3.3.2.6

ชื่อกระบวนการ Intersect Routine\*

คำอธิบาย เป็นการดำเนินการในโปรแกรมย่อย Intersect



### จากรูปที่ 4.13

กระบวนการที่ 6.3.3.3.1

ชื่อกระบวนการ Get two Table names

คำอธิบาย เป็นการรับชื่อตารางข้อมูล 2 ตาราง เพื่อนำไปใช้ในการดำเนินคำสั่งปฏิบัติการ Union

กระบวนการที่ 6.3.3.3.2

ชื่อกระบวนการ Check Status and Search Result name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบว่ามีการเรียกใช้ฐานข้อมูลหรือไม่ และตรวจสอบชื่อตารางข้อมูลผลลัพธ์ว่ามีอยู่แล้วในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.3.3

ชื่อกระบวนการ Search Table1 name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลที่ 1 ว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.3.4

ชื่อกระบวนการ Search Table2 name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลที่ 2 ว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.3.5

ชื่อกระบวนการ Check two table union-compatible

คำอธิบาย เป็นการตรวจสอบว่าตารางข้อมูลทั้งสองมีคุณสมบัติ union-compatible หรือไม่

กระบวนการที่ 6.3.3.3.6

ชื่อกระบวนการ Union Routine\*

คำอธิบาย เป็นการดำเนินการในโปรแกรมย่อย Union

## จากรูปที่ 4.14

กระบวนการที่ 6.3.3.3.4.1

ชื่อกระบวนการ Get two Table names

คำอธิบาย เป็นการรับชื่อตารางข้อมูล 2 ตาราง เพื่อนำไปใช้ในการดำเนินการคำนวณคำสั่งปฏิบัติการ Minus

กระบวนการที่ 6.3.3.3.4.2

ชื่อกระบวนการ Check Status and Search Result name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบว่ามีการเรียกใช้ฐานข้อมูลหรือไม่ และตรวจสอบชื่อตารางข้อมูลผลลัพธ์ว่ามีอยู่แล้วในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.3.4.3

ชื่อกระบวนการ Search Table1 name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลที่ 1 ว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.3.4.4

ชื่อกระบวนการ Search Table2 name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลที่ 2 ว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.3.4.5

ชื่อกระบวนการ Check two table union-compatible

คำอธิบาย เป็นการตรวจสอบว่าตารางข้อมูลทั้งสองมีคุณสมบัติ union-compatible หรือไม่

กระบวนการที่ 6.3.3.3.4.6

ชื่อกระบวนการ Minus Routine\*

คำอธิบาย เป็นการดำเนินการในโปรแกรมย่อย Minus

### จากรูปที่ 4.15

กระบวนการที่ 6.3.3.3.5.1

ชื่อกระบวนการ Get two Table names

คำอธิบาย เป็นการรับชื่อตารางข้อมูล 2 ตาราง เพื่อนำไปใช้ในการดำเนินการ Cross

กระบวนการที่ 6.3.3.3.5.2

ชื่อกระบวนการ Check Status and Search Result name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบว่ามีการเรียกใช้ฐานข้อมูลหรือไม่ และตรวจสอบชื่อตารางข้อมูลผลลัพธ์ว่ามีอยู่แล้วในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.3.5.3

ชื่อกระบวนการ Search Table1 name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลที่ 1 ว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.3.5.4

ชื่อกระบวนการ Search Table2 name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลที่ 2 ว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.3.5.5

ชื่อกระบวนการ Check two table union-compatible

คำอธิบาย เป็นการตรวจสอบว่าตารางข้อมูลทั้งสองมีคุณสมบัติ union-compatible หรือไม่

กระบวนการที่ 6.3.3.3.5.6

ชื่อกระบวนการ Cross Routine\*

คำอธิบาย เป็นการดำเนินการในโปรแกรมย่อย Cross

จากรูปที่ 4.16

กระบวนการที่ 6.3.3.4.2.1

ชื่อกระบวนการ Get Table name, Condition, Constant and Attribute

คำอธิบาย เป็นการรับชื่อตารางข้อมูล, ตัวดำเนินการเชิงเปรียบเทียบ, ค่าคงที่ และชื่อแอตทริบิวเพื่อนำไปใช้ในการดำเนินคำสั่งปฏิบัติการ Select

กระบวนการที่ 6.3.3.4.2.2

ชื่อกระบวนการ Check Status and Search Result name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบว่ามีการเรียกใช้ฐานข้อมูลหรือไม่ และตรวจสอบชื่อตารางข้อมูลผลลัพธ์ว่ามีอยู่แล้วในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.4.2.3

ชื่อกระบวนการ Search Table name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.4.2.4

ชื่อกระบวนการ Select Routine\*

คำอธิบาย เป็นการดำเนินการในโปรแกรมย่อย Select

จากรูปที่ 4.17

กระบวนการที่ 6.3.3.4.3.1

ชื่อกระบวนการ Get Table name and Attr. List for Project

คำอธิบาย เป็นการรับชื่อตารางข้อมูล และชื่อแอตทริบิวที่ต้องการทั้งหมดเพื่อนำไปใช้ในการดำเนินคำสั่งปฏิบัติการ Project

กระบวนการที่ 6.3.3.4.3.2

ชื่อกระบวนการ Check Status and Search Result name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบว่ามีการเรียกใช้ฐานข้อมูลหรือไม่ และตรวจสอบชื่อตารางข้อมูลผลลัพธ์ว่ามีอยู่แล้วในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.4.3.3

ชื่อกระบวนการ Search Table name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.4.3.4

ชื่อกระบวนการ Check all Attr. in Table

คำอธิบาย เป็นการตรวจสอบแอตทริบิวต์ทั้งหมดอยู่ในตารางข้อมูลหรือไม่

กระบวนการที่ 6.3.3.4.3.5

ชื่อกระบวนการ Project Routine\*

คำอธิบาย เป็นการดำเนินการในโปรแกรมย่อย Project

#### จากรูปที่ 4.18

กระบวนการที่ 6.3.3.4.4.1

ชื่อกระบวนการ Get two Table names, two Attr. and Condition for Join

คำอธิบาย เป็นการรับชื่อตารางข้อมูล 2 ตาราง ,ชื่อแอตทริบิวต์ 2 แอตทริบิวต์ และตัวดำเนินการเชิงเปรียบเทียบเพื่อนำไปใช้ในการดำเนินการคำสั่งปฏิบัติการ Join

กระบวนการที่ 6.3.3.4.4.2

ชื่อกระบวนการ Check Status and Search Result name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบว่ามีการเรียกใช้ฐานข้อมูลหรือไม่ และตรวจสอบชื่อตารางข้อมูลผลลัพธ์ว่ามีอยู่แล้วในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.4.4.3

ชื่อกระบวนการ Search two Table names in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลทั้งสองว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.4.4.4

ชื่อกระบวนการ Check each Attr. in Table

คำอธิบาย เป็นการตรวจสอบว่าแต่ละแอตทริบิวต์มีอยู่ในตารางข้อมูลอย่างถูกต้อง

กระบวนการที่ 6.3.3.4.4.5

ชื่อกระบวนการ Join Routine\*

คำอธิบาย เป็นการดำเนินการในโปรแกรมย่อย Join

**จากรูปที่ 4.19**

กระบวนการที่ 6.3.3.4.5.1

ชื่อกระบวนการ Get two Table names

คำอธิบาย เป็นการรับชื่อตารางข้อมูล 2 ตาราง เพื่อนำไปใช้ในการดำเนินการคำนวณค่าสัมประสิทธิ์การ Divide

กระบวนการที่ 6.3.3.4.5.2

ชื่อกระบวนการ Check Status and Search Result name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบว่ามีการเรียกใช้ฐานข้อมูลหรือไม่ และตรวจสอบชื่อตารางข้อมูลผลลัพธ์ว่ามีอยู่แล้วในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.4.5.3

ชื่อกระบวนการ Search Table1 name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลที่ 1 ว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.4.5.4

ชื่อกระบวนการ Check Table1 have two attribute

คำอธิบาย เป็นการตรวจสอบว่าตารางข้อมูลที่มีจำนวนแอตทริบิวต์เป็น 2 หรือไม่

กระบวนการที่ 6.3.3.4.5.5

ชื่อกระบวนการ Check two Table Equal

คำอธิบาย เป็นการตรวจสอบว่าตารางข้อมูลทั้งสองเป็นตารางข้อมูลเดียวกันหรือไม่

กระบวนการที่ 6.3.3.4.5.6

ชื่อกระบวนการ Search Table2 name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลที่ 2 ว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.4.5.7

ชื่อกระบวนการ Check Table2 have one attribute

คำอธิบาย เป็นการตรวจสอบว่าตารางข้อมูลที่มีจำนวนแอตทริบิวต์เป็น 1 หรือไม่

กระบวนการที่ 6.3.3.4.5.8

ชื่อกระบวนการ Divide Routine\*

คำอธิบาย เป็นการดำเนินการในโปรแกรมย่อย Divide

## จากรูปที่ 4.20

กระบวนการที่ 6.3.3.4.6.1

ชื่อกระบวนการ Get Table name and Attr. list

คำอธิบาย เป็นการรับชื่อตารางข้อมูล และชื่อแอตทริบิวต์ที่ต้องการทั้งหมดเพื่อนำไปใช้ในการดำเนินคำสั่งปฏิบัติการ Sequence

กระบวนการที่ 6.3.3.4.6.2

ชื่อกระบวนการ Check Status and Search Result name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบว่ามีการเรียกใช้ฐานข้อมูลหรือไม่ และตรวจสอบชื่อตารางข้อมูลผลลัพธ์ว่ามีอยู่แล้วในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.4.6.3

ชื่อกระบวนการ Search Table name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.4.6.4

ชื่อกระบวนการ Sequence Routine\*

คำอธิบาย เป็นการดำเนินการในโปรแกรมย่อย Sequence

## จากรูปที่ 4.21

กระบวนการที่ 6.3.3.4.7.1

ชื่อกระบวนการ Get Table name

คำอธิบาย เป็นการรับชื่อตารางข้อมูลเพื่อนำไปใช้ในการดำเนินคำสั่งปฏิบัติการ Unique

กระบวนการที่ 6.3.3.4.7.2

ชื่อกระบวนการ Check Status and Search Result name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบว่ามีการเรียกใช้ฐานข้อมูลหรือไม่ และตรวจสอบชื่อตารางข้อมูลผลลัพธ์ว่ามีอยู่แล้วในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.4.7.3

ชื่อกระบวนการ Search Table name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.4.7.4  
 ชื่อกระบวนการ Unique Routine\*  
 คำอธิบาย เป็นการดำเนินการในโปรแกรมย่อย Unique

#### จากรูปที่ 4.22

กระบวนการที่ 6.3.3.5.2.1  
 ชื่อกระบวนการ Get Case of function and Table name  
 คำอธิบาย เป็นการรับฟังก์ชันที่ต้องการคำนวณ และชื่อตารางข้อมูลเพื่อนำไปใช้ปฏิบัติการในรูปฟังก์ชันจากทุกแถวข้อมูลของตารางข้อมูล

กระบวนการที่ 6.3.3.5.2.2  
 ชื่อกระบวนการ Get Attr. name  
 คำอธิบาย รับชื่อแอตทริบิวต์ที่ใช้ในการหาฟังก์ชัน

กระบวนการที่ 6.3.3.5.2.3  
 ชื่อกระบวนการ Search Result name in Data Dictionary  
 คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลผลลัพธ์ว่ามีอยู่แล้วในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.5.2.4  
 ชื่อกระบวนการ Search Table name in Data Dictionary  
 คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.5.2.5  
 ชื่อกระบวนการ Check Attr. in table and Attr. Domain  
 คำอธิบาย ตรวจสอบแอตทริบิวต์ว่าอยู่ในตารางข้อมูลหรือไม่ และมีโดเมนตามที่ต้องการหรือไม่

กระบวนการที่ 6.3.3.5.2.6  
 ชื่อกระบวนการ AllFunction Routine\*  
 คำอธิบาย เป็นการดำเนินการในโปรแกรมย่อย All\_Function



### จากรูปที่ 4.23

กระบวนการที่ 6.3.3.5.3.1

ชื่อกระบวนการ Get Case of function and Table name

คำอธิบาย เป็นการรับฟังก์ชันที่ต้องการคำนวณ และชื่อตารางข้อมูลเพื่อนำไปใช้ปฏิบัติการในรูปฟังก์ชันสำหรับแต่ละกลุ่มย่อยของแถวข้อมูลของตารางข้อมูล

กระบวนการที่ 6.3.3.5.3.2

ชื่อกระบวนการ Get Attr. name and Attr. List

คำอธิบาย รับค่าแอตทริบิวต์ที่ใช้ในการหาฟังก์ชัน และแอตทริบิวต์ที่ใช้ในการแบ่งกลุ่มแถวข้อมูลในตารางข้อมูล

กระบวนการที่ 6.3.3.5.3.3

ชื่อกระบวนการ Search Result name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลผลลัพธ์ว่ามีอยู่แล้วในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.5.3.4

ชื่อกระบวนการ Search Table name in Data Dictionary

คำอธิบาย เป็นการตรวจสอบชื่อตารางข้อมูลว่าอยู่ในฐานข้อมูลที่เรียกใช้หรือไม่

กระบวนการที่ 6.3.3.5.3.5

ชื่อกระบวนการ Set F\_att point to Attr. function

คำอธิบาย กำหนดให้ตัวชี้ F\_att ชี้ไปยังแอตทริบิวต์ที่ใช้หาฟังก์ชัน

กระบวนการที่ 6.3.3.5.3.6

ชื่อกระบวนการ Make Attr. List for group

คำอธิบาย สร้างรายการโยงของแอตทริบิวต์จากแอตทริบิวต์ที่ใช้ในการแบ่งกลุ่มแถวข้อมูลในตารางข้อมูล

กระบวนการที่ 6.3.3.5.3.7

ชื่อกระบวนการ Check all Attr. in table

คำอธิบาย เป็นการตรวจสอบแอตทริบิวต์ทั้งหมดอยู่ในตารางข้อมูลหรือไม่

กระบวนการที่ 6.3.3.5.3.8

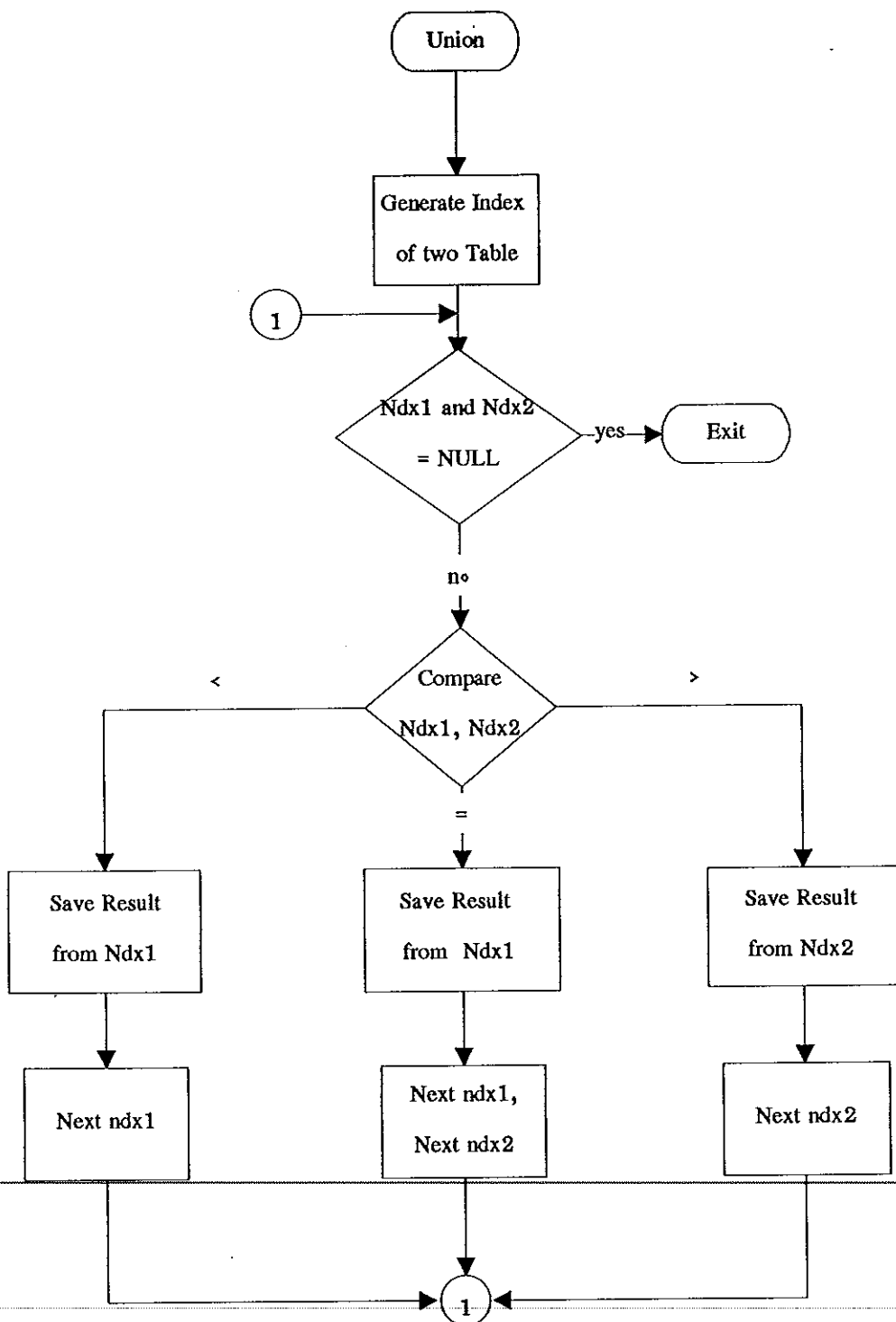
ชื่อกระบวนการ GroupFunction Routine\*

คำอธิบาย เป็นการดำเนินการในโปรแกรมย่อย Group\_Function

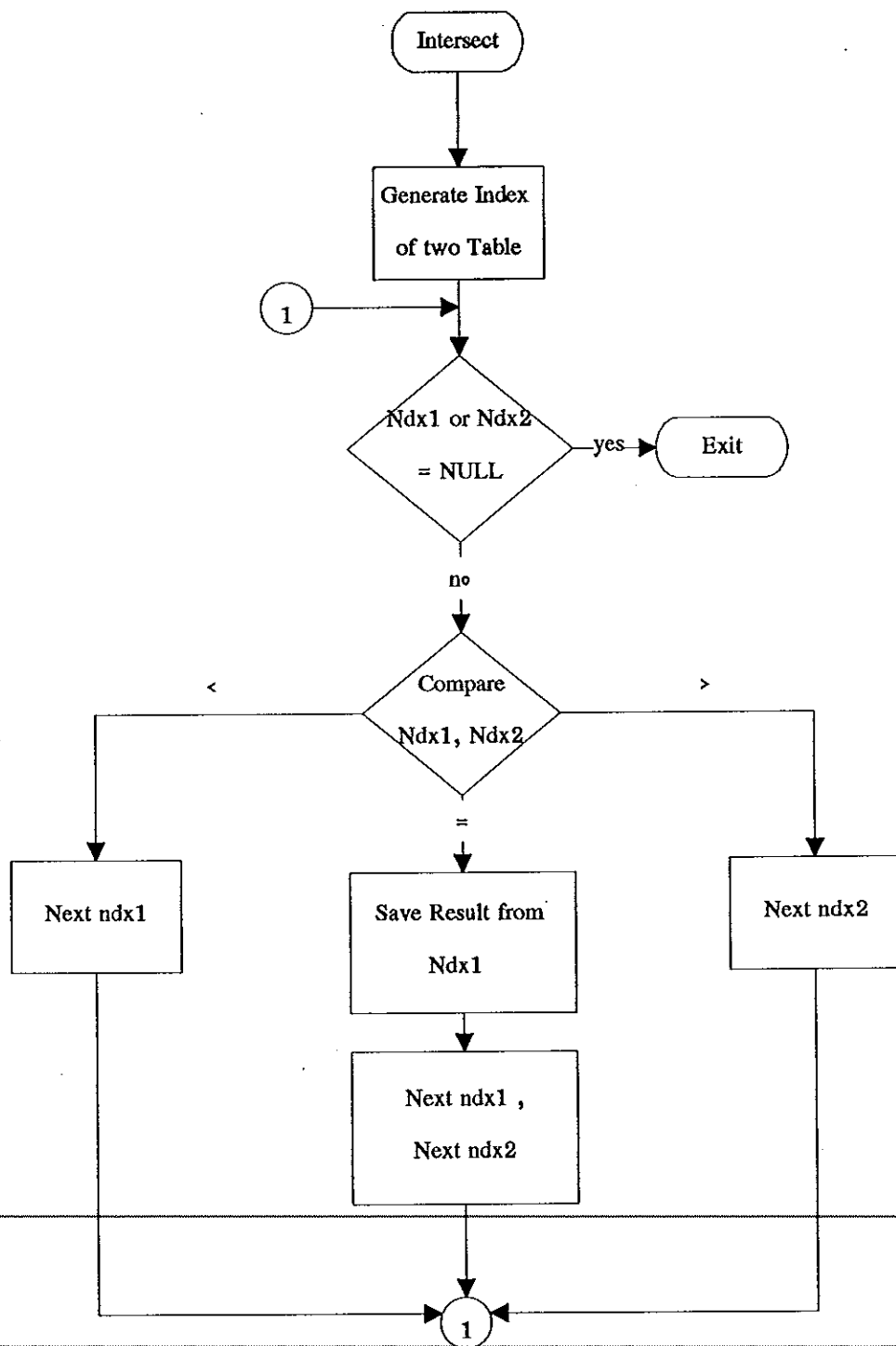
#### ผังงานการดำเนินงาน

ขั้นตอนวิธี (Algorithm) ของการดำเนินงานคำสั่งปฏิบัติการต่างๆ ในการพัฒนาโปรแกรมอย่างคร่าวๆ แสดงด้วย ผังงาน (Flow Chart) ดังรูปที่ 4.24 ถึง รูปที่ 4.35 โดย

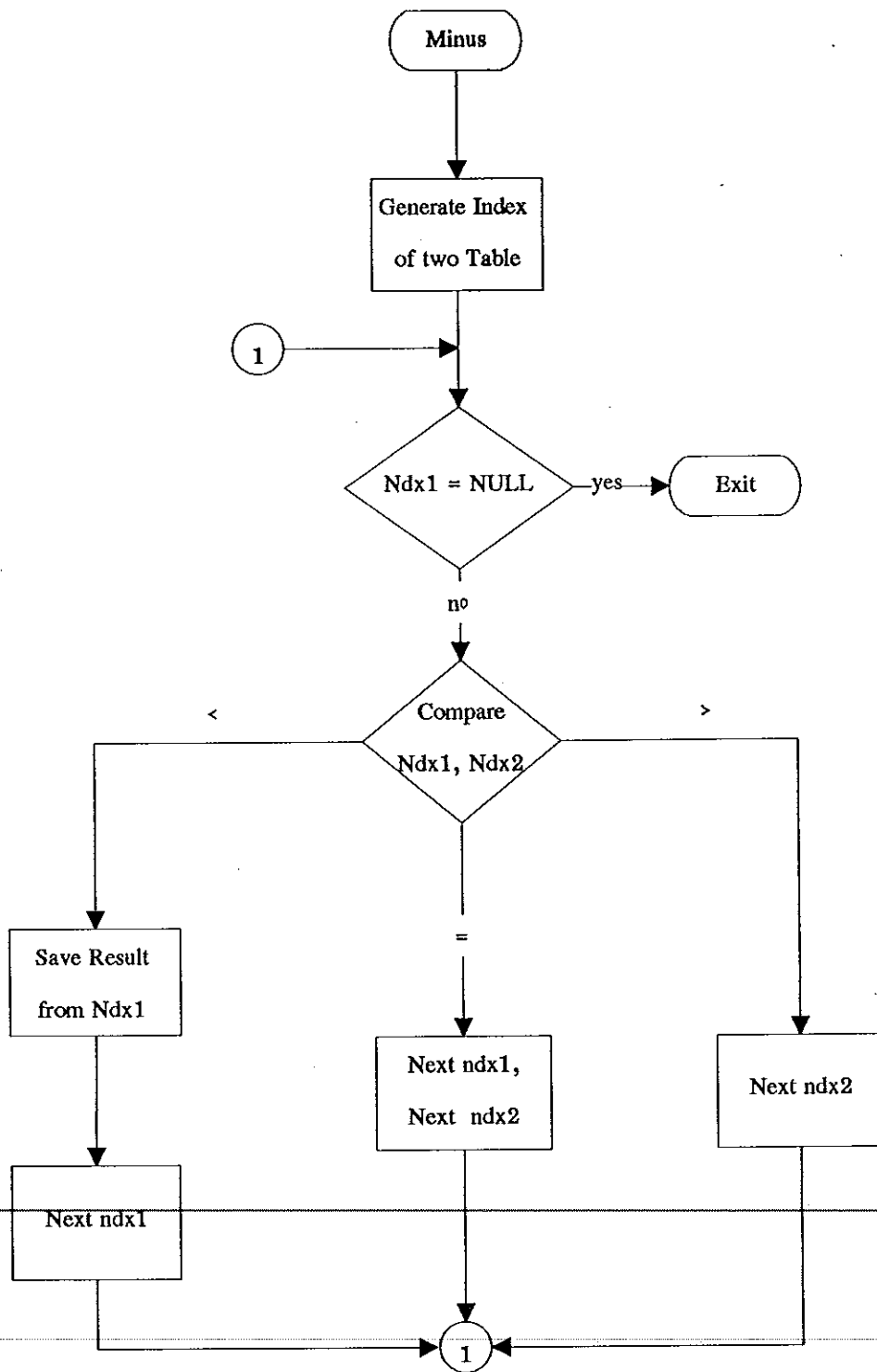
- Node เป็นตัวชี้ ที่ชี้ไปยังโหนดแรกของรายการโยงของทุเปิล
- Next node เป็นการย้ายตัวชี้ Node ไปยังโหนดถัดไปของรายการโยงของทุเปิลตามลำดับ
- Ndx1, Ndx2 เป็นตัวชี้ ที่ชี้ไปยังโหนดแรกของต้นไม้ของทุเปิลที่หนึ่ง และต้นไม้ของทุเปิลที่สองตามลำดับ
- Next ndx1, Next ndx2 เป็นการย้ายตัวชี้ Ndx1 และ Ndx2 ไปยังโหนดถัดไปของต้นไม้ของทุเปิลตามลำดับ
- Attr เป็นตัวชี้ ที่ชี้ไปยังโหนดแรกของรายการโยงของแอตทริบิวต์
- Next attr เป็นการย้ายตัวชี้ Attr ไปยังโหนดถัดไปของรายการโยงของแอตทริบิวต์ตามลำดับ



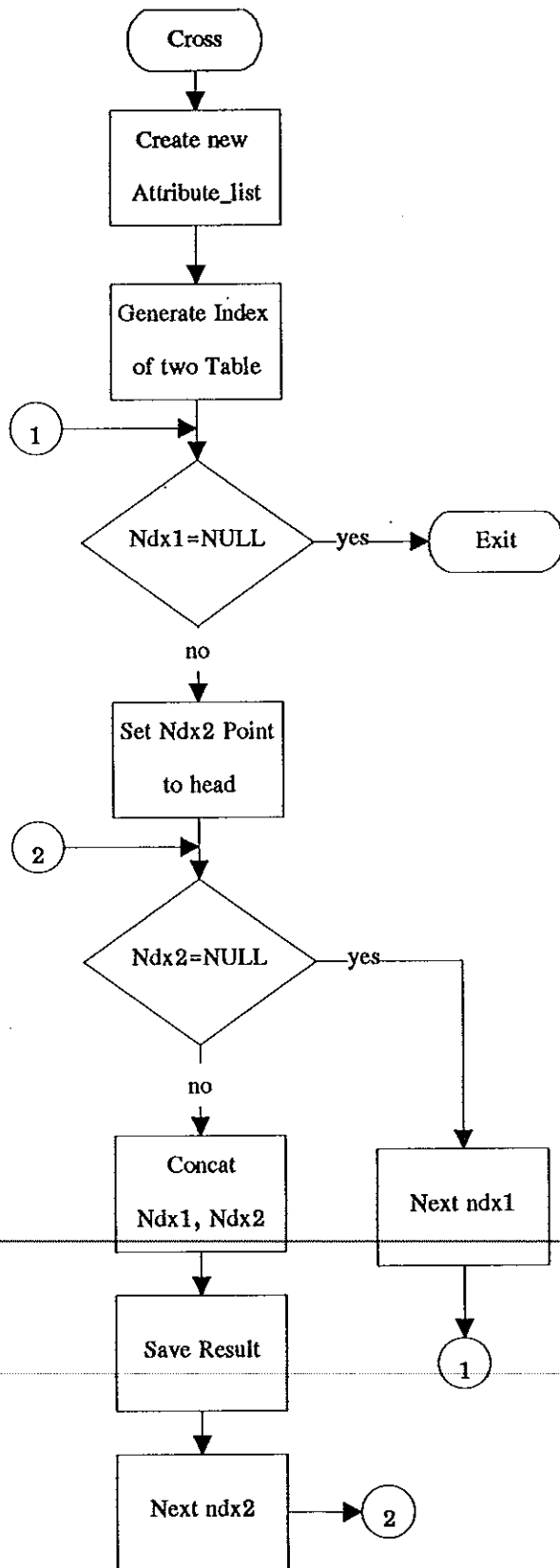
รูปที่ 4.24 ผังงานของโปรแกรมย่อย Union



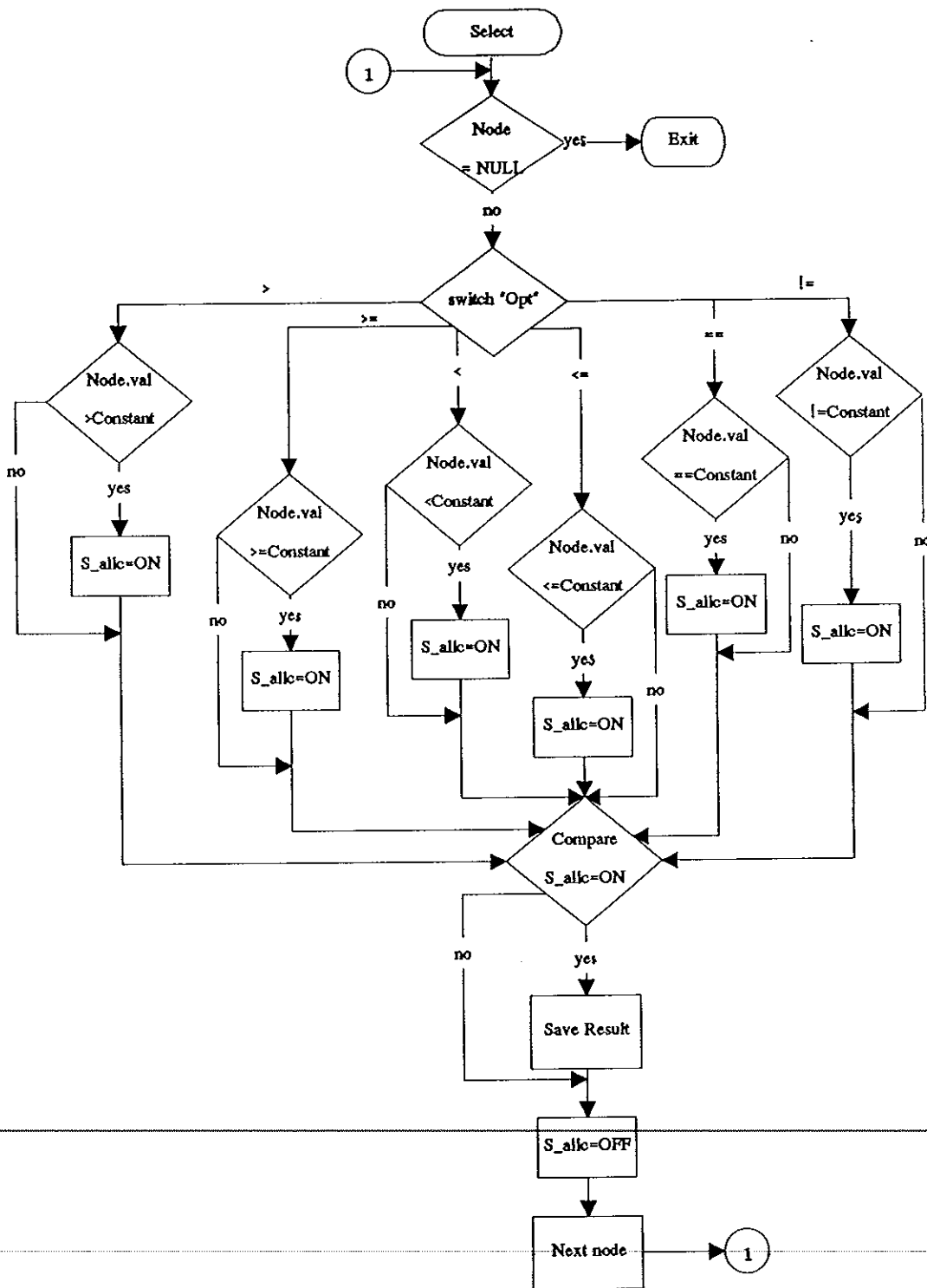
รูปที่ 4.25 ผังงานของโปรแกรมย่อย Intersect



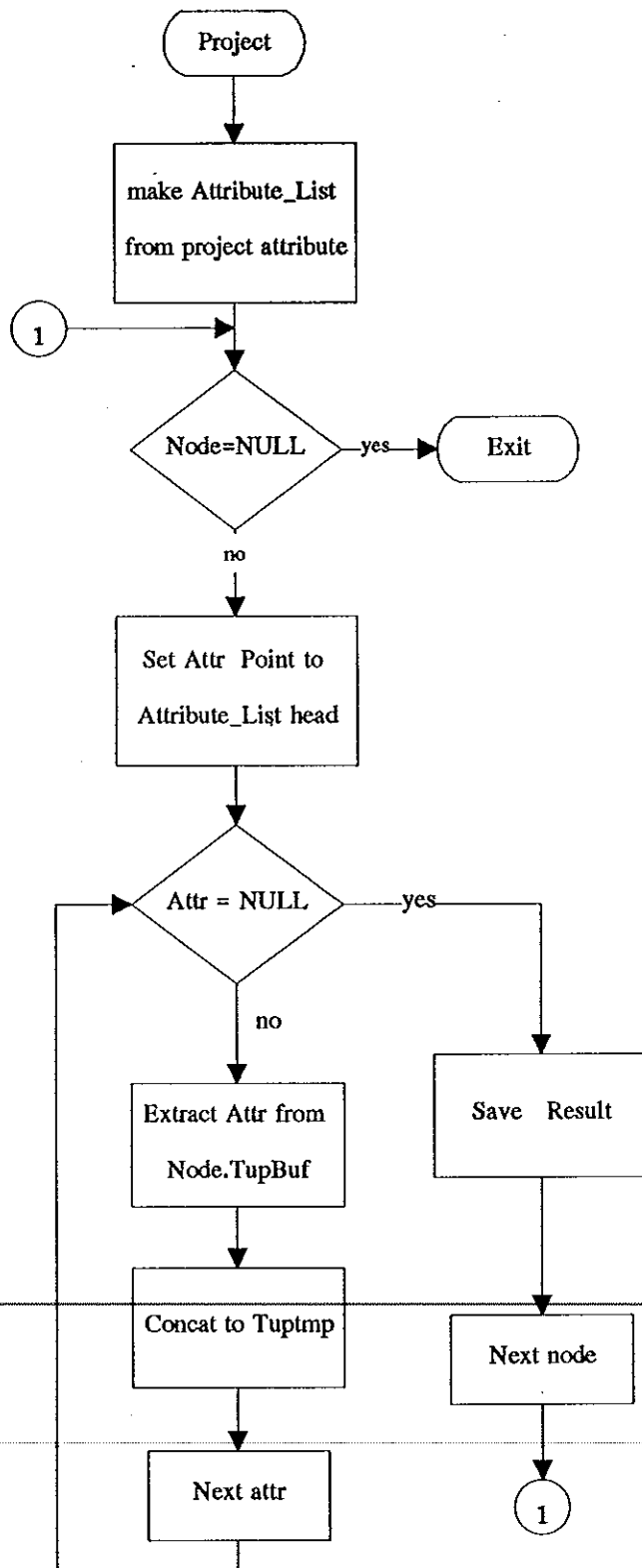
รูปที่ 4.26 ผังงานของโปรแกรมย่อย Minus



รูปที่ 4.27 ฟังก์ชันของโปรแกรมย่อย Cross

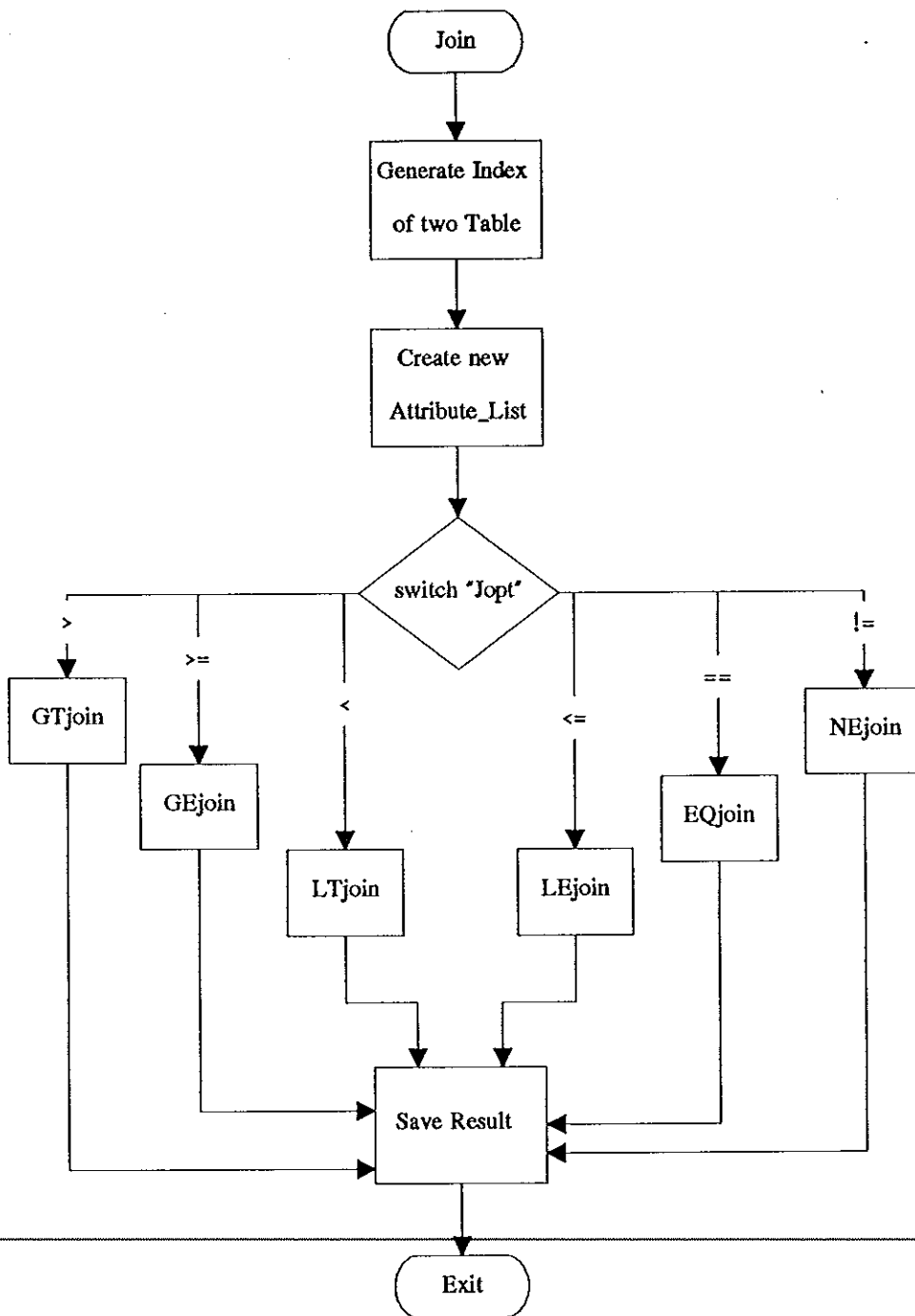


รูปที่ 4.28 ผังงานของโปรแกรมย่อย Select

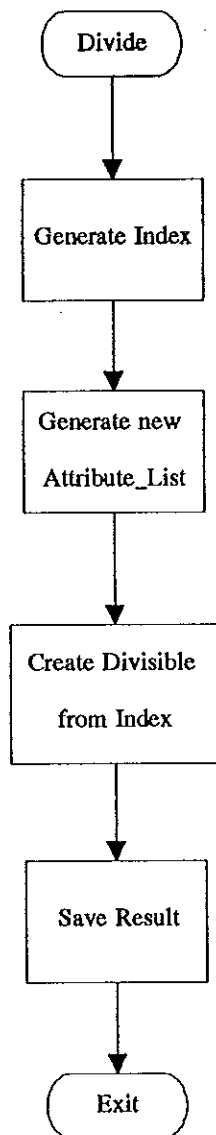


รูปที่ 4.29 ผังงานของโปรแกรมย่อย Project

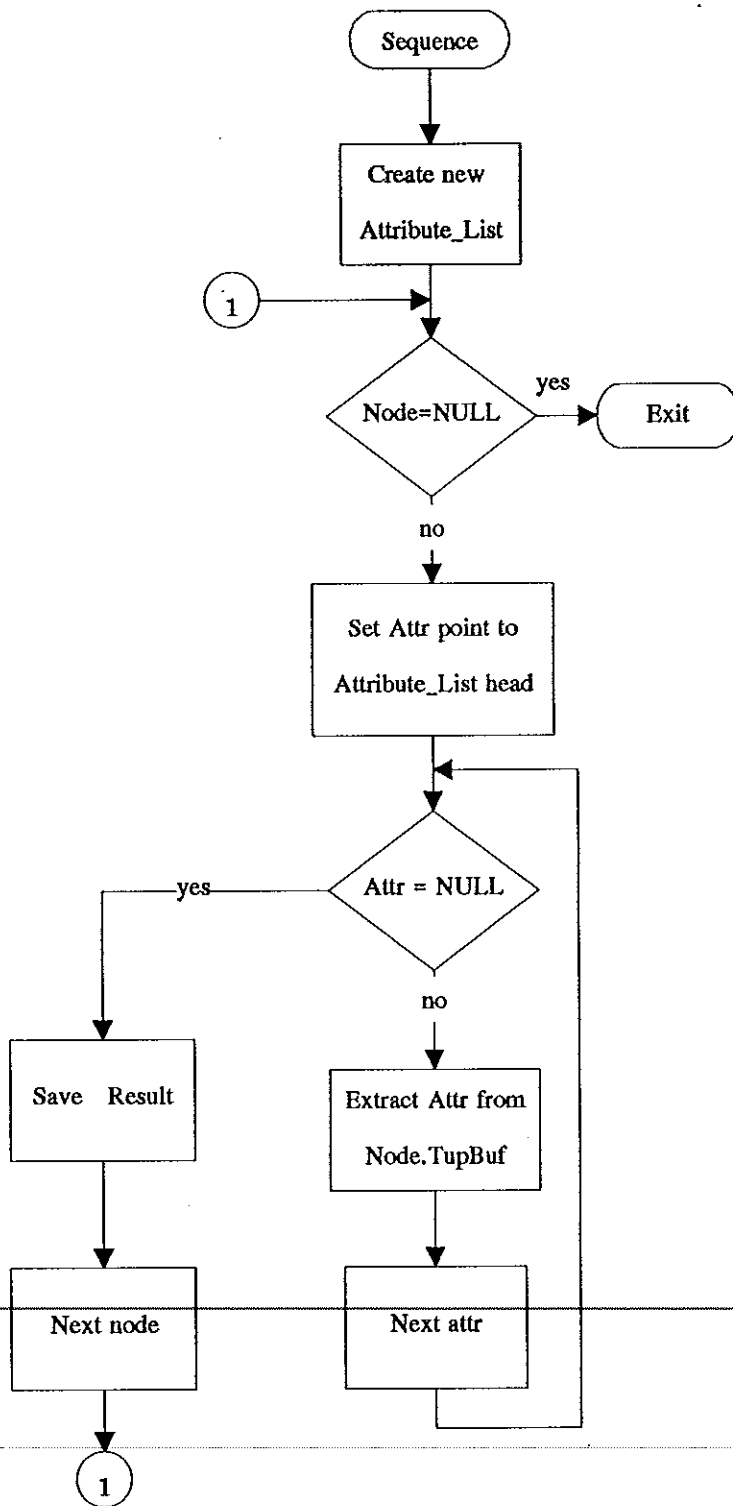




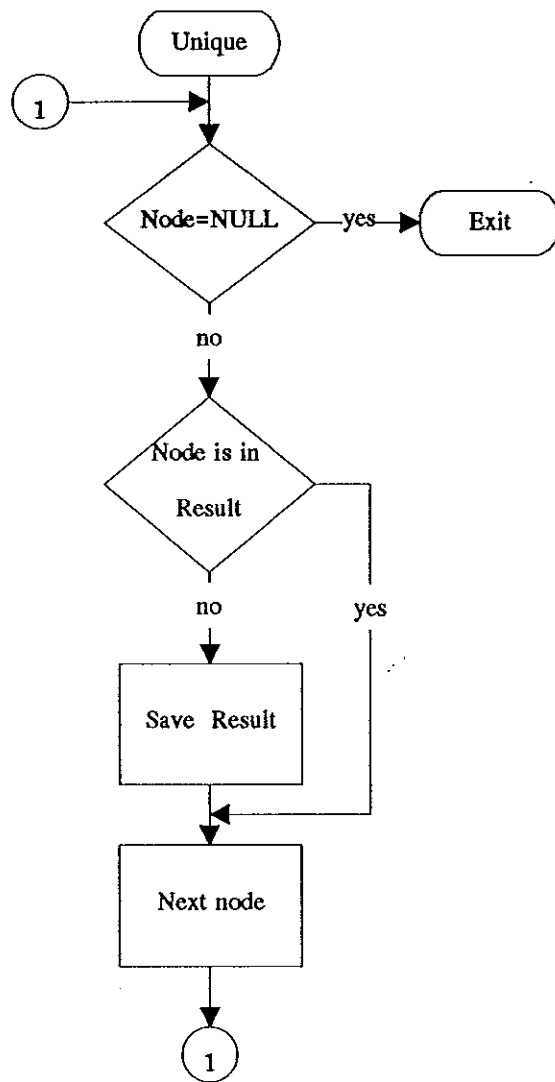
รูปที่ 4.30 ผังงานของโปรแกรมย่อย Join



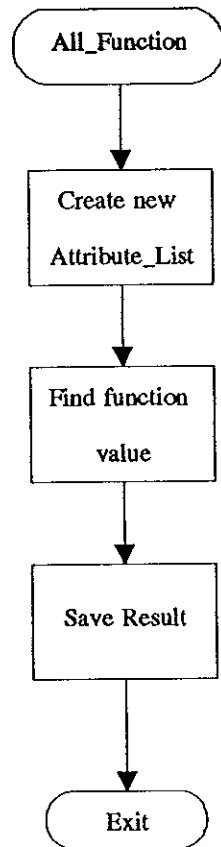
รูปที่ 4.31 ผังงานของโปรแกรมโดย Divide



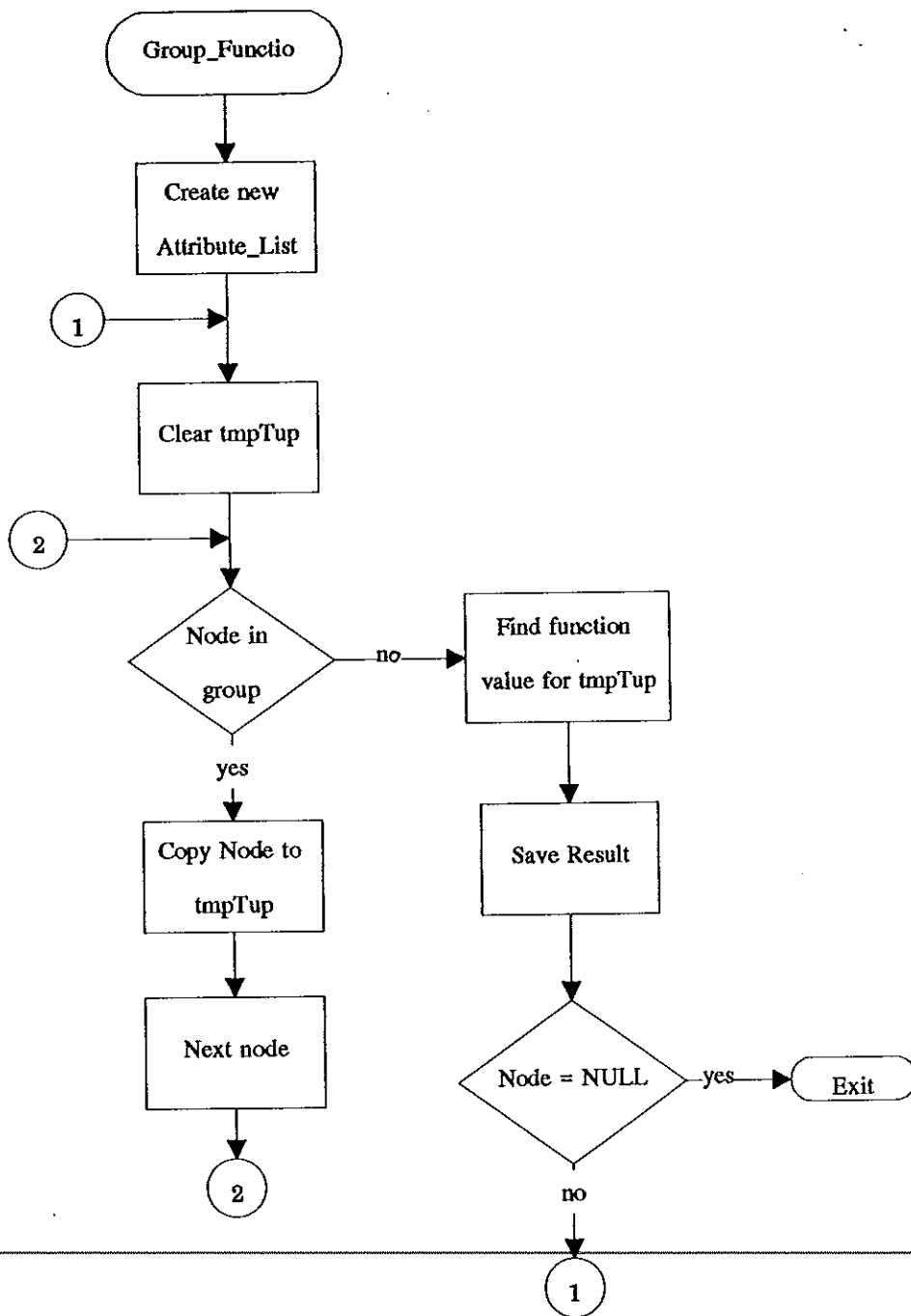
รูปที่ 4.32 ผังงานของโปรแกรมย่อย Sequence



รูปที่ 4.33 ฟังก์ชันของโปรแกรมย่อย Unique



รูปที่ 4.34 ผังงานของโปรแกรมย่อย All\_Function



รูปที่ 4.35 ผังงานของโปรแกรมย่อย Group\_Function

## บทที่ 5

### สรุป และข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

งานวิทยานิพนธ์นี้ได้ศึกษา และสร้างตัวแปลภาษาสอบถามฐานข้อมูลสัมพันธ์แบบพีชคณิต และได้ทำการย้ายโปรแกรมการดำเนินงานบางส่วนจากงานวิทยานิพนธ์ที่ได้ดำเนินการมาก่อนหน้านี้ [ทวีศักดิ์ (2534) และ เพียรพร (2534)] บนเครื่องคอมพิวเตอร์ VAX 11/785 มาแก้ไขปรับปรุง และเพิ่มเติม ให้สามารถทำงานได้บนเครื่องคอมพิวเตอร์ SUN SPARC IPX และสามารถเชื่อมโยงกับโปรแกรมแปลภาษาสอบถามฐานข้อมูลพีชคณิตสัมพันธ์ที่พัฒนา จากการทําวิจัยนี้ สามารถสรุปผลได้ดังนี้

- 1) ได้โปรแกรมแปลภาษาสอบถามฐานข้อมูลพีชคณิตสัมพันธ์ในลักษณะที่แต่ละข้อความสั่งการดำเนินงานอยู่ในรูปแถวคำสั่ง ซึ่งเป็นส่วนที่จำเป็นของระบบจัดการฐานข้อมูลในการเรียกใช้ข้อมูลในฐานข้อมูล
- 2) สามารถเชื่อมโยงโปรแกรมจากงานวิจัยที่ได้ดำเนินการมาก่อนหน้านี้กับโปรแกรมตัวแปลภาษาสอบถามฐานข้อมูลพีชคณิตสัมพันธ์ ERAO ให้สามารถดำเนินงานร่วมกันบนเครื่องคอมพิวเตอร์ SUN SPARC IPX ภายใต้ระบบปฏิบัติการ SunOS รุ่น 4.1.1 ได้
- 3) สามารถใช้โปรแกรมที่ได้เป็นโปรแกรมร่วมในการพัฒนาส่วนอื่น ๆ ของระบบจัดการฐานข้อมูล
- 4) สามารถนำโปรแกรมที่ได้ไปใช้ประกอบการสอนในเนื้อหาบางส่วนของวิชาระบบจัดการฐานข้อมูล

## 5.2 อุปสรรคและปัญหา

ปัญหาที่พบในงานวิทยานิพนธ์นี้ เนื่องจากงานวิทยานิพนธ์นี้เป็นงานต่อเนื่องจากงานวิทยานิพนธ์ที่ได้ดำเนินการมาก่อนหน้านี้โดยนำบางส่วนมาดำเนินการบนเครื่องคอมพิวเตอร์ SUN SPARC IPX ภายใต้ระบบปฏิบัติการ SunOS ดังนั้นจึงต้องใช้เวลาในการแก้ไขปรับปรุงงานวิทยานิพนธ์ที่ได้ดำเนินการมาก่อนหน้านี้ เพื่อสามารถดำเนินการได้บนเครื่องคอมพิวเตอร์ SUN SPARC IPX และเนื่องจากผู้วิจัยเองไม่เคยใช้เครื่องคอมพิวเตอร์ SUN SPARC IPX มาก่อน จึงยังไม่มี ความเข้าใจเกี่ยวกับภาษา และเครื่องมือที่ใช้ดีพอทำให้การพัฒนาโปรแกรมล่าช้า และไม่มีประสิทธิภาพมากเท่าที่ต้องการ นอกจากนี้ยังพบว่างานวิทยานิพนธ์เดิมใช้โปรแกรมช่วยจัดการเกี่ยวกับจอภาพชื่อ ncurses ซึ่งโปรแกรมดังกล่าวไม่มีใช้บนเครื่องคอมพิวเตอร์ SUN SPARC IPX แต่จะมีโปรแกรมที่มีการทำงานลักษณะเดียวกันคือ curses ซึ่งจะมีบางฟังก์ชันที่พบใน ncurses แต่ไม่พบใน curses และเนื่องจากตัวแปลภาษาซีบนเครื่องคอมพิวเตอร์ SUN SPARC IPX ของภาควิชาคณิตศาสตร์เป็น Classic C ซึ่งมีขีดความสามารถต่ำกว่า Standard C ที่ใช้ในงานวิทยานิพนธ์ที่ได้ดำเนินการมาก่อนหน้านี้ ดังนั้นจึงจำเป็นต้องแก้ไขโปรแกรมของงานวิทยานิพนธ์ที่ดำเนินการมาก่อนหน้านี้ให้สามารถทำงานได้บนเครื่องคอมพิวเตอร์ SUN SPARC IPX

## 5.3 ข้อเสนอแนะ

โปรแกรมที่ได้พัฒนาในงานวิทยานิพนธ์นี้มีวัตถุประสงค์เพื่อใช้ในการศึกษา ดังนั้นประสิทธิภาพในเรื่องของความเร็วในการทำงานต่าง ๆ จึงมิได้มีการคำนึงถึงมากนัก ผู้ที่มีความสนใจสามารถนำโปรแกรมย่อยหลัก ๆ ไปพัฒนาภาษาสอบถามฐานข้อมูลเชิงสัมพันธ์ในรูปแบบอื่น ๆ ต่อไป



## เอกสารอ้างอิง

จรมิต แก้วกัจจาล (2536) , การออกแบบและจัดการฐานข้อมูล,  
บริษัทซีเอ็ดยูเอ็ดชั่น จำกัด.

ดวงแก้ว สวามิกักดี (2534) , ระบบฐานข้อมูล, บริษัทซีเอ็ดยูเอ็ดชั่น จำกัด.

ทวีศักดิ์ ศิริฐานนท์ (2534) , การพัฒนาระบบจัดการฐานข้อมูลเชิงสัมพันธ์  
ส่วนภาษาฐานข้อมูล, วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต,  
มหาวิทยาลัยสงขลานครินทร์.

พงษ์พันธ์ พิณโท (2536) , การสร้างตัวแปลภาษาสอบถามฐานข้อมูล QBE,  
วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต, มหาวิทยาลัยสงขลานครินทร์.

เพ็ชรพร พรหมโชติ (2534) , โปรแกรมส่วนการกำหนดโครงสร้างฐานข้อมูล  
และการดำเนินงานพื้นฐาน, วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต,  
มหาวิทยาลัยสงขลานครินทร์.

อิ้ว ไอยรากาญจนกุล (2534) , การเขียนคอมไพเลอร์สำหรับ IBM PC,  
บริษัทซีเอ็ดยูเอ็ดชั่น จำกัด.

Alfred V. Aho, Jeffrey D. Ullman (1979) , *Principles of Compiler Design*,  
3<sup>rd</sup> edition, Addison-Wesley Publishing Company.

---

~~Alfred V. Aho, Ravi Sethi and Jeffrey D. Ullman (1986) , *Compilers Principles,  
Techniques, and Tools*, Addison-Wesley Publishing Company.~~

---

Date, C.J. (1989) , *A Guide to the SQL standard*, 5<sup>th</sup> edition vol.1,  
Addison-Wesley Publishing Company, London.

Date, C.J. (1989) , *An Introduction to Database System*, 5<sup>th</sup> edition vol.1,  
Addison-Wesley Publishing Company, London.

Elmasri, Ramez and Navathe, Sinankant B. (1989) , Fundamentals of Database Systems, Benjamin/Cummings Publishing.

Fliming, Candace C. (1989) , Handbook of Relational Database Design, Addison-Wesley Publishing Company.

George Koch (1990) , ORACLE : The Complete Reference, McGraw-Hill, Inc.

Horspool, R.N. (1986) , C Programming in the Berkeley UNIX Environment, Prentice-Hall.

Ken McDonell (1983) , Instructional Relational Algebra (IRA) User's Guide, Computer Science Department, Monash University, Clayton, Victoria, Australia.

Sun Microsystems Incorporated (1990) , Programmer's Overview Utilities&Libraries, Sun Microsystems.

Sun Microsystems Incorporated (1990) , SunOS Reference Manual, Sun Microsystems.

Theerachetmongkol, A. and Montgomery, A.Y., Semantic Integrity Constraints in the Query by Example Data Base Management Language, The Australian Computer Journal, Vol. 12, No. 1, February.

---

ภาคผนวก ก

---

---

## แผนภาพกระแสข้อมูล

แผนภาพกระแสข้อมูล (DFD:Data Flow Diagram) ซึ่งเป็นแผนภาพที่ใช้แสดงการส่งถ่ายข้อมูลที่เกิดขึ้นระหว่างกระบวนการในระบบ ทำให้ทราบถึงขั้นตอนของกระบวนการต่าง ๆ ว่ามีการส่งถ่ายข้อมูลอย่างไร ซึ่งเป็นการสะดวกในการทำความเข้าใจการดำเนินงานของระบบหนึ่ง ๆ ก่อนที่จะลงรายละเอียดต่าง ๆ ในการเขียน DFD นี้ผู้วิจัยได้ใช้โปรแกรมสำเร็จรูปที่ชื่อ ABC Flocharter version 1.2 ซึ่งทำงานภายใต้โปรแกรมสำเร็จรูป Microsoft Windows version 3.1 ช่วยในการเขียน โปรแกรมสำเร็จรูปนี้ดำเนินงานกับเครื่องไมโครคอมพิวเตอร์ขนาด 16 บิต สัญลักษณ์แสดงความหมายที่ปรากฏในแผนภาพกระแสข้อมูลมีดังนี้



แทนกระบวนการ (Process) ในการดำเนินงานตามข้อความที่ปรากฏภายในกรอบ



แทนกระบวนการในการดำเนินงานซึ่งมีขั้นตอนย่อย จะแสดงหรืออธิบายเพิ่มเติมอีก



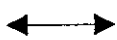
แทนแฟ้มข้อมูล ในที่นี้หมายถึงข้อมูลที่ถูกบันทึกลงสื่อบันทึกข้อมูลต่าง ๆ เช่น แผ่นดิสเก็ต เทปแม่เหล็ก จานแม่เหล็ก เป็นต้น



แทนส่วนการดำเนินงานนอกระบบที่ไม่รวมอยู่ในการออกแบบ



แทนกระแสข้อมูลที่มีการส่งผ่านข้อมูลในทิศทางเดียว



แทนกระแสข้อมูลที่มีการส่งผ่านข้อมูลในทิศทางทั้งไปและกลับ




ภาคผนวก ข

---

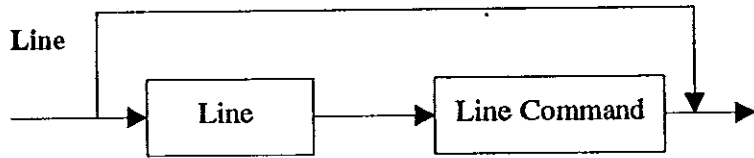
---

## แผนภาพวากยสัมพันธ์ของตัวแปลภาษา ERAO

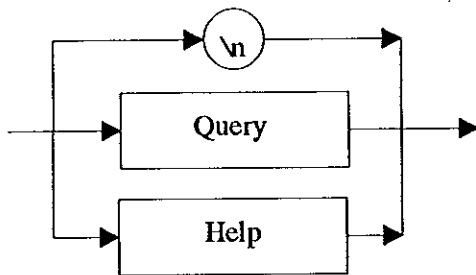
ในการอธิบายวากยสัมพันธ์ของตัวแปลภาษา ERAO ได้ใช้ แผนภาพวากยสัมพันธ์ (Syntax Diagram) ในการอธิบาย แผนภาพแสดงด้วยรูปภาพที่มีสัญลักษณ์ที่ถูกแทนที่ได้ หรือสัญลักษณ์ที่ถูกแทนที่ไม่ได้กำกับไว้ แต่ละรูปภาพจะมีลูกศรเข้าทางเดียว และออกทางเดียวเสมอ มีกฎเกณฑ์ ดังนี้

- 1) สิ่งใดๆ ที่ปรากฏในรูป  หรือ  จะต้องเขียนสิ่งนั้นในแถวคำสั่งเพื่อ สอบถามเหมือนที่ปรากฏในแผนภาพ โดยที่อักขระตัวใหญ่ หรือตัวเล็กไม่แตกต่างกัน
- 2) ทุกข้อความที่ปรากฏในรูป  เป็นสัญลักษณ์ที่ถูกแทนที่ได้
- 3) Identifier ในแผนภาพจะคือชื่อของตารางข้อมูล Attribute-name จะคือชื่อแอตทริบิวต์ และ Constant จะคือค่าคงที่

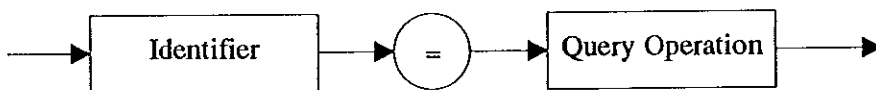
**หมายเหตุ** สามารถแทน Attribute-name ในแผนภาพ Function Parameter ด้วยสัญลักษณ์ “\*” ในกรณีนี้ Build-in Function คือ CNT

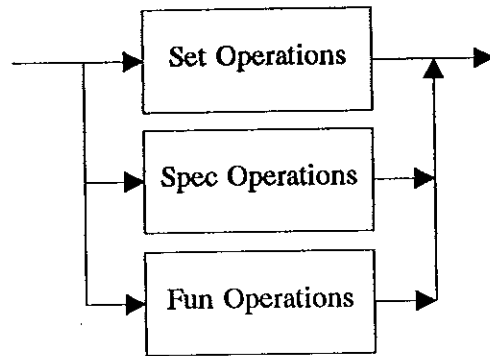
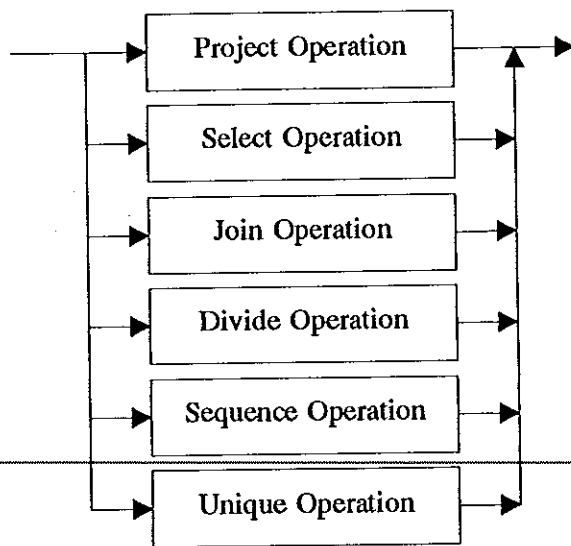


**Line Command**

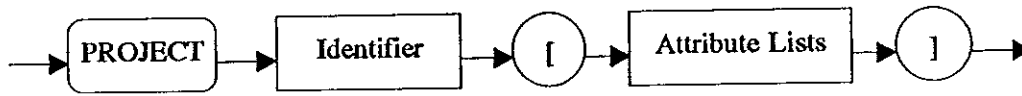
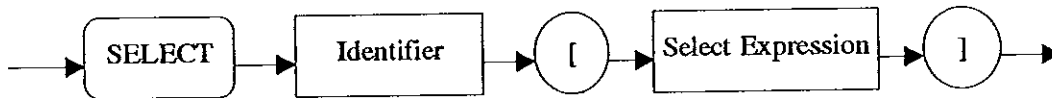
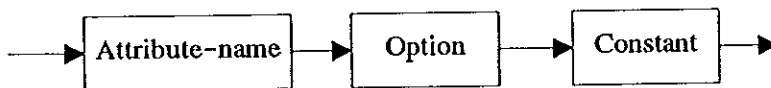
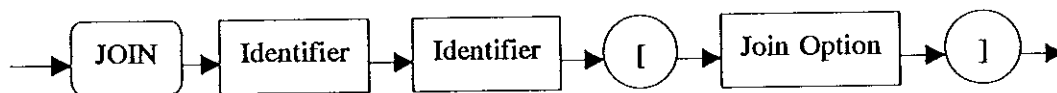
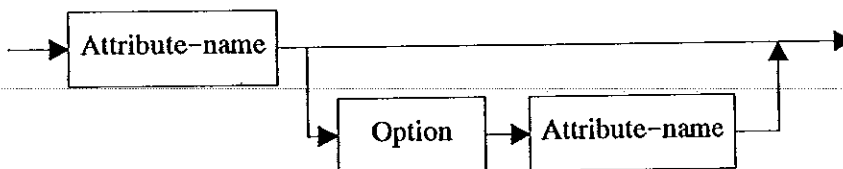


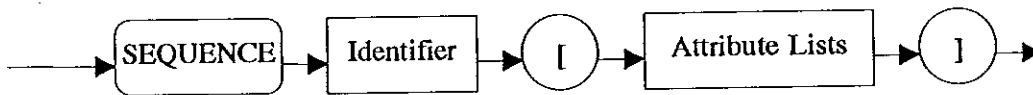
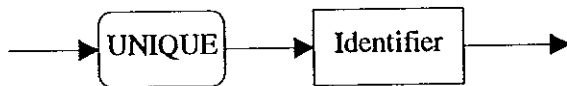
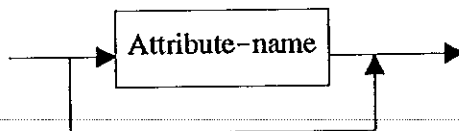
**Query**

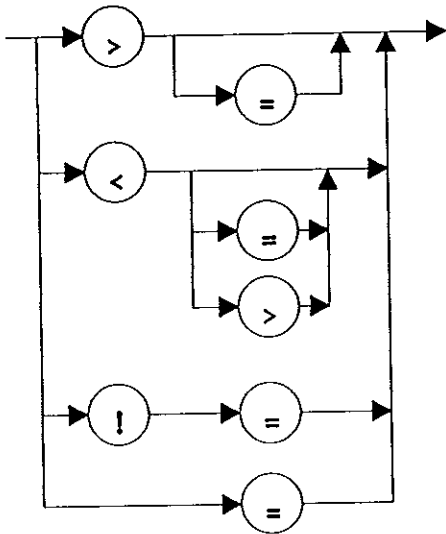
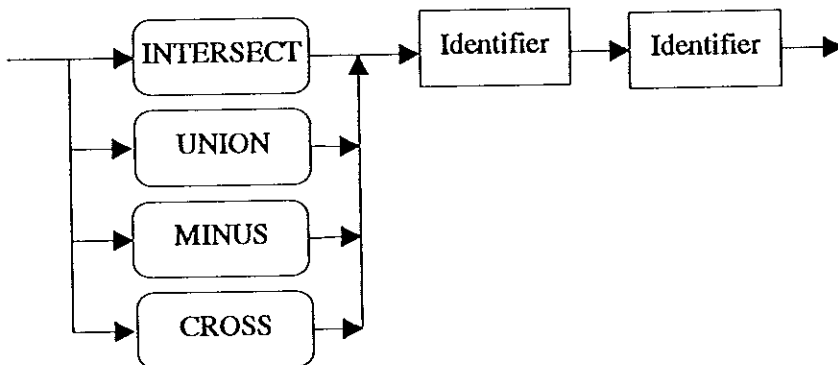
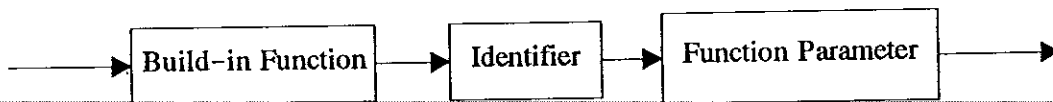


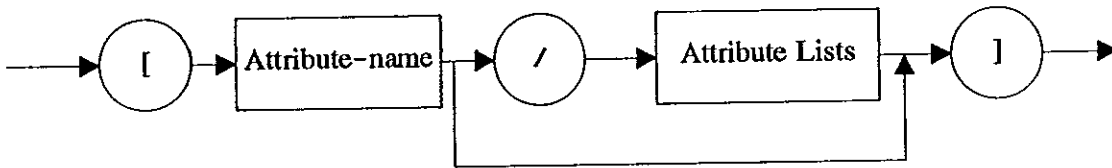
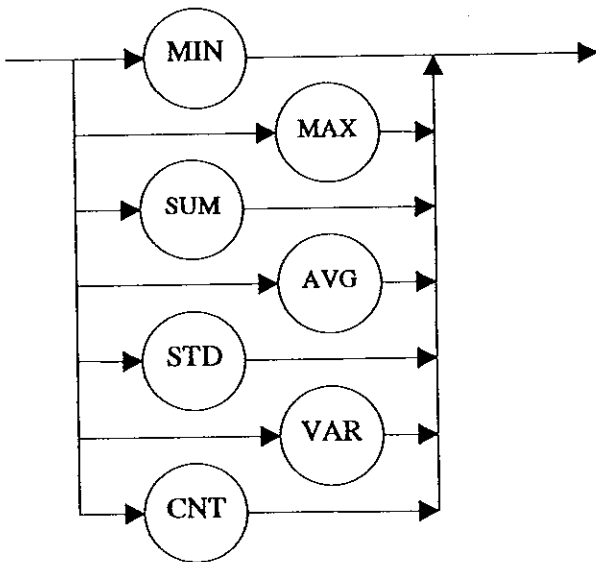
**Query Operation****Set Operations**

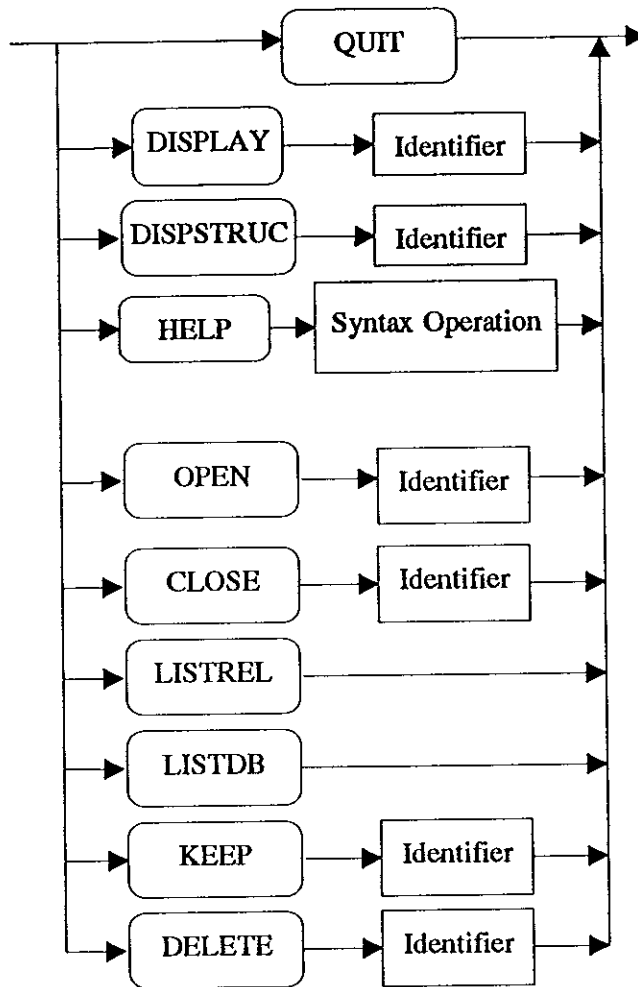


**Project Operation****Select Operation****Select Expression****Join Operation****Join**

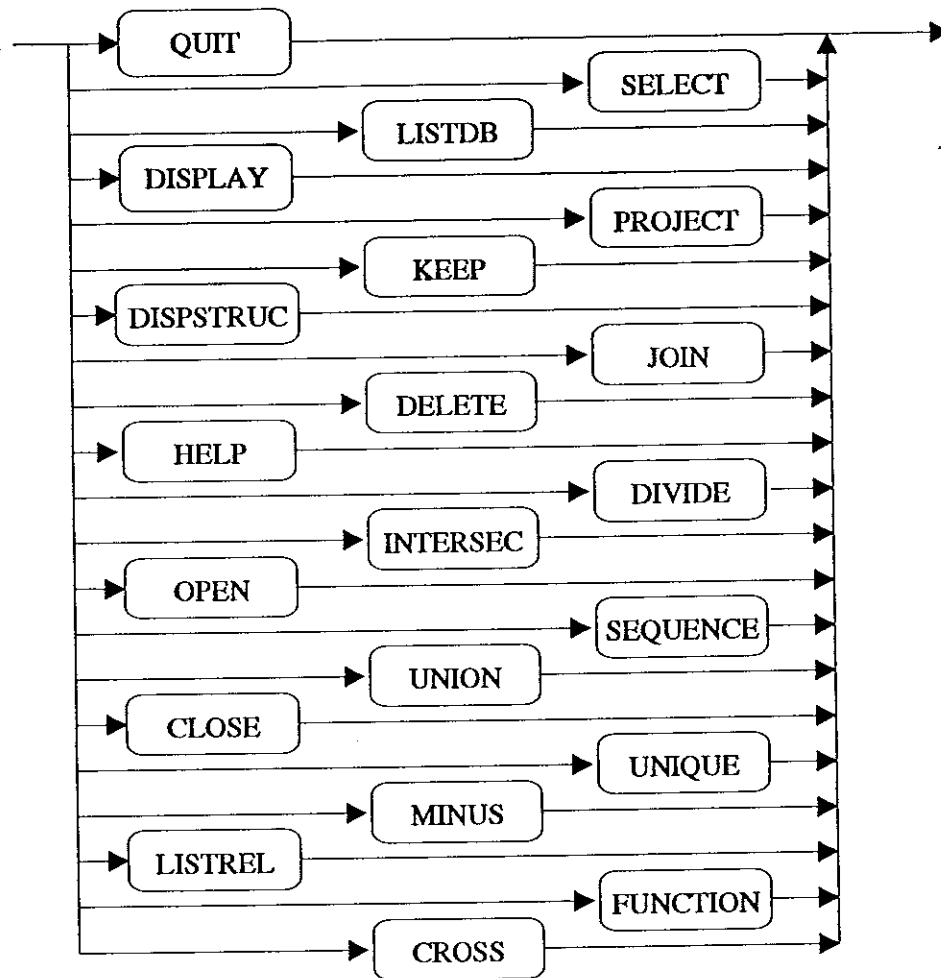
**Divide Operation****Sequence Operation****Unique Operation****Attribute Lists**

**Option****Spec Operations****Fun Operations**

**Function Parameter****Build-in Function**

**Help**

## Syntax Operatio



## ภาคผนวก ค

---

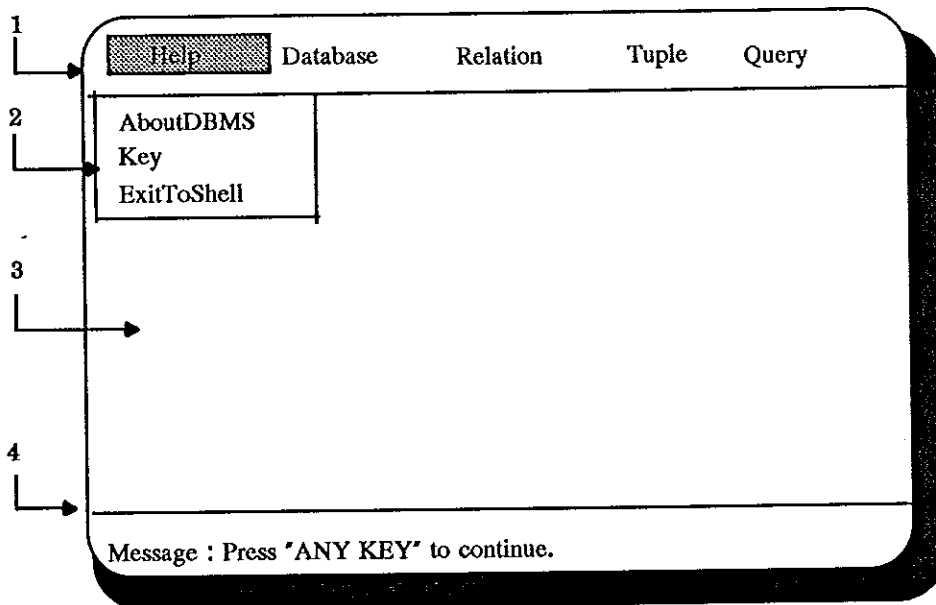
---

## เอกสารสำหรับผู้ใช้

การใช้งานระบบแบ่งออกเป็นสองส่วน ส่วนแรกคือส่วนการกำหนดโครงสร้างฐานข้อมูลและการดำเนินงานพื้นฐาน เป็นส่วนการใช้โปรแกรม Mrdbms ส่วนที่สองคือส่วนของภาษาฐานข้อมูลเป็นส่วนการใช้โปรแกรมตัวแปลภาษา ERAO

### การแสดงผลทางจอภาพ

โปรแกรม Mrdbms มีรูปแบบการทำงานที่เรียกว่า การจัดการหน้าต่าง (Window Manipulation) ลักษณะการแสดงผลทางจอภาพประกอบด้วยส่วนประกอบหลัก ดังรูปที่ ค.1



รูปที่ ค.1 ส่วนประกอบหลักของจอภาพ

ส่วนที่ 1 เป็นรายการหลัก (Main Menu) ซึ่งจะประกอบด้วยคำสั่งการทำงานทั้งหมดของโปรแกรม

ส่วนที่ 2 เป็นรายการย่อย (Sub Menu) ซึ่งจะอยู่ภายใต้รายการหลัก

ส่วนที่ 3 เป็นส่วนของการแสดงผล หรือจะใช้แสดงผลที่ได้จากการดำเนินงานต่าง ๆ

ส่วนที่ 4 เป็นส่วนที่ใช้แสดงข้อความผิดพลาด (Error Message) หรือข้อแนะนำการทำงานสำหรับผู้ใช้



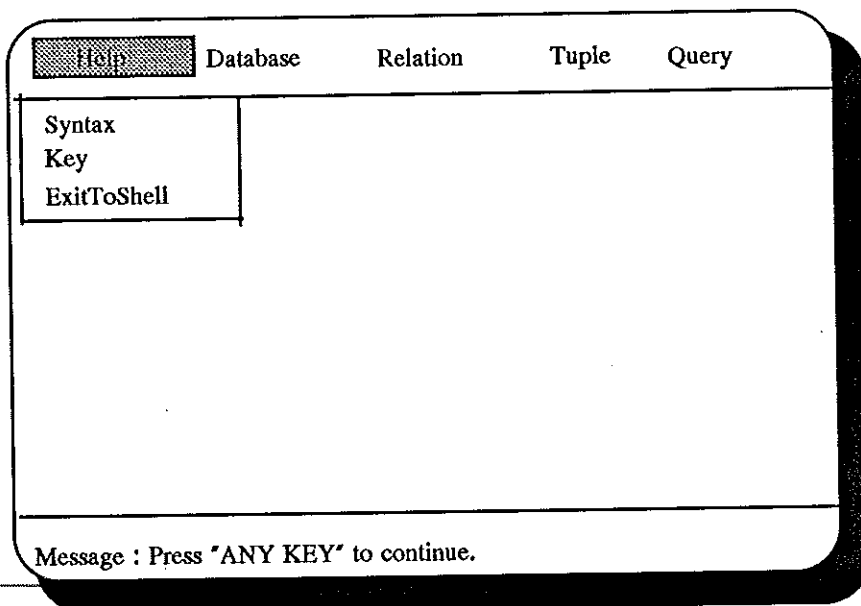
### การใช้แป้นพิมพ์ควบคุมการทำงาน

โปรแกรม Mrdbms ใช้แป้นพิมพ์ควบคุมการทำงานต่อไปนี้

แป้นพิมพ์	การทำงาน
l, L หรือ ->	เลือกรายการหลักที่อยู่ทางขวามือของรายการหลักปัจจุบัน
h, H หรือ <-	เลือกรายการหลักที่อยู่ทางซ้ายมือของรายการหลักปัจจุบัน
j, J หรือ	เลือกรายการย่อยที่อยู่ใต้รายการย่อยปัจจุบัน
k, K หรือ ^	เลือกรายการย่อยที่อยู่เหนือรายการย่อยปัจจุบัน
Enter	ยอมรับรายการหรือคำสั่งปัจจุบัน
Space bar	เปลี่ยนแปลงเงื่อนไขการทำงานของคำสั่ง
Esc	ยกเลิกการทำงานของคำสั่ง

### รายละเอียดรายการของรายการหลักและรายการย่อยของโปรแกรม Mrdbms

รายการหลัก Help มีรายการย่อยดังแสดงในรูปที่ ค.2

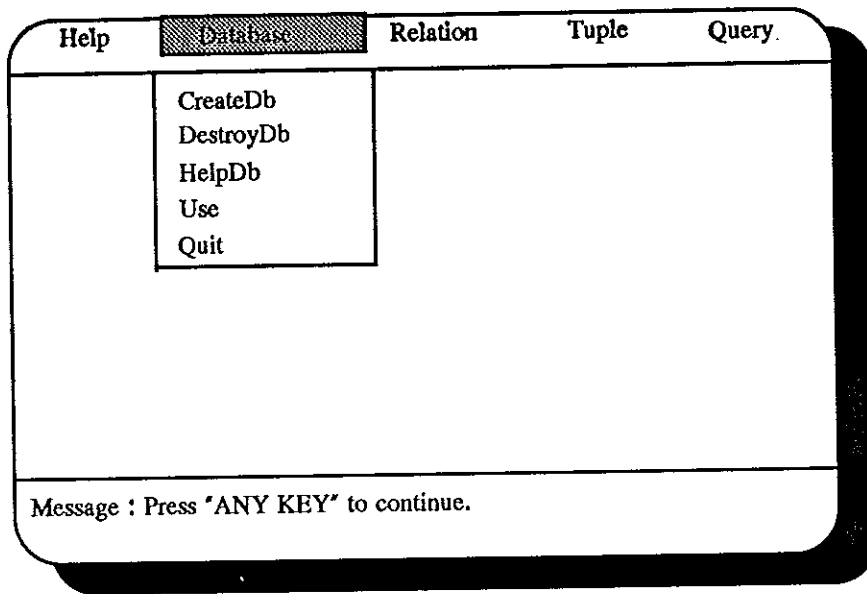


รูปที่ ค.2 ส่วนรายการหลัก Help

รายละเอียดของรายการย่อยมีดังนี้

- Syntax ขอดูรายละเอียดวากยสัมพันธ์ของแต่ละคำสั่งของโปรแกรม Mrdbms
  - Key บอกหน้าที่ของปุ่มต่าง ๆ ของแป้นพิมพ์ในการควบคุมการทำงานของโปรแกรม
  - ExitToShell ออกจากโปรแกรม Mrdbms ไปยัง Shell Command ของ SunOS
- ชั่วคราว

### รายการหลัก Database มีรายการย่อยดังแสดงในรูปที่ ค.3

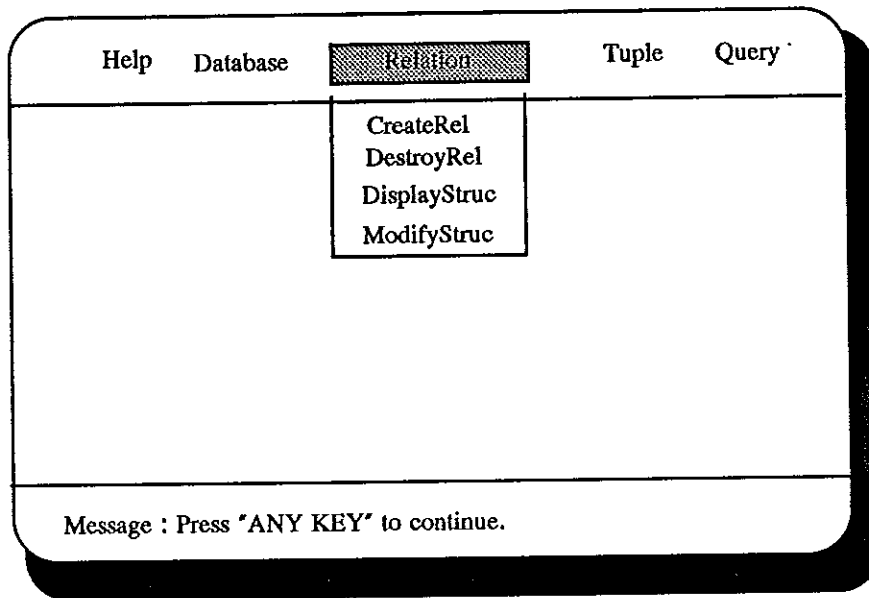


รูปที่ ค.3 ส่วนรายการหลัก Database

รายละเอียดของรายการย่อยมีดังนี้

- CreateDb สร้างฐานข้อมูลใหม่
- DestroyDb ลบฐานข้อมูล
- HelpDb แสดงรายละเอียดต่าง ๆ ของแต่ละฐานข้อมูล
- Use เรียกฐานข้อมูลที่มีอยู่แล้วขึ้นมาใช้
- Quit ออกจากโปรแกรม Mrdbms

รายการหลัก Relation มีรายการย่อยดังแสดงในรูปที่ ค.4

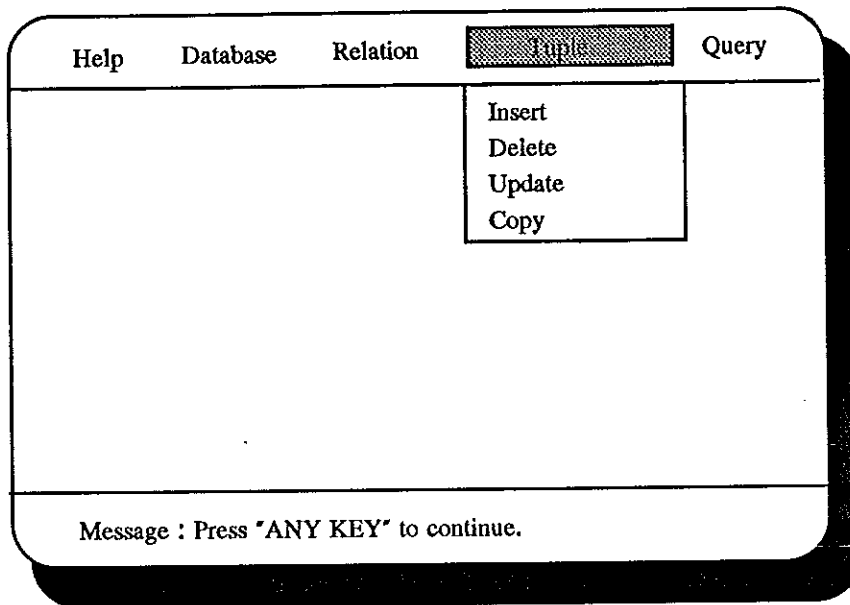


รูปที่ ค.4 ส่วนรายการหลัก Relation

รายละเอียดของรายการย่อยมีดังนี้

- CreateRel สร้างตารางข้อมูล
- DestroyRel ลบตารางข้อมูล
- DisplayStruc แสดงรายละเอียดต่าง ๆ ของแต่ละตารางข้อมูล
- ModifyStruc เปลี่ยนแปลงโครงสร้างของตารางข้อมูล

รายการหลัก Tuple มีรายการย่อยดังแสดงในรูปที่ ค.5

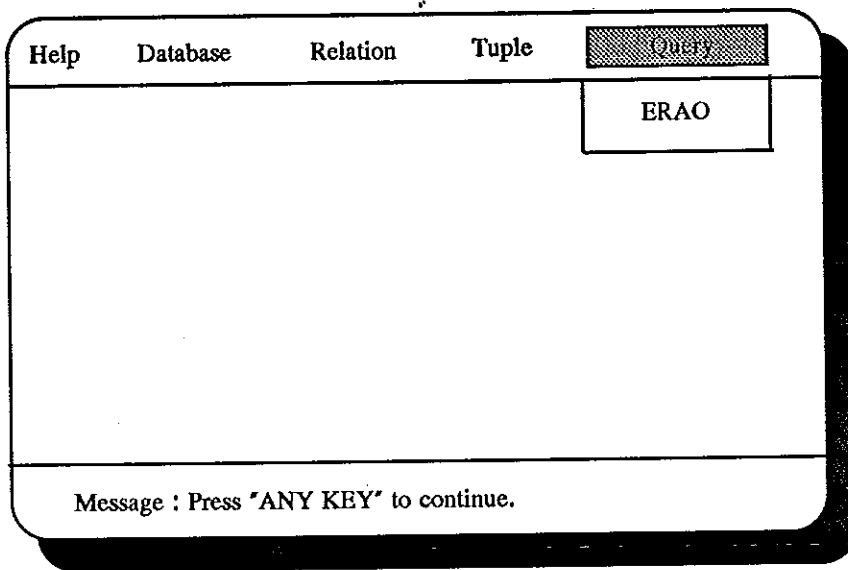


รูปที่ ค.5 ส่วนรายการหลัก Tuple

รายละเอียดของรายการย่อยมีดังนี้

- Insert เพิ่มแถวข้อมูลใหม่ให้กับตารางข้อมูล
- Delete ลบแถวข้อมูลออกจากตารางข้อมูล
- Update แก้ไขแถวข้อมูลบางส่วน
- Copy สำเนาเพิ่มข้อมูล

รายการหลัก Query มีรายการย่อยดังแสดงในรูปที่ ค.6



รูปที่ ค.6 ส่วนรายการหลัก Query

ในรายการหลัก Query จะประกอบด้วยรายการย่อยเดียวคือ ERAO ซึ่งเป็นส่วน  
ของภาษารฐานข้อมูลเพื่อเรียกใช้ตัวแปลภาษา ERAO

### การใช้โปรแกรมตัวแปลภาษา ERAO

เมื่อผู้ใช้เลือกรายการย่อย ERAO ในรายการหลัก Query การรับคำสั่งเพื่อปฏิบัติงาน จะเปลี่ยนเป็นแถวคำสั่ง โดยจะแสดงสัญลักษณ์ ERAO prompt "ERAO >" เพื่อให้ผู้ใช้ พิมพ์แถวคำสั่ง เกณฑ์การใช้โปรแกรมตัวแปลภาษา ERAO มีดังนี้

- 1) แต่ละแถวคำสั่งจะต้องอยู่ในบรรทัดเดียวกัน โปรแกรมตัวแปลภาษา ERAO รับแถวคำสั่งเมื่อผู้ใช้กดแป้นขึ้นบรรทัดใหม่ (ปุ่ม <enter>) จะแปลและประมวลผลทันที โดยผลลัพธ์ที่ได้จะแสดงทางจอภาพ
- 2) operator หรือคำสั่งปฏิบัติการภายในแถวคำสั่ง สามารถใช้เป็นอักขรภาษาอังกฤษตัวพิมพ์ใหญ่ หรือตัวพิมพ์เล็กทั้งคำ หรือมีทั้งตัวพิมพ์ใหญ่และตัวพิมพ์เล็กภายในคำ
- 3) แต่ละส่วนภายในแถวคำสั่งต้องแยกจากกันด้วยช่องว่างอย่างน้อย 1 ช่องว่างเช่น

OPEN part  
↑  
ช่องว่างอย่างน้อย 1 ช่อง

ยกเว้นส่วนที่อยู่ติดกับอักขระพิเศษอาจไม่มีช่องว่างก็ได้ เช่น

P1 = Project part[PNUM PNAME]  
↑                    ↑                    ↑  
เว้นหรือไม่เว้นช่องว่างก็ได้

- 4) ไม่นำคำสั่งวนมาตั้งเป็นชื่อของตารางข้อมูล

ต่อไปนี้เป็นตัวอย่างการใช้แถวคำสั่งปฏิบัติการ ERAO โดยใช้ตารางข้อมูลจาก  
ตัวอย่างฐานข้อมูลในรูปที่ ค.7

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

SP

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

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

SPJ

SNUM	PNUM	JNUM	QTY
S1	P1	J1	300
S1	P1	J2	200
S1	P1	J3	400
S1	P2	J1	100
S1	P2	J3	300
S1	P3	J1	200
S2	P1	J1	100
S2	P1	J2	300
S2	P2	J1	200

รูปที่ ค.7 ตัวอย่างฐานข้อมูล

**ตัวอย่าง ค.1** ตัวอย่างการใช้แฉวคำสั่งปฏิบัติการ Select พร้อมทั้งผลลัพธ์ที่ได้

ERAO >p1=select S[SNUM = S1]

Relation's name : p1

SNUM	SNAME	STATUS	CITY
=====	=====	=====	=====
S1	Smith	20	London

1 records selected

**ตัวอย่าง ค.2** ตัวอย่างการใช้แฉวคำสั่งปฏิบัติการ Project พร้อมทั้งผลลัพธ์ที่ได้

ERAO >p2=project S[SNUM SNAME]

Relation's name : p2

SNUM	SNAME
=====	=====
S1	Smith
S2	Jones
S3	Blake
S4	Clark
S5	Adams

5 records selected



ตัวอย่าง ค.3 ตัวอย่างการใช้แฉวคำสั่งปฏิบัติการ Join พร้อมทั้งผลลัพธ์ที่ได้

ERAO >p3=join S SP[SNUM]

Relation's name : p3

SNUM	SNAME	STATUS	CITY	SNUM	PNUM	QTY
=====	=====	=====	=====	=====	=====	=====
S1	Smith	20	London	S1	P1	300
S1	Smith	20	London	S1	P2	200
S1	Smith	20	London	S1	P3	400
S1	Smith	20	London	S1	P4	200
S1	Smith	20	London	S1	P5	100
S1	Smith	20	London	S1	P6	100
S2	Jones	10	Paris	S2	P1	300
S2	Jones	10	Paris	S2	P2	400
S3	Blake	30	Paris	S3	P2	200
S4	Clark	20	London	S4	P2	200
S4	Clark	20	London	S4	P4	300
S4	Clark	20	London	S4	P5	400

12 records selected

ตัวอย่าง ค.4 ตัวอย่างการใช้แถวคำสั่งปฏิบัติการ Divide พร้อมทั้งผลลัพธ์ที่ได้

ERAO >a1=project SP[SNUM PNUM]

ERAO>a2=project P[PNUM]

Relation's name : a1

Relation's name :a2

SNUM PNUM

PNUM

=====

=====

S1 P1

P1

S1 P2

P2

S1 P3

P3

S1 P4

P4

S1 P5

P5

S1 P6

P6

S2 P1

S2 P2

6 records selected

S3 P2

S4 P2

S4 P4

S4 P5

12 records selected

---

ERAO >a3=divide a1 a2

Relation's name : a3

SNUM

=====

S1

1 records selected

ตัวอย่าง ค.5 ตัวอย่างการใช้แกลวคำสั่งปฏิบัติการ Sequence กับตารางข้อมูล a1 ซึ่งได้  
จากตัวอย่าง ค.4 พร้อมทั้งผลลัพธ์ที่ได้

ERAO >a4=Sequence a1[PNUM]

Relation's name : a4

PNUM SNUM

=====

P1 S1

P1 S2

P2 S1

P2 S2

P2 S3

P2 S4

P3 S1

P4 S1

P4 S4

P5 S1

P5 S4

P6 S1

12 records selected

**ตัวอย่าง ค.6** ตัวอย่างการใช้แถวคำสั่งปฏิบัติการ Unique พร้อมทั้งผลลัพธ์ที่ได้

ERAO>q1=project SP[SNUM]

Relation's name : q1

SNUM

=====

S1

S1

S1

S1

S1

S1

S2

S2

S3

S4

S4

S4

12 records selected

ERAO >q2=unique q1

---

Relation's name : q2

SNUM

=====

S1

S2

S3

S4

4 records selected

**ตัวอย่าง ค.7 ตัวอย่างการใช้แควค่าสั่งปฏิบัติการ Intersect พร้อมทั้งผลลัพธ์ที่ได้**

ERAO>w1=select S[SNUM>S2]

Relation's name : w1

SNUM	SNAME	STATUS	CITY
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

3 records selected

ERAO>w2=select S[SNUM<S4]

Relation's name : w2

SNUM	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris

3 records selected

---

ERAO >w3=intersect w1 w2

Relation's name : w3

SNUM	SNAME	STATUS	CITY
S3	Blake	30	Paris

1 records selected

**ตัวอย่าง ค.8** ตัวอย่างการใช้แฉวคำสั่งปฏิบัติการ Union กับตารางข้อมูล w1 และ w2  
 ในตัวอย่าง ค.7 พร้อมทั้งผลลัพธ์ที่ได้

ERAO >w4=union w1 w2

Relation's name : w4

SNUM	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

5 records selected

**ตัวอย่าง ค.9** ตัวอย่างการใช้แฉวคำสั่งปฏิบัติการ Minus กับตารางข้อมูล w1 และ w2 ใน  
 ตัวอย่าง ค.7 พร้อมทั้งผลลัพธ์ที่ได้

ERAO >w5=minus w1 w2

Relation's name : w5

SNUM	SNAME	STATUS	CITY
S4	Clark	20	London
S5	Adams	30	Athens

2 records selected

ตัวอย่าง ค.10 ตัวอย่างการใช้แวงคำสั่งปฏิบัติการ Cross กับตารางข้อมูล w1 และ w2  
 ในตัวอย่าง ค.7 พร้อมทั้งผลลัพธ์ที่ได้

ERAO >w6=cross w1 w2

Relation's name : w6

SNUM	SNAME	STATUS	CITY	SNUM	SNAME	STATUS	CITY
S3	Blake	30	Paris	S1	Smith	20	London
S3	Blake	30	Paris	S2	Jones	10	Paris
S3	Blake	30	Paris	S3	Blake	30	Paris
S4	Clark	20	London	S1	Smith	20	London
S4	Clark	20	London	S2	Jones	10	Paris
S4	Clark	20	London	S3	Blake	30	Paris
S5	Adams	30	Athens	S1	Smith	20	London
S5	Adams	30	Athens	S2	Jones	10	Paris
S5	Adams	30	Athens	S3	Blake	30	Paris

9 records selected

**ตัวอย่าง ค.11** ตัวอย่างการใช้แถวคำสั่งปฏิบัติการเพื่อหาฟังก์ชัน Max สำหรับ  
ทุกแถวข้อมูล พร้อมทั้งผลลัพธ์ที่ได้

ERAO >f1=max SP[QTY]

Relation's name : f1

\_SUM

====

400

1 records selected

**ตัวอย่าง ค.12** ตัวอย่างการใช้แถวคำสั่งปฏิบัติการเพื่อหาฟังก์ชัน Sum สำหรับ  
แต่ละกลุ่มย่อยของแถวข้อมูล พร้อมทั้งผลลัพธ์ที่ได้

ERAO >f2=sum SPJ[QTY/SNUM PNUM]

Relation's name : f2

SNUM PNUM \_SUM

=====

S1 P1 900

S1 P2 400

S1 P3 200

S2 P1 400

---

S2 P2 200

5 records selected



เพื่ออำนวยความสะดวกในการใช้คำสั่งปฏิบัติการ ERAO มากขึ้น ดังนั้นจึงมีคำสั่งเพื่อดำเนินงานเพิ่มเติม โดยการดำเนินงานนี้จะไม่ทำให้ได้ตารางข้อมูลผลลัพธ์ คำสั่งเพื่ออำนวยความสะดวกในการใช้คำสั่งปฏิบัติการมีดังนี้

**รูปแบบ HELP**

**ความหมาย** เป็นคำสั่งเพื่อให้ความช่วยเหลือ หรือคำแนะนำ และดูรูปแบบคำสั่งปฏิบัติการ ERAO

**รูปแบบ QUIT**

**ความหมาย** เป็นคำสั่งเพื่อยกเลิกการทำงานในส่วนคำสั่งปฏิบัติการ ERAO ออกไปสู่ระบบงานเดิม

**รูปแบบ DISPLAY <table>**

**ความหมาย** เป็นคำสั่งเพื่อแสดงรายละเอียดข้อมูลจริงของตารางข้อมูล <table>

**ตัวอย่าง** display temp

เป็นการแสดงข้อมูลทั้งหมดของตารางข้อมูล temp

**รูปแบบ DISPSTRUC <table>**

**ความหมาย** เป็นคำสั่งเพื่อแสดงโครงสร้างของตารางข้อมูล <table>

**ตัวอย่าง** dispstruc temp

เป็นการแสดงโครงสร้างของตารางข้อมูล temp

**รูปแบบ OPEN <database>**

**ความหมาย** เป็นคำสั่งเพื่อเรียกใช้ฐานข้อมูล <database>

**ตัวอย่าง** open base

เป็นการเรียกใช้ฐานข้อมูล base

**รูปแบบ** CLOSE <database>

**ความหมาย** เป็นคำสั่งเพื่อยกเลิกการเรียกใช้ฐานข้อมูล <database>

**ตัวอย่าง** close base

เป็นการยกเลิกการเรียกใช้ฐานข้อมูล base

**รูปแบบ** LISTREL

**ความหมาย** เป็นคำสั่งเพื่อแสดงชื่อตารางข้อมูลทั้งหมดในฐานข้อมูลที่เรียกใช้

**รูปแบบ** LISTDB

**ความหมาย** เป็นคำสั่งเพื่อแสดงชื่อฐานข้อมูลทั้งหมดในระบบ

**รูปแบบ** KEEP <table>

**ความหมาย** เป็นคำสั่งเพื่อบรรจุตารางข้อมูล <table> ลงในแฟ้มข้อมูลต่าง ๆ ตารางข้อมูลจะสามารถใช้คำสั่งนี้ได้จะต้องเป็นตารางข้อมูลชั่วคราว (Temporary) ที่ได้จากคำสั่งปฏิบัติการ ERAO

**รูปแบบ** DELETE <table>

**ความหมาย** เป็นคำสั่งเพื่อกำจัดตารางข้อมูล <table> ออกจากแฟ้มข้อมูลต่าง ๆ

## ประวัติผู้เขียน

ชื่อ นางสาวสุภาภรณ์ กานต์สมเกียรติ  
วัน เดือน ปีเกิด วันศุกร์ที่ 8 เดือน สิงหาคม พ.ศ. 2512

วุฒิการศึกษา

วุฒิ

ชื่อสถาบัน

ปีที่สำเร็จการศึกษา

วิทยาศาสตร์บัณฑิต

มหาวิทยาลัยสงขลานครินทร์

2533

(คณิตศาสตร์)

วิทยาเขตหาดใหญ่

ทุนการศึกษา (ที่ได้รับระหว่างการศึกษา)

มูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสาร