



การสร้างตัวแปลภาษาสอบถามฐานข้อมูล QBE
An Implementation of QBE Query Language Interpreter

พงษ์พันธ์ พิณโก
Pongpan Pinto

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

2537


๗

เลขหมู่	QA๗๖๙.๒๓ ๑๒๕ ๘๙๘๗ ๓-๒
Bib Key	๑๑๒๘๕๗

ชื่อวิทยานิพนธ์ การสร้างตัวแปลภาษาสอตามฐานข้อมูล QBE
ชื่อผู้เขียน นายพงษ์พันธ์ พิณโท
สาขาวิชา วิทยาการคอมพิวเตอร์

คณะกรรมการที่ปรึกษา

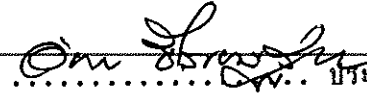
คณะกรรมการสอบ


..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. อัสฉา ชีร์เชษฐมงคล)


..... กรรมการ

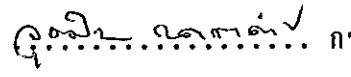
(ผู้ช่วยศาสตราจารย์ อิว ไอยรากาญจนกุล)



..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. อัสฉา ชีร์เชษฐมงคล)

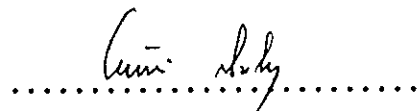

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ อิว ไอยรากาญจนกุล)


..... กรรมการ
(อาจารย์ วุฒิพงษ์ เตชะดำรงสิน)


..... กรรมการ
(อาจารย์ เลียง कुบุดนุ)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้เนิวิทยานิพนธ์ฉบับนี้ เป็น
ส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์


.....

(ดร. ไพรัตน์ สงวาน)

คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์	การสร้างตัวแปลภาษาสอบถามฐานข้อมูล QBE
ผู้เขียน	นายพงษ์พันธ์ พิณโท
สาขาวิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2536

บทคัดย่อ

QBE เป็นภาษาสอบถามฐานข้อมูลแบบแคลคูลัสสัมพันธ์ซึ่ง เป็นภาษาที่ไม่ต้องบอกลำดับขั้นตอนการดำเนินงาน เพื่อให้ได้มาซึ่งตารางข้อมูลค่าตอบจากตารางข้อมูลที่กำหนดให้ของระบบจัดการฐานข้อมูลเชิงสัมพันธ์ ผู้ใช้เพียงแต่ระบุว่าต้องการข้อมูลที่มีลักษณะหรือคุณสมบัติอย่างไรเท่านั้น

รูปแบบของภาษา QBE เป็นลักษณะการกรอกข้อความสอบถามในโครงร่างตารางข้อมูลที่เกี่ยวข้องกับสอบถามบนจอภาพคอมพิวเตอร์แทนการเขียนเป็นลำดับคำสั่ง ซึ่งง่ายการใช้สำหรับผู้ใช้ทั่ว ๆ ไปมาก แต่สามารถดำเนินงานสอบถามฐานข้อมูลที่ซับซ้อนได้เป็นอย่างดี

วิทยานิพนธ์นี้เป็นการสร้างตัวแปลภาษาสอบถามฐานข้อมูล QBE ในลักษณะที่เรียกว่า Interpreter กล่าวคือ เมื่อผู้ใช้กรอกข้อความสอบถามในตารางข้อมูลบนจอภาพคอมพิวเตอร์เสร็จเรียบร้อย และสั่งให้ระบบประมวลผล ระบบจะตรวจสอบวากยสัมพันธ์ของภาษาเมื่อพบว่าถูกต้องจะดำเนินการประมวลผลด้วยการแปลความหมายและดำเนินงานให้ผลลัพธ์ในรูปตารางข้อมูลบนจอภาพคอมพิวเตอร์ทันที โดยการดำเนินงานในส่วนการตรวจสอบความถูกต้องของวากยสัมพันธ์ของภาษาได้ใช้โปรแกรมอรรถประโยชน์ LEX และ YACC บนระบบปฏิบัติการยูนิกซ์ และในส่วนการประมวลผลเพื่อให้ได้มาซึ่งคำตอบได้ใช้โปรแกรมการดำเนินงานสอบถามแบบพีชคณิตสัมพันธ์ของงานวิทยานิพนธ์และโครงงานที่ทำมาก่อนหน้าวิทยานิพนธ์นี้ด้วย

Thesis Title An Implementation of QBE Query Language Interpreter
Author Mr. Pongpan Pinto
Major Program Computer Science
Academic Year 1993

Abstract

QBE is a relational calculus query language of a relational database management system. This language can be used to retrieve the desired data from a database without telling the steps of operations to the system. The user merely specifies characteristics of the desired data he wants to retrieve.

The syntax of QBE is based on the idea of making query entries in the structure of the relevant relation(s) in the tabular form(s) on the computer screen instead of writing a sequence of statements. Therefore, it is very simple and easy to be used by naive users. However, it is very efficient in retrieving even complex queries.

This thesis is aimed at a construction of QBE query language interpreter. When the user completes the query in table(s) on the screen and orders the system to perform the operation, the system will analyze and check the syntax. The valid query will be processed immediately and the result is shown in tabular form on the screen. LEX and YACC are the UNIX's utility programs used for lexical and syntax analysis of QBE interpreter. The relational algebra operations developed by the previous thesis and the project are used to process the semantics of the QBE query.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลงได้ด้วยดี ด้วยความช่วยเหลือและสนับสนุนจากบุคคล
หลายฝ่ายซึ่งผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูงมา ณ. โอกาสนี้ คือ

ผู้ช่วยศาสตราจารย์ ดร. อัญญา ชีรเชษฐมงคล อาจารย์ที่ปรึกษา ที่กรุณาให้
คำปรึกษา ข้อเสนอแนะทางวิชาการและแก้ไขปัญหิต่าง ๆ ตลอดจนตรวจทานวิทยานิพนธ์
ให้แก่ผู้วิจัยมาโดยตลอด

ผู้ช่วยศาสตราจารย์ อิว ไอยรากาญจนกุล อาจารย์ที่ปรึกษาร่วม ที่ได้กรุณาให้
คำปรึกษา ตรวจทานวิทยานิพนธ์ให้แก่ผู้วิจัย

คณะกรรมการสอบวิทยานิพนธ์ทุกท่าน ที่กรุณาช่วยตรวจและแก้ไขวิทยานิพนธ์

สำนักงานคณะกรรมการพัฒนาวิทยาศาสตร์และเทคโนโลยี สถาบันวิจัยวิทยาศาสตร์
และเทคโนโลยีแห่งประเทศไทย ซึ่งได้สนับสนุนทุนสำหรับการศึกษาและทำวิจัย

คุณพ่อ คุณแม่ พี่ชาย พี่สาว คุณชัชชุกฤ มณีฉาย น้องชายและน้องสาวของผู้วิจัย
ซึ่งได้สนับสนุน ช่วยเหลือและให้กำลังใจในการทำวิทยานิพนธ์มาโดยตลอด

พงษ์พันธ์ ฟิลิปโป

สารบัญ

หน้า

บทคัดย่อ	(3)
Abstract	(4)
กิตติกรรมประกาศ	(5)
สารบัญ	(6)
รายการตาราง	(8)
รายการภาพประกอบ	(9)
บทที่	
1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ขอบเขตของงานวิจัย	3
1.4 ขั้นตอนและระยะเวลาการดำเนินงาน	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	4
1.6 สถานที่และอุปกรณ์ที่ใช้	5
2 ภาษาสอบถามฐานข้อมูล QBE	6
2.1 ประวัติความเป็นมา	6
2.2 ภาษารฐานข้อมูล	7
2.3 ภาษาพีชคณิตสัมพันธ์	8
2.4 ภาษาแคลคูลัสสัมพันธ์	9
2.5 ภาษาสอบถามฐานข้อมูล QBE	12
3 การสร้างตัวแปลภาษาคำด้วยโปรแกรม LEX และ YACC	44
3.1 ตัวแปลภาษา	44
3.2 โปรแกรม LEX	47
3.3 โปรแกรม YACC	52

(6)

สารบัญ (ต่อ)

หน้า

4	การพัฒนาตัวแปลภาษาสอบถามฐานข้อมูล QBE	57
4.1	โครงสร้างของระบบงานเดิม	57
4.2	ตัวแปลภาษาสอบถามฐานข้อมูล QBE	64
4.3	โครงสร้างการดำเนินงาน	65
4.4	การวิเคราะห์ ออกแบบและพัฒนาระบบ	79
5	สรุปผล อุปสรรคปัญหาและข้อเสนอแนะ	117
5.1	สรุปผล	117
5.2	อุปสรรคและปัญหา	117
5.3	ข้อเสนอแนะ	118
	บรรณานุกรม	119
	ภาคผนวก ก ความหมายของแผนภาพกระแสข้อมูล	122
	ภาคผนวก ข แผนภาพวากยสัมพันธ์ของ QBE	125
	ภาคผนวก ค เอกสารสำหรับผู้ใช้งาน	131
	ค.1 วิธีใช้โปรแกรม Mrdbms	132
	ค.2 แผนภาพสรุปรายการหลักและรายการย่อยของ Mrdbms	139
	ค.3 วิธีใช้ตัวแปลภาษาฐานข้อมูล QBE	140
	ประวัติผู้เขียน	152

รายการตาราง

ตาราง

หน้า

4.1 สัญลักษณ์การกระทำและความหมายของถาดดำเนินงาน	74
---	----

รายการภาพประกอบ

ภาพประกอบ	หน้า
2.1 ตัวอย่างฐานข้อมูลที่ประกอบด้วยตารางข้อมูล S, P และ SP	10
3.1 ขั้นตอนการดำเนินงานของตัวแปลภาษาระดับสูง	45
3.2 ข้อมูลเข้า/ออก ของ LEX และ yylex()	47
3.3 ข้อมูลเข้า/ออก ของ YACC และ yyparse()	52
4.1 แผนภาพโอบอาร์ของโปรแกรมข้อมูลและตารางข้อมูลที่สัมพันธ์กัน	57
4.2 พังภาพสารบบเพิ่มข้อมูลของงานวิจัย	60
4.3 โครงสร้างข้อมูลของโปรแกรมข้อมูลขณะที่อยู่ในหน่วยความจำหลัก	62
4.4 แผนภูมิระบบแสดงการดำเนินงานในส่วนต่าง ๆ ของงานวิจัยเดิม	63
4.5 แผนภาพกระแสข้อมูลของระบบ Mrdbms เดิม	79
4.6 แผนภาพกระแสข้อมูลของตัวแปลภาษาสอบถามฐานข้อมูล QBE	81
4.7 แผนภาพกระแสข้อมูลของกระบวนการเลขที่ 7.2	82
4.8 แผนภาพกระแสข้อมูลของกระบวนการเลขที่ 7.4	83
4.9 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.4	84
4.10 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.8	85
4.11 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.8.2	86
4.12 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.8.4	87
4.13 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.8.5	88
4.14 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.8.5.4	89
4.15 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.8.5.5	90
4.16 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.8.5.6	91
4.17 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.5	92
4.18 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.6	93
4.19 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.7	94

รายการภาพประกอบ (ต่อ)

ภาพประกอบ

หน้า

4.20	รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.8	95
4.21	รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.9	97
4.22	รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.10	98
4.23	รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.11	99
4.24	รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.12	101
4.25	รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.13	102
4.26	รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.14	104
4.27	รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.15	105
4.28	รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.16	106
ค.1	ส่วนประกอบหลักของจอภาพ MRDBMS	132
ค.2	จอภาพรายการย่อยของรายการหลัก Help	134
ค.3	จอภาพรายการย่อยของรายการหลัก Database	135
ค.4	จอภาพรายการย่อยของรายการหลัก Relation	136
ค.5	จอภาพรายการย่อยของรายการหลัก Tuple	137
ค.6	จอภาพรายการย่อยของรายการหลัก Query	138
ค.7	โครงสร้างรายการหลักและรายการย่อยของโปรแกรม MRDBMS	139
ค.8	ตัวอย่างการสอบถามฐานข้อมูลด้วยภาษา QBE บนโครงสร้างตารางข้อมูลชื่อ S	140
ค.9	จอภาพรายการหลักของโปรแกรม Mrdbms	143
ค.10	จอภาพการระบุพื้นฐานข้อมูลที่ต้องการสอบถาม	144
ค.11	จอภาพสำหรับเลือกการสอบถามข้อมูลจากฐานข้อมูลด้วย QBE	145
ค.12	จอภาพเริ่มต้นการทำงานของโปรแกรมตัวแปลภาษาสอบถามฐานข้อมูล QBE	146
ค.13	จอภาพแสดงโครงสร้างตารางข้อมูลและชื่อตารางข้อมูลที่ต้องการใช้สอบถาม	147
ค.14	จอภาพแสดงชื่อแอตทริบิวของตารางข้อมูลที่ต้องการสอบถาม	147

ค.15 ตัวอย่างของข้อความสั่งสอบถามฐานข้อมูลด้วยภาษา QBE	148
ค.16 ตัวอย่างของข้อความสั่งสอบถามที่หนึ่งสคมีมากกว่า 10 ตัวอักษร	149
ค.17 ตัวอย่างจอภาพแสดงผลจากการประมวลผลการสอบถาม	150
ค.18 จอภาพเมื่อเลิกการทำงานของโปรแกรม Mrdbms	151

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

คอมพิวเตอร์ได้เข้ามามีบทบาทในชีวิตประจำวันของเรามากขึ้นในเกือบทุกด้าน ทั้งทางวิทยาศาสตร์และสังคมศาสตร์ ทำให้มีการผลิตคอมพิวเตอร์ที่มีประสิทธิภาพมากขึ้นเรื่อยๆ ขณะเดียวกันข้อมูลที่ใช้ประมวลผลมีมากขึ้นและการประมวลผลมีความยุ่งยากซับซ้อนขึ้น ดังนั้นในการพัฒนาโปรแกรมเพื่อประยุกต์ใช้กับระบบงานแต่ละอย่าง โปรแกรมเมอร์ต้องใช้เวลามาก ทั้งในการเขียนโปรแกรมและแก้ไขข้อผิดพลาดจากการทำงานของโปรแกรม การมีเครื่องมือในรูปโปรแกรมสำเร็จรูปจะช่วยจะทำให้ลดความยุ่งยากไปได้มาก

ระบบจัดการฐานข้อมูล (Database Management System - DBMS) เป็นโปรแกรมสำเร็จรูปซึ่งเป็นเครื่องมือช่วยลดความยุ่งยากซับซ้อนในการดำเนินงานกับข้อมูลที่ถูกจัดเก็บในลักษณะฐานข้อมูล ผู้ใช้ไม่จำเป็นต้องเขียนโปรแกรมดำเนินงานกับข้อมูล เพียงแต่เรียนรู้การเขียนภาษาฐานข้อมูล (Database language) เพื่อดำเนินงานกับฐานข้อมูล เช่น การเพิ่มข้อมูล การลบข้อมูล การปรับเปลี่ยนค่าข้อมูลเก่าที่มีอยู่และการเรียกดูข้อมูลและสารสนเทศที่ต้องการก็พอ

QBE (Query By Example) เป็นภาษาฐานข้อมูลที่สำคัญภาษาหนึ่งของระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational database management system) เสนอโดย M.H. Zloof ในปี ค.ศ. 1975 [Zloof (1975)] เพื่ออำนวยความสะดวกต่อผู้ใช้ในการดำเนินงานกับฐานข้อมูลเชิงสัมพันธ์ (Relational database) ผู้ใช้ซึ่งมีความรู้ทางคอมพิวเตอร์ไม่มากนักสามารถที่จะเริ่มต้นการเรียนรู้ภาษาได้อย่างรวดเร็วและสามารถเรียนรู้การใช้ภาษาทั้งหมดได้ด้วยตนเองในเวลาไม่มากนัก QBE เป็นภาษาที่มีรูปแบบง่ายแต่สามารถดำเนินงานที่ซับซ้อนได้มาก จากผลการศึกษาทางจิตวิทยากับผู้ให้ที่เป็นนักเรียนมัธยมปลาย นักศึกษาที่กำลังศึกษาและที่เพิ่งสำเร็จการศึกษาในระดับวิทยาลัย ซึ่งส่วนใหญ่ไม่เคยใช้คอมพิวเตอร์มาก่อนหรือบางส่วนเคยรับการศึกษาในระดับเบื้องต้นเท่านั้น

พบว่าสามารถใช้เวลาในการฝึกฝนและเรียนรู้การใช้ข้อความสั่งของภาษารฐานข้อมูล QBE โดยการสอบถามในรูปแบบที่ค่อนข้างซับซ้อนได้ภายในเวลาไม่เกิน 3 ชั่วโมง [Thomas and Gould (1975)]

งานวิจัยนี้ เป็นการสร้างตัวแปลภาษารฐานข้อมูล QBE ในลักษณะที่เรียกว่าอินเตอร์-พรีเตอร์ (Interpreter) โดยหลังจากการตรวจสอบความถูกต้องทางวากยสัมพันธ์แล้ว จะแปลความหมายและดำเนินการทันที โดยการดำเนินการจะเรียกใช้ปฏิบัติการพีชคณิต-สัมพันธ์ (Relational algebra operation) และการดำเนินการอื่นของงานวิจัยที่มีมาก่อนหน้านี้ [เพ็ชรพร (2534)] [ทวีศักดิ์ (2534)] และโครงการทางคณิตศาสตร์ของนักศึกษาเก่า [นิรันดร์และอรุณี (2534)] ตัวแปลภาษาที่สร้างในงานวิจัยนี้จะครอบคลุมเฉพาะส่วนของภาษาสอบถาม (Query language) เท่านั้น ไม่รวมส่วนของภาษาซึ่งใช้กำหนดโครงสร้างฐานข้อมูลและใช้ดำเนินการกับฐานข้อมูล

1.2 วัตถุประสงค์ของการวิจัย

- (1) เพื่อสร้างตัวแปลภาษาสอบถามฐานข้อมูล QBE
- (2) เพื่อเพิ่มเติมองค์ประกอบของงานวิจัยเดิมคือ "โปรแกรมส่วนกำหนดโครงสร้างฐานข้อมูลและการดำเนินการพื้นฐาน" และ "โปรแกรมเพื่อการพัฒนาาระบบจัดการฐานข้อมูลเชิงสัมพันธ์ส่วนภาษารฐานข้อมูล" ให้มีความสมบูรณ์มากยิ่งขึ้น โดยการเชื่อมโยงโปรแกรมที่พัฒนานี้กับโปรแกรมของงานวิจัยเดิมให้ทำงานร่วมกันได้

1.3 ขอบเขตของงานวิจัย

- (1) ศึกษารายละเอียดต่าง ๆ ของงานวิจัยเดิม ดังนี้
 - โครงสร้างปทานุกรมข้อมูลและโครงสร้างเพิ่มข้อมูล
 - รูปแบบคำสั่งและการดำเนินงานของแต่ละคำสั่งของภาษาสอบถามพีชคณิตสัมพันธ์
- (2) ศึกษาหลักการและเทคนิคต่าง ๆ ที่สามารถใช้ในการพัฒนาตัวแปลภาษาสอบถามฐานข้อมูล QBE
- (3) ตรวจสอบและจัดความกำกวมของไวยากรณ์ของภาษาสอบถามฐานข้อมูล QBE
- (4) พัฒนาโปรแกรมที่สามารถใช้ปทานุกรมข้อมูลและปฏิบัติการต่างๆร่วมกับงานวิจัยเดิม
- (5) เชื่อมโยงชุดคำสั่งของงานวิจัยที่พัฒนากับงานวิจัยเดิมให้สามารถทำงานร่วมกันได้
- (6) ผลลัพธ์ที่ได้จากการดำเนินงานสามารถแสดงได้ทางจอภาพ

1.4 ขั้นตอนและระยะเวลาการดำเนินงาน

ขั้นตอนการดำเนินงาน

- (1) ศึกษาและทำความเข้าใจนิยามและความหมาย รวมทั้งหาแนวทางในการดำเนินงานกับข้อความสั่งของ QBE ในรูปคำสั่งของภาษาพีชคณิตสัมพันธ์
- (2) ศึกษาและวิเคราะห์งานวิจัยเดิมในส่วนของโครงสร้างปทานุกรมข้อมูล โครงสร้างเพิ่มข้อมูล โครงสร้างข้อมูล รูปแบบคำสั่งและการดำเนินงานของภาษาพีชคณิตสัมพันธ์และรายละเอียดอื่น ๆ
- (3) ศึกษาการใช้โปรแกรมรรถประโยชน์ LEX, YACC, DBX และ NCURSES
- (4) เลือกและออกแบบวากยสัมพันธ์ของภาษาสอบถามฐานข้อมูล QBE ที่จะใช้
- (5) พัฒนาตัวแปลภาษาสอบถามฐานข้อมูล QBE โดยใช้ภาษาซี
- (6) ทดสอบและปรับปรุงแก้ไขโปรแกรมเพื่อให้ได้โปรแกรมที่ทำงานได้อย่างถูกต้องและสมบูรณ์
- (7) จัดทำเอกสารและรายงานวิทยานิพนธ์

ระยะเวลาในการดำเนินงาน

ใช้เวลาทั้งสิ้น 1 ปี 3 เดือน ดังแสดงด้วยผังภาพ

ระยะ เวลา	2535												2536		
	มค.	กพ.	มีค.	เมย.	พค.	มิย.	กค.	สค.	กย.	ตค.	พย.	ธค.	มค.	กพ.	มีค.
1	<->														
2		<->													
3			<->												
4				<---	---										
5					<---	---	---	---	---	---					
6											<---	---	---		
7											<---	---	---	---	---

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- (1) ได้โปรแกรมตัวแปลภาษาสอบถามฐานข้อมูล QBE ซึ่งเป็นส่วนหนึ่งของระบบจัดการฐานข้อมูลเชิงสัมพันธ์
- (2) ทำให้งานวิจัยเดิมมีความสมบูรณ์มากยิ่งขึ้น

1.6 สถานที่และอุปกรณ์ที่ใช้

สถานที่ทำการวิจัย

ห้องปฏิบัติการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์
มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

เครื่องมือที่ใช้

- (1) มินิคอมพิวเตอร์ VAX-11/785 ขนาดหน่วยความจำ 12 MB และใช้ระบบปฏิบัติการ Ultrix-32
- (2) เทอร์มินอลชนิด VT 220 หรือเทอร์มินอลซึ่งเป็นเครื่องไมโครคอมพิวเตอร์
- (3) ตัวแปลภาษาซี (C Compiler)
- (4) โปรแกรมรรถประโยชน์ช่วยตรวจจับกลุ่มคำ (Token) ชื่อ LEX
- (5) โปรแกรมรรถประโยชน์ช่วยตรวจไวยากรณ์ของภาษา QBE ชื่อ YACC
- (6) โปรแกรมรรถประโยชน์ช่วยจัดการเกี่ยวกับจอภาพ ชื่อ NCURSES
- (7) โปรแกรมรรถประโยชน์ช่วยตรวจและแก้ไขข้อผิดพลาดในขณะโปรแกรมกำลังถูกดำเนินการ ชื่อ DBX

ภาษาสอบถามฐานข้อมูล QBE

2.1 ประวัติความเป็นมา

ตั้งแต่ E.F. Codd ได้เสนอรูปแบบฐานข้อมูลเชิงสัมพันธ์ (Relational data model) เป็นต้นมา ระบบจัดการฐานข้อมูลเชิงสัมพันธ์เริ่มมีบทบาทมากขึ้นในแวดวงการใช้งานคอมพิวเตอร์จวบจนกระทั่งถึงปัจจุบัน อาจกล่าวได้ว่าเกือบทุกหน่วยงานมีความต้องการในการใช้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ทั้งนั้น [Korth and Silberschatz (1991)] ภาษาฐานข้อมูลเชิงสัมพันธ์ซึ่งเป็นตัวเชื่อมระหว่างผู้ใช้และระบบจัดการฐานข้อมูลเชิงสัมพันธ์จึงได้รับความสนใจในการศึกษาเพื่อการใช้งานเป็นอย่างมาก

QBE เป็นภาษาฐานข้อมูลหนึ่งของระบบจัดการฐานข้อมูลเชิงสัมพันธ์ ถูกออกแบบและเสนอโดย H.M. Zloof [Maier (1983)] นักวิจัยคอมพิวเตอร์ของบริษัท IBM ณ ศูนย์การวิจัยของ IBM ที่เมือง Yorktown ประเทศสหรัฐอเมริกา โดยมีเป้าหมายหลักเพื่อให้เป็นภาษาที่ง่ายต่อการใช้ของผู้ใช้ทั่ว ๆ ไปที่ไม่มีความรู้เรื่องการเขียนโปรแกรมหรือการใช้งานคอมพิวเตอร์ การใช้ภาษา QBE จะอยู่ในลักษณะการโต้ตอบระหว่างผู้ใช้และระบบผ่านทางแป้นพิมพ์และจอภาพโดยที่ผู้ใช้ไม่ต้องเรียนรู้รูปแบบของภาษามากนัก การเขียนคำถามในการสอบถามข้อมูลผู้ใช้เพียงแต่กรอกคำสั่งและข้อมูลเท่าที่จำเป็นเพียงเล็กน้อยในตารางที่แสดงบนจอภาพเท่านั้น QBE ถูกนำมาใช้อย่างเป็นทางการครั้งแรกโดยบริษัท IBM บนเครื่อง IBM 30xx และ IBM 43xx ภายใต้ระบบปฏิบัติการ VMS ต่อมาได้รับการพัฒนาให้สามารถทำงานได้บนเครื่องที่ผลิตโดยบริษัท IBM อื่นหลายรุ่น

2.2 ภาษาฐานข้อมูล

ภาษาฐานข้อมูลจัดว่าเป็นภาษาคอมพิวเตอร์ชนิดเฉพาะประสงค์ (Special purpose computer language) ซึ่งอำนวยความสะดวกแก่ผู้ใช้ในการติดต่อกับระบบจัดการฐานข้อมูลเพื่อกำหนดรูปแบบและลักษณะของข้อมูลที่จะเก็บในฐานข้อมูล รวมถึงการดำเนินงานต่าง ๆ กับฐานข้อมูล เช่น การเพิ่มข้อมูลเข้า การลบข้อมูลออก การเปลี่ยนแปลงค่าข้อมูลและการสอบถามข้อมูลจากฐานข้อมูล เป็นต้น โดยทั่วไปภาษาฐานข้อมูลอาจแบ่งได้เป็น 2 แบบตามรูปแบบและหน้าที่การดำเนินงาน [CODASYL DBTG (1970)] ดังนี้

- ภาษากำหนดโครงสร้าง/นิยาม (Data definition language - DDL)
- ภาษาคำเนิงาน (Data manipulation language - DML)

ภาษากำหนดโครงสร้าง/นิยาม

ภาษากำหนดโครงสร้าง/นิยามเป็นภาษาฐานข้อมูลที่ใช้ในการกำหนดชื่อและโครงสร้างฐานข้อมูลว่าประกอบด้วยรายละเอียดและประเภทข้อมูลอะไรบ้าง รวมทั้งข้อกำหนดหรือเงื่อนไขที่จะทำให้ข้อมูลในฐานข้อมูลมีความถูกต้องเป็นหนึ่งเดียวเสมอ สำหรับระบบจัดการฐานข้อมูลเชิงสัมพันธ์ ผลจากการทำงานจะได้โครงสร้างตารางสำหรับจัดเก็บข้อมูล และรายละเอียดเกี่ยวกับโครงสร้างที่ได้กำหนดสำหรับฐานข้อมูลนั้น ๆ จะถูกเก็บในฐานข้อมูลชนิดหนึ่งเรียกว่า ปทานุกรมข้อมูล (Data dictionary) เมื่อใดก็ตามที่มีการเปลี่ยนแปลงหรือเรียกใช้ข้อมูลจากฐานข้อมูลก็จะต้องอาศัยข้อมูลจากปทานุกรมข้อมูลนี้เสมอ

ภาษาคำเนิงาน

ภาษาคำเนิงานเป็นภาษาฐานข้อมูลที่ใช้ในการคำเนิงานกับข้อมูลในฐานข้อมูล อันได้แก่ การเพิ่มข้อมูลเข้าฐานข้อมูล (Insertion) การลบข้อมูลออกจากฐานข้อมูล (Deletion) การปรับปรุงเปลี่ยนแปลงแก้ไขข้อมูลเก่าในฐานข้อมูล (Update) รวมทั้งการค้นคืนข้อมูล (Retrieval) หรือการสอบถาม (Query) เพื่อถามหาข้อมูลที่มีคุณสมบัติเป็นไปตามเงื่อนไขที่บ่งบอกได้อย่างรวดเร็วทันที่ทันใจจากฐานข้อมูล การใช้ฐานข้อมูลในระดับของผู้ใช้ทั่วไปสามารถกระทำได้ด้วยการใช้ภาษาคำเนิงานผ่านระบบจัดการฐานข้อมูล

ในช่วงแรกของการพัฒนาการทางด้านภาษาฐานข้อมูล ภาษาซึ่งใช้ในการสอบถามฐานข้อมูลจะรวมอยู่กับภาษาคำเนิงานอื่น ๆ แต่ด้วยความสำคัญของการใช้ภาษานี้มีมาก เพราะเป็นส่วนที่ช่วยให้การประมวลผลกับฐานข้อมูลเป็นไปอย่างง่าย ภาษาในส่วนนี้จึงได้รับการพัฒนาและปรับปรุงให้ดียิ่ง ๆ ขึ้น สามารถใช้ได้เป็นอย่างดี ไม่ต้องขึ้นอยู่กับภาษาคอมพิวเตอร์ใด ๆ และมีชื่อเรียกพิเศษว่า "ภาษาสอบถาม" ในระบบจัดการฐานข้อมูลเชิงสัมพันธ์ ซึ่งฐานข้อมูลประกอบด้วยข้อมูลในรูปตารางข้อมูลที่เรียกว่า รีเลชัน (Relation) จำนวนมาก ภาษาสอบถามฐานข้อมูลอาจแบ่งได้เป็น 2 แบบ คือ

- ภาษาพีชคณิตสัมพันธ์ (Relational algebra language)
- ภาษาแคลคูลัสสัมพันธ์ (Relational calculus language)

2.3 ภาษาพีชคณิตสัมพันธ์

ภาษาพีชคณิตสัมพันธ์ เป็นภาษาที่ประกอบด้วยคำสั่งในการดำเนินการกับตารางข้อมูล ซึ่งก็คือเซต (Set) ในทางคณิตศาสตร์นั่นเอง ในการสอบถามข้อมูล ผู้ใช้จะต้องบอกขั้นตอนการดำเนินการด้วยคำสั่งเหล่านี้ตั้งแต่เริ่มต้น จนกระทั่งได้ผลลัพธ์ที่เป็นคำตอบที่ต้องการ หนึ่งคำสั่งของภาษาพีชคณิตสัมพันธ์ประกอบด้วย คำสั่งปฏิบัติการ (Operator) และตารางข้อมูลที่ใช้หนึ่งหรือสองตาราง ผลลัพธ์ที่ได้ของแต่ละคำสั่งก็คือตารางข้อมูล ซึ่งสามารถถูกใช้กับคำสั่งอื่น ๆ ต่อไปได้อีก

คำสั่งปฏิบัติการโดยทั่วไปของภาษาพีชคณิตสัมพันธ์ถูกแบ่งออกเป็น 2 กลุ่ม [Date (1986)] คือ

- ก. คำสั่งปฏิบัติการมาตรฐานในการดำเนินการกับเซต (Set operator)

คำสั่งในกลุ่มนี้ใช้ตารางข้อมูล 2 ตารางในการดำเนินการ ประกอบด้วยคำสั่งปฏิบัติการ ดังนี้

- Union
- Intersect
- Minus (Difference)
- Product

๒. คำสั่งปฏิบัติการพิเศษ (Special operator)

คำสั่งในกลุ่มนี้เป็นการดำเนินการพิเศษที่เพิ่มขึ้นมาเพื่อให้สามารถครอบคลุมการ

ดำเนินการที่คำสั่งปฏิบัติการมาตรฐานของเซตทำไม่ได้ ประกอบด้วยคำสั่งปฏิบัติการ ดังนี้

- Select
- Project
- Join
- Divide

2.4 ภาษาแคลคูลัสสัมพันธ์

ภาษาแคลคูลัสสัมพันธ์เป็นภาษาที่ไม่ต้องบอกลำดับขั้นตอนการดำเนินการ ในการสอบถามข้อมูล ผู้ใช้เพียงแต่ระบุว่าต้องการข้อมูลมีลักษณะหรือคุณสมบัติอย่างไร โดยไม่ต้องบอกวิธีการดำเนินการ ดังนั้น แต่ละข้อความสั่งของภาษาแคลคูลัสสัมพันธ์จะสัมพันธ์กับคำสั่งของภาษาพีชคณิตสัมพันธ์ชุดหนึ่ง (อาจเป็นคำสั่งเดียวหรือหลายคำสั่ง) เสมอ

รูปที่ 2.1 แสดงตัวอย่างฐานข้อมูลประกอบด้วยตารางข้อมูล 3 ตาราง ซึ่งจะใช้ประกอบการยกตัวอย่างที่ 2.1 - 2.24 ในหัวข้อนี้และหัวข้ออื่น ๆ โดยตัวอย่างที่ 2.1 แสดงชุดของคำสั่งในภาษาพีชคณิตสัมพันธ์ที่สัมพันธ์กับหนึ่งข้อความสั่งของภาษาแคลคูลัสสัมพันธ์ที่เรียกว่า SQL

S

P

SP

S#	SNAME	STATUS	CITY	P#	PNAME	COLOR	WEIGHT	CITY	S#	P#	QTY
S1	Smith	20	London	P1	Nut	Red	12	London	S1	P1	300
S2	Jones	10	Paris	P2	Bolt	Green	17	Paris	S1	P2	200
S3	Blake	30	Paris	P3	Screw	Blue	17	Rome	S1	P3	400
S4	Clark	20	London	P4	Screw	Red	14	London	S1	P4	200
S5	Adams	30	Athens	P5	Cam	Blue	12	Paris	S1	P5	100
				P6	Cog	Red	13	London	S1	P6	100
									S2	P1	300
									S2	P2	400
									S3	P2	200
									S4	P2	200
									S4	P4	300
									S4	P5	400

รูปที่ 2.1 ตัวอย่างฐานข้อมูลที่ประกอบด้วยตารางข้อมูล S, P และ SP

ตัวอย่างที่ 2.1

แสดงรหัสของ supplier และเมืองที่ supplier อยู่ของ supplier ที่ส่ง part มีรหัส "P2"

(Get supplier numbers and cities for suppliers who supply part P2)

เขียนเป็นลำดับคำสั่งของภาษาพีซีแอลดีเอ็มพีเอสได้ดังนี้

```
T1 = SELECT SP [P# = "P2"]
T2 = JOIN S [S#] T1
T3 = PROJECT T2 [S#, CITY]
DISPLAY T3
```

อาจเขียนแทนด้วยหนึ่งข้อความสั่งของภาษาแคลคูลัสสัมพันธ์ SQL ได้ดังนี้

```
SELECT S#, CITY
FROM S, SP
WHERE S.S# = SP.S#
AND SP.P# = "P2"
```

หมายเหตุ

จะเห็นว่า ข้อความสั่งที่เขียนด้วยภาษา SQL ผู้ใช้เพียงแต่กำหนดคุณลักษณะของเซตของข้อมูลที่ต้องการ ส่วนคำสั่งที่เขียนด้วยภาษาพีซีแอลดีเอ็มพีเอสจะประกอบด้วยคำสั่งชุดหนึ่งซึ่งเรียงลำดับตามการดำเนินงานที่ต้องกระทำเพื่อให้ได้มาซึ่งคำตอบที่ต้องการ

2.5 ภาษาสอบถามฐานข้อมูล QBE

QBE จัดเป็นภาษาฐานข้อมูลเคลคูลัสสัมพันธ์สำหรับค่าเนิงงานกับฐานข้อมูลเชิงสัมพันธ์ ภาษาหนึ่งที่มีวากสัมพันธ์ของภาษาง่ายแต่สามารถค่าเนิงงานกับคำถามในรูปแบบที่ซับซ้อนได้มาก โดยมีรูปแบบของข้อความสั่งในรูปแบบที่อ่านความสะดวกแก่ผู้ใช้ในการทำงานโต้ตอบกับจอภาพ (Screen oriented structure) ทั้งในการสอบถามฐานข้อมูล การกำหนดโครงสร้างฐานข้อมูล การค่าเนิงงานกับฐานข้อมูล และการควบคุมฐานข้อมูล [Zloof (1975)]

QBE ใช้ตารางข้อมูล 2 มิติช่วยในการเขียนข้อความสั่งสอบถามข้อมูล โดยให้ผู้ใช้ใส่ฟังก์ชันการค่าเนิงงาน ตัวค่าเนิงงาน "ค่าคงที่" (Constant) และ "ตัวอย่าง" (Example) ของข้อมูลที่ต้องการถามหาลงในตาราง ผลลัพธ์ที่ได้จากการค่าเนิงงานจะถูกแสดงออกมาในรูปแบบของตารางข้อมูลเช่นเดียวกัน แนวคิดพื้นฐานของ QBE คือ ความแตกต่างระหว่าง "ค่าคงที่" และ "ตัวอย่าง" โดย "ตัวอย่าง" จะแสดงถึงตัวอย่างของคำตอบที่เป็นไปได้จากการสอบถาม ในขณะที่ "ค่าคงที่" ใช้ในการกำหนดเงื่อนไขของการสอบถาม [Luk (1984)] ลักษณะที่แตกต่างกันของการเขียนสัญลักษณ์ "ค่าคงที่" และ "ตัวอย่าง" ในตาราง 2 มิติเพื่อการสอบถามข้อมูล คือ "ตัวอย่าง" จะมีเส้นตรงใต้สัญลักษณ์ ส่วน "ค่าคงที่" ไม่มี เช่น ต้องการสอบถามข้อมูลที่มีใจความว่า "ต้องการรหัสของsupplier สำหรับ supplier ที่อยู่ในเมืองชื่อ Paris" อาจเขียนรูปแบบของการสอบถามได้ ดังนี้

S	S#	SNAME	STATUS	CITY
	<u>p.s7</u>			Paris

p. เป็นฟังก์ชันการดำเนินการเพื่อแสดงหรือพิมพ์ค่าข้อมูลของสดมภ์ที่ต้องการ (ที่ p. อยู่) s7 คือ "ตัวอย่าง" แสดงถึงตัวอย่างของค่าตอบที่เป็นไปได้จากการสอบถาม ส่วน Paris คือ "ค่าคงที่" แสดงถึงค่าในเงื่อนไขที่ต้องการสอบถาม (ในที่นี้คือ CITY='Paris') รูปแบบของการสอบถามดังกล่าวถ่ายทอดออกมาเป็นข้อความได้ว่า "แสดงรหัส (เช่น "s7") ของ supplier ทุกคนที่อยู่ในเมือง Paris" โดย "s7" ไม่จำเป็นแต่จะปรากฏในเซตของผลลัพธ์หรือในตารางข้อมูล S ก็ได้ "ตัวอย่าง" อาจเป็นค่าอะไรก็ได้ เช่น pig, 7 หรือ x เป็นต้น โดยไม่ทำให้ความหมายของการสอบถามเสียไป ฟังก์ชันการดำเนินการอื่น ๆ เช่น ao. do. all. gr. และ un. เป็นฟังก์ชันที่ใช้ในการเรียงลำดับข้อมูลจากน้อยไปมาก เรียงลำดับข้อมูลจากมากไปน้อย ฟังก์ชันที่เกี่ยวกับการรวมสมาชิกภายในเซต ฟังก์ชันที่ใช้ในการจัดกลุ่ม และฟังก์ชันสำหรับจัดข้อมูลที่มีค่าซ้ำออกตามลำดับ นอกจากนี้ยังมี Built-in function ที่ใช้หาค่าต่าง ๆ อีก คือ cnt. sum. avg. max. min. std. และ var. ซึ่งเป็นฟังก์ชันที่ใช้กับจำนวนข้อมูล หาผลรวม หาค่าเฉลี่ย หาค่าสูงสุด หาค่าต่ำสุด หาค่าส่วนเบี่ยงเบนมาตรฐาน และหาค่าความแปรปรวนของข้อมูล ตามลำดับ

ส่วนตัวดำเนินการเปรียบเทียบประกอบด้วยสัญลักษณ์ "!" (ไม่เท่ากับ), "=" (เท่ากับ), ">" (มากกว่า), "<" (น้อยกว่า), ">=" (มากกว่าหรือเท่ากับ) และ "<=" (น้อยกว่าหรือเท่ากับ) ใช้เปรียบเทียบค่าของสดมภ์ที่สัญลักษณ์เปรียบเทียบอยู่กับค่าทางขวามือของสัญลักษณ์ สำหรับสัญลักษณ์ "=" อาจเขียนหรือไม่เขียนก็ได้ ดังตัวอย่างคำว่า "Paris" ในสดมภ์ชื่อ CITY ข้างต้น หมายความว่าค่า CITY เท่ากับ "Paris"

การเขียนข้อความสั่งสอบถามของ QBE โดยการให้ "ค่าคงที่" "ตัวอย่าง" ฟังก์ชัน คำนิยามและ Built-in function สามารถสื่อความหมายต่อผู้ใช้ได้ เนื่องจากการเขียนฟังก์ชันและค่าต่าง ๆ กระทำบนสดมภ์ของตารางสอบถามทำให้มองเห็นตำแหน่งและการเชื่อมโยงได้ชัดเจน ในขณะที่การเขียนข้อความสั่งสอบถามของ SQL ต้องเขียนเป็นคำสั่งบนบรรทัดเดียวทำให้การสื่อความหมายไม่ชัดเจนเหมือนอย่าง QBE

M.H. Zloof ได้ออกแบบการดำเนินการของ QBE ซึ่งผู้ใช้สามารถเขียนชุดข้อความสั่งสอบถามโดยการให้ฟังก์ชันการดำเนินการ ค่าคงที่ ตัวอย่าง และ Built-in function เพื่อให้สอบถามข้อมูลจากตารางข้อมูลต่าง ๆ ในฐานะข้อมูล ผู้วิจัยได้ศึกษารูปแบบและการ

ทำงานของฟังก์ชันการดำเนินงานเหล่านี้ พร้อมทั้งปรับปรุง เพิ่มเติมหลายประการ ดังนี้

ก. ฟังก์ชันการดำเนินงาน gr.

M.M. Zloof ใช้เส้นคู่ใต้ค่าคงที่ เพื่อให้แสดงรูปแบบของการจัดกลุ่มข้อมูล เช่น แสดงผลรวมปริมาณของ part ทั้งหมดของแต่ละ supplier ซึ่ง สามารถเขียนด้วยข้อความสั่งสอบถามดังนี้

SP	S#	P#	QTY
	p. <u>SX</u>		p.sum.all.

การใช้รูปแบบดังกล่าวไม่สามารถสื่อความหมายต่อผู้ใช้ในการเขียนข้อความสั่งสอบถาม ผู้วิจัยจึงใช้ฟังก์ชันการดำเนินงาน gr. แทนรูปแบบดังกล่าว เพื่อจัดกลุ่มแถวข้อมูล ดังนี้

SP	S#	P#	QTY
	p.gr. <u>SX</u>		p.sum.all.

ข. ฟังก์ชันการดำเนินงาน all.

M.M. Zloof ใช้ฟังก์ชัน all. ในการสอบถามข้อมูลเพื่อเปรียบเทียบชุดข้อมูล เช่น ต้องการแสดงรหัสของ supplier สำหรับ supplier ซึ่งอย่างน้อยส่งทุก part ที่มีค่า WEIGHT น้อยกว่า 19 สามารถเขียนข้อความสั่งสอบถามได้ดังนี้

SP	S#	P#	QTY
	p.gr.sx	[all.px] *	

P	P#	PNAME	COLOR	WEIGHT	CITY
	all.px			<19	

นิพจน์ all.px แสดงถึงกลุ่มของข้อมูลซึ่งอาจมีค่าซ้ำกันได้

สัญลักษณ์ $[\text{all.px}]$ แสดงความหมายว่า ในกลุ่มข้อมูลนี้ อย่างน้อยมีกลุ่มข้อมูลที่ได้จาก all.px อยู่ นั่นคือ $\text{all.px} \subseteq [\text{all.px}]$

ผู้วิจัยได้เปลี่ยนรูปแบบดังกล่าวด้วยการยกเลิกการใช้สัญลักษณ์ $[\text{all.px}]$ และใช้ $[\text{all.px}]$ แทน นั่นคือ $\text{all.px} \subseteq [\text{all.px}]$

ค. Built-in function

Built-in function ซึ่ง H.M. Zloof ได้นำเสนอไว้ด้วยกัน 5 ฟังก์ชันมาตรฐาน คือ cnt., sum., avg., max. และ min. ผู้วิจัยได้เพิ่มเติมฟังก์ชันที่ใช้ในการคำนวณค่าทางสถิติ 2 ฟังก์ชัน คือ std. และ var. ซึ่งใช้ในการหาค่าส่วนเบี่ยงเบนมาตรฐาน และ ค่าความแปรปรวนของชุดข้อมูล ตามลำดับ เนื่องจากเป็นค่าที่ใช้กันมากในทางสถิติเช่นเดียวกับค่าผลรวม และ ค่าเฉลี่ย

นอกจากนี้ ผู้วิจัยได้ศึกษารูปแบบข้อความสั่ง วิธีการดำเนินงานและการพัฒนาโปรแกรมชื่อ iqbe (Instructional query by example) ซึ่งออกแบบและพัฒนาโดย Dr. Ken J. McDonell และ Rojer Luk และทำงานภายใต้ระบบปฏิบัติการ Ultrix-32 บนเครื่องคอมพิวเตอร์ตระกูล VAX-11 เพื่อเป็นแนวทางในการออกแบบและ

พัฒนาตัวแปรภาษา QBE ในงานวิจัยนี้ โดยปรับปรุงในส่วนของการรับข้อความสิ่งสอบถาม และการแสดงผลทางจอภาพเล็กน้อย พร้อมทั้งเพิ่มฟังก์ชันการดำเนินการที่โปรแกรม iqbe ไม่มี

ต่อไปนี้เป็นตัวอย่างแสดงการเขียนภาษาสอบถามฐานข้อมูล QBE แบบต่าง ๆ คำตอบในตัวอย่างที่สามารถแปลงจากภาษาสอบถามฐานข้อมูล QBE เป็นภาษาพีชคณิตสัมพันธ์ที่สมนัยกันได้ก็จะแสดงการสอบถามด้วยภาษาพีชคณิตสัมพันธ์ที่สมนัยด้วย

การสอบถามชื่อตารางข้อมูลในฐานข้อมูล

ตัวอย่างที่ 2.2

แสดงรายชื่อของตารางข้อมูลทั้งหมดที่มีอยู่ในฐานข้อมูลที่กำลังดำเนินการ

p.	

คำอธิบาย

1. การสอบถามชื่อตารางข้อมูลจากฐานข้อมูลจะต้องกระทำบนส่วนของชื่อตารางข้อมูลบนตารางว่าง ผลการสอบถามจะได้ชื่อของตารางข้อมูลทั้งหมดที่อยู่ในฐานข้อมูลที่กำลังทำงานอยู่
2. p. อาจตามด้วย "ตัวอย่าง" ใดๆ ก็ได้ เช่น p.x ซึ่งให้ความหมายอย่างเดียวกัน

ผลการสอบถาม

Relation name

S

P

SP

การสอบถามชื่อตารางข้อมูลพร้อมรายชื่อแอตทริบิวต์

ตัวอย่างที่ 2.3

แสดงรายชื่อของตารางข้อมูลพร้อมรายชื่อแอตทริบิวต์ทั้งหมดที่มีอยู่ในฐานข้อมูลที่กำลังดำเนินงาน

p.p.	

คำอธิบาย

- เช่นเดียวกัน การสอบถามต้องกระทำบนตารางว่างบนส่วนของชื่อตารางข้อมูล ผลการสอบถามจะได้ชื่อตารางข้อมูลพร้อมทั้งชื่อแอตทริบิวต์ทั้งหมดของแต่ละตาราง
- การสอบถามชื่อตารางข้อมูลและแอตทริบิวต์อาจเขียนเป็น p.x p. ก็ได้

ผลการสอบถาม

<u>Relation name</u>	<u>Attribute name</u>			
S	S#	SNAME	STATUS	CITY
P	P#	PNAME	COLOR	WEIGHT CITY
SP	S#	P#	QTY	

การสอบถามรายละเอียดเฉพาะตารางข้อมูลที่ต้องการ

ตัวอย่างที่ 2.4

แสดงรายละเอียดทุกอย่างของตารางข้อมูล S

S p.	

คำอธิบาย

เป็นการสอบถามรายละเอียดต่าง ๆ ของตารางข้อมูลชื่อ "S" จากปกานุกรมฐานข้อมูลที่ถูกสร้างด้วยภาษากำหนดโครงสร้าง/นิยาม

ผลการสอบถาม

Database name : sale

Relation name : S

Create date : 03-04-91

Tuple width : 24

Relation type : *

Last update : 03-04-91

Number of tuple : 5

Att name	Key-type	Format	Length
SNUM	P	C	2
SNAME	N	C	10
STATUS	N	I	2
CITY	N	C	10

การสอบถามข้อมูลอย่างง่ายตัวอย่างที่ 2.5

แสดงรหัสของ part สำหรับทุก part ที่ถูกส่ง

(Get part number for all parts supplied.)

SP	S#	P#	QTY
		p.px	

คำอธิบาย

px ไม่จำเป็นต้องมีก็ได้ และรหัสของ part ที่ซ้ำกันจะไม่แสดงให้เห็น

ภาษาพีซีดีดีสัมพันธ์ที่สัมพันธ์กัน

T = PROJECT SP [P#]

DISPLAY T

ผลการสอบถาม

SP	P#
	P1
	P2
	P3
	P4
	P5
	P6

ตัวอย่างที่ 2.6

แสดงรายละเอียดของข้อมูลทั้งหมดของทุก supplier
(Get full details of all suppliers.)

S	S#	SNAME	STATUS	CITY
	<u>p.sx</u>	<u>p.sn</u>	<u>p.st</u>	<u>p.sc</u>

หรือเขียนรูปแบบการสอบถามให้สั้นได้ดังนี้

S	S#	SNAME	STATUS	CITY
P.				

ภาษาพีซีดีบีเอ็มที่สนับสนุน

DISPLAY S

ผลการสอบถาม

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

การสอบถามข้อมูลโดยใช้เงื่อนไข "และ" / "หรือ"

ตัวอย่างที่ 2.7

แสดงรหัสของ supplier สำหรับ supplier ที่อยู่ใน Paris และ STATUS > 20

(Get supplier numbers for suppliers in Paris with status > 20.)

S	S#	SNAME	STATUS	CITY
	p.sx		>20	Paris

คำอธิบาย

1. การสร้างเงื่อนไข "STATUS > 20" ใช้ตัวดำเนินการเปรียบเทียบ ">" เพื่อระบุว่ารหัสของ supplier ที่ได้จะต้องมีค่าของ STATUS มากกว่า 20
2. โดยทั่วไปตัวดำเนินการ "=", "!=", "<", "<=" และ ">=" สามารถใช้ได้ ในทำนองเดียวกันนี้ แต่ถ้าไม่มีตัวดำเนินการใด ๆ ก็ะหมายถึง "="

ภาษาที่ผลิตสัมพันธ์ที่สัมพันธ์กัน

T1 = SELECT S [STATUS > 20]

T2 = SELECT S [CITY = "Paris"]

T3 = T1 INTERSECT T2

T4 = PROJECT T3 [S#]

DISPLAY T4

ผลการสอบถาม

S	S#
	S3

ตัวอย่างที่ 2.8

แสดงรหัสของ supplier สำหรับ supplier ที่ส่งทั้ง Part P1 และ P2

(Get supplier numbers for supplier who supply both part P1 and part P2.)

SP	S#	P#	QTY
	p.sx	P1	
	sx	P2	

ภาษาที่คิดค้นขึ้นที่สมมติกัน

T1 = SELECT SP [P# = "P1"]

T2 = SELECT SP [P# = "P2"]

T3 = T1 INTERSECTION T2

T4 = PROJECT T3 [S#]

DISPLAY T4

ผลการสอบถาม

SP	S#
	S1
	S2

ตัวอย่างที่ 2.9

แสดงรหัสของ supplier สำหรับ supplier ที่อยู่ใน Paris หรือ ที่มีค่า STATUS > 20

(Get supplier numbers for suppliers who are located in Paris or have status > 20 (or both).)

S	S#	SNAME	STATUS	CITY
	p.sx			Paris
	p.sy		>20	

ภาษาพีซีดีดีสัมพันธ์ที่สมนัยกัน

T1 = SELECT S [CITY = "Paris"]

T2 = SELECT S [STATUS > 20]

T3 = T1 UNION T2

T4 = PROJECT T3 [S#]

DISPLAY T4

ผลการสอบถาม

S	S#
	S2
	S3
	S5

ข้อสังเกต

1. การสอบถามข้อมูลที่มีเงื่อนไขมากกว่าหนึ่งเงื่อนไข อาจเชื่อมเงื่อนไขแต่ละเงื่อนไขเข้าด้วยกันด้วยตัวดำเนินการทางตรรกะ "และ" หรือ "หรือ"
2. ตัวดำเนินการทางตรรกะ "และ" และ "หรือ" ในภาษาสอบถามฐานข้อมูล QBE ไม่มีการเขียนแบ่งบอกอย่างชัดเจน แต่จะเป็นไปโดยปริยายตามรูปแบบของภาษาดังแสดงในตัวอย่างที่ 2.6 - 2.8
3. การสอบถามข้อมูลที่มีเงื่อนไขมากกว่าหนึ่งเงื่อนไข โดยเชื่อมเงื่อนไขแต่ละเงื่อนไขเข้าด้วยกันด้วยตัวดำเนินการทางตรรกะ "และ" อาจทำได้สองรูปแบบคือ
 - การใช้ "ค่าคงที่" มากกว่าหนึ่งค่าบนแถวเดียวกับฟังก์ชัน p. บนตารางใด ๆ (ตัวอย่าง 2.7)
 - การใช้ "ตัวอย่าง" ที่เหมือนกันมากกว่าหนึ่งค่าซึ่งอยู่ในสดมภ์เดียวกับฟังก์ชัน p. แต่อยู่บนแถวที่ต่างกันภายในตารางเดียวกัน โดยฟังก์ชัน p. จะมีเพียงหนึ่งฟังก์ชันและอยู่แถวใดก็ได้ที่มี "ตัวอย่าง" อยู่ แต่ละแถวที่มี "ตัวอย่าง" จะมีเงื่อนไขของการสอบถามโดยการใช้ "ค่าคงที่" หนึ่งหรือมากกว่าหนึ่งค่า (ตัวอย่าง 2.8)
4. การสอบถามข้อมูลที่มีเงื่อนไขมากกว่าหนึ่งเงื่อนไข โดยเชื่อมเงื่อนไขแต่ละเงื่อนไขเข้าด้วยกันด้วยตัวดำเนินการทางตรรกะ "หรือ" ทำได้โดยการสร้างเงื่อนไขบนแถวที่ต่างกัน ซึ่งทุกแถวจะมีฟังก์ชัน p. ของสดมภ์ที่เหมือนกันและจะต้องมี "ตัวอย่าง" ที่ต่างกัน (ตัวอย่าง 2.9)

ตัวอย่างที่ 2.10

แสดงรหัสของ supplier สำหรับ supplier ซึ่งมีอย่างน้อย 1 part ที่ส่งถูกส่งโดย supplier "S2"

(Get supplier numbers for suppliers who supply at least one part supplied by supplier S2.)

SP	S#	P#	QTY
	<u>P.SX</u>	<u>PX</u>	
	S2	<u>PX</u>	

ภาษาพีชคณิตสัมพันธ์ที่สมนัยกัน

T1 = SELECT SP [S# = "S2"]

T2 = JOIN SP [P#] T1

T3 = PROJECT T2 [S#]

DISPLAT T3

ผลการสอบถาม

SP	S#
	S1
	S2
	S3
	S4

ตัวอย่างที่ 2.11

แสดงรหัสของ part ที่ถูกส่งโดย supplier มากกว่าหนึ่งราย

(Get part numbers for all parts supplied by more than one supplier.)

SP	S#	P#	QTY
	<u>sx</u>	p. <u>px</u>	
	<u>!=sx</u>	<u>px</u>	

คำอธิบาย

Part ที่เป็นคำตอบจะเป็น Part ที่มีค่า P# ปรากฏในตาราง SP มากกว่าหนึ่งแถวข้อมูล (แสดงในตารางด้วยค่า px ที่ปรากฏในตารางด้วยค่า 2 แถวที่มีค่า S# ไม่เท่ากัน)

ภาษาพีชคณิตสัมพันธ์ที่สมนัยกัน

T1 = JOIN SP [S# != S#] SP

T2 = JOIN SP [P#] SP

T3 = T1 INTERSECT T2

T4 = PROJECT T3 [P#]

DISPLAY T4

ผลการสอบถาม

SP	P#
	P1
	P2
	P4
	P5

ตัวอย่างที่ 2.12

แสดงชื่อของ supplier ที่ส่ง part รหัส "P2"

(Get supplier names for suppliers who supply part P2.)

S	S#	SNAME	STATUS	CITY	SP	S#	P#	QTY
	<u>sx</u>	<u>p.sn</u>				<u>sx</u>	P2	

คำอธิบาย

"ตัวอย่าง" sx ถูกใช้เป็นตัวเชื่อมระหว่างตารางข้อมูล S และ SP แสดงให้เห็นว่า ชื่อของ supplier ที่ต้องการเป็นคำตอบ เป็นชื่อที่มีค่า S# ปรากฏในตาราง SP ของแถวข้อมูลที่มีค่า P# เป็น "P2"

ภาษาพีซีดีบีเอ็มพื้นฐานที่สัมพันธ์กัน

T1 = SELECT SP [P# = "P2"]

T2 = JOIN S [S#] T1

T3 = PROJECT T2 [SNAME]

DISPLAY T3

ผลการสอบถาม

S	SNAME
	Smith
	Jones
	Blake
	Clark

ตัวอย่างที่ 2.13

แสดงชื่อของ supplier ซึ่งอย่างน้อยที่สุด part ที่ส่งมี part. สีแดงอยู่

(Get supplier names for suppliers who supply at least one red part)

S	S#	SNAME	STATUS	CITY	SP	S#	P#	QTY
	<u>sx</u>	p. <u>sn</u>				<u>sx</u>	<u>px</u>	

P	P#	PNAME	COLOR	WEIGHT	CITY
	<u>px</u>		Red		

คำอธิบาย

"ตัวอย่าง" sx และ px เป็นตัวเชื่อมระหว่างตาราง S กับ SP และ SP กับ P ตามลำดับ แสดงให้เห็นว่า "ชื่อ supplier ที่ต้องการเป็น supplier ที่มีค่า S# ปรากฏอยู่ในตาราง SP ซึ่งแสดงว่า supplier นี้ส่ง part โดย part ที่ส่งมีค่า P# ปรากฏอยู่ในตาราง P และมีค่าของ COLOR ที่สมนัยกันเป็นสีแดง (Red)"

ภาษาพีชคณิตสัมพันธ์ที่สมนัยกัน

T1 = SELECT P [COLOR = "Red"]

T2 = JOIN S [S#] SP

T3 = JOIN T2 [P#] T1

T4 = PROJECT T3 [SNAME]

DISPLAY T4

ผลการสอบถาม

S	SNAME
	Smith
	Jones
	Clark

ตัวอย่างที่ 2.14

แสดงชื่อของ supplier ที่ไม่ได้ส่ง part ซึ่งมีรหัส "P1"

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

S	S#	SNAME	STATUS	CITY	SP	S#	P#	QTY
	<u>SX</u>	p. <u>sn</u>			!	<u>SX</u>	P1	

คำอธิบาย

- สัญลักษณ์ "!" แทนการปฏิเสธ (Negation)
- ชื่อ supplier ซึ่งจะเป็นคำตอบต้องเป็น supplier ที่มีค่า S# ไม่ปรากฏขึ้นในแถวข้อมูลของตาราง SP ที่มีค่า P# เป็น "P1"

ภาษาพีชคณิตสัมพันธ์ที่สมมูลกัน

T1 = SELECT SP [P# = "P1"]

T2 = PROJECT T1 [SNUM]

T3 = JOIN S [S#] T2

T4 = S MINUS T3

T5 = PROJECT T4 [SNAME]

DISPLAY T5

ผลการสอบถาม

S	SNAME
	Blake
	Clark
	Adams

ตัวอย่างที่ 2.15

สำหรับแต่ละ part ที่ถูกส่ง ให้แสดงรหัสของ part และชื่อของเมืองของ supplier ที่ส่ง part นั้น ๆ

(For each part supplied, get part number and names of all cities supplying the part.)

S	S#	SNAME	STATUS	CITY	SP	S#	P#	QTY
	<u>sx</u>			<u>p.sc</u>		<u>sx</u>	<u>p.px</u>	

ภาษาที่คิดสัมพันธ์กัน

T1 = JOIN S [S#] SP

T2 = PROJECT T1 [CITY,P#]

DISPLAY T2

ผลการสอบถาม

Result	CITY	P#
	London	P1
	London	P2
	London	P3
	London	P4
	London	P5
	London	P6
	Paris	P1
	Paris	P2

การสอบถามข้อมูลโดยใช้ Built-in function

ตัวอย่างที่ 2.16

แสดงจำนวน supplier ทั้งหมดที่มีอยู่

S	S#	SNAME	STATUS	CITY
	p.ent.all. <u>sx</u>			

คำอธิบาย

1. all. เป็น Built-in function ซึ่งดำเนินการรวบรวมเป็นชุดข้อมูลค่าของ สดมภ์ที่ all. อยู่ (ในที่นี้จะคือค่า S# ทั้งหมดในตาราง S)
2. cnt. เป็น Built-in function ซึ่งนับจำนวนของข้อมูลทั้งหมดของชุดข้อมูล ที่ได้จากการดำเนินการ all.
3. สามารถเขียน "p.ent.all." แทน "p.ent.all.sx" โดยความหมายยังคงเหมือนเดิม

ผลการสอบถาม

S	S#.cnt
	5

ตัวอย่างที่ 2.17

นับจำนวน supplier ทั้งหมดที่กำลังส่ง part

SP	S#	P#	QTY
	p.cent.un.all.sx		

คำอธิบาย

การทำงานของฟังก์ชัน cnt. ในข้อนี้จะนับค่าของชุดข้อมูลที่ได้รับการดำเนินงาน all.sx ที่มีค่าไม่ซ้ำกัน (จากการใช้ฟังก์ชัน un. (unique) ในการหาค่าซ้ำจากการดำเนินงาน all.sx)

ผลการสอบถาม

SP	S#.cnt
	4

ตัวอย่างที่ 2.18

หาผลรวมของค่าปริมาณของ part ทุก part ซึ่งถูกส่งโดย supplier ที่มีรหัส "S1"

SP	S#	P#	QTY
	S1		p.sum.all.100

1. การทำงานของฟังก์ชัน sum. ในข้อนี้จะบวกค่าของชุดข้อมูลที่ได้จาก all.100
2. จากการสอบถามด้วยภาษา QBE ข้างต้นสามารถถ่ายทอดเป็นข้อความได้ว่า "สำหรับ supplier ที่มีรหัส "S1" ให้หาผลบวกของค่าปริมาณของ part ทุก part ที่ "S1" เป็นผู้ส่ง

ผลการสอบถาม

SP	QTY.sum
	1300

ตัวอย่างที่ 2.19

แสดงผลรวมปริมาณของ part ทั้งหมดที่แต่ละ supplier ส่ง

SP	S#	P#	QTY
	p.gr. <u>sx</u>		p.sum.all.

คำอธิบาย

1. gr. เป็น Built-in function ดำเนินการจัดกลุ่มแถวข้อมูลตามค่าสมาชิกที่ gr. อยู่
2. sum. เป็น Built-in function ดำเนินการบวกค่าทั้งหมดที่ได้จากการดำเนินงาน all. กับทุกกลุ่มของแถวข้อมูลที่ได้จากข้อ 1

ผลการสอบถาม

SP	S#	QTY.sum
	S1	1300
	S2	700
	S3	200
	S4	900

การสอบถามข้อมูลโดยการเปรียบเทียบชุดข้อมูล

ตัวอย่างที่ 2.20

แสดงรหัสของ supplier ที่ส่ง part ทุก part ที่มีอยู่

(Get supplier numbers for suppliers who supply all parts.)

SP	S#	P#	QTY
	p.gr.sx	all.px	

P	P#	PNAME	COLOR	WEIGHT	CITY
	all.px				

คำอธิบาย

1. gr.sx ในสัดมภ์ S# เป็นการจับกลุ่มค่าสัดมภ์อื่น ๆ ตามค่า S# แต่ละค่า
2. all.px ในสัดมภ์ P# ของตารางข้อมูล SP และ P เป็นการเปรียบเทียบชุดของค่า P# ในตาราง SP (ซึ่งถูกจัดเป็นกลุ่มตามค่าแต่ละ S# ในข้อ 1) ว่าเท่ากับชุดของค่า S# ในตาราง P

ภาษาพีชคณิตสัมพันธ์ที่สมนัยกัน

T1 = PROJECT SP [S#,P#]

T2 = PROJECT P [P#]

T3 = DIVIDE T1 [(P#) = (P#)] T2

DISPLAY T3

ผลการสอบถาม

SP	S#
	S1

ตัวอย่างที่ 2.21

แสดงรหัสของ supplier สำหรับแต่ละ supplier ซึ่งอย่างน้อยส่งทุก part ที่มีค่า WEIGHT น้อยกว่า 19

SP	S#	P#	QTY
	p.gr.sx	[all.px]	

P	P#	PNAME	COLOR	WEIGHT	CITY
	all.px			<19	

คำอธิบาย

1. ข้อความในตาราง P ให้ค่า P# ชุดหนึ่งซึ่งเป็น P# ของแถวข้อมูลที่มีค่า WEIGHT น้อยกว่า 19
2. all.px \subset [all.px]

ภาษาที่ผลิตสัมพันธ์ที่สัมพันธ์กัน

T1 = PROJECT SP [S#,P#]

T2 = SELECT P [WEIGHT < 19]

T3 = PROJECT T2 [P#]

T4 = DIVIDE T1 [(P#) >= (P#)] T3

DISPLAY T4

ผลการสอบถาม

SP	S#
	S1

ตัวอย่างที่ 2.22

แสดงรหัสของ supplier สำหรับแต่ละ supplier ซึ่งส่ง part ที่มี WEIGHT น้อยกว่า

19

SP	S#	P#	QTY
	p.gr.sx	all.px	

P	P#	PNAME	COLOR	WEIGHT	CITY
	[all.px]			<19	

ภาษาพีซีดีดีสัมพันธ์ที่สมมูลกัน

T1 = SELECT P [WEIGHT < 19]

T2 = PROJECT SP [S#,P#]

T3 = PROJECT T1 [P#]

T4 = DIVIDE T2 [(P#) <= (P#)] T3

DISPLAY T4

ผลการสอบถาม

SP	S#
	S2
	S3
	S4

ตัวอย่างที่ 2.23

แสดงรหัสของ supplier ที่ในบรรดา part ที่ส่งมีทุก part ที่ supplier S2 ส่ง

(Get supplier numbers for suppliers who supply at least all those part supplied by supplier S2.)

SP	S#	P#	QTY
	p.gr.sx	[all.px]	
	S2	all.px	

ภาษาที่ซัดคณิศสัมพันธ์ที่สมนัยกัน

T1 = SELECT SP [S# = "S2"]

T2 = PROJECT SP [S#,P#]

T3 = PROJECT T1 [P#]

T4 = DIVIDE T2 [(P#) >= (P#)] T3

DISPLAY T4

ผลการสอบถาม

SP	S#
	S1
	S2

ตัวอย่างที่ 2.24

แสดงรหัสและ STATUS ของ supplier ที่อยู่ในเมือง Paris โดยเรียงลำดับค่างาน
น้อยไปหามากตามค่างาน

(Get supplier number and status for suppliers in Paris, in ascending order of supplier number.)

S	S#	SNAME	STATUS	CITY
	p.ao.all.		p.	Paris

คำอธิบาย

1. ao. เป็นฟังก์ชันในการเรียงลำดับผลลัพธ์ข้อมูลที่ได้จากการสอบถามจากค่าน้อยไปหาค่ามาก ก่อนที่จะแสดงออกทางจอภาพ
2. ตรงกันข้ามกับ ao. ฟังก์ชัน do. จะทำให้มีการเรียงลำดับจากค่ามากไปหาค่าน้อย

ผลการสอบถาม

S	S#	STATUS
	S2	10
	S3	20

การสร้างตัวแปลภาษาคิวโปรแกรม LEX และ YACC

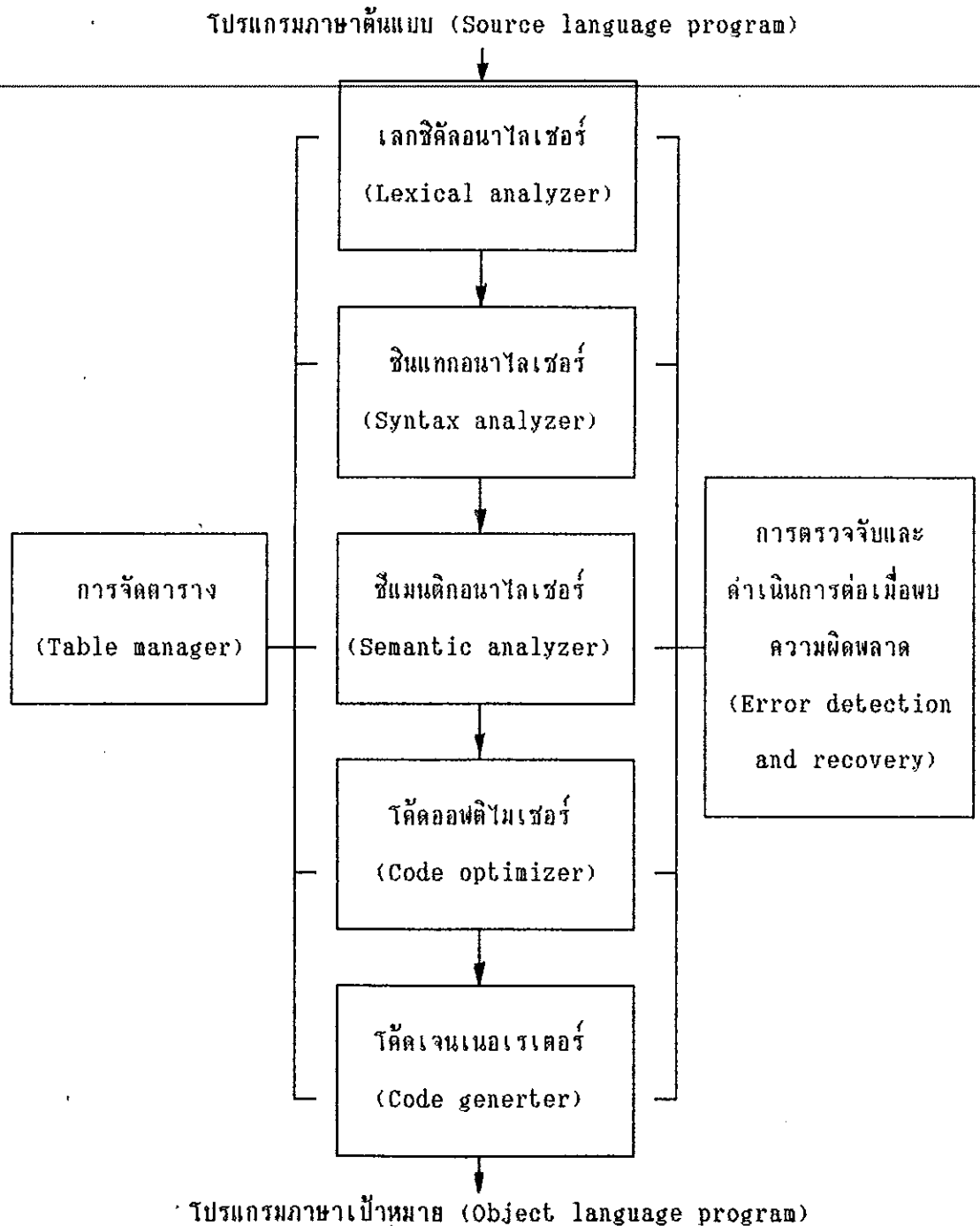
3.1 ตัวแปลภาษา

เมื่อแรกเริ่มที่มีการใช้คอมพิวเตอร์ การเขียนโปรแกรมเพื่อให้สามารถติดต่อกับเครื่องคอมพิวเตอร์ได้ ผู้เขียนโปรแกรมจำเป็นต้องเขียนด้วยภาษาเครื่อง (Machine language) ซึ่งเป็นลำดับของเลข 0 และ 1 ที่เรียงต่อกันในรูปแบบบิต (Bit pattern) ซึ่งผู้เขียนโปรแกรมมักประสบกับความยุ่งยากอันเนื่องมาจากข้อผิดพลาดของการแทนค่าสิ่งในรูปแบบบิตเสมอ อีกทั้งยังยากต่อการเปลี่ยนแปลงปรับปรุงโปรแกรมในภายหลัง จึงมีการพัฒนาภาษาแอสเซมบลี (Assembly language) ขึ้นมาเพื่อแก้ปัญหาดังกล่าว อย่างไรก็ตามความยุ่งยากในการเขียนโปรแกรมด้วยภาษาแอสเซมบลีก็ยังคงมีอยู่ เนื่องจากภาษาแอสเซมบลียังคงมีคำสั่งต่าง ๆ ที่เป็นการทำงานขั้นพื้นฐานเช่นเดียวกับภาษาเครื่อง จึงมีการพัฒนาต่อมาเป็นภาษาระดับสูง (High level language) โดยโปรแกรมที่เขียนด้วยภาษาระดับสูงจะเขียนได้สะดวกและง่ายกว่าเพราะใกล้เคียงภาษามนุษย์ ทำให้มีข้อผิดพลาดน้อยลงและง่ายต่อการปรับปรุงแก้ไขในภายหลัง แต่ภาษาระดับสูงจำเป็นต้องมีตัวแปลภาษา (Translator) เพื่อแปลงโปรแกรมที่เขียนด้วยภาษาระดับสูงให้เป็นภาษาเครื่อง

ตัวแปลภาษา หมายถึง โปรแกรมซึ่งทำหน้าที่แปลโปรแกรมที่เขียนด้วยภาษาระดับสูง เรียกว่า โปรแกรมต้นแบบ (Source program) ไปเป็นโปรแกรมที่อยู่ในรูปของอีกภาษาหนึ่งเรียกว่า โปรแกรมเป้าหมาย (Object program) ถ้าโปรแกรมต้นแบบเป็นโปรแกรมเขียนด้วยภาษาระดับสูง เช่น ฟอรัทแรน (FORTRAN) หรือ พีแอลวัน (PL/1) เป็นต้น ก็จะใช้ตัวแปลภาษาเรียกว่า คอมไพเลอร์ (Compiler) แปลจากภาษาระดับสูงไปเป็นภาษาเครื่องหรือภาษาแอสเซมบลี ส่วนโปรแกรมที่เขียนด้วยภาษาแอสเซมบลีจะใช้ตัวแปลภาษาเรียกว่า แอสเซมเบลเลอร์ (Assembler) แปลภาษาแอสเซมบลีไปเป็นภาษาเครื่อง นอกจากนี้ยังมีตัวแปลภาษาซึ่งมีหน้าที่แปลโปรแกรมต้นแบบไปเป็นรหัสระหว่างกลาง (Intermediate code) แล้วปฏิบัติการกับรหัสระหว่างกลางนั้นโดยตรง เรียกว่า อินเทอร์พรีเตอร์ เช่น ตัวแปลภาษาชื่อเบสิก (BASIC) หรือ คำสั่งที่ใช้ติดต่อกับระบบ

ปฏิบัติการ (Operating system commands) เป็นต้น

โครงสร้างการดำเนินงานของตัวแปลภาษาระดับสูงค่อนข้างซับซ้อน อาจแบ่งขั้นตอนการดำเนินงานได้ 5 ขั้นตอน ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการดำเนินงานของตัวแปลภาษาระดับสูง

จากรูปที่ 3.1 อธิบายขั้นตอนต่าง ๆ โดยสรุปได้ดังนี้

ก. ขั้นตอนเลขชั้คคณาไลเซอร์ ทำหน้าที่อ่านอักขระจากโปรแกรมภาษาต้นแบบ แล้วจัดการแยกอักขระเหล่านั้นออกเป็นกลุ่ม ๆ เรียกว่า โทเคน (Token) ตามหลักภาษาต้นแบบที่กำหนดไว้ เช่น คำหลัก ชื่อตัวแปร ค่าคงที่ ตัวดำเนินการ เป็นต้น

ข. ขั้นตอนซินแทกอนาไลเซอร์ ทำหน้าที่ตรวจสอบว่าโทเคนที่ได้จากขั้นตอนเลขชั้คคณาไลเซอร์ได้เข้ามาเขียนเรียงกันถูกต้องตามวากยสัมพันธ์ (Syntax) ของภาษาต้นแบบหรือไม่

ค. ขั้นตอนซีแมนติคอนาไลเซอร์ ทำหน้าที่ตรวจสอบความหมายของภาษาและตีความเพื่อสร้างรหัสระหว่างกลางที่ทำงานเหมือนภาษาต้นแบบ

ง. ขั้นตอนโค้ดคอมไพเลอร์ ทำหน้าที่ปรับปรุงรหัสระหว่างกลางที่ได้จากขั้นตอนซีแมนติคอนาไลเซอร์ให้มีประสิทธิภาพมากขึ้น เช่น การจัดการค่านวที่เข้าข้อและไม่มี ความจำเป็นออกไปเพื่อให้ทำงานได้รวดเร็วขึ้นและเปลืองเนื้อที่ในหน่วยความจำหลักน้อยลง

จ. ขั้นตอนโค้ดเจนเนอเรเตอร์ ทำหน้าที่เปลี่ยนโปรแกรมในรูปแบบระหว่างกลางให้เป็นโปรแกรมภาษาเป้าหมายที่ต้องการ

นอกจากนี้ ตัวแปลภาษายังมีหน้าที่สำคัญอีก 2 ประการคือ การจัดตาราง ซึ่งทำหน้าที่บันทึกรายละเอียดต่าง ๆ ที่ปรากฏในโปรแกรมภาษาต้นแบบไว้ในตารางเพื่อใช้ในการคำนวณงานของขั้นตอนต่าง ๆ ของตัวแปลภาษา และ การตรวจจับและดำเนินการต่อเมื่อพบความผิดพลาดโดยมีข้อความบอกข้อผิดพลาดให้ผู้เขียนโปรแกรมทราบ ตัวแปลภาษาที่ดีควรจะตรวจจับความผิดพลาดให้ได้มากที่สุดและใกล้เคียงกับความผิดพลาดที่เกิดขึ้นมากที่สุด ทั้งจะไม่หยุดการทำงานเมื่อตรวจพบความผิดพลาดด้วย

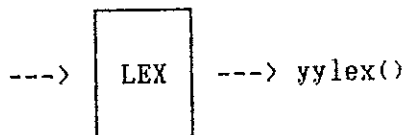
สำหรับตัวแปลภาษาสอบถาม QBE ที่พัฒนาในงานวิจัยนี้เป็นตัวแปลภาษาชนิดที่เรียกว่า อินเทอร์พรีเตอร์ ขั้นตอนการคำนวณงานจะมีเพียง 3 ขั้นตอนแรก โดยในขั้นตอนที่ 3 เป็นการตีความและสร้างรหัสระหว่างกลางเพื่อใช้ในการคำนวณงานตามข้อความสั่งต่าง ๆ ในโปรแกรม

โปรแกรมช่วยการตรวจจับกลุ่มคำ LEX และโปรแกรมช่วยการตรวจไวยากรณ์ YACC เป็นเครื่องมือที่เรียกว่าโปรแกรมอรรถประโยชน์ (Utility program) ของระบบปฏิบัติการ UNIX ที่มีอยู่บนเครื่องมินิคอมพิวเตอร์ VAX-11/785 ซึ่งผู้เขียนโปรแกรมสามารถนำมาใช้ประโยชน์ช่วยในการสร้างตัวแปลภาษาได้รวดเร็วขึ้น

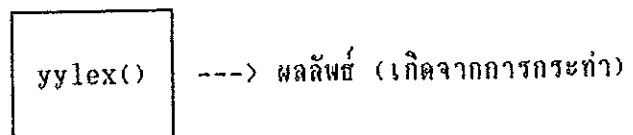
3.2 โปรแกรม LEX

โปรแกรม LEX (Lexical analyzer generator) เป็นโปรแกรมซึ่งอ่านรูปแบบไวยากรณ์หรือรูปแบบวากยสัมพันธ์ของภาษาระดับสูงซึ่งเป็นภาษาต้นแบบและสร้างโปรแกรมย่อยชื่อ `yylex()` เพื่อตรวจจับกลุ่มคำหรือโทเคนจากข้อความสิ่งของโปรแกรมต้นแบบที่อ่านเข้ามาทีละอักขระ พร้อมทั้งปฏิบัติตามคำสั่งบ่งบอกการกระทำซึ่งเขียนต่อจากรูปแบบเหมือน (Pattern matching) ที่ตรวจพบ รูปที่ 3.2 แสดงภาพรวมการดำเนินงานของโปรแกรม LEX และโปรแกรมย่อย `yylex()`

รูปแบบวากยสัมพันธ์ของ
ภาษาต้นแบบในรูปแบบ
นิพจน์ปกติ+การกระทำ



ลำดับอักขระของนิพจน์ปกติ



รูปที่ 3.2 ข้อมูลเข้า/ออก ของ LEX และ `yylex()`

3.2.1 LEX

จากรูปที่ 3.2 จะเห็นว่าข้อมูลเข้าของ LEX คือรูปแบบวากยสัมพันธ์ในรูปนิพจน์ปกติ (Regular expression) และการกระทำ (Action) รวมทั้งคำสั่งอื่น ๆ ที่สามารถเพิ่มเติมเข้าไปได้ของภาษาดั้งเดิม โดยเค้าโครงของรูปแบบข้อมูลเข้าของ LEX ซึ่งเป็นนิพจน์ปกติ + การกระทำมีรูปแบบ ดังนี้

```
{ definitions }
%%
{ rules }
%%
{ user subroutines }
```

โดยที่

{ definitions } เป็นส่วนที่ใช้กำหนดเพิ่มข้อมูลบรรจุโปรแกรมภาษาที่ต้องการใช้ร่วม และกำหนดนิยามของนิพจน์ปกติที่ใช้ในภาษา โดยนิยามของนิพจน์ปกติจะประกอบด้วยชื่อและนิยาม (อยู่ในเครื่องหมาย "[" และ "]") และชื่อของนิพจน์สามารถถูกอ้างอิงถึงได้ในส่วนของ {rules} โดยการเขียนชื่อนิพจน์ภายในเครื่องหมาย "(" และ ")"

{ rules } ประกอบด้วยนิพจน์ปกติและการกระทำที่สามารถเขียนเป็นภาษาซีหรือฟอร์แทรนได้ เช่น การตรวจจ็บบรูปแบบ "integer" เมื่อพบให้แสดงข้อความ "Found integer" สามารถเขียนเป็นกฎได้ดังนี้

```
{integer} printf("Found integer");
```

ในที่นี้ {integer} เป็นนิพจน์ปกติ ส่วนการกระทำคือคำสั่ง printf("Found integer");

(user subroutines) เป็นส่วนที่ผู้ใช้สามารถเขียนคำสั่งภาษาซีหรือภาษาฟอร์แทรนที่มีการทำงานนอกเหนือจากการตรวจจับรูปแบบและการกระทำในส่วนที่เป็น การกระทำของ rules คำสั่งเหล่านี้จะนำไปต่อท้ายในชุดคำสั่ง yylex()

สำหรับนิพจน์ปกติซึ่ง LEX ใช้เป็นตัวกำหนดรูปแบบที่ต้องการตรวจจับประกอบด้วยอักขระที่อยู่ในรูปแบบและตัวดำเนินการ (Operator) สามารถแบ่งการกำหนดอักขระและตัวดำเนินการออกเป็นกลุ่ม ๆ ดังนี้

ก. อักขระในรูปแบบ

อักขระหรือสายอักขระใด ๆ ที่อยู่ในตัวเครื่องหมาย " " จะหมายถึงอักขระหรือสายอักขระนั้น ๆ แม้ว่าอักขระนั้น ๆ จะเป็นตัวดำเนินการก็ตาม ยกเว้น อักขระ หลีก (Escape character) ซึ่งจะกล่าวในข้อ ข. เช่น xyz"++" มีความหมายเหมือนกับ "xyz++" ซึ่งหมายถึงรูปแบบที่ประกอบด้วย อักขร x, y, z และเครื่องหมายบวกสองตัว

ข. อักขระหลีก (\)

มีรูปแบบ \x เมื่อ x เป็นอักขระใด ๆ ต่อไปนี้ n, r, b, f, t, หรือตัวเลขฐานแปด (หมายถึงอักขระ newline, return, backspace, form feed, tab และอักขระที่มีรหัสเท่ากับตัวเลขฐานแปด ตามลำดับ) หรือตัวดำเนินการนั้น ๆ เช่น xyz\+\+ มีความหมายอย่างเดียวกับ xyz"++"

ค. กลุ่มอักขระ

การอ้างถึงกลุ่มอักขระทำได้โดยการใส่ตัวดำเนินการ [] เช่น [abc] หมายถึงอักขระ a หรือ b หรือ c เป็นต้น อักขระในเครื่องหมาย [] จะหมายถึงอักขระนั้น ๆ เสมอ ยกเว้นอักขระ 3 ตัว คือ "-" "^" และ "\" ซึ่งทำหน้าที่เป็นตัวดำเนินการและมีความหมายดังนี้

- ตัวดำเนินการ "-" ใช้เพื่อกำหนดช่วงของอักขระเช่น [a-z0-9_] หมายถึงอักขร a ถึง z หรือ ตัวเลข 0 ถึง 9 หรือ _ เป็นต้น
- ตัวดำเนินการ "^" ต้องปรากฏเป็นตัวแรกหลังเครื่องหมาย "[" ใช้เพื่อกำหนดกลุ่มอักขระที่ยกเว้น เช่น [^abc] หมายถึง อักขระใด ๆ ที่ไม่ใช่ a หรือ b หรือ c

- ตัวดำเนินการ "\" มีความหมายเหมือนข้อ ข.

ง. อักขระใด ๆ

ตัวดำเนินการ "." หมายถึง อักขระใด ๆ ยกเว้น อักขระ newline

จ. การให้เลือกอักขระ

ตัวดำเนินการ "?" หมายถึงอักขระ หรือ กลุ่มอักขระ หรือสาย

อักขระที่อยู่หน้า "?" จะมีหรือไม่มีก็ได้ เช่น $ab?c$ หมายถึง จะมีอักขระ b หรือไม่มีก็ได้

ในเคอร์รูปแบบ ac หรือ abc

ฉ. อักขระซ้ำแบบไม่จำกัดจำนวน

การกำหนดการซ้ำกันของอักขระ กลุ่มอักขระหรือสายอักขระ จำนวนศูนย์ครั้งขึ้นไป (ไม่มีหรือมีตั้งแต่หนึ่งครั้งขึ้นไป) และจำนวนหนึ่งครั้งขึ้นไป ใช้ตัวดำเนินการ

"*" และ "+" ตามลำดับ เช่น

a^* หมายถึง รูปแบบที่ไม่มี a เลย หรือมี a ตั้งแต่หนึ่งตัวขึ้นไป

a^+ หมายถึง รูปแบบที่มี a ตั้งแต่หนึ่งตัวขึ้นไป

ช. การเลือกอย่างใดอย่างหนึ่ง

ตัวดำเนินการ "|" หมายถึง "หรือ" เช่น $ab|cd$ หมายถึงรูปแบบ

ab หรือ cd

ซ. การรวมกลุ่มหรือจัดลำดับ

กรณีที่นิพจน์ปกติมีความซับซ้อนและจำเป็นต้องรวมกลุ่มหรือจัดลำดับก่อน

หลังสามารถทำได้โดยใช้ตัวดำเนินการ "(" และ ")" เช่น $(ab|cd)^+(ef)^*$

ฅ. การฝังบริบท

- ถ้าอักขระตัวแรกของนิพจน์ปกติเป็นตัวดำเนินการ "^" หมายถึง ต้องการตรวจจับรูปแบบที่อยู่เริ่มต้นของบรรทัด (รูปแบบที่อยู่หลัง อักขระ newline หรือ อยู่ตรงเริ่มต้นข้อมูลเข้า)

- ถ้าอักขระตัวสุดท้ายของนิพจน์ปกติเป็นตัวดำเนินการ "\$" จะหมายถึงต้องการตรวจจับรูปแบบที่อยู่ท้ายสุดของบรรทัด (รูปแบบที่ตามหลังด้วยอักขระnewline ทันที) ส่วนตัวดำเนินการ "/" เช่น x/y หมายถึงรูปแบบที่ " x " ต้องตามด้วย " y " เท่านั้น

ญ. การซ้ำแบบจำกัดจำนวนและการแทนที่

ตัวดำเนินการ $\{ \}$ มีสองความหมาย ความหมายแรก ถ้าใน $\{ \}$ เป็นเลขจำนวนเต็ม m, n ($m \leq n$) จะหมายถึงการซ้ำกันของอักขระหรือกลุ่มอักขระหรือสายอักขระข้างหน้าจำนวน m ถึง n ครั้ง เช่น $a\{1,3\}$ หมายถึง รูปแบบ a หรือ aa หรือ aaa และอีกความหมายคือ ถ้าใน $\{ \}$ เป็นชื่อที่กำหนดไว้ก่อนแล้วในส่วน definitions จะหมายถึงการแทนที่

ส่วนที่เขียนต่อท้ายนิพจน์ปกติเรียกว่า "การกระทำ" เป็นส่วนที่ใช้เขียนคำสั่งที่ใช้ในการคำนวณงานเมื่อพบรูปแบบที่ตรงกับข้อกำหนดของนิพจน์ปกติ ถ้าการกระทำมีมากกว่าหนึ่งคำสั่งจะเขียนอยู่ภายในเครื่องหมาย "{" กับ "}" คำสั่งดังกล่าวอาจเขียนด้วยคำสั่งภาษาซี ภาษาฟอร์แทรนหรืออาจเป็นตัวแปรหรือชุดคำสั่งย่อยของโปรแกรม LEX ที่สามารถนำมาใช้ในส่วนการกระทำ

3.2.2 วิธีใช้ LEX

โปรแกรม LEX มีวิธีใช้ตามรูปแบบคำสั่ง ดังนี้

```
lex [-tvfn] [file]
```

ค่าที่อยู่ใน [] จะมีหรือไม่ก็ได้ โดย

- t ต้องการให้ผลลัพธ์จากการคำนวณงานของ LEX ออกทางจอภาพ ถ้าไม่มีค่านี้ข้อมูลออกจะเก็บอยู่ในแฟ้มข้อมูลชื่อ lex.yy.c
 - v ให้พิมพ์สถิติของค่าบางค่า ถ้าไม่กำหนดจะไม่พิมพ์ให้
 - n ยกเลิกค่า -v ไม่ต้องพิมพ์ค่าสถิติ
 - f ไม่ต้องให้ LEX ย่อขนาดของตาราง ถ้าไม่มีค่านี้ LEX จะย่อตารางให้ ทำให้ขนาดโปรแกรมเล็กลง
- file เป็นชื่อแฟ้มข้อมูลที่เก็บข้อมูลเข้าของ LEX ถ้าไม่มีแสดงว่าแฟ้มข้อมูลเข้าคือแป้นพิมพ์

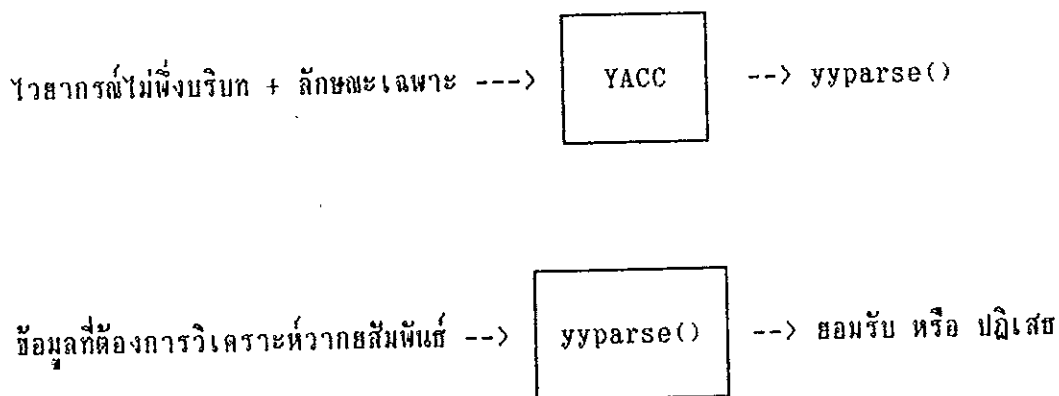
3.2.3 yylex()

ชุดคำสั่งย่อย `yylex()` ที่เก็บอยู่ในแฟ้มข้อมูล `lex.yy.c` ต้องแปลด้วยตัวแปลชุดคำสั่งภาษาซีและเชื่อมโยงกับชุดคำสั่งย่อยอื่น ๆ ที่ LEX จัดไว้ให้ใน library สำหรับระบบปฏิบัติการ Ultrix-32 โดยผู้ใช้จะต้องใช้ option `-ll` เพื่ออ้างถึง library ดังกล่าวเช่น

```
cc lex.yy.c -ll
```

3.3 โปรแกรม YACC

YACC (Yet another compiler-compiler) เป็นโปรแกรมซึ่งอ่านไวยากรณ์ลักษณะเฉพาะ (Attributed grammar) ประกอบด้วยไวยากรณ์ไม่พึ่งบริบท (Context-free grammar) และลักษณะเฉพาะ (Attribute) ของภาษาระดับสูงและสร้างโปรแกรมย่อยชื่อ `yyparse()` เพื่อวิเคราะห์และตรวจสอบวากยสัมพันธ์ของภาษาซึ่งวากยสัมพันธ์ของภาษาคอมพิวเตอร์ระดับสูงส่วนใหญ่สามารถอธิบายได้ด้วยไวยากรณ์ไม่พึ่งบริบท ผลจากการดำเนินงาน `yyparse()` จะมีค่า 0 เมื่อไวยากรณ์ที่นำมาวิเคราะห์เขียนถูกต้องตามวากยสัมพันธ์ แต่ถ้าไม่ถูกต้องจะมีค่าเป็น 1 จากรูปที่ 3.3 แสดงให้เห็นข้อมูลเข้า/ออกของโปรแกรม YACC และโปรแกรมย่อย `yyparse()`



รูปที่ 3.3 ข้อมูลเข้า/ออก ของ YACC และ `yyparse()`

3.3.1 YACC

ไวยากรณ์ไม่เพียงรับทเขียนเป็นกฎ (Rule หรือ Production) ตั้งแต่หนึ่งกฎขึ้นไปโดยแต่ละกฎมีรูปแบบ

lhs : rhs ;

เมื่อ lhs เป็นสัญลักษณ์ไม่สิ้นสุด (Nonterminal symbol) ส่วน rhs เป็นสายอักขระ (String) ซึ่งประกอบด้วย สัญลักษณ์ไม่สิ้นสุดหรือสัญลักษณ์สิ้นสุด (Terminal symbol) หรือเป็นสายอักขระว่าง (Empty string) โดยมีเครื่องหมาย ":" ใช้แบ่ง lhs กับ rhs และเครื่องหมาย ";" บอกการสิ้นสุดของแต่ละกฎและจะมีสัญลักษณ์ไม่สิ้นสุดอยู่ตัวหนึ่ง เรียกว่าสัญลักษณ์เริ่มต้น (Starting symbol) ตัวอย่างเช่น กฎต่อไปนี้เป็น

program : PROGRAM statements END ; (1)

statements : statement 'statements' ; (2)

statements : ; (3)

statement : STOP ';' ; (4)

สัญลักษณ์ไม่สิ้นสุดคือ program, statements และ statement ส่วนสัญลักษณ์สิ้นสุด คือ PROGRAM, END, STOP และ ';' โดยมีสัญลักษณ์ไม่สิ้นสุดตัวแรกของกฎแรกคือ program เป็นสัญลักษณ์เริ่มต้น กฎที่ (3) ทางขวาของเครื่องหมาย ":" เป็นสายอักขระว่าง กฎที่ (2) และ กฎที่ (3) สามารถเขียนรวมกันได้เพื่อให้กระชับและเข้าใจง่าย โดยใช้เครื่องหมาย "|" ดังนี้

statements : statement statements | ;

ภาษาที่อธิบายด้วยไวยากรณ์ข้างต้นเป็นภาษาที่เขียนขึ้นต้นด้วย PROGRAM ตามด้วยคำสั่ง STOP ; ก็จริงก็ได้ หรือไม่ก็ได้ และสิ้นสุดด้วยคำว่า END

ข้อมูลเข้าของ YACC มีรูปแบบดังนี้

```
{ declarations }
```

```
%%
```

```
{ rules }
```

```
%%
```

```
{ programs }
```

{ declaration } ใช้เขียนกลุ่มคำสั่งภาษาซีใด ๆ ซึ่งอยู่ระหว่างเครื่องหมาย % (กับ %) ผลจากการดำเนินงานของโปรแกรม YACC ทำให้กลุ่มคำสั่งดังกล่าวอยู่ก่อนชุดคำสั่งย่อย yyparse()

{ rules } ต้องเขียนอยู่ในรูปแบบดังที่กล่าวมาแล้วข้างต้น ชื่อของสัญลักษณ์ไม่สิ้นสุดและชื่อของสัญลักษณ์สิ้นสุด ประกอบด้วยอักษร a-z หรือ A-Z (ชื่อที่ประกอบด้วยอักษรตัวเล็กจะต่างกับชื่อที่ประกอบด้วยอักษรตัวใหญ่) หรือตัวเลข หรือเครื่องหมาย "." หรือ "_" โดยที่อักขระตัวแรกของชื่อต้องไม่เป็นตัวเลข

{ programs } ใช้เขียนกลุ่มคำสั่งภาษาซีใด ๆ ผลจากการดำเนินงานของโปรแกรม YACC ทำให้กลุ่มคำสั่งดังกล่าวอยู่ก่อนชุดคำสั่งย่อย yyparse() ใช้เขียนคำสั่งภาษาซีหรือฟอร์แทรน โดยคำสั่งเหล่านี้จะถูกนำไปเขียนข้างนอกชุดคำสั่งย่อย yyparse()

{ declarations } และ { programs } จะมีหรือไม่มีก็ได้ และถ้าไม่มีส่วน programs เครื่องหมาย "%%" อันที่สองจะมีหรือไม่มีก็ได้ การเขียนหมายเหตุ ให้เขียนอยู่ภายในเครื่องหมาย "/*" กับ "*/" เหมือนภาษาซี

สัญลักษณ์สิ้นสุดสามารถกำหนดเป็นอักขระใด ๆ หนึ่งตัวเขียนอยู่ภายในเครื่องหมาย ' ' เช่น '+' หรืออาจเขียนด้วยอักขระหลักแบบเดียวกับภาษาซีก็ได้ เช่น '\n' หมายถึงอักขระขึ้นบรรทัดใหม่

YACC จะให้คำสั่ง %token, %left, %right หรือ %nonassoc เพื่อกำหนดชื่อที่เป็นสัญลักษณ์สิ้นสุด และใช้คำสั่ง %start ในการกำหนดสัญลักษณ์เริ่มต้น โดยปกติจะถือว่าสัญลักษณ์ไม่สิ้นสุดตัวแรกของกฎแรกเป็นสัญลักษณ์เริ่มต้นเสมอหากไม่มีคำสั่งดังกล่าว

3.3.2 วิธีใช้ YACC

YACC มีรูปแบบคำสั่งที่ใช้ ดังนี้

```
yacc [-vd] grammar
```

ค่าที่อยู่ในเครื่องหมาย [] จะมีหรือไม่ก็ได้ โดย

-v พิมพ์รายงานการกระทำและสถิติบางอย่างเก็บอยู่ในแฟ้มข้อมูลชื่อ

y.output

-d การกำหนดให้คำสั่ง #define ซึ่งกำหนดรหัสให้กับสัญลักษณ์สิ้นสุดเก็บ

อยู่ในแฟ้มข้อมูลชื่อ y.tab.h

grammar เป็นชื่อแฟ้มข้อมูลที่บรรจุข้อมูลเข้าของ YACC

3.3.3 yyparse()

ชุดคำสั่งย่อย yyparse() ที่ได้จาก YACC จะเก็บอยู่ในแฟ้มข้อมูลชื่อ y.tab.c ก่อนที่จะนำไปใช้ต้องแปลด้วยตัวแปลภาษา C ก่อน และถ้าต้องการเชื่อมโยงกับชุดคำสั่งย่อย main() และ yyerror() ที่มีให้ใช้ ให้เชื่อมโยงโดยการใส่ -ly เช่น

```
cc y.tab.c -ly
```

ถ้าต้องการดูขั้นตอนการทำงานของ yyparse() ในขณะทำการวิเคราะห์กระจายให้แก่ไวยากรณ์ของตัวแปร yydebug ในแฟ้มข้อมูล y.tab.c ให้มีค่าเท่ากับ 1 แล้วช่วงการแปลให้ใส่ option -DYYDEBUG เช่น

```
cc -DYYDEBUG y.tab.c -ly
```

กรณีที่ใช้ LEX สร้างชุดคำสั่งย่อย yylex() สามารถนำชุดคำสั่งย่อย

ดังกล่าว ซึ่งเก็บอยู่ในแฟ้มข้อมูล lex.yy.c มาแปลหรือเชื่อมโยงกับ yyparse() ซึ่งเก็บ
อยู่ในแฟ้มข้อมูล y.tab.c ได้ ดังนี้

```
cc -DYYDEBUG lex.yy.c y.tab.c -ly -ll
```

สมมติว่ามีรูปแบบที่ต้องการตรวจหาเป็นข้อมูลเข้าของ LEX เก็บไว้ในแฟ้ม
ข้อมูลชื่อ lex.yy.c และไวยากรณ์ลักษณะเฉพาะเป็นข้อมูลเข้าของ YACC เก็บไว้ในแฟ้ม
ข้อมูลชื่อ ex.y ซึ่งต้องการสร้างโปรแกรมที่ใช้ในการทำงานบางอย่างที่ example จะมี
ลำดับการทำงานดังนี้

```
yacc -d ex.y
```

```
lex ex.l
```

```
cc -o example lex.yy.c y.tab.c -ly -ll
```

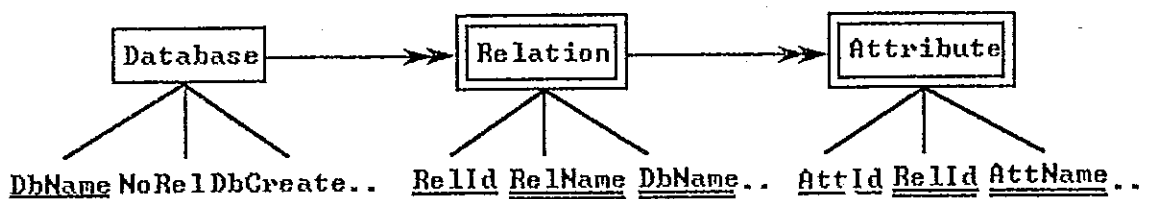
การพัฒนาตัวแปลภาษาสอบถามฐานข้อมูล QBE

4.1 โครงสร้างของระบบงานเดิม

ดังที่กล่าวมาแต่ต้นว่า งานวิทยานิพนธ์นี้เป็นการสร้างตัวแปลภาษา QBE ให้สามารถทำงานร่วมกันได้กับระบบจัดการฐานข้อมูลเบื้องต้นซึ่งมีอยู่ก่อน [เพ็ชรพร (2534) และ ทวีศักดิ์ (2534)] ผู้วิจัยจึงจำเป็นต้องศึกษาโครงสร้างการดำเนินงานและรายละเอียดบางอย่างของงานวิจัยเดิมเพื่อทำความเข้าใจและสามารถใช้โครงสร้างบางส่วนของระบบงานเดิมได้

4.1.1 ปทานุกรมข้อมูลและโครงสร้างแฟ้มข้อมูล

ปทานุกรมข้อมูล (Data dictionary) เป็นส่วนหนึ่งของระบบจัดการฐานข้อมูล ซึ่งใช้จัดเก็บรายละเอียดทั้งหมดของข้อมูลที่ระบบจัดการฐานข้อมูลต้องการใช้ในการดำเนินงานต่าง ๆ ในงานวิจัยเดิมได้ใช้แผนภาพโออาร์ช่วยในการออกแบบเพื่ออธิบายความหมายของโครงสร้างฐานข้อมูลที่เป็นของปทานุกรมข้อมูล โดยแปลงเป็นแต่ละตารางข้อมูลที่สัมพันธ์กันได้ดังแสดงในรูปที่ 4.1



ตารางข้อมูลที่สัมพันธ์กัน

DataBase(DbName, NoRel, DbCreate)

Relation(RelId, RelName, DbName ..)

Attribute(AttId, RelId, AttName ..)

รูปที่ 4.1 แผนภาพโออาร์ของปทานุกรมข้อมูลและตารางข้อมูลที่สัมพันธ์กัน

ตารางข้อมูลจะถูกจัดเก็บข้อมูลในรูปแบบข้อมูลซึ่งมีรายละเอียดดังนี้

DataBase คือเพิ่มข้อมูลสำหรับเก็บรายละเอียดข้อมูลของฐานข้อมูลทั้งหมดที่มีในระบบในขณะเวลาหนึ่ง ๆ โดยหนึ่งเรคอร์ด (Record) แทนข้อมูลของหนึ่งฐานข้อมูล รายละเอียดของโครงสร้างเพิ่มข้อมูลมีดังนี้

ชื่อฟิลด์	ชนิด	รูปแบบ	ขนาด	คำอธิบาย
DbName	P	C	15	ชื่อของฐานข้อมูล

NoRel	N	I	2	จำนวนตารางข้อมูลที่มีในฐานข้อมูล
-------	---	---	---	----------------------------------

DbCreate	N	D	8	วันที่ที่มีการสร้างฐานข้อมูล
----------	---	---	---	------------------------------

Relation คือ เพิ่มข้อมูลสำหรับเก็บรายละเอียดของทุกตารางข้อมูลของแต่ละฐานข้อมูลที่ปรากฏในเพิ่มข้อมูล 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	จำนวนทูปเปิล
-------	---	---	---	--------------

TupLngth	N	I	3	ขนาดทูปเปิล (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 คือ แฟ้มข้อมูลสำหรับเก็บรายละเอียดของแอตทริบิวต์ที่มีในแต่ละตารางข้อมูลในฐานข้อมูลหนึ่ง ๆ มีรายละเอียดดังนี้

ชื่อฟิลด์	ชนิด	รูปแบบ	ขนาด	คำอธิบาย
AttId	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

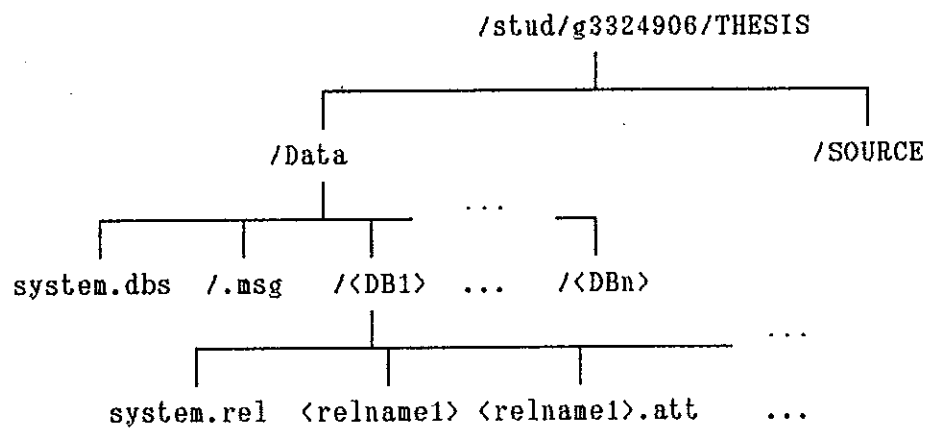
N : Non Key]

หมายเหตุ

- ชนิด คือข้อมูลบอกชนิดหรือประเภทของชื่อแอตทริบิวต์ว่าเป็น Primary Key (P) Secondary Key (S) หรือ Non Key (N)
- รูปแบบ คือประเภทของข้อมูลว่าเป็นตัวอักษร (C) เลขจำนวนเต็ม (I) หรือ วันที่ (D)
- ขนาด คือจำนวนอักขระที่มากที่สุดที่สามารถแทนข้อมูลชิ้นหนึ่ง ๆ
- เครื่องหมาย * หมายถึง ยังไม่มีการพัฒนาในส่วนนี้จากงานวิจัยเดิม

4.1.2 สารบบแฟ้มข้อมูล

สารบบแฟ้มข้อมูล (File directory) ซึ่งจัดเก็บแฟ้มข้อมูลที่เกี่ยวข้องกับการดำเนินงานของงานวิจัยเดิมได้ถูกออกแบบเป็นแผนผังต้นไม้ (Tree diagram) และนำมาดำเนินงานต่อในงานวิจัยนี้ โดยเปลี่ยนจากสารบบเดิมคือ /stud/g302402/THESIS เป็น /stud/g3324906/THESIS ดังแสดงในรูปที่ 4.2



รูปที่ 4.2 แผนผังสารบบแฟ้มข้อมูลของงานวิจัย

สารบบ "/stud/g3324906/THESIS/SOURCE" ใช้เก็บโปรแกรมที่พัฒนาทั้งหมด

สารบบ "/stud/g3324906/THESIS/Data" ใช้เก็บพยานุกรมข้อมูลของระบบ รวมทั้งข้อมูลของฐานข้อมูลต่าง ๆ ที่ถูกสร้างขึ้นในระบบ

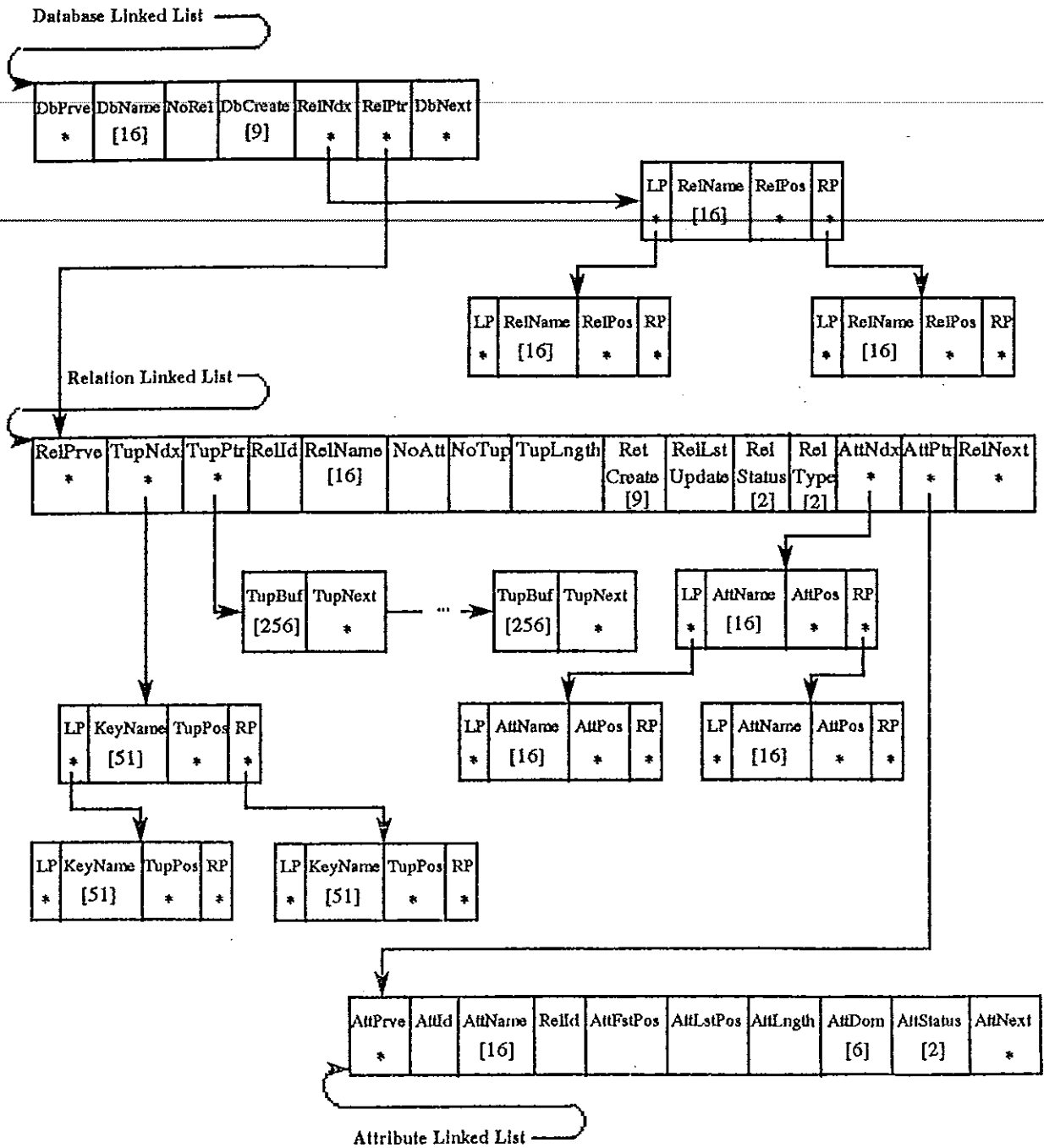
- แฟ้มข้อมูล "system.dbs" ใช้เก็บรายละเอียดของฐานข้อมูล
- สารบบ "/stud/g3324906/THESIS/Data/.msg" ใช้เก็บแฟ้มข้อมูลเกี่ยวกับคำแนะนำในการใช้โปรแกรม
- สารบบ "/stud/g3324906/THESIS/Data/<DBn>" โดย $n = 1, 2, \dots$ ใช้เก็บรายละเอียดโครงสร้างของแต่ละฐานข้อมูลพร้อมข้อมูลที่ถูกรสร้างขึ้น โดยแต่ละสารบบประกอบด้วยแฟ้มข้อมูลต่าง ๆ ของหนึ่งฐานข้อมูล คือ

- system.rel เป็นแฟ้มข้อมูลใช้เก็บรายละเอียดของทุกตารางข้อมูลในฐานข้อมูล
- <relnamen> โดย $N = 1, 2, \dots$ เป็นชื่อแฟ้มข้อมูลจริงของแต่ละตารางข้อมูลในฐานข้อมูล
- <relnamen>.att โดย $n = 1, 2, \dots$ เป็นชื่อแฟ้มข้อมูลที่จัดเก็บรายละเอียดของแอตทริบิวต์ของแต่ละตารางข้อมูลในฐานข้อมูล

4.1.3 โครงสร้างข้อมูล

โครงสร้างข้อมูล (Data structure) ส่วนใหญ่ที่ใช้เป็นแบบรายการโยง (Linked list) และรูปต้นไม้แบบค้นหาภาค (Binary search tree) ดังแสดงในรูปที่ 4.3 โดยมีรายละเอียดดังนี้

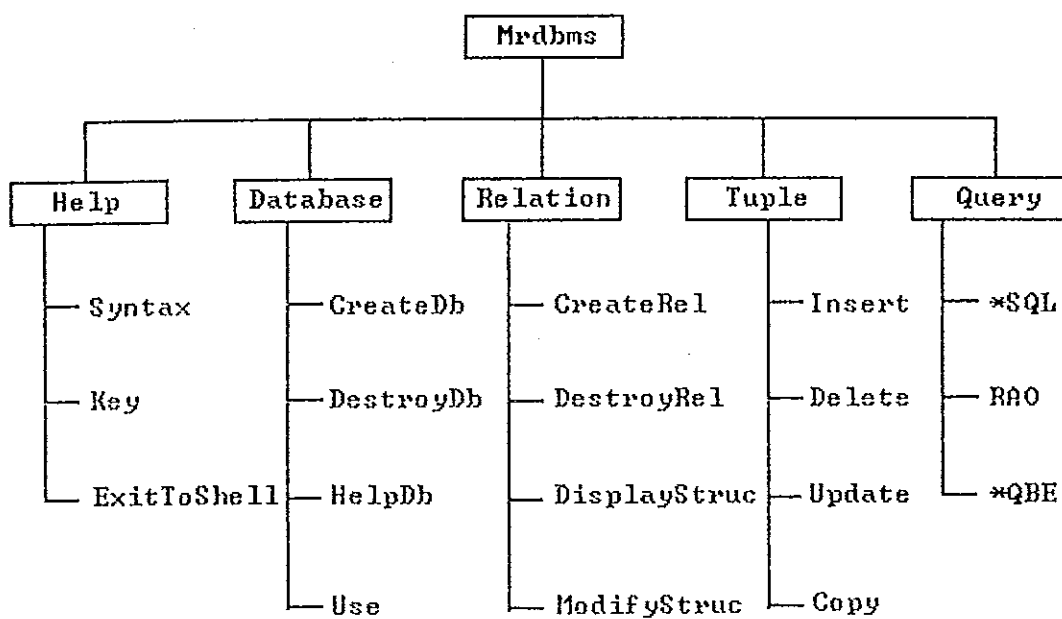
- ก. รายการโยงของฐานข้อมูล (Database linked list) เป็นแบบหลายตัวโยง (Multilink) สร้างจากข้อมูลที่ปรากฏในแฟ้มข้อมูล system.dbs ซึ่งแต่ละโหนดจะแทนรายละเอียดต่าง ๆ ของหนึ่งฐานข้อมูลในระบบ
- ข. รายการโยงของตารางข้อมูล (Relation linked list) เป็นแบบหลายตัวโยงสร้างจากข้อมูลที่ปรากฏในแฟ้มข้อมูล system.rel ซึ่งแต่ละโหนดแทนรายละเอียดต่าง ๆ ของหนึ่งตารางข้อมูล
- ค. รายการโยงของแอตทริบิวต์ (Attribute linked list) เป็นแบบรายการโยงคู่ (Doubly linked list) สร้างจากข้อมูลในแฟ้มข้อมูล <relnamen>.att โดยหนึ่งโหนดของรายการใช้แทนข้อมูลหนึ่งแอตทริบิวต์
- ง. รายการโยงของทูเปิล (Tuple linked list) เป็นแบบรายการโยงเดี่ยว (Singly linked list) สร้างจากข้อมูลในแฟ้มข้อมูล <relnamen> ซึ่งเป็นแฟ้มข้อมูลที่ใช้สำหรับเก็บข้อมูลจริง ๆ หนึ่งโหนดของรายการใช้แทนข้อมูลหนึ่งแถว



รูปที่ 4.3 โครงสร้างข้อมูลของโปรแกรมข้อมูลขณะที่อยู่ในหน่วยความจำหลัก

4.1.4 แผนภูมิระบบของระบบงานเดิม

แผนภูมิระบบ (System chart) ของระบบงานเดิมซึ่งให้ชื่อว่า "Mrdbms" แสดงในรูปที่ 4.5 แสดงให้เห็นการดำเนินงานในส่วนต่าง ๆ และส่วนที่จะนำมาเชื่อมโยงกับงานวิจัยนี้คือส่วนของรายการย่อยชื่อ "QBE" ซึ่งอยู่ภายใต้รายการ "Query" โดยในแผนภูมิของระบบงานเดิมได้ใส่เครื่องหมาย "*" ไว้ข้างหน้าเพื่อบอกให้ทราบว่ายังไม่ได้มีการดำเนินงานใด ๆ กับส่วนของรายการย่อยนี้



รูปที่ 4.4 แผนภูมิระบบแสดงการดำเนินงานในส่วนต่าง ๆ ของงานวิจัยเดิม

4.2 ตัวแปลภาษาสอบถามฐานข้อมูล QBE

ตัวแปลภาษาสอบถามฐานข้อมูล QBE ที่พัฒนาในงานวิจัยนี้ประกอบด้วยชุดของอักขระ "ค่าคงที่" "ตัวอย่าง" ตัวดำเนินการ และ Built-in function โดยมีรายละเอียดดังนี้

ก. ชุดของอักขระ (Character sets)

ชุดของอักขระที่ใช้ในภาษาสอบถามฐานข้อมูล QBE ประกอบด้วย

- ตัวอักษร (Letter) ประกอบด้วยตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ A-Z และตัวพิมพ์เล็ก a-z จำนวน 52 ตัว
- ตัวเลข (Digit) ประกอบด้วยตัวเลข 0-9 จำนวน 10 ตัว
- อักขระพิเศษ (Special characters) ประกอบด้วยตัวอักขระทั้งหมด 33 ตัว คือ # \$ % & ' () * + , - / : ; ? @ ^ _ { | } ~ ! . = < > \ [] _ " และ ช่องว่าง (Blank)

ข. ค่าคงที่ (Constant element)

ค่าคงที่ประกอบด้วยอักขระใด ๆ จากชุดของอักขระในข้อ ก. ตั้งแต่หนึ่งตัวขึ้นไปอาจเขียนอยู่ภายในเครื่องหมาย " " หรือไม่ได้ โดยมีข้อแตกต่างคือ ถ้าหากเขียนอยู่ภายในเครื่องหมาย " " อักขระทั้งหมดที่อยู่ภายในเครื่องหมายดังกล่าวไม่ว่าจะเป็นตัวอักษร ตัวเลขหรืออักขระพิเศษจะถือว่าเป็นค่าคงที่ทั้งหมด เช่น "Hello ! test" แต่ถ้าให้เครื่องหมาย " " เป็นส่วนหนึ่งของค่าคงที่ด้วย จะต้องเขียนอักขระ " สองตัวติดกันแทนหนึ่งตัว เช่น "It""s" จะหมายถึง It"s แต่ถ้าไม่เขียนอักขระดังกล่าวอยู่ในเครื่องหมาย " " อักขระพิเศษทุกตัวรวมทั้งอักขระว่างจะต้องมีเครื่องหมาย \ นำหน้าทุกครั้ง เช่น It\s หมายถึง It"s มิฉะนั้นอักขระพิเศษดังกล่าวอาจถูกตีความเป็นตัวดำเนินการหรือถ้าเป็นอักขระว่างก็จะถูกขจัดออกไป

ค. "ตัวอย่าง" (Example element)

"ตัวอย่าง" ประกอบด้วยอักขระที่เป็นตัวอักษรหรือตัวเลข โดยมีอักขระพิเศษ "_" (Under score) เป็นอักขระตัวแรกเสมอและ "ตัวอย่าง" ที่แสดงบนจอภาพจะเป็นชุดตัวอักษรและ/หรือตัวเลขที่มีการขีดเส้นใต้ "ตัวอย่าง" ใช้แทนความหมายของตัวอย่างของคำตอบที่เป็นไปได้ทั้งหมด เช่น เมื่อป้อนตัวอย่าง "_Red" จะแสดงทางจอภาพเป็น Red แทนความหมายของสีใด ๆ เช่น สีแดง เป็นต้น

ง. ตัวดำเนินการ

- ตัวดำเนินการเปรียบเทียบ ประกอบด้วย $>$ $>=$ $<$ $<=$ $!$ และ $!=$ โดย $!$ มีความหมายเช่นเดียวกับ $!=$ คือ "ไม่เท่ากับ" ตัวดำเนินการเปรียบเทียบใช้เปรียบเทียบค่าข้อมูลของสแต็กจากฐานข้อมูลกับค่าคงที่หรือค่า "ตัวอย่าง"
- ตัวดำเนินการ p. บ่งถึงความต้องการแสดงค่าของแอดทรีวิวที่ p. อยู่ออกทางจอภาพ
- ตัวดำเนินการ ao. และ do. ค่าเนินการเรียงลำดับข้อมูลในแอดทรีวิวที่ต้องการแสดงค่าทางจอภาพจากค่าน้อยไปมากและค่ามากไปน้อย ตามลำดับ
- ตัวดำเนินการ gr. ค่าเนินการแบ่งค่าข้อมูลของแอดทรีวิวที่ gr. อยู่ออกเป็นกลุ่มของชุดข้อมูล
- ตัวดำเนินการ all. ค่าเนินการรวมข้อมูลซึ่งเป็นค่าของแอดทรีวิวที่ all. อยู่ให้เป็นหนึ่งชุดข้อมูล
- ตัวดำเนินการ un. ค่าเนินการจัดข้อมูลที่เข้าชื้อจากกลุ่มข้อมูลที่มีอยู่
- Built-in function เป็นฟังก์ชันที่ใช้ในการดำเนินการกับกลุ่มค่าของแอดทรีวิวที่ฟังก์ชันอยู่ ประกอบด้วยฟังก์ชัน cnt. sum. avg. max. min. std. และ var. ซึ่งดำเนินการต่อไปนี้เป็นคือ นับจำนวนข้อมูล หาค่าผลรวม หาค่าเฉลี่ย หาค่าสูงสุด หาค่าต่ำสุด หาค่าเบี่ยงเบนมาตรฐาน และหาค่าความแปรปรวน ตามลำดับ

4.3 โครงสร้างการดำเนินงาน

ในหัวข้อนี้จะกล่าวถึงโครงสร้างการดำเนินงานของตัวแปลภาษาสอบถามฐานข้อมูล QBE ซึ่งผู้วิจัยได้ดำเนินการตาม 2 ขั้นตอนแรกของโครงสร้างการดำเนินงานทั่วไปของตัวแปลภาษาตามรูปที่ 3.1 โดยขั้นตอนเลขที่คลอนาไลเซอร์จะใช้โปรแกรม LEX ช่วย ขั้นตอนในแทกซ์อานาไลเซอร์จะใช้โปรแกรม YACC ช่วย จากนั้นจะเป็นการตีความข้อความสั่ง QBE ที่มีรูปแบบถูกต้องและดำเนินการด้วยการเรียกใช้โปรแกรมย่อยของการดำเนินงานของแต่ละคำสั่งของภาษาพีชคณิตสัมพันธ์ของงานวิจัยและโครงงานที่มีมาก่อน

4.3.1 ขั้นตอนเลขที่คัลลนาไลเซอร์

ขั้นตอนนี้ใช้โปรแกรม LEX สร้างโปรแกรมตรวจจับรูปแบบเหมือน โดยใช้นิพจน์ปกติเป็นตัวกำหนดรูปแบบที่ต้องการตรวจจับ ข้อมูลเข้าของ LEX ของภาษาสอบถามฐานข้อมูล QBE มีดังนี้

```

/* ----- definition part ----- */
/* C code that will be included in lex.yy.c */
%{
/* Include the declaration of test character function */
#include <ctype.h>
/* Include the terminal symbols */
#include "y.tab.h"

#define token(x) x
extern str[255];
extern FILE *fp;
int token_code;
%}

/* The definition for text replacement of regular expression */
alpha      [a-zA-Z]
alphanum   [a-zA-Z0-9]
SpecialChar  [^{alphanum}\n]
const_element  ({alphanum};\{SpecialChar})
str_const_element  \"([^\n]{1}\")+\n
whitespace  [ \t]

```

```

%%
/* ----- rule part ----- */
{whitespace}+      ;
">="              return (token_code = token(GE));
">"               return (token_code = token(GT));
"<="              return (token_code = token(LE));
"<"               return (token_code = token(LT));
"!="              return (token_code = token(NE));
"!"               return (token_code = token(NE));
"="               return (token_code = token(EQ));
"["               return (token_code = token(LB));
"]"               return (token_code = token(RB));
{alpha}+"."       return (token_code = lookup());
"_{alphanum}+"    {
                    strcpy(str,yytext);
                    return (token_code = token(EXAMPLE));
                };
{str_const_element} {
                    delete_open_and_end_str_notation();
                    strcpy(str,yytext);
                    return (token_code = token(CONST));
                };
{const_element}+ {
                    delete_backslash_of_special_char();
                    strcpy(str,yytext);
                    return (token_code = token(CONST));
                };
                    return (token_code = token(yytext[0]));

```

```
%%
```

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

```
InputFilePointer ()
```

```
{
```

```
    yyin = fp;
```

```
}
```

```
struct keyword {
```

```
    char *name;
```

```
    int   token_yylex; }
```

```
keytable[] = {
```

```
    "p.", token(P),
```

```
    "ao.", token(AO),
```

```
    "do.", token(DO),
```

```
    "max.", token(MAX),
```

```
    "min.", token(MIN),
```

```
    "cnt.", token(CNT),
```

```
    "avg.", token(AVG),
```

```
    "std.", token(STD),
```

```
    "var.", token(VAR),
```

```
    "sum.", token(SUM),
```

```
    "all.", token(ALL),
```

```
    "un.", token(UN),
```

```
    "gr.", token(GR),
```

```
    0,0
```

```
};
```

```
char *lower(n)
char *n;
{
    static char name[128];
    register char *p;
    p = (char *)name;
    while (*n)
        ( isupper(*n) ? (*p++ = *n++ + 'a'-'A') : (*p++ = *n++); );
    *p = '\0';
    return (char *)name;
}

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



```
delete_backslash_of_special_char()
```

```
{  
    register int i, j=0;  
    register char tmp_str[255];  
    for (i=0; yytext[i] != '\0'; i++)  
        tmp_str[j++]=(yytext[i]!='\') ? yytext[i]:yytext[i+1];  
    tmp_str[j] = '\0';  
    strcpy(yytext,tmp_str);  
}
```

```
delete_open_and_end_str_notation()
```

```
{  
    register int i=1, j=0;  
    register char tmp_str[255];  
    for (; i < strlen(yytext); i++)  
        tmp_str[j++]=(yytext[i]!='\"') ? yytext[i]:yytext[i+1];  
    tmp_str[j] = '\0';  
    strcpy(yytext,tmp_str);  
}
```

4.3.2 ขั้นตอนอินเทกซ์นาไลเซอร์

ขั้นตอนนี้ใช้โปรแกรม YACC สร้างโปรแกรมตรวจสอบวากยสัมพันธ์ของภาษาสอบถามฐานข้อมูล QBE โดยปกติแต่ละการสอบถามฐานข้อมูลด้วย QBE จะแบ่งการตรวจสอบออกเป็น 2 ส่วน คือ การตรวจสอบเกี่ยวกับชื่อตาราง ชื่อแอตทริบิว และรายละเอียดอื่น ๆ จากโปรแกรมข้อมูลและตรวจสอบข้อความสั่งของภาษาสอบถาม QBE ซึ่งข้อมูลเข้าของ YACC จะเป็นการตรวจสอบในส่วนที่สอง ข้อมูลเข้าของ YACC ซึ่งเป็นภาษาสอบถามฐานข้อมูล QBE มีดังนี้

```

/* ----- declaration part ----- */

/* C code that is included in y.tab.c */
{%
/* Include the function of input/output data */
#include <stdio.h>
/* Include the declaration of test character function */
#include <ctype.h>
/* Include the function of string */
#include <string.h>
/* Include the definition and variables of QBE.program */
#include "scr.ext"

FILE *fp

extern int    token_code;
extern char  yytext[];
extern void  InputFilePointer();

%}

```

```
/* The definition of a terminal symbol name */
```

```
%token GE      GT      LE      LT      NE      EQ
        P      GR
        AO      DO
        MAX     MIN     CNT     AVG     VAR     STD     SUM
        ALL     UN
        CONST   EXAMPLE
        LB      RB
```

```
%%
```

```
/* ----- rules part ----- */
```

```
QBE command : #100 project function operator const_example
```

```
! LB ALL EXAMPLE #611 RB #800 ;
```

```
project      : ! P #210 ;
```

```
function     : ! order
```

```
! GR #312
```

```
! built_in_func unique ALL #321 ;
```

```
order       : AO #310
```

```
! DO #311 ;
```

```

built_in_func : | MAX #313
                | MIN #314
                | CNT #315
                | SUM #316
                | AVG #317
                | STD #318
                | VAR #319 ;

unique        : | UN #320 ;

operator      :          #410
                | GE #411
                | GT #412
                | LE #413
                | LT #414
                | EQ #410
                | NE #415 ;

const_example : | CONST #610
                | EXAMPLE #611 ;

```

สัญลักษณ์ #ddd (เมื่อ d เป็นตัวเลข 0-9) คือ สัญลักษณ์การกระทำ (Action symbol) และ ddd คือหมายเลขฟังก์ชันการดำเนินงาน ทำหน้าที่เรียกใช้ฟังก์ชันการดำเนินงาน (Action routine function) เพื่อทำงานอย่างใดอย่างหนึ่งที่ต้องการ หมายเลขฟังก์ชันการดำเนินงานและความหมายของการดำเนินงานของงานวิจัยนี้ จะเป็นไปตามตารางที่ 4.1

หมายเลข ฟังก์ชันการดำเนินงาน	การดำเนินงาน
100 210	กำหนดค่าเริ่มต้นให้กับตัวแปรที่ระบบการใช้ Built-in function เพิ่มข้อมูลให้กับตาราง Project symbol โดยมีรายละเอียดดังนี้
	<ul style="list-style-type: none"> - ชื่อตารางที่แสดงบนจอภาพ - หมายเลขของตารางที่แสดงบนจอภาพ - หมายเลขของแถวของตารางที่แสดงบนจอภาพ - ข้อมูลของแอตทริบิวต์ที่เป็นชื่อเก่า ประกอบด้วย <ul style="list-style-type: none"> - หมายเลขของสคีม่าที่แสดงบนจอภาพ - ชื่อแอตทริบิวต์ตรงกับชื่อของแอตทริบิวต์ของตาราง - รูปแบบของการเรียงลำดับข้อมูล - รูปแบบของการใช้ Built-in function - รูปแบบของการตัดค่าที่ซ้ำออก - รูปแบบซึ่งแสดงว่าแอตทริบิวต์ดังกล่าวมีการสอบถามแบบกลุ่ม - ข้อมูลของแอตทริบิวต์ที่เป็นชื่อใหม่ <ul style="list-style-type: none"> - หมายเลขของสคีม่าที่แสดงบนจอภาพ - ชื่อแอตทริบิวต์ที่ได้เปลี่ยนแปลงเป็นชื่อใหม่ - รูปแบบของการเรียงลำดับข้อมูล - รูปแบบของการใช้ Built-in function - รูปแบบของการตัดค่าที่ซ้ำออก

ตารางที่ 4.1 สัญลักษณ์การกระทำและความหมายของกวรดำเนินงาน

หมายเลข ฟังก์ชันการดำเนินงาน	* การดำเนินงาน
	- รูปแบบซึ่งแสดงว่าแอดทริบิวต์ดังกล่าวมีการสอบตามแบบกลุ่ม
310-311	<p>รูปแบบของการเรียงลำดับข้อมูล มีค่าเป็น 0, 1 และ 2 ซึ่งหมายถึงไม่มีการเรียงลำดับข้อมูลจากค่าน้อยไปหาค่ามากและเรียงลำดับข้อมูลจากค่ามากไปหาค่าน้อย ตามลำดับ</p> <p>รูปแบบของการใช้ Built-in function มีค่า 0-7 ซึ่งนั่นหมายถึงไม่มีการใช้ Built-in function หากค่าสูงสุด หากค่าต่ำสุด นับจำนวนแถวข้อมูล หากผลรวมของชุดข้อมูล หากค่าเฉลี่ย หากค่าเบี่ยงเบนมาตรฐาน และหากค่าความแปรปรวน ตามลำดับ</p> <p>รูปแบบของการตัดค่าที่เข้าออกมีค่า 0 และ 1 ซึ่งหมายถึงไม่มีและมีการตัดค่าที่เข้าออก ตามลำดับ</p> <p>รูปแบบซึ่งแสดงว่าแอดทริบิวต์ดังกล่าวมีการสอบตามแบบกลุ่มมีค่า 0 และ 1 ซึ่งหมายถึง ไม่มีและมีการสอบตามแบบกลุ่ม ตามลำดับ</p> <p>กำหนดรูปแบบของการเรียงลำดับข้อมูลของตาราง Project symbol โดยเรียงข้อมูลจากค่าน้อยไปหาค่ามากและจากค่ามากไปหาค่าน้อย ตามลำดับ</p>

ตารางที่ 4.1 สัญลักษณ์การกระทำและความหมายของการดำเนินงาน (ต่อ)

หมายเลข ฟังก์ชันการดำเนินงาน	การดำเนินงาน
312	เพิ่มข้อมูลให้กับตารางการใช้ฟังก์ชัน "gr." โดยมีรายละเอียดดังนี้
313-319 320 321 และ 800	<ul style="list-style-type: none"> - หมายเลขของตารางที่แสดงบนจอภาพ - หมายเลขของแถวของตารางที่แสดงบนจอภาพ - หมายเลขของสัณฐานของตารางที่แสดงบนจอภาพ - ชื่อของแอตทริบิวต์ <p>กำหนดชนิดของ Built-in function</p> <p>กำหนดรหัสที่สั่งให้มีการตัดค่าซ้ำออก</p> <p>เพิ่มข้อมูลให้กับตารางการใช้ฟังก์ชัน all. โดยมีรายละเอียดดังนี้</p> <ul style="list-style-type: none"> - หมายเลขของตารางที่แสดงบนจอภาพ - หมายเลขของแถวของตารางที่แสดงบนจอภาพ - หมายเลขของสัณฐานของตารางที่แสดงบนจอภาพ - ชื่อของแอตทริบิวต์ - ชนิดของกลุ่ม (ชนิดของกลุ่มเป็น 0 ถ้าถูกเรียกใช้โดยฟังก์ชันการดำเนินงานหมายเลข 321 และมีค่าเป็น 1 ถ้าถูกเรียกใช้โดยฟังก์ชันการดำเนินงานหมายเลข 800) - ชนิดของ Built-in function มีค่าเช่นเดียวกับรูปแบบของการใช้ Built-in function ของตาราง Project symbol

ตารางที่ 4.1 สัญลักษณ์การกระทำและความหมายของการดำเนินงาน (ต่อ)

หมายเลข ฟังก์ชันการดำเนินการ	การดำเนินการ
	<p>- รหัสที่สั่งให้มีการตัดค่าที่เข้าออก (0 คือ ตัดออก และ 1 ไม่ตัด)</p>
410-414	<p>เก็บค่าตัวดำเนินการเปรียบเทียบ $=$ $>$ $<$ $<=$ และ $>=$ เพื่อนำค่าดังกล่าวไปไว้ในตารางสัญลักษณ์ของ "ค่าคงที่" หรือ "ตัวอย่าง" ต่อไป</p>
415	<p>เก็บค่าตัวเปรียบเทียบ ! หรือ $!=$ เพื่อนำค่าดังกล่าวไปไว้ในตารางสัญลักษณ์ของ "ค่าคงที่" หรือ "ตัวอย่าง" ต่อไป แต่ถ้าหากค่าดังกล่าวอยู่ในสดมภ์เดียวกับชื่อตารางก็จะเพิ่มข้อมูลให้กับตาราง Minus symbol โดยมีรายละเอียดดังนี้</p> <ul style="list-style-type: none"> - หมายเลขของตารางที่แสดงบนจอภาพ - หมายเลขของแถวของตารางที่แสดงบนจอภาพ
610	<p>เพิ่มข้อมูลให้กับตารางของ "ค่าคงที่" โดยมีรายละเอียดดังนี้</p> <ul style="list-style-type: none"> - หมายเลขของตารางที่แสดงบนจอภาพ - หมายเลขของแถวของตารางที่แสดงบนจอภาพ - หมายเลขของสดมภ์ของตารางที่แสดงบนจอภาพ - ตัวดำเนินการเปรียบเทียบ ($>$, $>=$, $<$, $<=$, $!$, หรือ $!=$) - ค่าคงที่

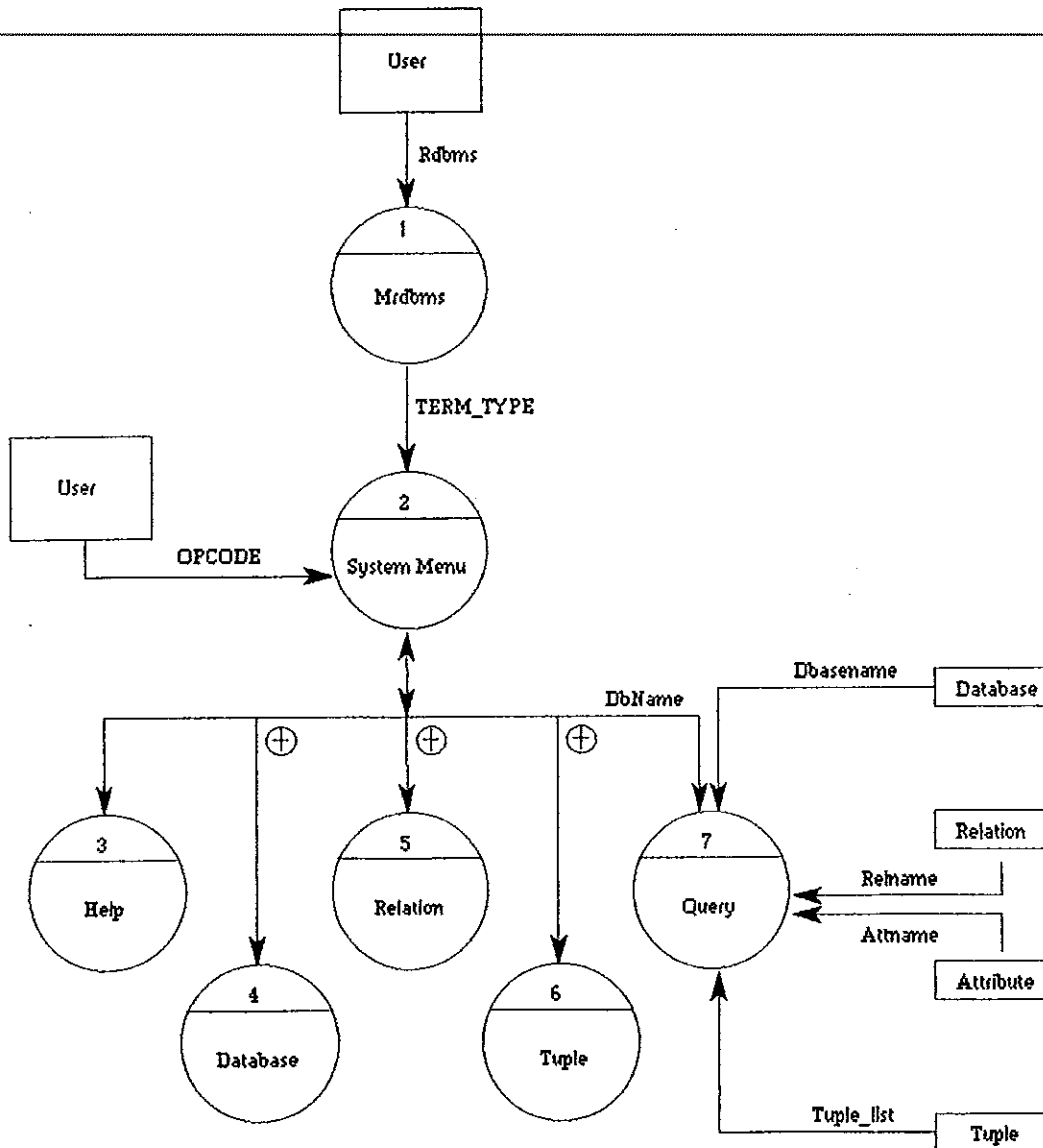
ตารางที่ 4.1 สัญลักษณ์การกระทำและความหมายของการดำเนินการ (ต่อ)

หมายเลข ฟังก์ชันการดำเนินงาน	การดำเนินงาน
611	<p>เพิ่มข้อมูลให้กับตารางของ "ตัวอย่าง" โดยมีรายละเอียดดังนี้</p> <ul style="list-style-type: none"> - หมายเลขของตารางที่แสดงบนจอภาพ - หมายเลขของแถวของตารางที่แสดงบนจอภาพ - หมายเลขของสัณฐานของตารางที่แสดงบนจอภาพ - ตัวดำเนินการเปรียบเทียบ (>, >=, <, <=, !=, หรือ !=) - "ตัวอย่าง" - ชื่อแอตทริบิวต์

ตารางที่ 4.1 สัญลักษณ์การกระทำและความหมายของการดำเนินงาน (ต่อ)

4.4 การวิเคราะห์ ออกแบบและพัฒนาระบบ

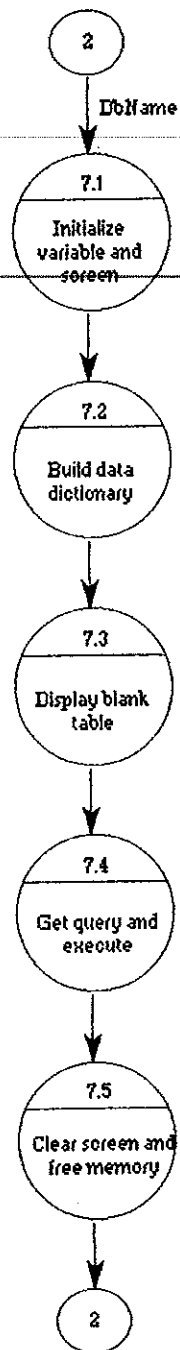
จากการศึกษาระบบงานเดิมดังรูปที่ 4.4 พบว่า ตัวแปลภาษาสอบถามฐานข้อมูล QBE ที่พัฒนาในงานวิจัยานพิเศษนี้ต้องเข้าไปเชื่อมต่อกับเมนูระบบงานเดิมตรงส่วนรายการย่อย "Query" ของรูปที่ 4.4 รูปที่ 4.5 แสดงแผนภาพกระแสข้อมูลของระบบงานเดิม



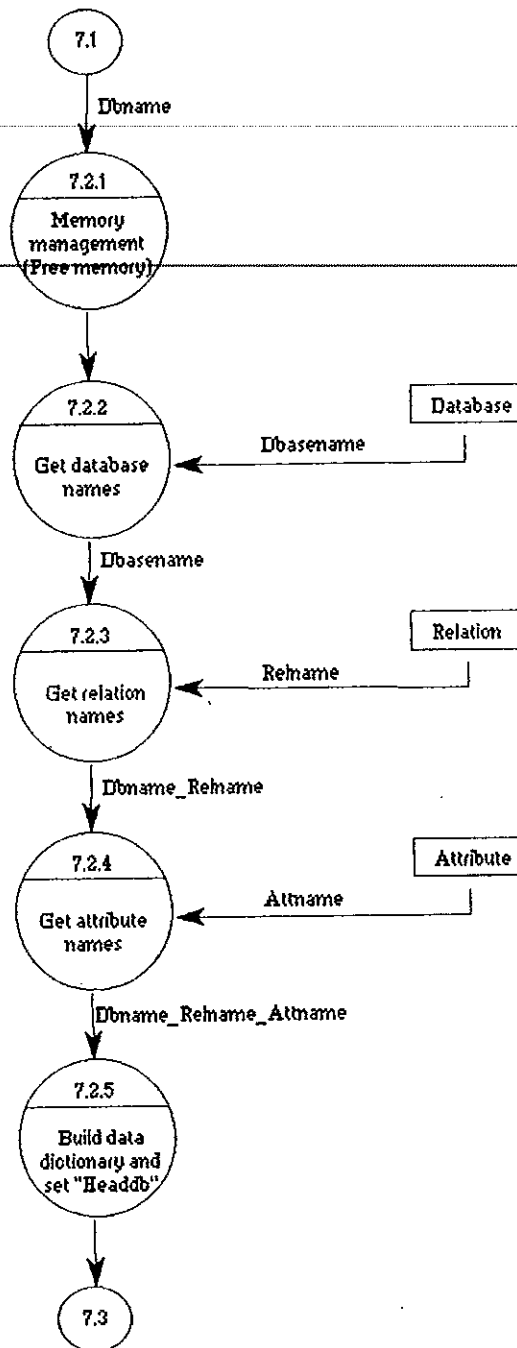
รูปที่ 4.5 แผนภาพกระแสข้อมูลของระบบ Mrdbms เดิม

4.4.1 แผนภาพกระแสข้อมูล

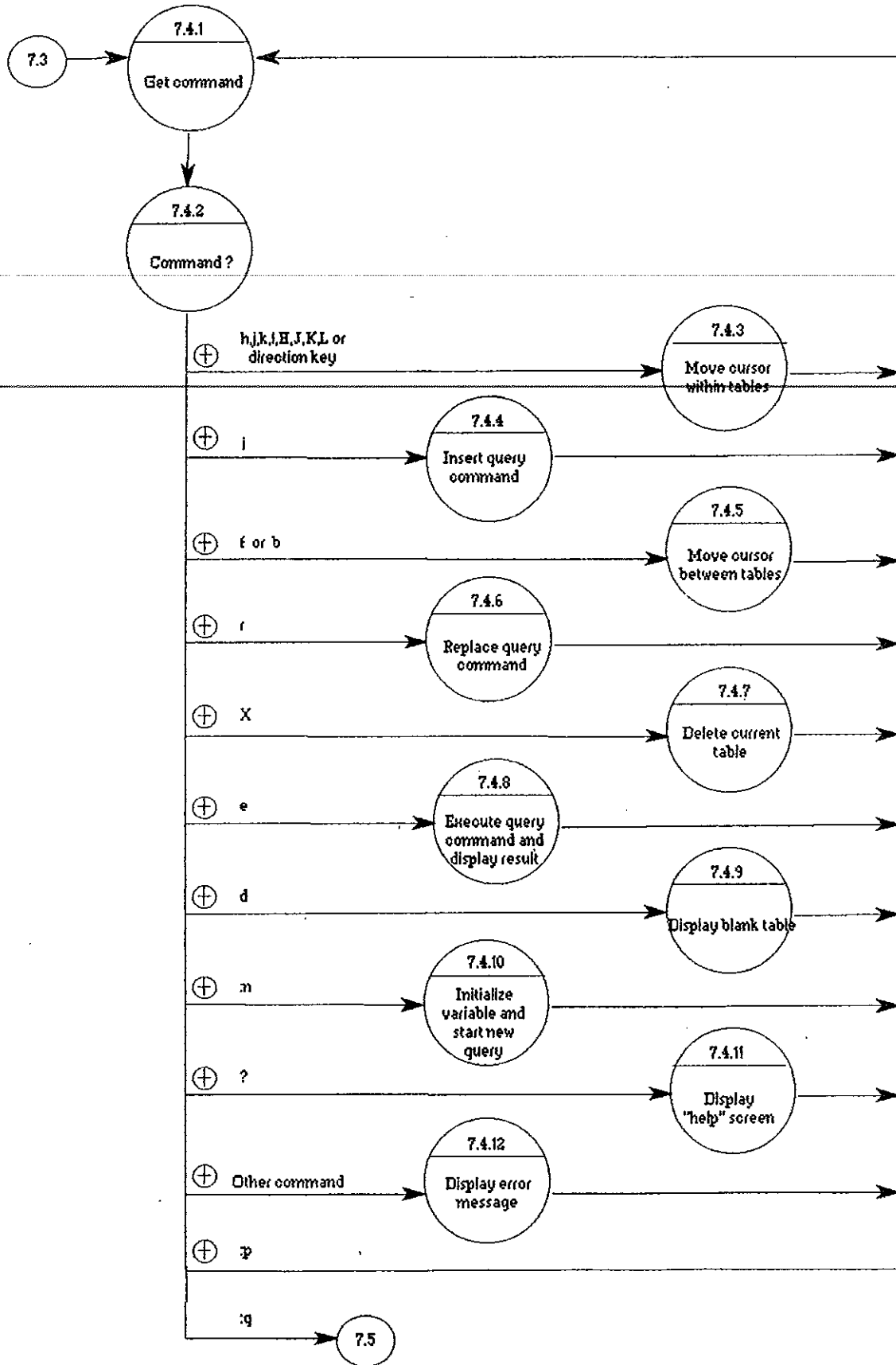
จากรูปที่ 4.5 กระบวนการหมายเลข 7 เป็นการค้าเงินงานการสอบถามข้อมูลประกอบด้วยการสอบถามข้อมูลด้วยภาษาฐานข้อมูล SQL, RAO และ QBE รูปที่ 4.6 ถึงรูปที่ 4.16 แสดงแผนภาพกระแสข้อมูลที่ได้จากการวิเคราะห์และออกแบบระบบงานการทำงานของตัวแปลภาษาสอบถามฐานข้อมูล QBE จากกระบวนการหมายเลข 7 ในส่วนของภาษาฐานข้อมูล QBE และรูปที่ 4.17 ถึงรูปที่ 4.28 จะเป็นคำอธิบายรายละเอียดการทำงานของแต่ละกระบวนการในแผนภาพกระแสข้อมูลตั้งแตรูปที่ 4.5 ถึงรูปที่ 4.16



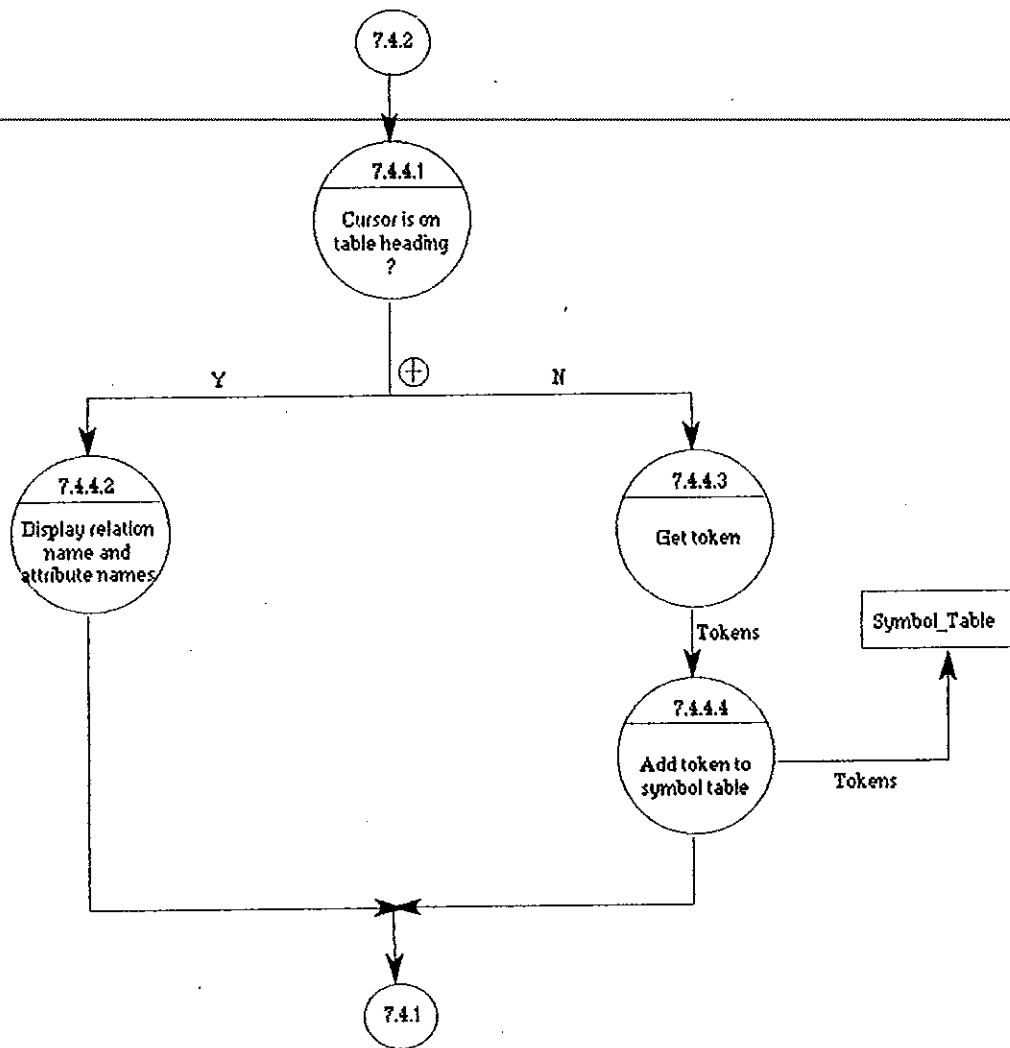
รูปที่ 4.6 แผนภาพกระแสข้อมูลของตัวแปลภาษาสอบถามฐานข้อมูล QBE



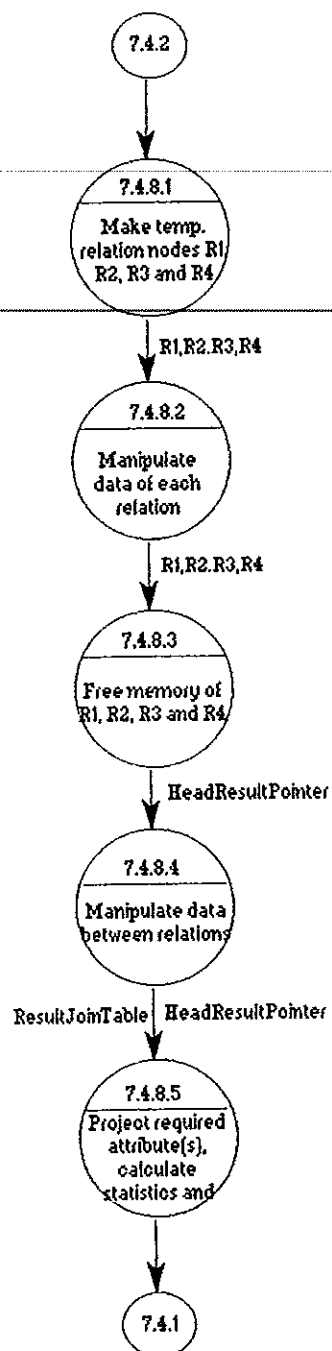
รูปที่ 4.7 แผนภาพกระแสข้อมูลของกระบวนการเลขที่ 7.2



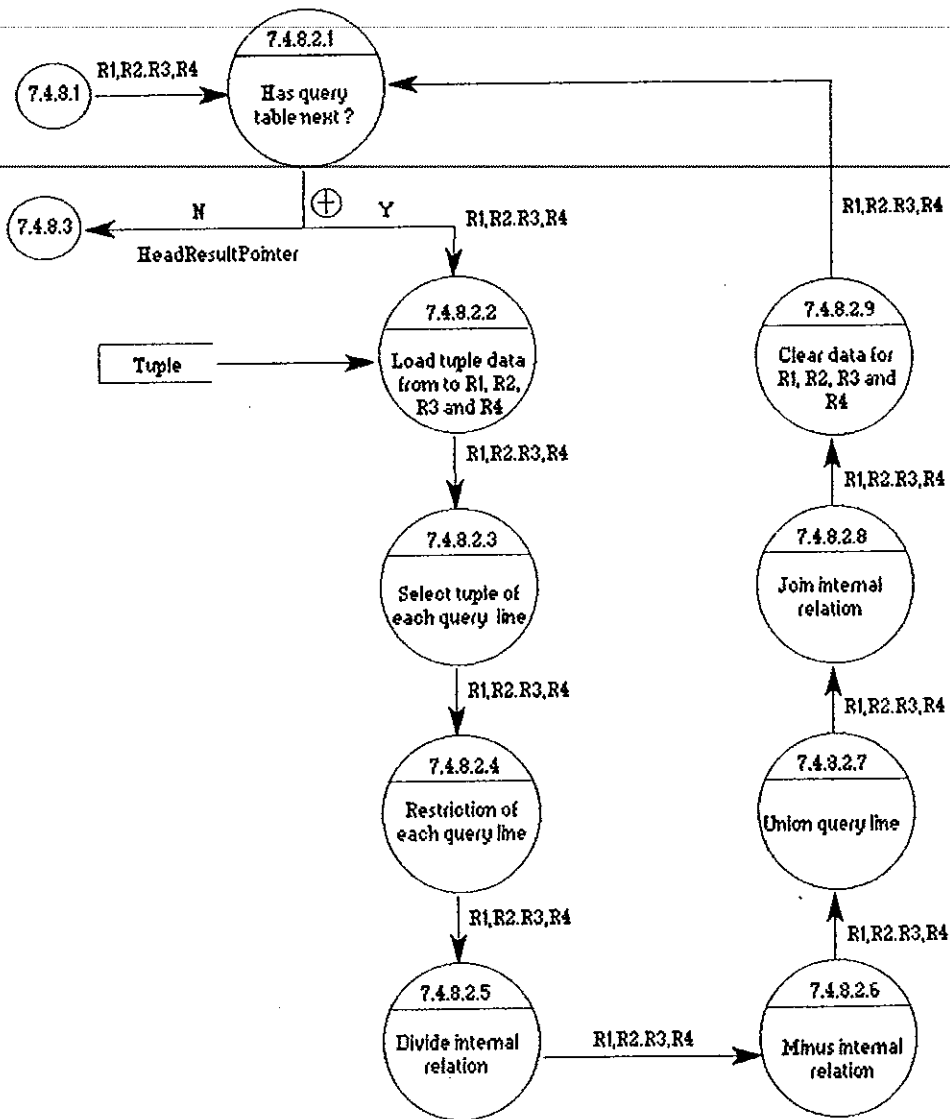
รูปที่ 4.8 แผนภาพกระแสข้อมูลของกระบวนการเลขที่ 7.4



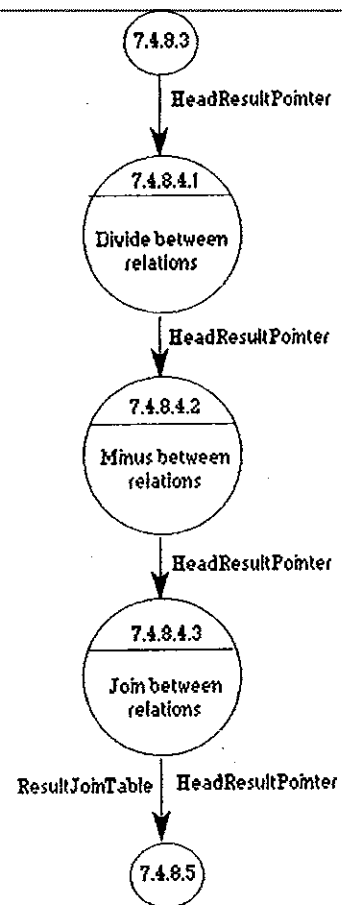
รูปที่ 4.9 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.4



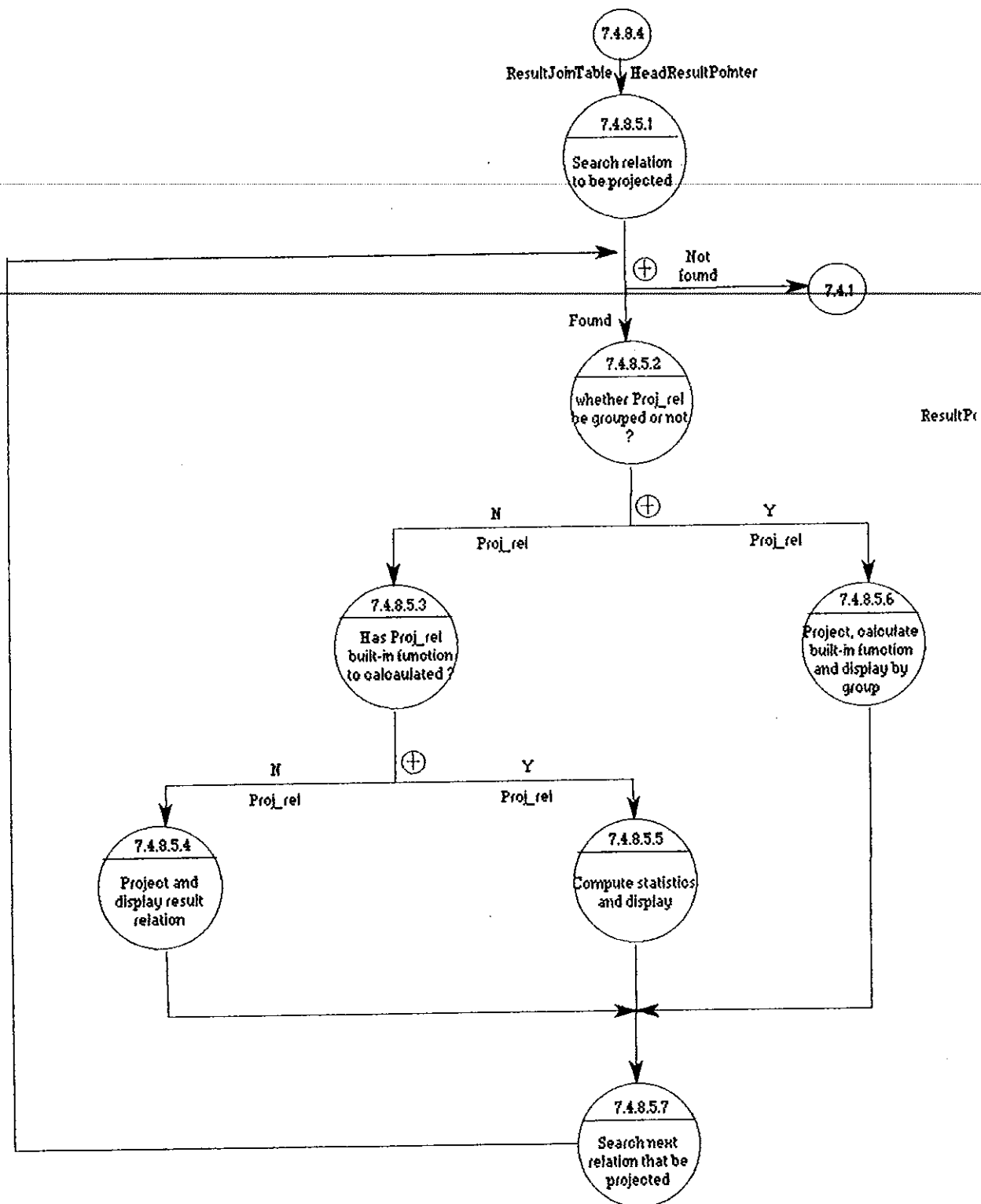
รูปที่ 4.10 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.8



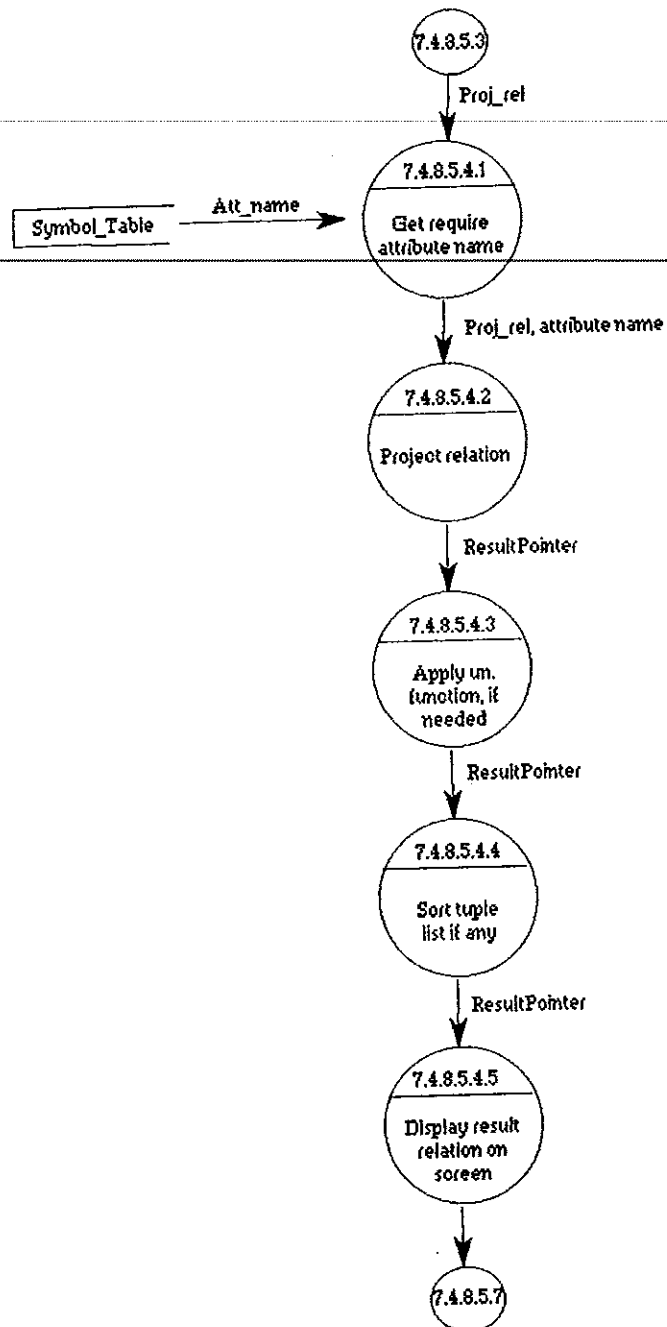
รูปที่ 4.11 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.8.2



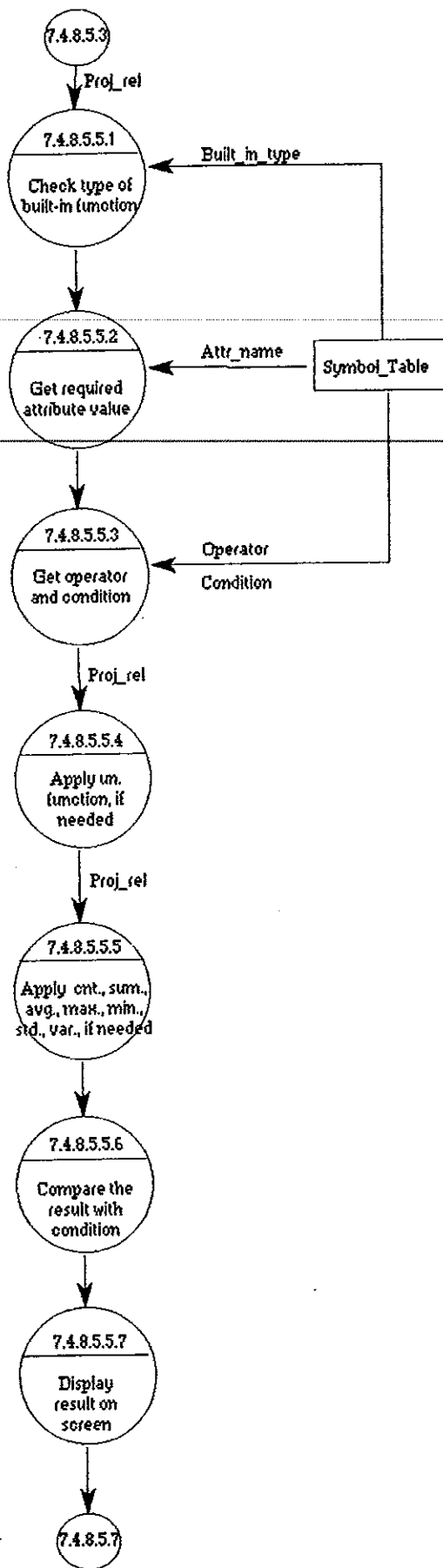
รูปที่ 4.12 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.8.4



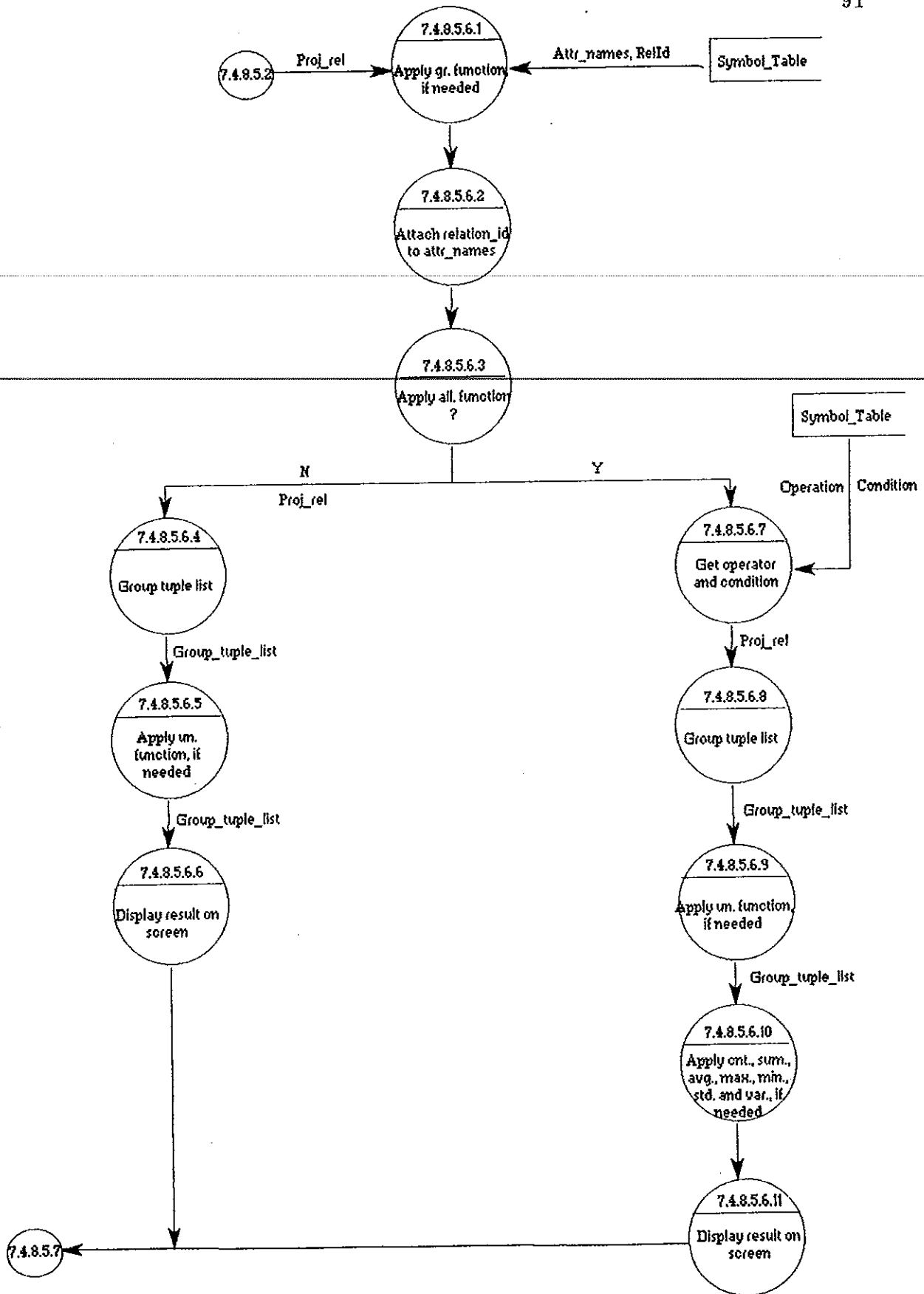
รูปที่ 4.13 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.8.5



รูปที่ 4.14 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.8.5.4



รูปที่ 4.15 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.8.5.5



รูปที่ 4.16 แผนภาพกระแสข้อมูลของกระบวนการ 7.4.8.5.6

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.5</p>
<p>หมายเลขกระบวนการ : 1</p>
<p>การทำงาน : เป็นการดำเนินงานของระบบ Mrdbms โดยรับข้อมูลจากผู้ใช้เพื่อปรับสภาพของจอภาพให้เหมาะสมกับการแสดงผลต่าง ๆ ซึ่งเลือกการแสดงผลการทำงานได้กับจอภาพ 2 ชนิดคือจอภาพแบบ VT และจอภาพแบบ Monochrome</p>
<p>หมายเลขกระบวนการ : 2</p> <p>การทำงาน : เป็นการดำเนินงานของระบบ Mrdbms มีการแสดงผลแบบ Pull down menu ซึ่งผู้ใช้สามารถเลือกการทำงานได้จากรายการหลักของระบบ</p>
<p>หมายเลขกระบวนการ : 3-6</p> <p>การทำงาน : เป็นการดำเนินงานต่าง ๆ ในส่วนรายการหลักของระบบ Mrdbms ศึกษารายละเอียดได้จากงานวิจัยก่อนหน้านี้</p>
<p>หมายเลขกระบวนการ : 7</p> <p>การทำงาน : เป็นการดำเนินงานให้ผู้ใช้เลือกรูปแบบของภาษาสอบถามฐานข้อมูล ซึ่งมี 3 ภาษาคือ RAO, SQL และ QBE สำหรับงานวิจัยครั้งนี้ผู้ใช้สามารถเลือกภาษาสอบถามฐานข้อมูลแบบ QBE</p>

รูปที่ 4.17 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.5

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.6</p>
<p>หมายเลขกระบวนการ : 7.1</p> <p>การทำงาน : เป็นการกำหนดค่าเริ่มต้นให้กับตัวแปรต่าง ๆ ที่จะนำมาใช้กับโปรแกรม พร้อมทั้งกำหนดจอภาพให้สามารถทำงานกับโปรแกรมได้</p>
<p>หมายเลขกระบวนการ : 7.2</p> <p>การทำงาน : เป็นการสร้างโปรแกรมข้อมูล สำหรับการปฏิบัติการของส่วนภาษาสอบถามฐานข้อมูล QBE</p>
<p>หมายเลขกระบวนการ : 7.3</p> <p>การทำงาน : เป็นการแสดงตารางเริ่มต้นที่ว่างเปล่าเพื่อรับข้อความสอบถามฐานข้อมูล</p>
<p>หมายเลขกระบวนการ : 7.4</p> <p>การทำงาน : เป็นการรับข้อความสอบถามฐานข้อมูลและค่าเงินงานเพื่อให้ได้ผลลัพธ์ตามที่ต้องการ</p>
<p>หมายเลขกระบวนการ : 7.5</p> <p>การทำงาน : เป็นการคืนหน่วยความจำทั้งหมดจากการทำงานของตัวแปลภาษาสอบถามฐานข้อมูล QBE พร้อมทั้งกำหนดจอภาพให้สามารถดำเนินการกับโปรแกรม Mrdbms ต่อไป</p>

รูปที่ 4.18 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.6

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.7</p>
<p>หมายเลขกระบวนการ : 7.2.1</p> <p>การทำงาน : เป็นการคืนหน่วยความจำบางส่วนจากการทำงานของระบบให้แก่ระบบปฏิบัติการ</p>
<p>หมายเลขกระบวนการ : 7.2.2</p> <p>การทำงาน : เป็นการรับชื่อฐานข้อมูลทั้งหมดที่มีอยู่ในแฟ้มข้อมูล Database</p>
<p>หมายเลขกระบวนการ : 7.2.3</p> <p>การทำงาน : เป็นการรับชื่อตารางข้อมูลทั้งหมดที่มีอยู่ในแฟ้มข้อมูล Relation</p>
<p>หมายเลขกระบวนการ : 7.2.4</p> <p>การทำงาน : เป็นการรับชื่อแอตทริบิวต์ทั้งหมดที่มีอยู่ในแฟ้มข้อมูล Attribute</p>
<p>หมายเลขกระบวนการ : 7.2.5</p> <p>การทำงาน : เป็นการนำข้อมูลที่ได้จากกระบวนการที่ 7.2.2 ถึง 7.2.4 มาสร้างปฏิทินกรรมบนหน่วยความจำหลักและกำหนดให้ HeadDb เป็นตำแหน่งเริ่มต้นของรายการโยงของทุเป็ลที่สร้างจากข้อมูลในแฟ้มข้อมูลของระบบ</p>

รูปที่ 4.19 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.7

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.8</p>
<p>หมายเลขกระบวนการ : 7.4.1</p> <p>การทำงาน : เป็นการรับคำสั่งเพื่อดำเนินงานในการรับข้อความสอบถาม การเคลื่อนย้ายเคอร์เซอร์ การประมวลผลการสอบถามและแสดงผล เป็นต้น</p>
<p>หมายเลขกระบวนการ : 7.4.2</p> <p>การทำงาน : เป็นการตรวจสอบว่าคำสั่งจากกระบวนการที่ 7.4.1 เป็นคำสั่งใด</p>
<p>หมายเลขกระบวนการ : 7.4.3</p> <p>การทำงาน : เป็นการเคลื่อนย้ายตำแหน่งของเคอร์เซอร์ภายในตารางสอบถาม</p>
<p>หมายเลขกระบวนการ : 7.4.4</p> <p>การทำงาน : เป็นการรับข้อความสอบถาม QBE ที่ต้องการสอบถามข้อมูลจากฐานข้อมูล</p>
<p>หมายเลขกระบวนการ : 7.4.5</p> <p>การทำงาน : เป็นการเคลื่อนตำแหน่งของเคอร์เซอร์ระหว่างตารางสอบถาม</p>
<p>หมายเลขกระบวนการ : 7.4.6</p> <p>การทำงาน : เป็นการแก้ไขข้อความสอบถาม QBE เก่า</p>
<p>หมายเลขกระบวนการ : 7.4.7</p> <p>การทำงาน : เป็นการลบตารางสอบถาม ณ ตำแหน่งที่เคอร์เซอร์ปรากฏ</p>

รูปที่ 4.20 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.8

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.8 (ต่อ)</p>
<p>หมายเลขกระบวนการ : 7.4.8</p> <p>การทำงาน : เป็นการประมวลข้อความสอบถาม QBE พร้อมทั้งแสดงผลของการดำเนินการทางจอภาพ</p>
<p>หมายเลขกระบวนการ : 7.4.9</p> <p>การทำงาน : เป็นการแสดงตารางว่างเปล่าตารางใหม่เพื่อใส่ข้อความสอบถาม</p>
<p>หมายเลขกระบวนการ : 7.4.10</p> <p>การทำงาน : เป็นการยกเลิกข้อความสอบถาม QBE เดิมทั้งหมดเพื่อเริ่มการสอบถามใหม่</p>
<p>หมายเลขกระบวนการ : 7.4.11</p> <p>การทำงาน : เป็นการแสดงรายละเอียดของการใช้คำสั่งต่าง ๆ การจัดการบนจอภาพ</p>
<p>หมายเลขกระบวนการ : 7.4.12</p> <p>การทำงาน : เป็นการแสดงข้อความแทนความผิดพลาดจากการดำเนินการต่าง ๆ</p>

รูปที่ 4.20 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.8 (ต่อ)

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.9</p>
<p>หมายเลขกระบวนการ : 7.4.4.1</p> <p>การทำงาน : เป็นการตรวจสอบตำแหน่งของเคอร์เซอร์ว่าอยู่บนส่วนของชื่อตารางหรือไม่</p>
<p>หมายเลขกระบวนการ : 7.4.4.2</p> <p>การทำงาน : เป็นการแสดงชื่อตารางข้อมูลและชื่อของแอตทริบิวต์ของตารางข้อมูลที่จะใช้ในการเขียนข้อความสอบถาม QBE</p>
<p>หมายเลขกระบวนการ : 7.4.4.3</p> <p>การทำงาน : เป็นการแยกสายอักขระของข้อความสอบถาม QBE ออกเป็นโทเคนต่างๆ</p>
<p>หมายเลขกระบวนการ : 7.4.4.4</p> <p>การทำงาน : เป็นการนำค่าแทนโทเคนที่ได้จากกระบวนการที่ 7.4.4.3 ไปเก็บในตารางสัญลักษณ์ต่าง ๆ</p>

รูปที่ 4.21 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.9

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.10</p>
<p>หมายเลขกระบวนการ : 7.4.8.1</p> <p>การทำงาน : เป็นการสร้างตารางข้อมูลชั่วคราวชื่อ R1, R2, R3 และ R4 เพื่อใช้ในเก็บผลลัพธ์ที่ได้จากการประมวลผลคำสั่ง QBE ในกระบวนการที่ 7.4.8.2 โดยแต่ละตารางข้อมูลจะประกอบด้วยแถวข้อมูลจำนวน 4 แถว</p>
<p>หมายเลขกระบวนการ : 7.4.8.2</p> <p>การทำงาน : เป็นการประมวลผลคำสั่งพีชคณิตสัมพันธ์ที่สัมพันธ์กับข้อความสอบถาม QBE บนหนึ่งตารางข้อมูล</p>
<p>หมายเลขกระบวนการ : 7.4.8.3</p> <p>การทำงาน : เป็นการคืนความจำหลักของตารางข้อมูล R1, R2, R3 และ R4 ให้กับระบบ</p>
<p>หมายเลขกระบวนการ : 7.4.8.4</p> <p>การทำงาน : เป็นการประมวลผลคำสั่งพีชคณิตสัมพันธ์ที่สัมพันธ์กับข้อความสอบถามของ QBE บนหลายตารางข้อมูลที่สัมพันธ์กัน</p>
<p>หมายเลขกระบวนการ : 7.4.8.5</p> <p>การทำงาน : เป็นการดำเนินงานกับตารางข้อมูลผลลัพธ์ที่ได้จากกระบวนการที่ 7.4.8.2 และ กระบวนการที่ 7.4.8.4 โดยการใช้ปฏิบัติการพีชคณิตสัมพันธ์ของ Project, gr., un. และ Built-in function ต่าง ๆ</p>

รูปที่ 4.22 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.10

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.11</p>
<p>หมายเลขกระบวนการ : 7.4.8.2.1</p> <p>การทำงาน : เป็นการตรวจสอบว่าตารางข้อมูลของข้อความสอบถาม QBE ปัจจุบันประกอบด้วยตารางข้อมูลเดี่ยวหรือหลายตารางข้อมูล</p>
<p>หมายเลขกระบวนการ : 7.4.8.2.2</p> <p>การทำงาน : เป็นการนำข้อมูลจากแฟ้มข้อมูล Tuple มาไว้ที่ตารางข้อมูลชั่วคราว R1, R2, R3 และ R4 เพื่อนำไปใช้กับกระบวนการที่ 7.4.8.2.3 - 7.4.8.2.8</p>
<p>หมายเลขกระบวนการ : 7.4.8.2.3</p> <p>การทำงาน : เป็นการเรียกใช้ปฏิบัติการ Select เพื่อดำเนินงานกับข้อมูลของตารางข้อมูลชั่วคราว R1, R2, R3 และ R4</p>
<p>หมายเลขกระบวนการ : 7.4.8.2.4</p> <p>การทำงาน : เป็นการเรียกใช้ปฏิบัติการ Restriction เพื่อดำเนินงานกับข้อมูลของตารางข้อมูลชั่วคราว R1, R2, R3 และ R4</p>
<p>หมายเลขกระบวนการ : 7.4.8.2.5</p> <p>การทำงาน : เป็นการเรียกใช้ปฏิบัติการ Divide เพื่อดำเนินงานกับข้อมูลของตารางข้อมูลชั่วคราว R1, R2, R3 และ R4</p>

รูปที่ 4.23 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.11

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.11 (ต่อ)</p>
<p>หมายเลขกระบวนการ : 7.4.8.2.6</p> <p>การทำงาน : เป็นการเรียกใช้ปฏิบัติการ Minus เพื่อดำเนินการกับข้อมูลของตารางข้อมูลชั่วคราว R1, R2, R3 และ R4</p>
<p>หมายเลขกระบวนการ : 7.4.8.2.7</p> <p>การทำงาน : เป็นการเรียกใช้ปฏิบัติการ Union เพื่อดำเนินการกับข้อมูลของตารางข้อมูลชั่วคราว R1, R2, R3 และ R4</p>
<p>หมายเลขกระบวนการ : 7.4.8.2.8</p> <p>การทำงาน : เป็นการเรียกใช้ปฏิบัติการ Join เพื่อดำเนินการกับข้อมูลของตารางข้อมูลชั่วคราว R1, R2, R3 และ R4</p>
<p>หมายเลขกระบวนการ : 7.4.8.2.9</p> <p>การทำงาน : เป็นการคืนเนื้อที่ความจำของตารางข้อมูลชั่วคราว R1, R2, R3 และ R4 ให้กับระบบ</p>

รูปที่ 4.23 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.11 (ต่อ)

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.12</p>
<p>หมายเลขกระบวนการ : 7.4.8.4.1</p> <p>การทำงาน : เป็นการเรียกใช้ปฏิบัติการ Divide ดำเนินการกับข้อมูลของตาราง สอบถามที่มีความสัมพันธ์กัน</p>
<p>หมายเลขกระบวนการ : 7.4.8.4.2</p> <p>การทำงาน : เป็นการเรียกใช้ปฏิบัติการ Minus ดำเนินการกับข้อมูลของตาราง สอบถามที่มีความสัมพันธ์กัน</p>
<p>หมายเลขกระบวนการ : 7.4.8.4.3</p> <p>การทำงาน : เป็นการเรียกใช้ปฏิบัติการ Join ดำเนินการกับข้อมูลของตาราง สอบถามที่มีความสัมพันธ์กัน</p>

รูปที่ 4.24 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.12

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.13</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.1</p> <p>การทำงาน : เป็นการค้นหาตารางข้อมูลที่จะใช้ในปฏิบัติการ Project ว่ามีหรือไม่</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.2</p> <p>การทำงาน : เป็นการตรวจสอบว่าตารางข้อมูลที่ได้จากกระบวนการที่ 7.4.8.5.2 จะต้องมีการจัดกลุ่มข้อมูลหรือไม่</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.3</p> <p>การทำงาน : เป็นการตรวจสอบว่าตารางข้อมูลที่ได้จากกระบวนการที่ 7.4.8.5.3 มีการใช้ Built-in function หรือไม่</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.4</p> <p>การทำงาน : เป็นการเรียกใช้ปฏิบัติการ Project กับตารางข้อมูลที่ได้จากกระบวนการที่ 7.4.8.4 และแสดงผลทางจอภาพ</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.5</p> <p>การทำงาน : เป็นการเรียกใช้ Built-in function กับตารางข้อมูลที่ได้จากกระบวนการที่ 7.4.8.5.4 และแสดงผลทางจอภาพ</p>

รูปที่ 4.25 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.13

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.13 (ต่อ)</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.6</p> <p>การทำงาน : เป็นการจัดกลุ่มข้อมูลของตารางข้อมูลที่ได้จากกระบวนการที่ 7.4.8.5.3 เรียกใช้ Built-in function เรียกใช้ปฏิบัติการ Project และ แสดงผลทางจอภาพ</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.6</p> <p>การทำงาน : เป็นการค้นหาตารางข้อมูลต่อไปที่จะใช้ปฏิบัติการ Project</p>

รูปที่ 4.25 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.13 (ต่อ)

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.14</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.4.1</p> <p>การทำงาน : เป็นการรับชื่อแอดทริบิวที่จะแสดงผล</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.4.2</p> <p>การทำงาน : เป็นการเรียกใช้ปฏิบัติการ Project กับชื่อแอดทริบิวของตารางข้อมูลที่ได้จากกระบวนการที่ 7.4.8.4</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.4.3</p> <p>การทำงาน : เป็นการจัดข้อมูลซ้ำที่ได้จากการเรียกใช้ปฏิบัติการ project ออกเมื่อต้องการให้ตารางข้อมูลผลลัพธ์มีค่าที่ไม่ซ้ำ</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.4.4</p> <p>การทำงาน : เป็นการเรียงข้อมูลจากน้อยไปมาก หรือ มากไปน้อย เมื่อต้องการให้ตารางข้อมูลผลลัพธ์มีการเรียงลำดับข้อมูล</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.4.5</p> <p>การทำงาน : เป็นการแสดงตารางข้อมูลผลลัพธ์ที่ได้ทางจอภาพ</p>

รูปที่ 4.26 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.14

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.15</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.5.1</p> <p>การทำงาน : เป็นการตรวจสอบรูปแบบของ Built-in function ที่ใช้ในข้อความ สอบถาม</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.5.2</p> <p>การทำงาน : เป็นการรับชื่อแอตทริบิวต์ที่ต้องการคำนวณค่าทางสถิติ</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.5.3</p> <p>การทำงาน : เป็นการรับตัวดำเนินการและค่าของข้อมูลที่เป็นเงื่อนไขของการสอบถาม</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.5.4</p> <p>การทำงาน : เป็นการจัดข้อมูลที่เข้าออกเมื่อต้องการให้ตารางข้อมูลผลลัพธ์ไม่มีค่าที่ว่าง</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.5.5</p> <p>การทำงาน : เป็นการเรียกใช้ Built-in function คำนวณค่าทางสถิติ</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.5.6</p> <p>การทำงาน : เป็นการเปรียบเทียบผลการคำนวณค่าทางสถิติกับเงื่อนไขของการสอบถาม</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.5.7</p> <p>การทำงาน : เป็นการแสดงตารางข้อมูลผลลัพธ์ของค่าทางสถิติที่ได้จากการสอบถาม</p>

รูปที่ 4.27 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.15

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.16</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.6.1</p>
<p>การทำงาน : เป็นการรับข้อแอดทริบิวที่ต้องการจัดกลุ่มค่า</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.6.2</p>
<p>การทำงาน : เป็นการค้นหาเซตของตารางข้อมูลที่ได้จากกระบวนการที่ 7.4.8.4 ซึ่งมีการจัดกลุ่ม</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.6.3</p>
<p>การทำงาน : เป็นการตรวจสอบตารางข้อมูลที่ได้จากกระบวนการที่ 7.4.8.4 ซึ่งมีการจัดกลุ่มว่ามีการสอบถามแบบกลุ่มหรือไม่</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.6.4</p>
<p>การทำงาน : เป็นการจัดกลุ่มข้อมูลของตารางข้อมูล</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.6.5</p>
<p>การทำงาน : เป็นการขจัดข้อมูลที่ซ้ำกันของข้อมูลแต่ละกลุ่มที่ได้จากกระบวนการที่ 7.4.8.5.6.4 เมื่อต้องการให้ตารางผลลัพธ์ไม่มีค่าที่ซ้ำ</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.6.6</p>
<p>การทำงาน : เป็นการแสดงตารางข้อมูลผลลัพธ์ที่ได้จากการสอบถามทางจอภาพ</p>

รูปที่ 4.28 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.16

<p>แผนภูมิลำดับข้อมูลรูปที่ : 4.16 (ต่อ)</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.6.7</p> <p>การทำงาน : เป็นการรับตัวดำเนินการและค่าของข้อมูลที่เป็นเงื่อนไขของการสอบถาม</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.6.8</p> <p>การทำงาน : เป็นการจัดกลุ่มข้อมูลของตารางข้อมูล</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.6.9</p> <p>การทำงาน : เป็นการขจัดข้อมูลที่ซ้ำกันของข้อมูลแต่ละกลุ่มที่ได้จากกระบวนการที่ 7.4.8.5.6.9 เมื่อต้องการให้ตารางผลลัพธ์ไม่มีค่าที่ซ้ำ</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.6.10</p> <p>การทำงาน : เป็นการคำนวณค่าทางสถิติสำหรับตารางข้อมูลที่ได้จากกระบวนการที่ 7.4.8.5.6.9</p>
<p>หมายเลขกระบวนการ : 7.4.8.5.6.11</p> <p>การทำงาน : เป็นการแสดงตารางผลลัพธ์ที่ได้จากการสอบถามทางจอภาพ</p>

รูปที่ 4.28 รายละเอียดของแต่ละกระบวนการของแผนภูมิลำดับข้อมูลรูปที่ 4.16 (ต่อ)

4.4.2 โครงสร้างข้อมูล ตารางสัญลักษณ์ ค่าคงที่และตัวแปร

การพัฒนาตัวแปลภาษาสอบถามฐานข้อมูล QBE ในงานวิจัยนี้ จำเป็นต้องใช้โครงสร้างข้อมูลของปกานุกรมข้อมูล ค่าคงที่ และตัวแปรที่สำคัญของงานวิจัยเดิม พร้อมทั้งสร้างค่าคงที่ ตัวแปร ตารางสัญลักษณ์ (Symbol table) สำหรับฟังก์ชันการดำเนินการ all., p., gr. และคำสั่งพีชคณิตสัมพันธ์ Minus เพิ่มเติมในการสร้างตัวแปลภาษาด้วย นอกจากนี้ จะทำการจัดตารางในรูปแบบของโครงสร้างของตารางต่าง ๆ โดยบันทึกรายละเอียดที่สำคัญเพื่อใช้ในการดำเนินการของขั้นตอนต่าง ๆ ของตัวแปลภาษา

4.4.2.1 รายละเอียดของโครงสร้างข้อมูล

โครงสร้างข้อมูลของปกานุกรมข้อมูลจะอยู่ในหน่วยความจำหลักทั้งหมดเป็นของงานวิจัยเดิม แสดงในรูปที่ 4.3 รายละเอียดของแต่ละโหนดในโครงสร้างข้อมูลดังกล่าวอาจสรุปได้ดังนี้

โครงสร้างโหนดของ Tuple linked list

```
typedef struct tuple {
    char          TupBuf[TSIZE+1]; /*Buffer area for tuple      */
    struct tuple *TupNext;         /*Pointer to next tuple buffer*/
} TUPNODE;
```

โครงสร้างโหนดของ Tuple tree

```
typedef struct tupndx {
    struct tupndx *LP;             /*Left pointer of tuple index */
    char          KeyName[51];     /*Key name of tuple           */
    TUPNODE       *TupPos;         /*Pointer to tuple in tuple list*/
    struct tupndx *RP;             /*Right pointer of tuple index */
} TUPNDXNODE;
```

โครงสร้างข้อมูลของ Attribute linked list

```

typedef struct att {
    struct att *AttPrev;          /*Pointer to previous attribute*/
    int      AttId;              /*Attribute identity code    */
    char     AttName[ANAME+1];   /*Name of attribute          */
    int      RelId;             /*Relation owner attribute   */
    int      AttFstPos;         /*First position in tuple    */
    int      AttLstPos;         /*Last position in tuple     */
    int      AttLngth;         /*Size of attribute          */
    char     AttDom[6];         /*Domain of attribute        */
    char     AttStatus[2];     /*Type of attribute (P,A,N)  */
    struct att *AttNext;        /*Pointer to next attribute  */
} ATTNODE;

```

โครงสร้างข้อมูลของ Attribute tree

```

typedef struct attndx {
    struct attndx *LP;          /*Left pointer of attribute index*/
    char     AttName[ANAME+1]; /*Key name of attribute        */
    ATTNODE  *AttPos;          /*Pointer to attribute in attribute list*/
    struct attndx *RP;          /*Right pointer of attribute index*/
} ANDXNODE;

```


โครงสร้างข้อมูล Relation linked list

```

typedef struct rel {
    struct rel *RelPrev;           /*Pointer to previous relation*/
    TUPNDXNODE *TupNdx;           /*Pointer to tuple index */
    TUPNODE *TupPtr;              /*Pointer to tuple buffer */
    int RelId;                    /*Relation identity code */
    char RelName[RNAME+1];       /*Name of relation */
    int NoAtt;                    /*Number of attributes */
    int NoTup;                    /*Number of tuple */
    int TupLngth;                /*Size of tuple */
    char RelCreate[9];           /*Create date */
    char RelLstUpdate[9];       /*Last update */
    char RelStatus[2];          /*Relation status (I,D,A) */
    char RelType[2];            /*Relation type (B,K,T) */
    ANDXNODE *AttNdx;           /*Pointer to attribute index */
    ATTNODE *AttPtr;            /*Pointer to list of attribute*/
    struct rel *RelNext;        /*Pointer to next relation */
} RELNODE;

```

```

typedef struct relndx {
    struct relndx *LP;          /*Left pointer of relation index */
    char RelName[RNAME+1];     /*Name of relation */
    RELNODE *RelPos;          /*Pointer to relation in relation list*/
    struct relndx *RP;          /*Right pointer of relation index */
} RNDXNODE;

```

โครงสร้างโหนดของ Database linked list

```

typedef struct db {
    struct db *DbPrev;          /*Pointer to prev. database node*/
    char      DbName[DBNAME+1]; /*Name of database          */
    int       NoRel;           /*Number of relation        */
    char      DbCreate[9];     /*Create date                */
    RNDXNODE *RelNdx;         /*Pointer to relation index  */
    RELNODE  *RelPtr;         /*Pointer to list of relation */
    struct db *DbNext;        /*pointer to next database node */
} DBNODE;

```

4.4.2.2 รายละเอียดของโครงสร้างของตารางสัญลักษณ์

โครงสร้างของตารางสัญลักษณ์ใช้เก็บข้อมูลต่าง ๆ ที่ได้จากขั้นตอนเลขชี้คัลลอแนไลเซอร์และซินแทกกอแนไลเซอร์ เพื่อใช้เป็นข้อมูลสำหรับประมวลผลข้อความซึ่งสอบถามสรุปรายละเอียดได้ดังนี้

โครงสร้างตารางของ "ค่าคงที่"

```

typedef struct ConstSymTab {
    int      TabNo;           /*Table number              */
    int      RowNo;          /*Row number                 */
    int      ColNo;          /*Column number              */
    char     Opr[3];         /*Operator of const. element*/
    char     ConstVal[255]; /*Value of const. element   */
    struct ConstSymTab *ConstNext; /*Pointer to next const. node*/
} CONST_NODE;

```

โครงสร้างของตารางของ "ตัวอย่าง"

```

typedef struct ExpSymTab {
    int          TabNo;          /*Table number */
    int          RowNo;         /*Row number */
    int          ColNo;         /*Column number */
    char        Opr[3];         /*Operator of exp. element */
    char        ExpVal[255];    /*Value of exp. element */
    char        AttName[16];    /*Attr. name of exp. element */
    struct ExpSymTab *ExpNext;  /*Pointer to next exp. element*/
} EXAMPLE_NODE;

```

โครงสร้างของตารางของการใช้ฟังก์ชัน "all."

```

typedef struct SetSymTab {
    int          TabNo;          /* Table number */
    int          RowNo;         /* Row number */
    int          ColNo;         /* Column number */
    char        AttName[16];    /*Attr. name of set symbol table*/
    int          Type;          /* type=0->subset; type=1->set */
    int          Built_inFlag; /* Flag for bulid-in func. */
    int          Flag_Unq;     /* Flag for unique func. */
    struct SetSymTab *SetNext;
} SET_NODE;

```

โครงสร้างของตาราง Project symbol

```

typedef struct OldAttName {
    int          ColNo;          /*Column number          */
    char         OldName[16]; /*Old attr. name of proj. node */
    int          OrderFlag; /*Flag for sort result of query */
    int          AggrFlag; /*Flag for calculate build-in func*/
    int          UnqFlag; /*Flag for calculate build-in func*/
    int          AllFlag; /*Flag for set if any */
    struct OldAttName *AttNext; /*Pointer to next attr. node */
} OLDDATTNAME;

typedef struct NewAttName {
    int          ColNo;          /*Column number          */
    char         NewName[16]; /*New attr. name of proj. node */
    int          OrderFlag; /*Flag for sort result of query */
    int          Built_inFlag; /*Flag for calculeat build-in func*/
    int          UnqFlag; /*Flag for calculeat build-in func*/
    int          AllFlag; /*Flag for set if any */
    struct NewAttName *AttNex /*Pointer to next attr. node */
} NEWATTNAME;

typedef struct ProjSymTab {
    char         TabName[16]; /*Table name */
    int          TabNo; /*Table number */
    int          RowNo; /*Row number */
    OLDDATTNAME *OldAtt; /*Pointer to attr. linked list*/
    NEWATTNAME *NewAtt; /*Pointer to attr. linked list*/
    struct ProjSymTab *ProjNext; /*Pointer to next proj. node */
} PROJ_NODE;

```

โครงสร้างของตารางฟังก์ชัน "gr."

```

typedef struct GroupSymTab {
    int          TabNo;          /*Table number          */
    int          RowNo;         /*Row number            */
    int          ColNo;         /*Column number         */
    char         AttName[16];   /*Attr. name of group symbol table*/

    struct GroupSymTab *GroupNext;
} GROUP_NODE;

```

โครงสร้างของตาราง Minus symbol

```

typedef struct MinusSymTab {
    int          TabNo;          /*Table number          */
    int          RowNo;         /*Row number            */

    struct MinusSymTab *MinusNext;
} MINUS_NODE;

```

4.4.2.3 รายละเอียดของค่าคงที่

ค่าคงที่ใช้เก็บขนาดสำหรับชื่อฐานข้อมูล ชื่อตารางข้อมูล ชื่อ tuple ความยาวของแถวของข้อมูล และข้อมูลสำหรับตารางสอบถามที่แสดงบนจอภาพ สรุป รายละเอียดได้ดังนี้

```

#define RNAME          15          /*Length of relation name          */
#define ANAME          15          /*Length of attribute name          */
#define DBNAME         15          /*Length of database name          */
#define TSIZE          255         /*Length of tuple                    */
#define MaxTab         6           /*Maximum of query table            */

```

```

#define MaxRow      5      /*Maximum of line of query table */
#define MaxCol      15     /*Maximum of attr. of query table*/
#define MaxAttrName 31     /*Length of attr. name          */
#define MaxqField   11     /*Maximum of query field        */

```

4.4.2.4 รายละเอียดตัวแปรที่สำคัญ

ตัวแปรใช้เก็บข้อมูลสำหรับรายการโยงของฐานข้อมูล ตารางข้อมูล tuple และตารางสัญลักษณ์ต่าง ๆ สรุปรายละเอียดได้ดังนี้

```

DBNODE      *usedb;      /*Pointer to database that used */
DBNODE      *Headdb;     /*Database pointer point to     */
              /*header of database list      */
RELNODE     *userel;     /*Pointer to relation that used */

CONST_NODE  *Hconst_sym  /*Head pointer of const. list   */
EXAMPLE_NODE *Hexp_sym   /*Head pointer of exp. list     */
PROJ_NODE   *Hproj_sym   /*Head pointer of proj. list    */
GROUP_NODE  *HG          /*Head pointer of group list    */
SET_NODE    *HS          /*Head pointer of set list      */
MINUS_NODE  *Hminus_sym  /*Head pointer of minus list    */

CONST_NODE  *const_sym   /*Pointer to const. node that used*/
EXAMPLE_NODE *exp_sym    /*Pointer to exp. node that used */
PROJ_NODE   *proj_sym    /*Pointer to proj. node that used*/
GROUP_NODE  *G          /*Pointer to group node that used*/
SET_NODE    *S          /*Pointer to set node that used */
MINUS_NODE  *minus_sym  /*Pointer to minus node that used*/

```

```

struct query {
    char qlntab [MaxqField];
} qtab [MaxTab][MaxRow][MaxCol] /*Array of string of head table,*/
                                /* attr. name and query line */

char head [MaxTab] [MaxCol] [MaxAttrName]
                                /* -- The name of table and attr. */

int  x [MaxTab]                /* -- position of line and column */
int  y [MaxTab]                /*  on screen of current table -- */
int  row [MaxTab]              /* -- Current row and column of */
int  col [MaxTab]              /*  qtab var. of current table -- */
int  IndTab                    /* Index to current screen table */

int  FstCol [MaxTab]           /* First screen column */
int  LstCol [MaxTab]           /* Last screen column */
int  TopTab                    /* Top of screen table */
int  BottomTab                 /* Bottom of screen table */
int  LstTab                    /* Amount of query table */
int  scrcol [MaxTab]           /* Amount of column on current table */
int  CountCol [MaxTab]        /* The counter of screen column */

```

สรุปผล อุปสรรคปัญหาและข้อเสนอแนะ

5.1 สรุปผล

งานวิทยานิพนธ์นี้เป็นการศึกษาสร้างตัวแปลภาษาฐานข้อมูล QBE เฉพาะส่วนของการสอบถามฐานข้อมูลเชิงสัมพันธ์ โดยตัวแปลภาษาที่สร้างจะเป็นตัวแปลชนิดอินเทอร์พรีเตอร์ให้ทำงานร่วมกับโปรแกรมของงานวิจัยเดิมซึ่งทำงานเกี่ยวกับการสร้างตารางข้อมูล การเพิ่มแถวข้อมูลเข้า การลบแถวข้อมูลออก การปรับปรุงเปลี่ยนแปลงค่าข้อมูล และการดำเนินการโดยใช้ปฏิบัติการที่ผลิตสัมพันธ์ โดยรวมโปรแกรมทั้งหมดเข้าด้วยกัน วิทยาการของภาษาสอบถามฐานข้อมูล QBE ที่ใช้ในงานวิจัยนี้ นอกจากเป็นรูปแบบเดิมของ M.M. Zloof แล้ว ผู้วิจัยได้เพิ่มเติมบางรูปแบบ เพื่อให้ภาษา QBE ที่พัฒนามีขีดความสามารถมากขึ้น โดยใช้โปรแกรม LEX และ YACC ช่วยในการแยกคำศัพท์และตรวจสอบความถูกต้องของรูปแบบของข้อความสอบถาม ตามลำดับ ผลที่ได้จากการสอบถามจะแสดงตารางข้อมูลผลลัพธ์ออกทางจอภาพคอมพิวเตอร์

5.2 อุปสรรคและปัญหา

ในการดำเนินงานวิทยานิพนธ์นี้ ผู้วิจัยประสบอุปสรรคและปัญหาดังต่อไปนี้

- (1) ความยุ่งยากในการออกแบบการทำงานของตัวแปลภาษา เนื่องจากภาษาสอบถามฐานข้อมูล QBE เป็นภาษาที่สามารถเขียนได้หลายรูปแบบและหนังสือที่ใช้ในการค้นคว้ายังมีไม่มากนัก ทำให้ต้องมีการปรับปรุงเปลี่ยนแปลงโปรแกรมบ่อยครั้ง
- (2) การทำงานร่วมกันของโปรแกรมที่พัฒนาขึ้นกับโปรแกรมของงานวิจัยเดิม และการใช้คำสั่งต่าง ๆ ของภาษาซี เนื่องจากผู้วิจัยไม่มีความเข้าใจเกี่ยวกับการทำงานของบางคำสั่งและเครื่องมือต่าง ๆ มาก่อน จึงทำให้การพัฒนาโปรแกรมล่าช้า
- (3) เสียเวลาในการดำเนินงานแปลภาษาซีไปเป็นภาษาเครื่องและการเชื่อมโยงโปรแกรมเข้าด้วยกัน เนื่องจากโปรแกรมที่พัฒนาค่อนข้างใหญ่และมีคอมพิวเตอร์ VAX-

11/785 ทำงานภายใต้ระบบปฏิบัติการ Ultrix-32 ซึ่งมีการทำงานแบบแบ่งกันใช้ (Time sharing) เมื่อมีผู้ใช้มากทำให้ต้องมีการแบ่งเวลาการทำงานของผู้ใช้กับผู้ใช้จำนวนมาก ด้วย ทำให้การทำงานแต่ละงานค่อนข้างช้า

5.3 ข้อเสนอแนะ

ผู้จัดทำวิทยานิพนธ์ฯ ขอเสนอแนะการแก้ปัญหาและการปรับปรุงโปรแกรมให้ดีขึ้นบาง

ประการดังนี้

- (1) ในการพัฒนาโปรแกรมเพื่อการวิจัยควรมีเครื่องมือและอุปกรณ์ที่ใช้ในการวิจัยโดยเฉพาะ เนื่องจากคอมพิวเตอร์ของศูนย์คอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ จัดให้บริการแก่ผู้ใช้ทั้งที่เป็นเจ้าหน้าที่ที่ทำงานในการเก็บข้อมูลต่าง ๆ ของมหาวิทยาลัย และผู้ใช้ที่เป็นนักศึกษาซึ่งลงวิชาเรียนต่าง ๆ โดยเฉพาะวิชาที่ต้องใช้ภาษาโคบอล ทำให้คอมพิวเตอร์ต้องจัดแบ่งเวลาการทำงานของผู้ใช้กับผู้ใช้ต่าง ๆ มากขึ้น ส่งผลให้การดำเนินงานกับโปรแกรมต่าง ๆ ล่าช้ามาก
- (2) ห้องสมุดของมหาวิทยาลัยควรจัดหาหนังสือหรือบทความที่เกี่ยวกับคอมพิวเตอร์เพิ่มมากขึ้น เพื่อประโยชน์ต่อการศึกษาและค้นคว้าวิจัย
- (3) ต้องปรับปรุงการทำงานของฟังก์ชันต่างๆ ของงานวิจัยเดิม เพื่อให้การทำงานเร็วและมีประสิทธิภาพมากขึ้น

บรรณานุกรม

เกษมสิงห์ แก้วอิม (2531) โปรแกรมคำสั่งปฏิบัติการฐานข้อมูลแบบพีซีชนิดสัมพันธ์,
วิทยานิพนธ์มหาบัณฑิต บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์.

ดวงแก้ว สวามิภักดิ์ (2534) ระบบฐานข้อมูล, ซีเอ็ดดูเคชั่น.

ทวีศักดิ์ ศิริฐานนท์ (2534) การพัฒนาระบบจัดการฐานข้อมูลเชิงสัมพันธ์ ส่วนภาษาฐานข้อมูล,
วิทยานิพนธ์มหาบัณฑิต บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์.

เน็รุษ ดิยะวิโรจน์และอรุณี เลิศนาคร (2534) Aggregate Function in RAO,
โครงการทางคณิตศาสตร์ ภาควิชาคณิตศาสตร์ มหาวิทยาลัยสงขลานครินทร์.

เพียรพร พรหมโพธิ์ (2534) โปรแกรมส่วนการกำหนดโครงสร้างฐานข้อมูลและการ
ดำเนินงานพื้นฐาน, วิทยานิพนธ์มหาบัณฑิต บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์.

วิทยา วัชรวิทยากุล (2534) ภาษาและการโปรแกรม C, ซีเอ็ดดูเคชั่น.

อิว ไยสรากาญจนกุล (2534) การเขียนคอมไพเลอร์สำหรับ IBM PC, ซีเอ็ดดูเคชั่น.

Aho, A.V. and Ullman, J.D. (1977) Principles of Compiler Design,
Addison-Wesley.

Borland International (1987) TURBO PASCAL : Owner's Handbook.

Cardenas, A.F. (1985) Database Management Systems, 2d ed. Allyn
and Bacon.

Christian, K. (1983) The Unix Operating System, Wiley-Interscience.

Date, C.J. (1977) An Introduction to Database System, 2d ed.
vol. 1, Addison-Wesley Publishing.

Date, C.J. (1986) An Introduction to Database System, 4th ed.
vol. 1, Addison-Wesley Publishing.

Davis, G.B. and Olson, M.H. (1985) Management Information Systems,
2d ed., McGraw-Hill.

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

Ken J. McDonnell (1983) An Overview of the Relational Test Bed (RTB),
Department of Computer Science, Monash University.

Korth, H.F. and Silberschatz, A. (1991) Database System Concept,
2d ed., McGraw-Hill.

Luk, R.W.K. (1983) Instructional Query-By-Example (IQBE) :
Implementation Guide, Monash University.

Luk, R.W.K. (1984) Instructional Query-By-Example (IQBE) : User's
Guide, Monash University.

Maier, D. (1983) The Theory of Relational Database, Computer Science Press.

Martin, J. (1985) Fourth-Generation Language, vol. 1, Prentice-Hall.

Schreiner, A.T. and Friedman, H.G. (1985) Introduction to Compiler Construction with UNIX, Prentice-Hill.

Thomas, J.C. and Gould, J.D. (1975) A Psychological Study of Query by Example, IBM T.J. Watson Center.

Tremblay, J. and Sorenson, P. (1985) The Theory and Practice of Compiler Writing. McGraw-Hill.

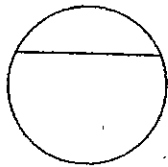
Zloof, M.M. (1975) Query By Example : The Invocation and Definition of Tables and Forms, Proc. of the Int. and VLDBs, Sept 22-24.

ព័ត៌មាន

ความหมายของแผนภาพกระแสข้อมูล

แผนภาพกระแสข้อมูล (Data flow diagram) เรีกสั้น ๆ ว่า DFD คือ แผนภาพที่ใช้สัญลักษณ์ต่าง ๆ นำมาเรียงรวมกันเข้าอย่างมีกฎเกณฑ์เพื่อแสดงการส่งถ่ายของข้อมูลที่เกิดขึ้นระหว่างกระบวนการ (Process) ในระบบงาน โดยแสดงให้เห็นว่ากระบวนการต่าง ๆ มีข้อมูลเข้าคืออะไร และผลจากการดำเนินงานของแต่ละกระบวนการจะได้ข้อมูลอะไรเป็นผลลัพธ์ ซึ่งอาจเป็นข้อมูลที่นำไปใช้ในกระบวนการอื่น ๆ ต่อไป แต่ละกระบวนการสามารถแบ่งย่อยออกเป็นกระบวนการย่อยในระดับต่าง ๆ ได้ ทำให้ทราบขั้นตอนของการส่งถ่ายข้อมูลของกระบวนการต่าง ๆ และการดำเนินงานทั้งหมดของระบบงาน

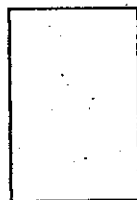
สำหรับงานวิจัยครั้งนี้ ผู้วิจัยได้นำโปรแกรมสำเร็จรูปชื่อ ABC flowcharter version 1.1 ซึ่งทำงานภายใต้โปรแกรมสำเร็จรูป Microsoft windows version 3.1 ดำเนินงานกับเครื่องไมโครคอมพิวเตอร์ขนาด 16 บิต มาช่วยในการออกแบบ โดยใช้สัญลักษณ์ดังต่อไปนี้



แทนกระบวนการในการทำงานของระบบงาน



แทนแฟ้มข้อมูลที่ไว้เก็บข้อมูลของระบบงาน



แทนแหล่งกำเนิดหรือสิ้นสุดของข้อมูลที่ไม่รวมอยู่ในการออกแบบซึ่งจะต้องมีการติดต่อกับระบบงาน

→ แทนกระแสของข้อมูลที่มีการส่งถ่ายข้อมูลในทิศทางเดียวตามหัวลูกศร

↔ แทนกระแสของข้อมูลที่มีการส่งถ่ายข้อมูลในสองทิศทางทั้งไปและกลับ

⊕ ⊗

แทนค่าตรรก "and" และ "or" ตามลำดับ




ភាគដេញ ១

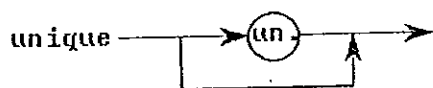
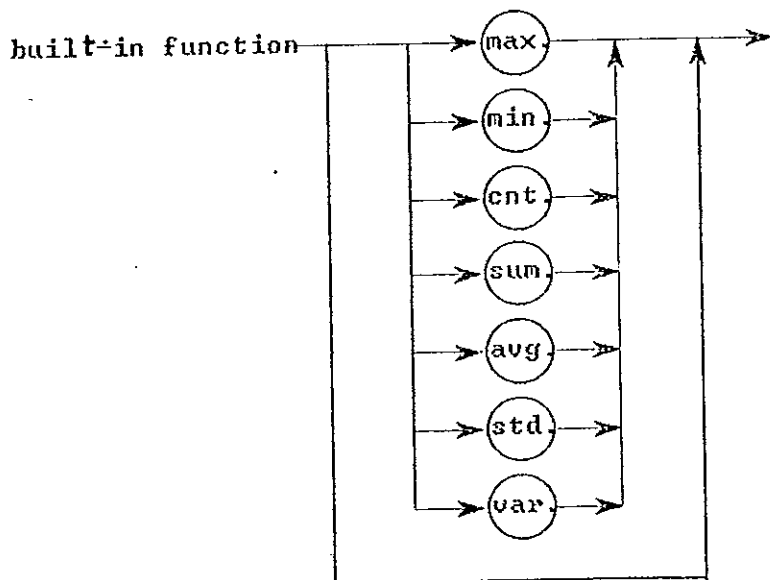
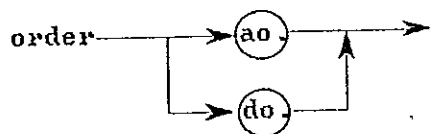
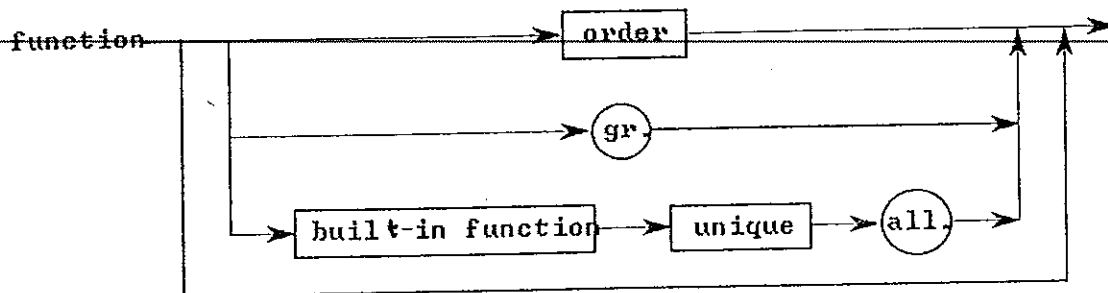
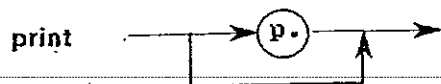
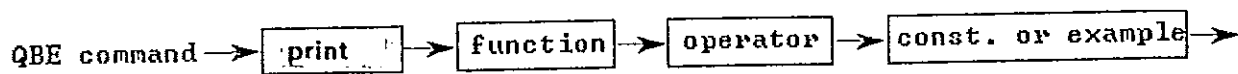
แผนภาพวากยสัมพันธ์ของ QBE

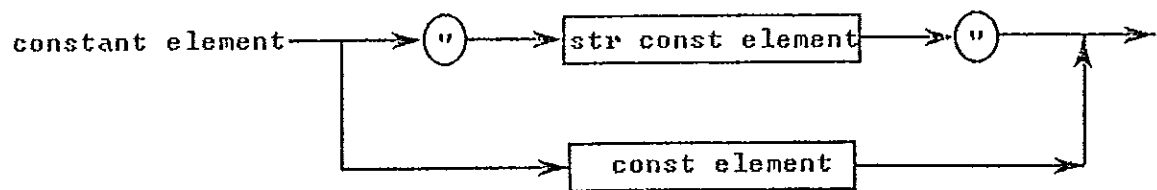
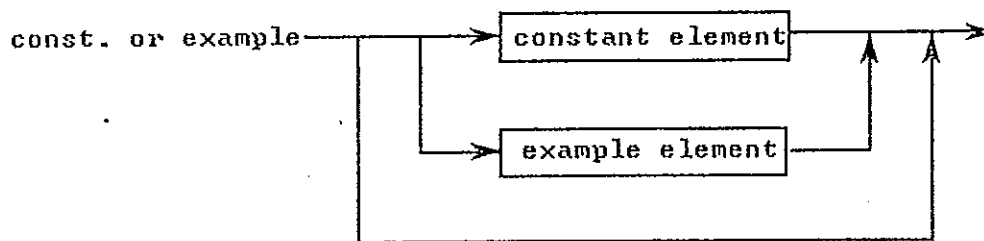
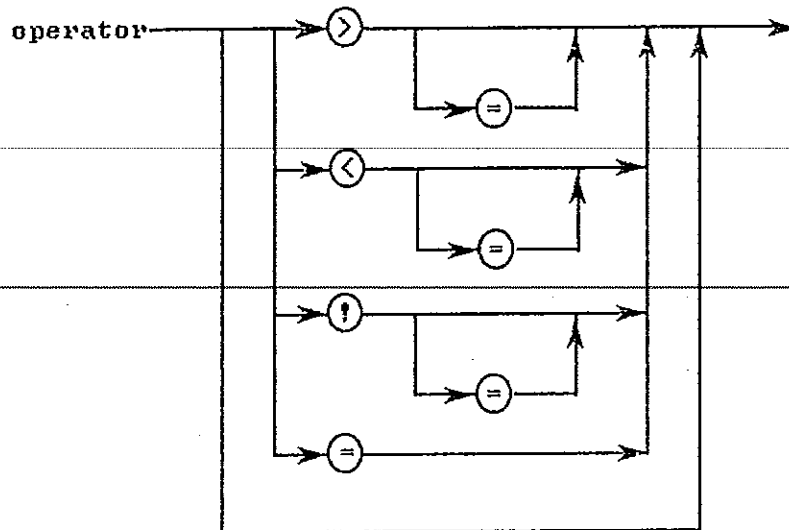
(Syntax diagram of QBE)

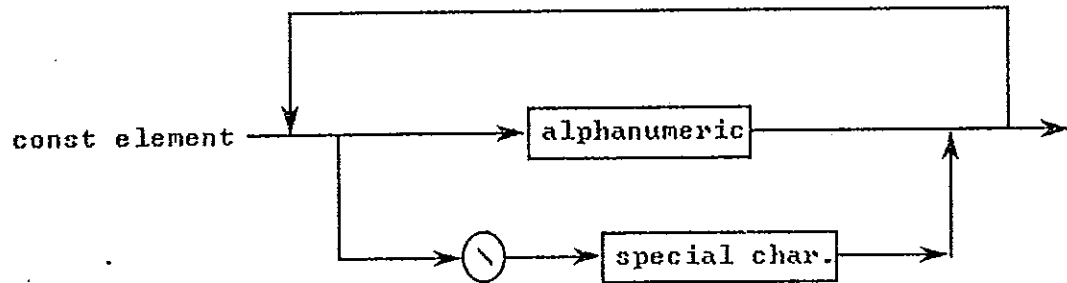
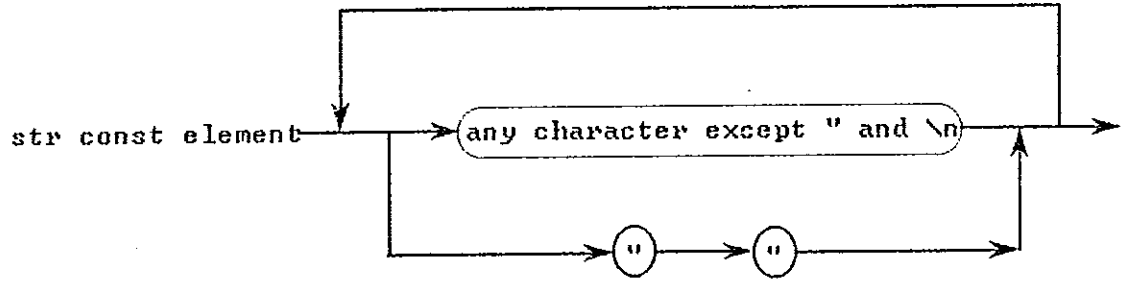
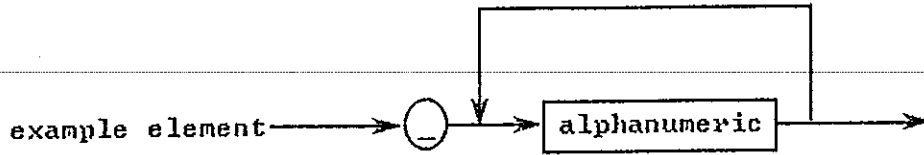
แผนภาพวากยสัมพันธ์ใช้อธิบายรูปแบบหรือไวยากรณ์ของภาษา โดยแต่ละแผนภาพเขียนแทนด้วยรูปภาพและลูกศร โดยรูปภาพใช้แทนสัญลักษณ์แทนที่ได้และสัญลักษณ์ที่แทนไม่ได้ และลูกศรจะมีทางเข้ารูปภาพและทางออกจากรูปภาพได้ทิศทางเดียวเสมอ

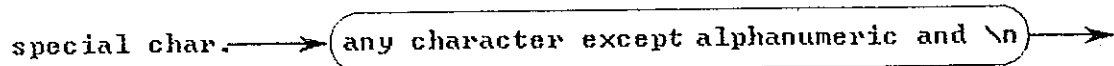
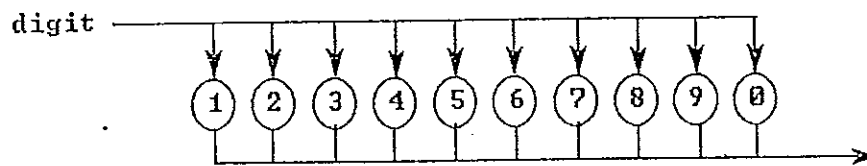
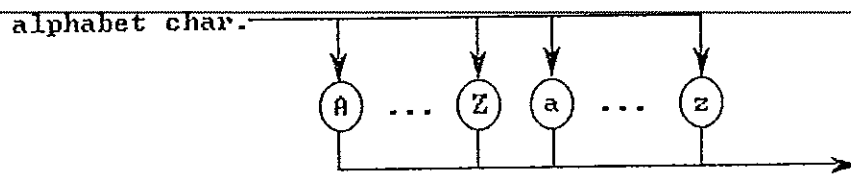
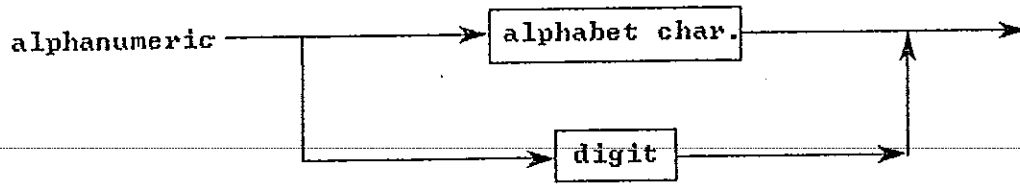
แผนภาพวากยสัมพันธ์ของภาษาสอบถามฐานข้อมูล QBE มีกฎเกณฑ์ดังนี้

1. รูปภาพ  หรือ  แทนสัญลักษณ์ที่แทนไม่ได้ ดังนั้นข้อความภายในรูปภาพจะเป็นข้อความที่ใช้ในการเขียนข้อความสอบถามของ QBE
2. รูปภาพ  แทนสัญลักษณ์ที่แทนที่ได้ ดังนั้นข้อความภายในรูปภาพจะสามารถถูกแทนที่ด้วยแผนภาพอื่นได้









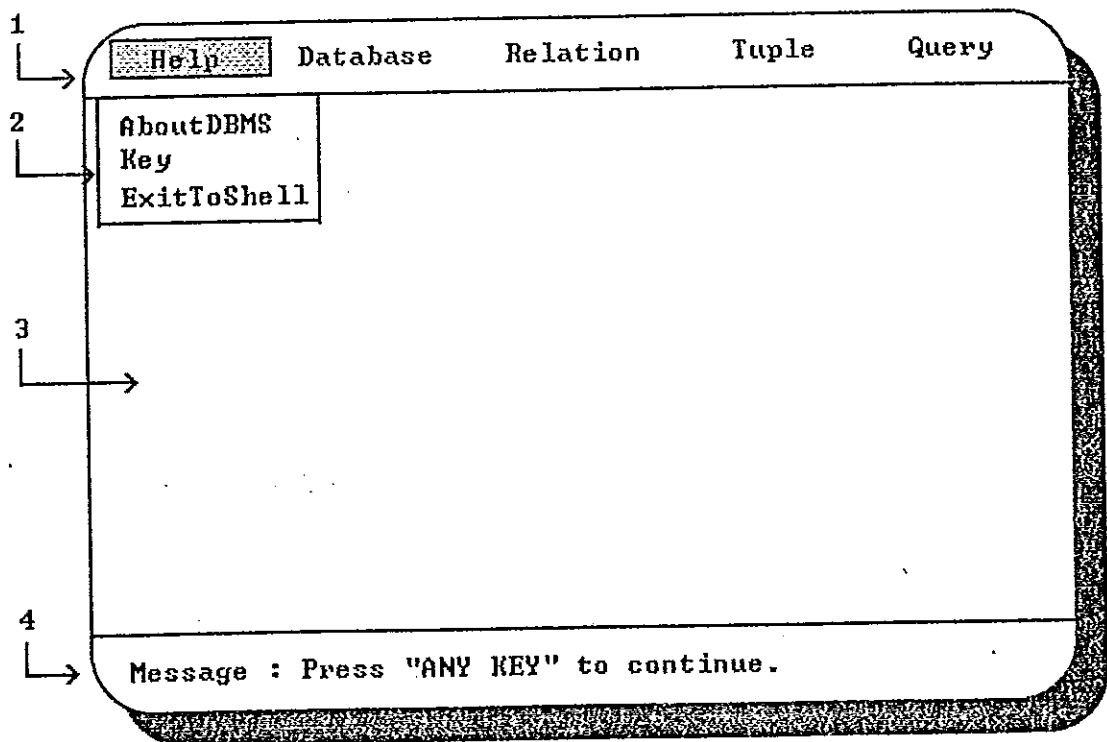
ᠠᠨᠠᠨᠠᠨᠠ ᠠ

เอกสารสำหรับผู้ใช้

1. วิธีใช้โปรแกรม Mrdbms

1.1 การแสดงผลทางจอภาพ

โปรแกรม Mrdbms มีรูปแบบการทำงานที่เรียกว่า การจัดการหน้าต่าง (Window manipulation) โดยสามารถแสดงคำสั่งและผลจากการดำเนินงานทั้งหมดได้ภายในหนึ่งจอภาพที่กำหนดให้ทำให้ผู้ใช้ได้รับความสะดวกต่อการใช้ ลักษณะจอภาพของ Mrdbms ประกอบด้วยส่วนประกอบหลักดังรูปที่ ค.1



รูปที่ ค.1 ส่วนประกอบหลักของจอภาพ Mrdbms

ส่วนที่ 1 เป็นรายการหลักของคำสั่งการทำงานทั้งหมดของโปรแกรม

ส่วนที่ 2 เป็นรายการย่อยของการทำงานของแต่ละรายการหลัก

ส่วนที่ 3 เป็นส่วนของการแสดงข้อความเพื่อการใช้งานของระบบและการแสดงผลที่ได้จากการดำเนินงานต่าง ๆ

ส่วนที่ 4 เป็นส่วนที่ใช้แสดงข้อความผิดพลาดหรือข้อแนะนำการทำงานสำหรับผู้ให้

1.2 การใช้แป้นพิมพ์ควบคุมการทำงาน

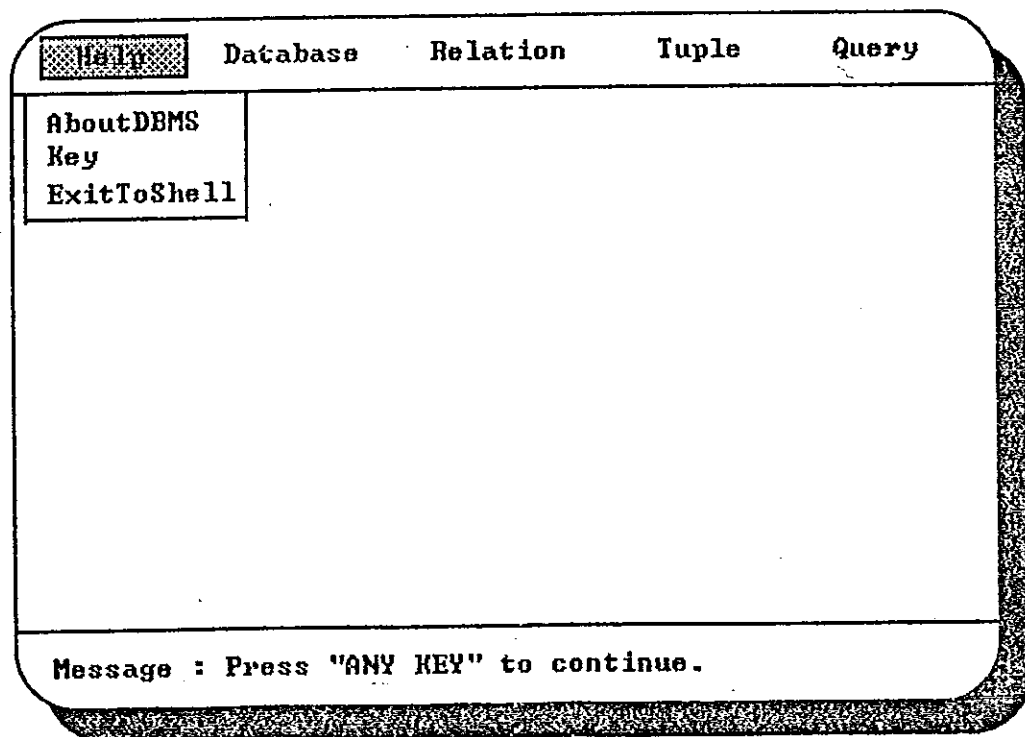
โปรแกรม Mrdbms ใช้แป้นพิมพ์ควบคุมการทำงานดังตารางต่อไปนี้

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

1.3 รายละเอียดรายการหลักของ Mrdbms

1.3.1 รายการหลัก Help

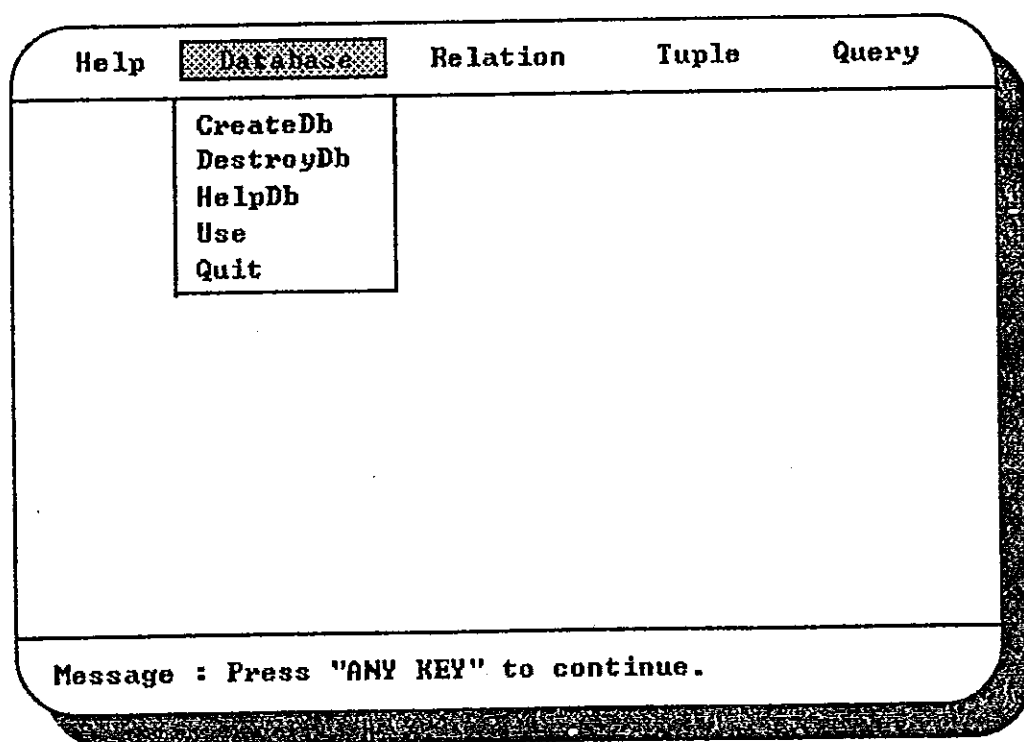
รายการนี้เป็นคำแนะนำทั่วไปเพื่ออำนวยความสะดวกแก่ผู้ใช้ ประกอบด้วยรายการย่อย 3 รายการ คือ Syntax, Key, และ ExitToShell ซึ่งคำแนะนำอธิบายคำสั่งต่าง ๆ ที่จำเป็นในระบบ อธิบายรายละเอียดเกี่ยวกับการใช้เพิ่มเติมควบคุมการทำงานและออกจากโปรแกรมชั่วคราว Mrdbms เพื่อคำแนะนำกับระบบปฏิบัติการ UTRIX-32 ตามลำดับ ดังแสดงในรูปที่ ค.2



รูปที่ ค.2 จอภาพรายการย่อยของรายการหลัก Help

1.3.2 รายการหลัก Database

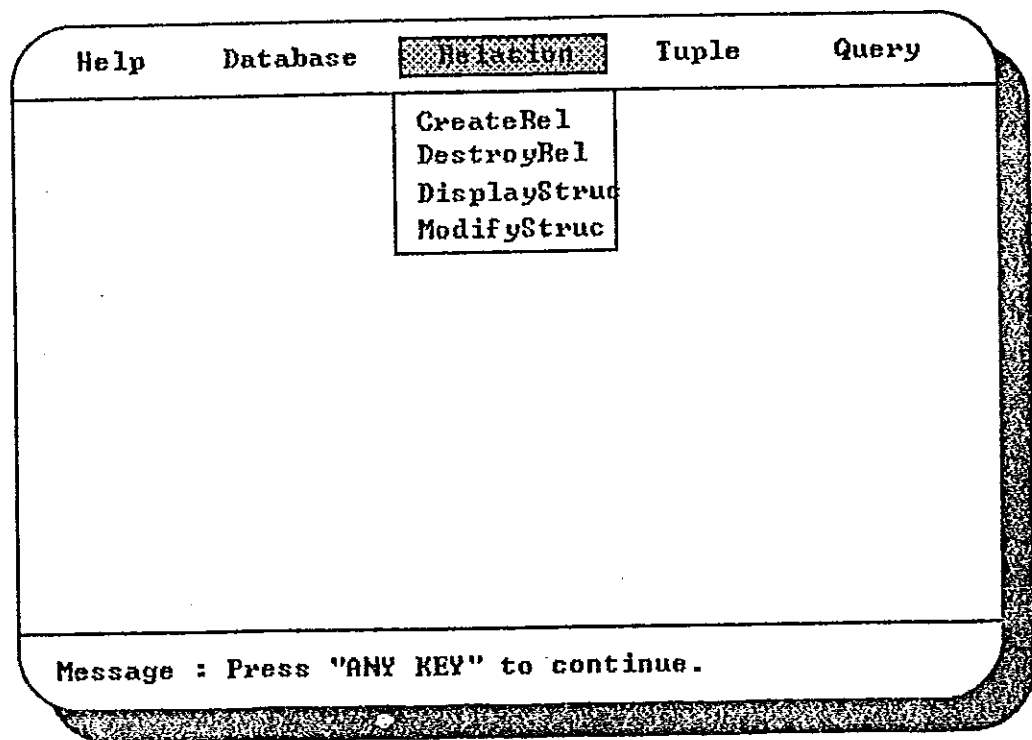
รายการนี้สำหรับการดำเนินงานต่าง ๆ กับฐานข้อมูลของผู้ใช้ ประกอบด้วยรายการย่อย 5 รายการ คือ CreateDb, DestroyDb, HelpDb, Use และ Quit สำหรับดำเนินงานสร้างฐานข้อมูล ลบฐานข้อมูล แสดงรายละเอียดของฐานข้อมูล เรียกใช้ฐานข้อมูล และออกจากการทำงานของโปรแกรมตามลำดับ ดังแสดงในรูปที่ ค.3



รูปที่ ค.3 จอภาพรายการย่อยของรายการหลัก Database

1.3.3 รายการหลัก Relation

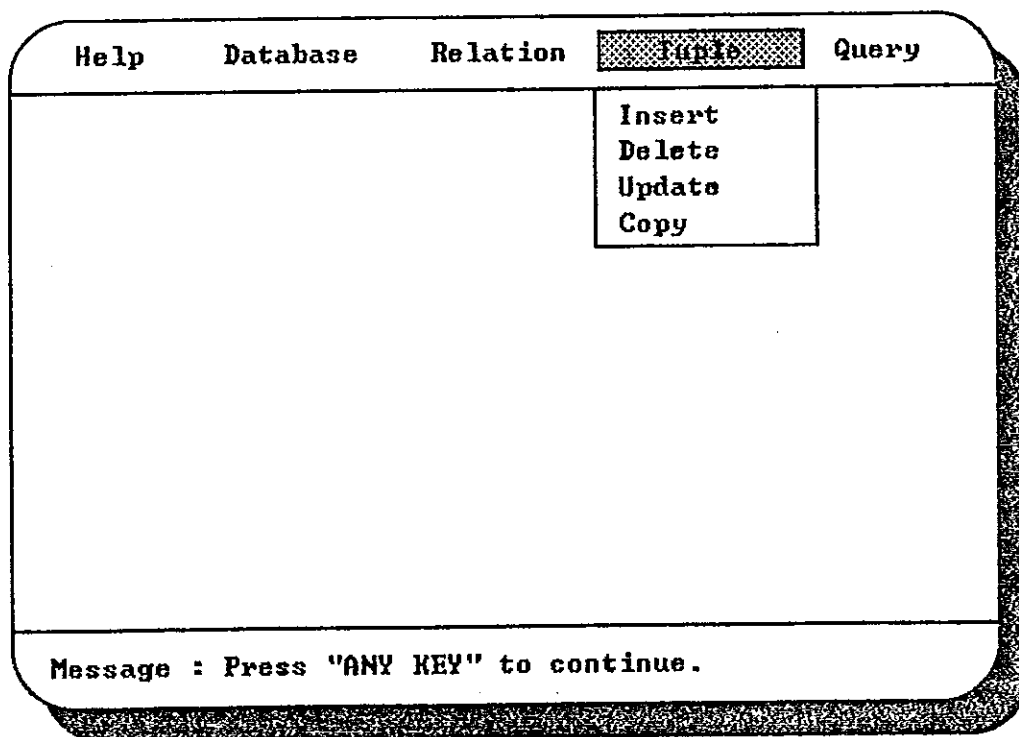
รายการนี้สำหรับให้ผู้ใช้ดำเนินงานต่าง ๆ กับตารางข้อมูลของฐานข้อมูลในจุดที่ถูกระบุไว้ ประกอบด้วยรายการย่อย 4 รายการ คือ CreateRel, DestroyRel, DisplayStruc และ ModifyStruc สำหรับดำเนินงานสร้างตารางข้อมูล ลบตารางข้อมูล แสดงรายละเอียดโครงสร้างตารางข้อมูลและเปลี่ยนแปลงรายละเอียดของโครงสร้างตารางข้อมูลตามลำดับ ดังแสดงในรูปที่ ค.4



รูปที่ ค.4 จอภาพรายการย่อยของรายการหลัก Relation

1.3.4 รายการหลัก Tuple

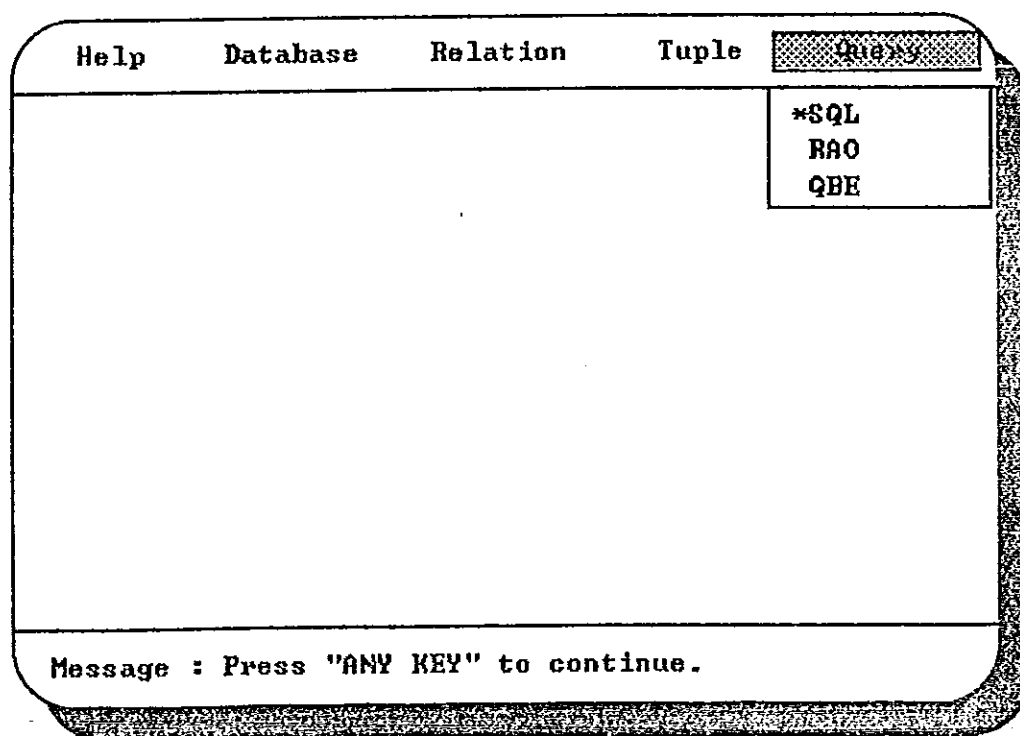
รายการนี้สำหรับให้ผู้ใช้ดำเนินการเกี่ยวกับแถวข้อมูลของตารางข้อมูลของฐานข้อมูลปัจจุบันที่ถูกเรียกใช้ ประกอบด้วยรายการย่อย 4 รายการ คือ Insert, Delete, Update และ Copy สำหรับดำเนินการเพิ่มข้อมูล ลบข้อมูล แก้ไขข้อมูล และสำเนาเพิ่มข้อมูลระหว่างระบบปฏิบัติการ ULTRIX-32 กับ Mrdbms ตามลำดับ ดังแสดงในรูปที่ ค.5



รูปที่ ค.5 จอภาพรายการย่อยของรายการหลัก Tuple

1.3.5 รายการหลัก Query

รายการนี้สำหรับให้ผู้ใช้สอบถามข้อมูลที่อยู่ในฐานข้อมูลโดยใช้ภาษาฐานข้อมูลต่าง ๆ ประกอบด้วยรายการย่อย 3 รายการ คือ SQL, RAO และ QBE สำหรับสอบถามฐานข้อมูลโดยใช้ ภาษาฐานข้อมูล SQL, คำสั่งปฏิบัติการพิเศษคณิตสัมพันธ์ RAO และภาษาฐานข้อมูล QBE ตามลำดับ ดังแสดงในรูปที่ ค.6

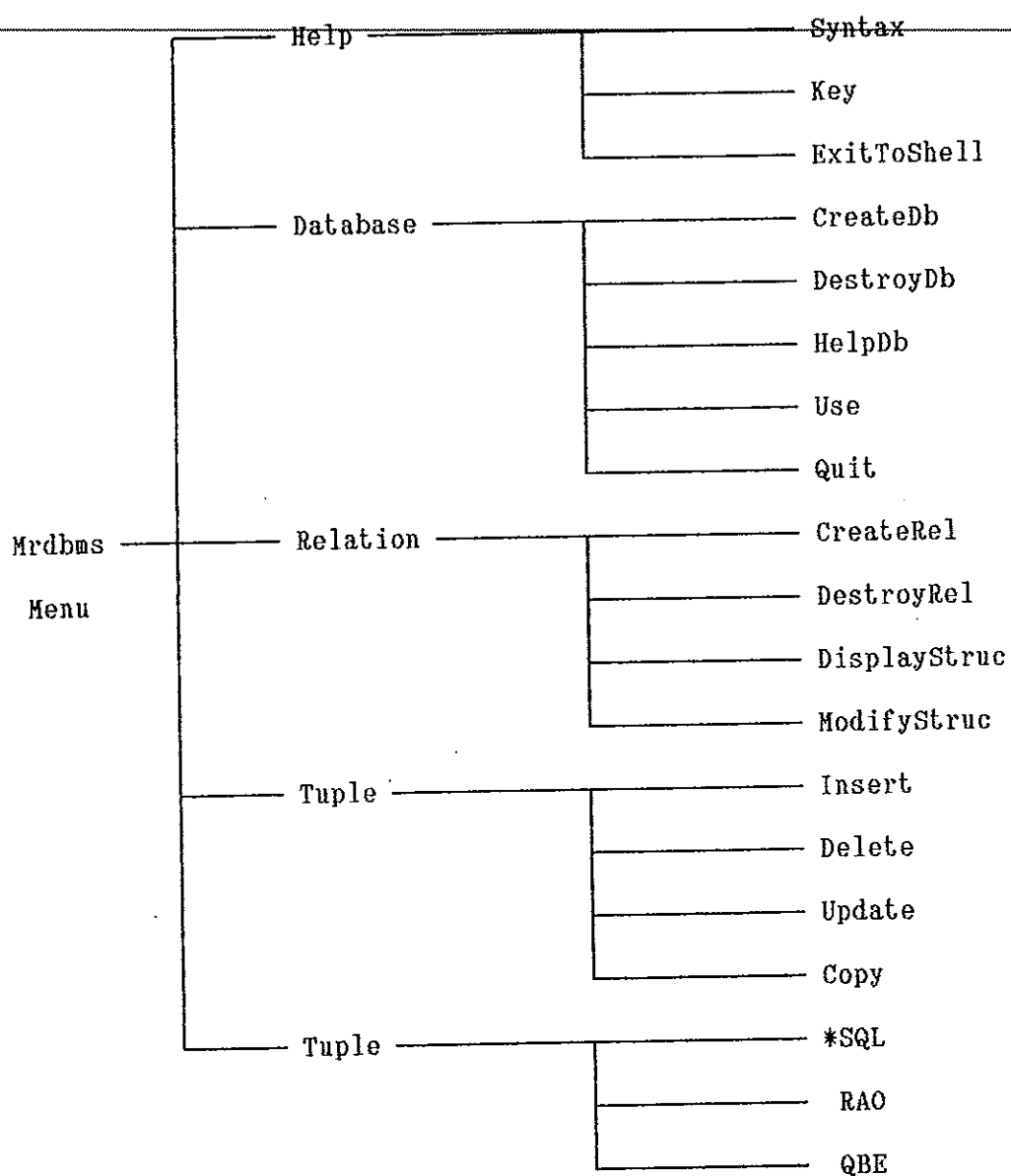


รูปที่ ค.6 จอภาพรายการย่อยของรายการหลัก Query

หมายเหตุ *SQL หมายถึงกำลังอยู่ในระหว่างการพัฒนาโปรแกรม

2. แผนภาพสรุปโครงสร้างรายการหลักและรายการย่อยของ Mrdbms

ภาพรวมรายการหลักและรายการย่อยของโปรแกรม Mrdbms อาจเห็นให้อยู่ในรูปแบบโครงสร้างต้นไม้ดังรูปที่ ค.7

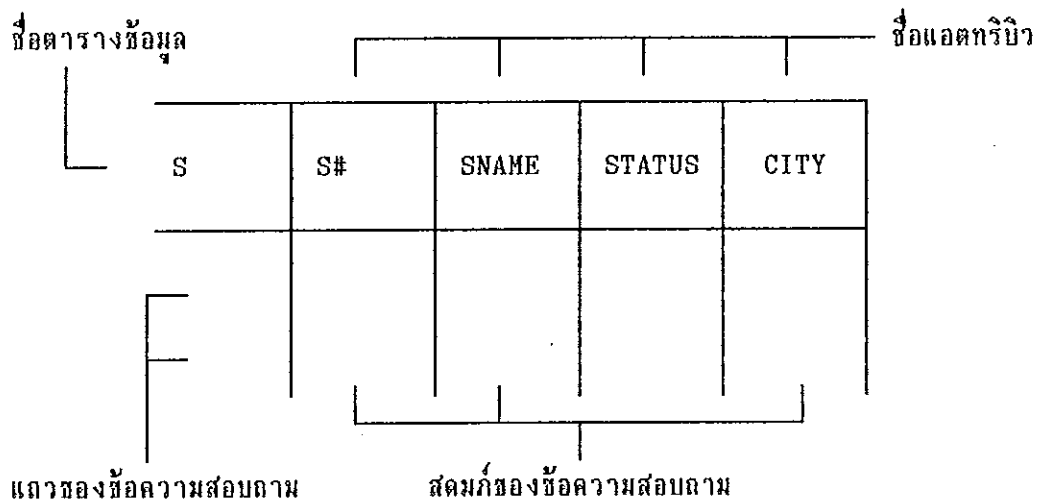


รูปที่ ค.7 โครงสร้างรายการหลักและรายการย่อยของโปรแกรม Mrdbms

3. วิธีใช้ตัวแปลภาษาฐานข้อมูล QBE

3.1 การแสดงผลทางจอภาพ

โปรแกรมการดำเนินงานของตัวแปลภาษา QBE มีรูปแบบการทำงานลักษณะเป็นการโต้ตอบระหว่างผู้ใช้กับระบบผ่านจอภาพคอมพิวเตอร์ เมื่อผู้ใช้ต้องการสอบถามข้อมูลจากตารางข้อมูลใดก็สามารถเรียกโครงร่างของตารางข้อมูลนั้น ๆ มาเพื่อกรอกข้อความสอบถามตามหลักวากยสัมพันธ์ของภาษา QBE รูปที่ ค.8 เป็นตัวอย่างโครงร่างตารางข้อมูลและตัวอย่างของการสอบถามฐานข้อมูลด้วยภาษา QBE โดยรายละเอียดส่วนประกอบต่าง ๆ ของโครงร่างตารางข้อมูลและข้อความสอบถามมีดังนี้



รูปที่ ค.8 ตัวอย่างการสอบถามฐานข้อมูลด้วยภาษา QBE บนโครงร่างตารางข้อมูลชื่อ S

รายละเอียดส่วนประกอบต่าง ๆ ดังนี้

- ชื่อตารางข้อมูล เป็นส่วนที่แสดงชื่อตารางข้อมูลที่ต้องการสอบถาม
- ชื่อแอตทริบิว เป็นส่วนที่แสดงชื่อของแอตทริบิวของตารางข้อมูลที่ต้องการสอบถาม โดยโปรแกรมจะนำชื่อดังกล่าวมาแสดงบนจอภาพหลังจากรับชื่อตารางข้อมูลจากผู้ใช้และชื่อตารางข้อมูลดังกล่าวปรากฏอยู่ในฐานข้อมูลที่ต้องการสอบถาม
- แถวและสดมภ์ของการสอบถาม เป็นส่วนที่ผู้ใช้กรอกข้อความสอบถาม เพื่อสอบถามฐานข้อมูลด้วยรูปแบบของภาษาสอบถาม QBE

3.2 การใช้แป้นพิมพ์ควบคุมการทำงาน

ผู้ใช้สามารถกดแป้นพิมพ์ "7" เพื่อดูรายละเอียดของการใช้แป้นพิมพ์ในการควบคุมการเลื่อนตำแหน่งสำหรับกรอกข้อความสอบถามและการทำงานของโปรแกรม รายละเอียดของแป้นพิมพ์ที่สามารถใช้ได้และการทำงานมีดังนี้

แป้นพิมพ์	การทำงาน
h หรือ ->	เลื่อนตำแหน่งไปยังสดมภ์ทางซ้าย 1 สดมภ์
l หรือ <-	เลื่อนตำแหน่งไปยังสดมภ์ทางขวา 1 สดมภ์
k หรือ ↑	เลื่อนตำแหน่งขึ้น 1 แถว
j หรือ ↓	เลื่อนตำแหน่งลง 1 แถว
H	เลื่อนตำแหน่งไปยังสดมภ์ของแอตทริบิวแรก
L	เลื่อนตำแหน่งไปยังสดมภ์ของแอตทริบิวสุดท้าย
K	เลื่อนตำแหน่งขึ้นไปอยู่แถวของการสอบถามแถวแรก
J	เลื่อนตำแหน่งลงไปอยู่แถวของการสอบถามแถวสุดท้าย
f	เลื่อนตำแหน่งไปยังตารางข้อมูลสอบถามถัดไป
b	เลื่อนตำแหน่งไปยังตารางข้อมูลสอบถามที่อยู่ก่อนหน้า 1 ตาราง
e	ประมวลผลข้อความสอบถามและแสดงผลการดำเนินงานทางจอภาพ
x	ลบตารางข้อมูลและข้อความสอบถามปัจจุบันที่เคอร์เซอร์ปรากฏอยู่
d	แสดงตารางว่างเพื่อใส่คำสั่งสอบถาม 1 ตาราง

- :n ยกเลิกตารางข้อมูลปัจจุบันที่ปรากฏบนจอภาพพร้อมข้อความสอบถาม ทั้งหมด และแสดงตารางข้อมูลให้ใหม่ร่างเปล่าให้ใหม่เพื่อการกรอกข้อความสอบถามใหม่
- :p แสดงตารางสอบถามเดิม
- :q เลิกการสอบถามและกลับไปยังโปรแกรม Mrdbms
- i เริ่มรับข้อความสอบถาม ณ. ตำแหน่งที่เคอร์เซอร์ปรากฏ
- r ยกเลิกข้อความสอบถามเก่าและรับข้อความสอบถามใหม่ ณ. ตำแหน่งที่เคอร์เซอร์ปรากฏ

3.3 การนำระบบเข้าสู่หน่วยความจำและการใช้งาน

การเรียกใช้โปรแกรมตัวแปลภาษาสอบถามฐานข้อมูล QBE จะต้องเรียกผ่านโปรแกรม Mrdbms โดยเมื่ออยู่บนระบบปฏิบัติการของ UNIX ใช้คำสั่งดังนี้

```
% Mrdbms
```

โปรแกรม Mrdbms เริ่มการดำเนินงานโดยแสดงข้อความเพื่อให้ผู้ใช้กำหนดชนิดของจอภาพที่กำลังใช้งานว่าเป็นแบบใดด้วยการกดแป้นพิมพ์ตัวอักษร "v" หรือ "m" เพื่อให้มีการปรับคุณสมบัติจอภาพให้สอดคล้องกับการทำงานของโปรแกรม ข้อความดังกล่าวมีดังนี้

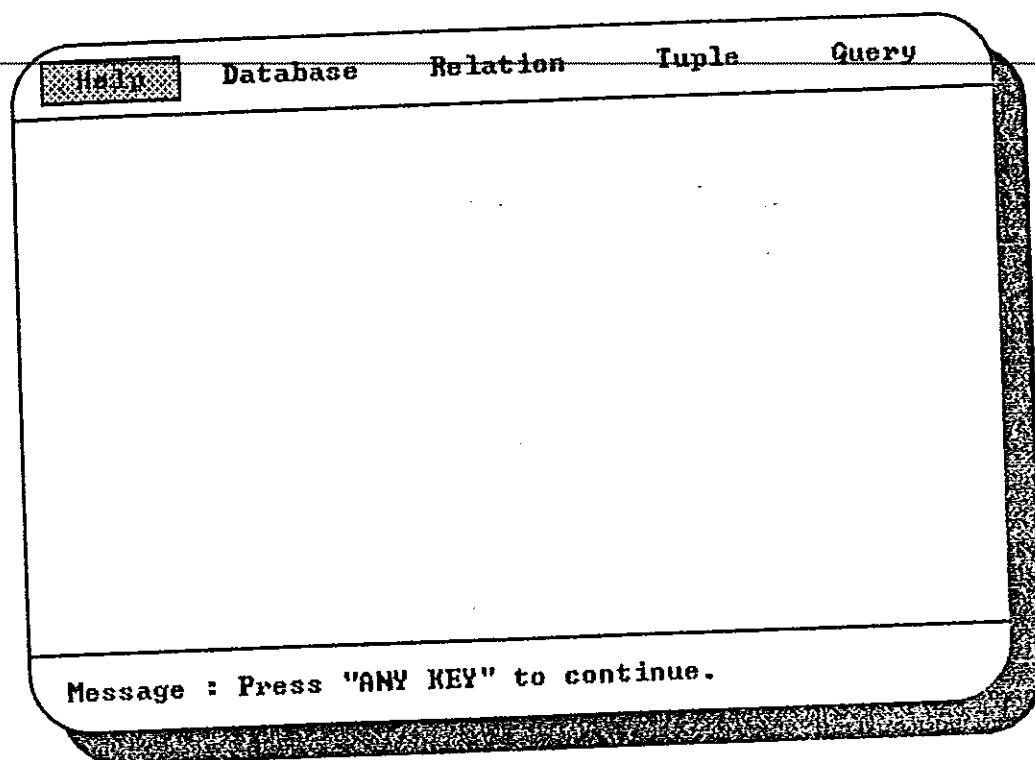
```
WHAT KIND OF TERMINAL ?
```

```
VT terminal
```

```
Micro terminal
```

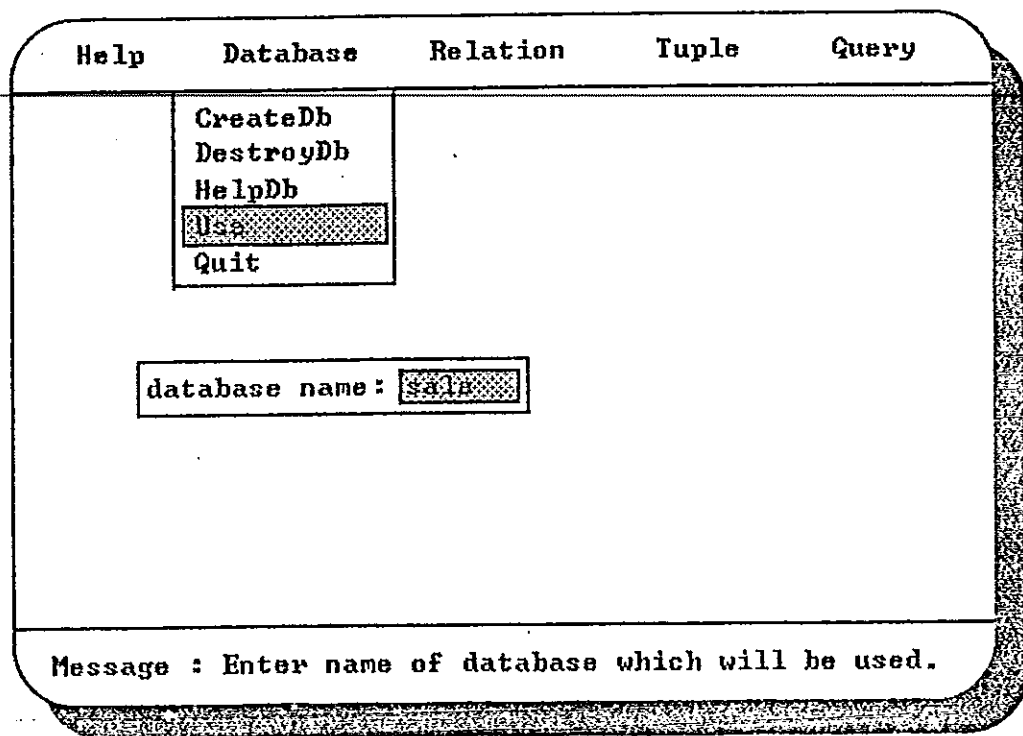
```
SELECT [_]
```

โปรแกรม Mrdbms เริ่มการทำงานโดยแสดงจอภาพดังรูปที่ ค.9



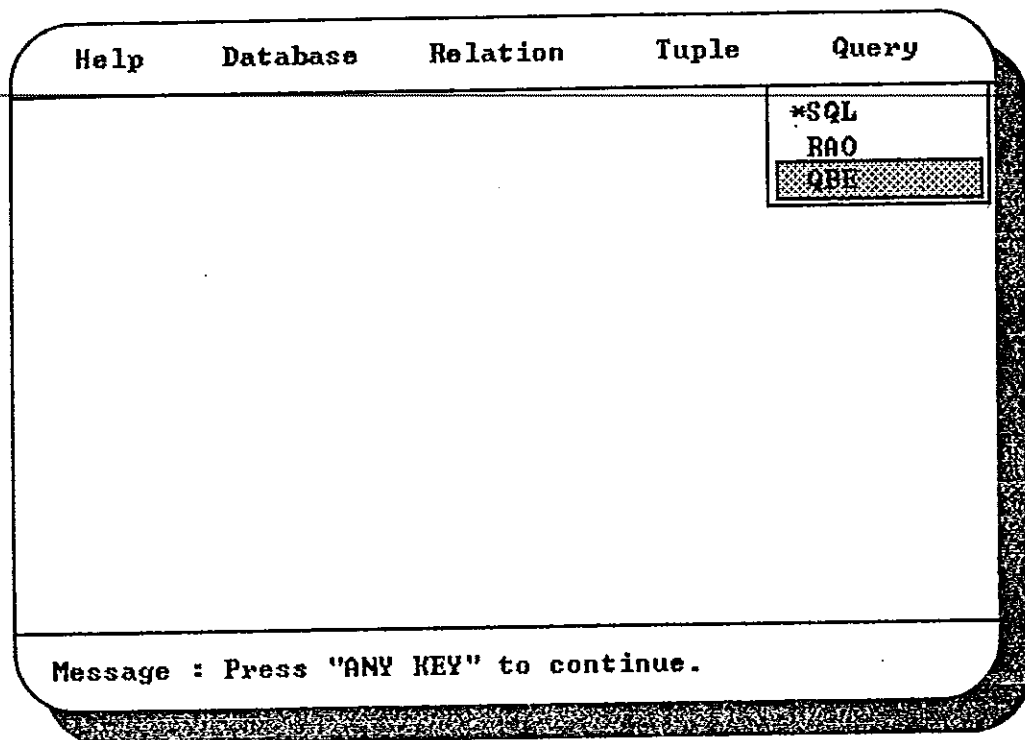
รูปที่ ค.9 จอภาพรายการหลักของโปรแกรม Mrdbms

เลือกรายการย่อย Use จากรายการหลัก Database เพื่อระบุชื่อของฐานข้อมูลที่ต้องการสอบถาม รูปที่ ค.10 แสดงการเลือกให้ฐานข้อมูลชื่อ "sale"



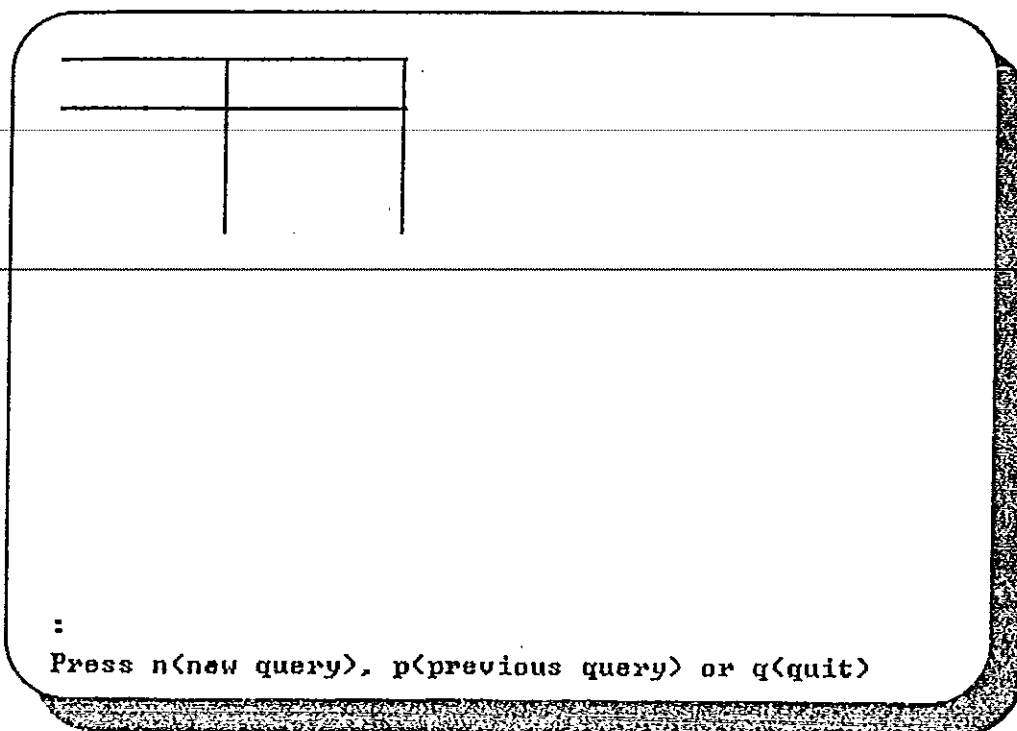
รูปที่ ค.10 จอภาพการระบุชื่อฐานข้อมูลที่ต้องการสอบถาม

คำเนนให้เลือกรายการย่อย QBE จากรายการหลัก Query เพื่อสอบถามข้อมูลจากฐานข้อมูลดังกล่าวด้วยโปรแกรมตัวแปลภาษาสอบถามฐานข้อมูล QBE ดังแสดงในรูปที่ ค.11



รูปที่ ค.11 จอภาพสำหรับเลือกการสอบถามข้อมูลจากฐานข้อมูลด้วย QBE

ผลที่ได้จากการดำเนินงานดังกล่าว ทำให้โปรแกรมตัวแปลภาษาสอบถามฐานข้อมูล QBE เริ่มการทำงานด้วยการแสดงจอภาพที่มีดังแสดงในรูปที่ ค.12 โครงร่างตารางว่างซึ่งพร้อมรับคำสั่งที่ใช้ในการสอบถาม



รูปที่ ค.12 จอภาพเริ่มต้นการทำงานของโปรแกรมตัวแปลภาษาสอบถามฐานข้อมูล QBE

ก่อนดำเนินการใด ๆ ต่อไป ผู้ใช้ต้องใส่ชื่อของตารางข้อมูลที่จะสอบถามลงใน ส่วนของชื่อตารางข้อมูล โดยตารางข้อมูลดังกล่าวจะต้องมีอยู่ในฐานข้อมูลแล้ว หลังจากนั้น โปรแกรมจะแสดงชื่อตารางข้อมูลและชื่อแอตทริบิวต์ทั้งหมดตารางดังกล่าวและพร้อมที่จะรับ ข้อความสั่งที่ใช้ในการสอบถามต่อไป เช่น ถ้าผู้ใช้ต้องการสอบถามข้อมูลจากตารางข้อมูล ชื่อ "S" ทำได้โดยใส่คำว่า S ลงในส่วนของชื่อตารางข้อมูล ดังแสดงในรูปที่ ค.13 จากนั้นระบบจะตรวจสอบว่ามีตารางข้อมูลชื่อ "S" อยู่ในเป้าหมายข้อมูลหรือไม่ ถ้ามีก็จะ แสดงโครงสร้างตารางข้อมูลชื่อ "S" ให้ผู้ใช้กรอกข้อความสอบถามต่อไป ดังแสดงในรูปที่ ค.14

S	

:
 Press n(new query), p(previous query) or q(quit)

รูปที่ ค.13 จอภาพแสดงโครงสร้างตารางข้อมูลและชื่อตารางที่ต้องการใช้สอบถาม

S	S#	SNAME	STATUS	CITY

:
 Press n(new query), p(previous query) or q(quit)

รูปที่ ค.14 จอภาพแสดงชื่อแอตทริบิวต์ของตารางข้อมูลที่ต้องการสอบถาม

จากนั้น ผู้ใช้สามารถกรอกข้อความลงในโครงร่างตารางข้อมูลดังกล่าวดังตัวอย่างแสดงในรูปที่ ค.15

S	S#	SNAME	STATUS	CITY
	p.s7			Paris

=
Press n(new query), p(previous query) or q(quit)

รูปที่ ค.15 ตัวอย่างของข้อความสั่งสอบถามฐานข้อมูลด้วยภาษา QBE

- p. ในสัดมภ์ชื่อ "S#" แทนข้อความสั่งให้พิมพ์ค่าข้อมูลในสัดมภ์นี้
- s7 คือ "ตัวอย่าง" โดยอักษรแรกจะต้องเป็นเครื่องหมาย "_" นั่นคือ "s7" จะได้จากการใส่ลำดับตัวอักษร "_s7" แทนข้อความ "เช่นตัวอย่าง s7 ในสัดมภ์ข้อมูลนี้"
- Paris คือ ค่าคงที่ แทนข้อความ "แถวข้อมูลของตารางข้อมูล S ที่มีค่าของ CITY เท่ากับ Paris"

โปรแกรมตัวแปลภาษาสอบถามฐานข้อมูล QBE กำหนดให้แต่ละสดมภ์สามารถรับตัวอักษรที่ใช้ในการสอบถามได้มากที่สุด 10 ตัว ดังนั้นหากการสอบถามมีจำนวนตัวอักษรมากกว่า 10 ตัว อักษรตัวที่ 10, 20 และ 30 จะต้องเป็นตัวอักษร '*' เพื่อให้เป็นเครื่องหมายว่าข้อความสอบถามดังกล่าวยังมีต่อ ทำให้หลังจากนั้นเคอร์เซอร์จะไปรอรับตัวอักษรตัวถัดไปในแถวไป เมื่อมีการประมวลข้อความสอบถาม เครื่องหมาย '*' จะถูกจัดออกไป

S	S#	SNAME	STATUS	CITY
	p.cnt.all*			
	.sx			

:
 Press n(new query), p(previous query) or q(quit)

รูปที่ ค.16 ตัวอย่างของข้อความสั่งสอบถามที่หนึ่งสดมภ์มีมากกว่า 10 ตัวอักษร

- เมื่อจัดเครื่องหมาย '*' จะได้ 'p.cnt.all.sx'
- เมื่อผู้ใช้กรอกข้อความสอบถามครบถ้วนแล้ว สามารถสั่งประมวลผลการสอบถามด้วยการกดแป้นพิมพ์ "e" ผลลัพธ์ที่ได้จากการประมวลผลการสอบถามนี้แสดงในรูปที่ ค.17

S	S#
	S2
	S3

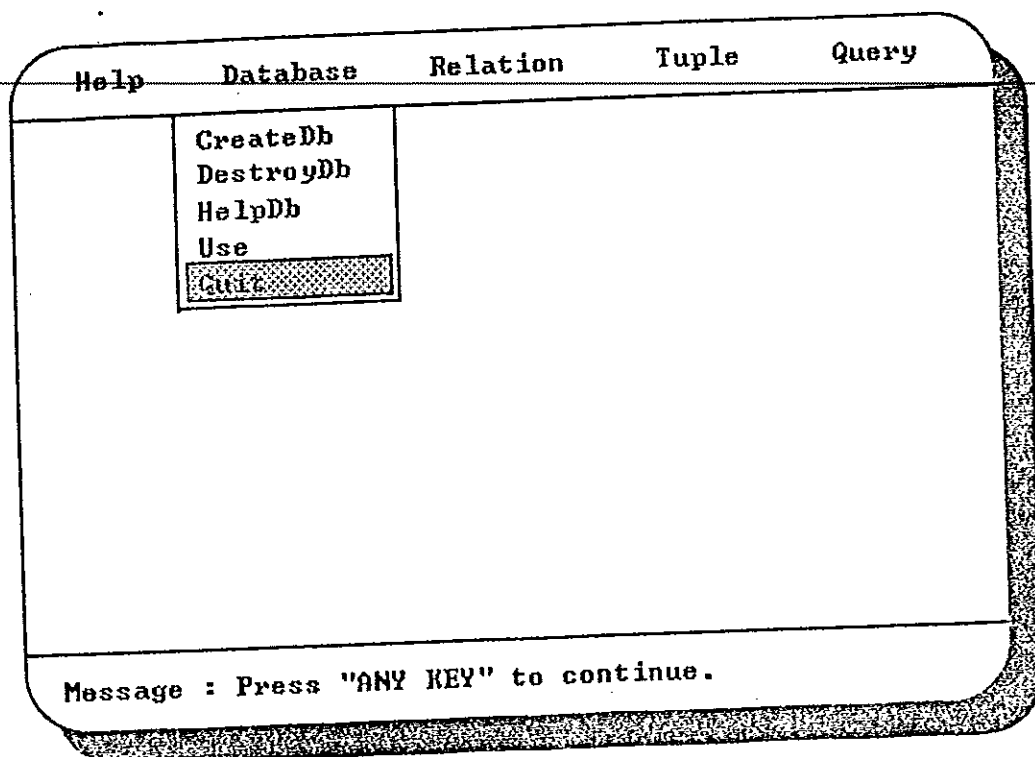
:
 Press n(new query), p(previous query) or q(quit)

รูปที่ ค. 17 ตัวอย่างจอภาพแสดงผลลัพธ์จากการประมวลผลการสอบถาม

หลังจากเสร็จสิ้นการประมวลผลแต่ละการสอบถาม ผู้ใช้สามารถที่จะเลือกดำเนินการต่อได้ดังนี้

- ถ้าผู้ใช้กดแป้นพิมพ์ "n" ระบบจะเริ่มการสอบถามใหม่ด้วยการยกเลิกจอภาพและข้อความสอบถามเดิม พร้อมทั้งแสดงจอภาพดังรูปที่ ค.12 เพื่อรับข้อความสั่งสอบถามใหม่
- ถ้าผู้ใช้กดแป้นพิมพ์ "p" จะแสดงจอภาพของการสอบถามเดิมเพื่อให้ผู้ใช้ปรับปรุงเปลี่ยนแปลงใหม่
- ถ้าผู้ใช้กดแป้นพิมพ์ "q" ระบบจะยกเลิกการทำงานของโปรแกรมตัวแปลภาษาสอบถามฐานข้อมูล QBE และเข้าสู่การดำเนินงานของโปรแกรม Mrdbms ต่อไป โดย

แสดงจอกภาพดังรูปที่ ค.9 เมื่อผู้ใช้ต้องการออกจากโปรแกรม Mrdbms ให้เลือกรายการย่อย Quit จากรายการหลัก Database



รูปที่ ค.18 จอภาพเมื่อเลิกการทำงานของโปรแกรม Mrdbms

การดำเนินการดังรูปที่ ค.18 สามารถทำได้เช่นเดียวกับเมื่ออยู่ในหน้าจอของการเขียนคำสั่งสอบถาม โดยกดแป้นพิมพ์ ":" แล้วจึงกดแป้นพิมพ์ "n", "p" หรือ "q"

ประวัติผู้เขียน

ชื่อ นายพงษ์พันธ์ ฟิลโท

วัน เดือน ปีเกิด วันศุกร์ที่ 28 เดือน กรกฎาคม พ.ศ. 2510

วุฒิการศึกษา

วุฒิ

ชื่อสถาบัน

ปีที่สำเร็จการศึกษา

วิทยาศาสตร์บัณฑิต มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

2532

(คณิตศาสตร์)

ทุนการศึกษา (ที่ได้รับระหว่างการศึกษา)

สถาบันวิจัยวิทยาศาสตร์และเทคโนโลยีแห่งประเทศไทย