



การพัฒนาภาษาสอบถามฐานข้อมูล SQL

An Implementation of SQL

วรากรณ์ วีระวิทยา

Varaporn Veravithaya

1

วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

มหาวิทยาลัยสงขลานครินทร์

Master of Science Thesis in Computer Science

Prince of Songkla University

2538

๒

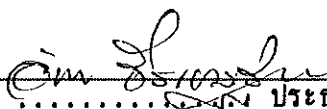
เลขหมู่	QA96.9.D3	A46	8538	Q.2
Bib Key	84515 84514			

(1)

ชื่อวิทยานิพนธ์ การพัฒนาภาษาสอบถามฐานข้อมูล SQL
ผู้เขียน นางสาววราภรณ์ วีระวิทยา
สาขาวิชา วิทยาการคอมพิวเตอร์

คณะกรรมการที่ปรึกษา

คณะกรรมการสอบ


..... ประธานกรรมการ


..... ประธานกรรมการ

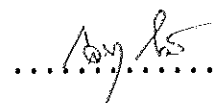
(ผู้ช่วยศาสตราจารย์ ดร. อัจฉา ชีร์เชษฐมงคล) (ผู้ช่วยศาสตราจารย์ ดร. อัจฉา ชีร์เชษฐมงคล)

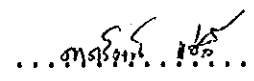
...(ลาศึกษาต่อ).. กรรมการ

...(ลาศึกษาต่อ).. กรรมการ

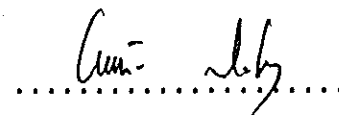
(ผู้ช่วยศาสตราจารย์ อิว ไอบรากาศญจนกุล)

(ผู้ช่วยศาสตราจารย์ อิว ไอบรากาศญจนกุล)


..... กรรมการ
(อาจารย์ เลขา ไชยสร)


..... กรรมการ
(อาจารย์ คารารัตน์ แซ่ลี่)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้บัณฑิตวิทยานิพนธ์ ฉบับนี้เป็น
ส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์


.....

(ดร. ไพรัตน์ สงวนไพร)

คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์	การพัฒนาภาษาสอบถามฐานข้อมูล SQL
ผู้เขียน	นางสาววรารภรณ์ วีระวิทยา
สาขาวิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2538

บทคัดย่อ

ภาษาสอบถามฐานข้อมูลเชิงสัมพันธ์ถูกคิดขึ้นมามากมายหลายรูปแบบ แต่ที่เป็นที่นิยมใช้กันมากที่สุดได้แก่ ภาษา SQL ซึ่งเป็นภาษาฐานข้อมูลที่ใช้ดำเนินงานกับระบบจัดการฐานข้อมูลเชิงสัมพันธ์หลายระบบในปัจจุบัน และเป็นที่ยอมรับจาก ANSI ให้เป็นภาษาฐานข้อมูลมาตรฐาน

งานวิทยานิพนธ์นี้สร้างตัวแปลภาษาสอบถามฐานข้อมูล SQL ที่เป็นมาตรฐานของ ANSI และเพิ่มเติมบางรูปแบบของข้อความสอบถาม เพื่อให้สามารถครอบคลุมการดำเนินงานสอบถามฐานข้อมูลได้มากขึ้น โดยรูปแบบของข้อความสอบถามที่เพิ่มเติมได้ออกแบบและเลือกมาจาก SQL ที่ใช้ในระบบจัดการฐานข้อมูลโอราเคิล โดยเน้นการใช้รูปแบบที่สั้นและเข้าใจความหมายได้ง่าย ตัวแปลภาษาที่ถูกสร้างจะครอบคลุมการตรวจสอบวากยสัมพันธ์และการตีความหมายของแต่ละข้อความสั่ง ส่วนการดำเนินงานประมวลผลตามความหมายจะเรียกใช้โปรแกรมการดำเนินงานกับฐานข้อมูลด้วยคำสั่งปฏิบัติการของภาษาสอบถามฐานข้อมูลเชิงสัมพันธ์แบบที่ซคณิตสัมพันธ์ของงานวิทยานิพนธ์ที่มีมาก่อนหน้านี้

Thesis Title	An Implementation of SQL
Author	Miss Varaporn Veravithaya
Major Program	Computer Science
Academic Year	1995

Abstract

The relational database query language has been invented in various forms. The most popular one is "SQL". The SQL is a database language which is implemented presently on many relational database management systems in the market and is accepted by the ANSI as a standard database language.

This thesis presents an interpreter for the ANSI's standard "SQL" query language, which is normally called "selected statement", with some extension to cover more detailed queries. The pattern (syntax) of each extended query has been designed on and selected from the SQL which is being used in the ORACLE RDBMS. Usage of short pattern, as well as being easy to understand have been emphasized. The developed interpreter covers only the syntax checking and semantics interpreting. The processing of the statements is done by calling various routines of the relational algebra operations which are the programs developed by the former thesis in the Department.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงไปด้วยดี ด้วยความช่วยเหลือและสนับสนุนจากบุคคล
หลายฝ่าย ซึ่งผู้วิจัยขอกราบขอบพระคุณอย่างสูงมา ณ. โอกาสนี้ ได้แก่

ผู้ช่วยศาสตราจารย์ ดร. อัญญา ชีรเชษฐมงคล อาจารย์ที่ปรึกษา ที่กรุณาให้คำ
ปรึกษา ชี้แนวทาง แก้ไขและตรวจทานวิทยานิพนธ์ให้แก่ผู้วิจัยมาโดยตลอด

ผู้ช่วยศาสตราจารย์ อิว ไอยราภาณจนกุล อาจารย์ที่ปรึกษาร่วม ที่กรุณาให้คำ
ปรึกษาแก่ผู้วิจัย

คณะกรรมการสอบวิทยานิพนธ์ทุกท่าน ที่กรุณาช่วยตรวจและแก้ไขวิทยานิพนธ์
คุณพ่อ คุณแม่ และทุกคนในครอบครัว ที่ให้ความสนับสนุนทุนการศึกษา เป็นกำลัง
ใจในการทำวิทยานิพนธ์มาโดยตลอด

เจ้าหน้าที่ศูนย์คอมพิวเตอร์ และภาควิชาคณิตศาสตร์ทุกท่านที่ให้ความช่วยเหลือ
อำนวยความสะดวกความสะดวกตลอดเวลา

เพื่อนๆ พี่ๆ ที่น่ารักทุกคน ที่ให้กำลังใจในการทำวิทยานิพนธ์

วราภรณ์ วีระวิทยา

สารบัญ

	หน้า
บทคัดย่อ	(3)
Abstract	(4)
กิตติกรรมประกาศ	(5)
สารบัญ	(6)
รายการรูป	(9)
รายการตาราง	(10)
บทที่	
1 บทนำ	1
1.1 วัตถุประสงค์ของงานวิจัย	2
1.2 ขอบเขตของงานวิจัย	2
1.3 ขั้นตอนและระยะเวลาการดำเนินงาน	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ	5
1.5 สถานที่และอุปกรณ์ที่ใช้	5
1.6 สรุปเนื้อหา	6
2 ภาษารฐานข้อมูล SQL	7
2.1 ประวัติความเป็นมา	7
2.2 ลักษณะของภาษา	8
2.2.1 การกำหนดโครงสร้าง/นิยาม	12
2.2.2 ภาษาคำเนิงาน	13
2.2.3 ภาษาสอบถามฐานข้อมูล	14

สารบัญ (ต่อ)

	หน้า
2.3 ภาษา SQL มาตรฐานและภาษา SQL ที่ใช้ในซอฟต์แวร์ ORACLE	19
2.4 การประมวลผลข้อความสอบถามฐานข้อมูลของ SQL	25
3 โครงสร้างของระบบ Mrdbms	53
3.1 โครงสร้างของระบบ	53
- ปทานุกรมข้อมูล	53
- โครงสร้างแฟ้มข้อมูลของปทานุกรมข้อมูล	54
- สารบบแฟ้มข้อมูล	57
- โครงสร้างข้อมูล	58
- แผนภูมิระบบ	60
3.2 โครงสร้างระบบ Mrdbms ใหม่	61
4 ตัวแปลภาษา SQL	62
4.1 ตัวแปลภาษาระดับสูง	62
4.2 โปรแกรม Lex และ Yacc	66
4.2.1 โปรแกรม Lex	66
4.2.2 โปรแกรม Yacc	71
4.3 องค์ประกอบของตัวแปลภาษา SQL	74

สารบัญ (ต่อ)

	หน้า
4.4 โครงสร้างของตัวแปลภาษา SQL	80
- ขั้นตอนการวิเคราะห์ศัพท์	80
- ขั้นตอนการวิเคราะห์วากยสัมพันธ์	87
- ขั้นตอนการวิเคราะห์ความหมายและตีความ	92
4.5 การดำเนินงานของระบบ	94
4.6 การดำเนินงานเชื่อมโยงกับคำสั่งปฏิบัติการของภาษาพีชคณิตสัมพันธ์	104
4.7 การตรวจสอบความผิดพลาด	112
5 สรุปและข้อเสนอแนะ	115
5.1 ปัญหาและอุปสรรค	116
5.2 ข้อเสนอแนะ	116
บรรณานุกรม	118
ภาคผนวก ก วากยสัมพันธ์ของข้อความสอบถาม SQL ในรูปผังภาพวากยสัมพันธ์	121
ภาคผนวก ข เอกสารสำหรับผู้ไ้	132
ประวัติผู้เขียน	145

รายการรูป

รูปที่	หน้า
2.1 ตารางข้อมูล S	10
2.2 ตารางข้อมูล P	10
2.3 ตารางข้อมูล SP	11
2.4 ตารางข้อมูล A	16
2.5 ตารางข้อมูล B	16
3.1 แผนภาพไออาร์ของปทานุกรมข้อมูลและตารางข้อมูลที่สัมพันธ์กัน	54
3.2 การจัดเก็บสารบบแฟ้มข้อมูลของ Mrdbms	57
3.3 โครงสร้างข้อมูลของปทานุกรมข้อมูลขณะที่อยู่ในหน่วยความจำหลัก	59
3.4 แผนภูมิระบบ Mrdbms	60
3.5 แผนภูมิระบบ Mrdbms ใหม่	61
4.1 ขั้นตอนการดำเนินงานทั่วไปของคอมไพเลอร์และอินเทอร์พรีเตอร์	63
4.2 ขั้นตอนต่างๆ ของอินเทอร์พรีเตอร์ที่ใช้ในงานวิทยานิพนธ์นี้	65
4.3 ข้อมูลเข้า/ออกของ Lex และ yylex()	66
4.4 ข้อมูลเข้า/ออกของ Yacc และ yyparse()	71
4.5 รูปแบบของข้อความสอบถาม SQL	77
4.7 แผนภาพกระแสข้อมูลระบบ Mrdbms	99
4.8 แผนภาพกระแสข้อมูลของระบบ SQL ในรายการหลัก	100
4.9 แผนภาพกระแสข้อมูลของกระบวนการ Select Statement	101
4.10 แผนภาพกระแสข้อมูลของกระบวนการ Additional Command	102
ข.1 จอภาพรายการหลัก Query ของระบบ Mrdbms	133
ข.2 การใช้โปรแกรมตัวแปลภาษา	134

รายการรูป (ต่อ)

รูปที่		หน้า
ข.3	รายการ HELP	135
ข.4	รูปแบบข้อความสอบถาม SELECT	138
ข.5	จอภาพขณะใช้งานของรายการย่อย Quit	144

รายการตาราง

ตารางที่		หน้า
2.6	ข้อความสั่งของภาษาสอบถามฐานข้อมูล SQL และลำดับของข้อความสั่งของภาษา พีรคณิตสัมพันธ์ที่สมนัยกัน	27
4.6	หมายเลขรหัสแทนการดำเนินงานของตัวแปลภาษา SQL	92
4.11	ข้อความแสดงข้อผิดพลาดและความเสียหายและความเสียหาย	113

บทที่ 1

บทนำ

งานวิทยานิพนธ์นี้เป็นงานพัฒนาส่วนหนึ่งของโปรแกรมระบบจัดการฐานข้อมูลซึ่งต่อเนื่องจากงานวิทยานิพนธ์ที่ดำเนินงานมาก่อนหน้านี้สองชิ้นคือ "โปรแกรมส่วนการกำหนดโครงสร้างฐานข้อมูลและการดำเนินงานพื้นฐาน" [เพียรพร (2534)] ซึ่งพัฒนาโปรแกรมในส่วนการกำหนดโครงสร้างฐานข้อมูลและการดำเนินงานพื้นฐานกับฐานข้อมูล เช่น การลบ การเพิ่มและการปรับปรุงเปลี่ยนแปลงค่าข้อมูลในฐานข้อมูล ตลอดจนปรับปรุงเปลี่ยนแปลงรายละเอียดต่างๆ ของโครงสร้างฐานข้อมูล และงานวิทยานิพนธ์ "การพัฒนาโปรแกรมจัดการฐานข้อมูลเชิงสัมพันธ์ ส่วนภาษาฐานข้อมูล" [ทวีศักดิ์ (2534)] ซึ่งพัฒนาโปรแกรมดำเนินงานกับฐานข้อมูลที่ถูกสร้างจากงานวิทยานิพนธ์แรกด้วยคำสั่งปฏิบัติการของภาษาฐานข้อมูลที่ชคณิตสัมพันธ์ (Relational Algebra Database Language) โดยโปรแกรมระบบจัดการฐานข้อมูลที่ถูกพัฒนามาก่อนหน้านี้มีชื่อว่า Mrdbms (Mini-relational database management system) และงานวิทยานิพนธ์นี้ได้พัฒนาเพิ่มเติมจากงานวิทยานิพนธ์เดิมในส่วนของการสร้างตัวแปลภาษาสอบถามฐานข้อมูล (Database Query Language) ชื่อ SQL (Structural Query Language) ซึ่งเป็นภาษาที่ใช้ในการสอบถามข้อมูลจากฐานข้อมูลเชิงสัมพันธ์เพื่อให้ได้มาซึ่งคำตอบที่ต้องการได้อย่างรวดเร็วและถูกต้อง โดยมีวิธีการป้อนคำถามในลักษณะรูปแบบที่กำหนดไว้ตามไวยากรณ์ของภาษาแบบง่าย ๆ ผู้ใช้ไม่จำเป็นต้องมีความรู้ด้านภาษาคอมพิวเตอร์ทั่วไป เช่น ภาษาโคบอล (COBOL) หรือภาษาปาสคาล (Pascal)

ภาษาสอบถามฐานข้อมูลเชิงสัมพันธ์ถูกคิดค้นขึ้นมามากมายหลายรูปแบบในช่วงสองทศวรรษที่ผ่านมา แต่ที่เป็นที่นิยมใช้กันมากได้แก่ ภาษา SQL ซึ่งเป็นภาษาฐานข้อมูลที่ใช้ดำเนินงานกับระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) หลายระบบในปัจจุบัน ภาษา SQL ได้รับการยอมรับจาก ANSI (American National Standard Institute) ให้เป็นภาษาฐานข้อมูลมาตรฐานเมื่อปี ค.ศ. 1982

ถึงแม้ว่า ภาษาสอบถามฐานข้อมูล SQL จะถูกยอมรับเป็นภาษามาตรฐานโดย ANSI แล้ว แต่การดำเนินงานสอบถามข้อมูลจากฐานข้อมูลยังกระทำไม่ได้ไม่ครบสมบูรณ์ทุกรูปแบบที่ต้องการ ทำให้บริษัทผู้ผลิตซอฟต์แวร์ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ทั้งหลายซึ่งใช้ SQL ได้นำรูปแบบของ SQL มาตรฐานไปปรับปรุง เปลี่ยนแปลงและเพิ่มเติมเพื่อให้สอดคล้องกับการดำเนินงานตามที่ต้องการเหมาะสม

งานวิทยานิพนธ์นี้ศึกษารูปแบบโครงสร้างภาษาและดำเนินการสร้างตัวแปลภาษา

สอบถาม SQL ที่เป็นมาตรฐานของ ANSI และเพิ่มเติมบางรูปแบบที่มีใช้ในระบบจัดการฐานข้อมูล ORACLE ที่ไม่มีใน SQL มาตรฐาน เพื่อให้สามารถครอบคลุมการดำเนินงานการสอบถามฐานข้อมูลได้มากขึ้น

1.1 วัตถุประสงค์ของงานวิจัย

1. ออกแบบและพัฒนาโปรแกรมแปลภาษา SQL โดยใช้โปรแกรมมอรรถประโยชน์ Lex และ Yacc ของระบบปฏิบัติการ UNIX ช่วยในการพัฒนา
2. เชื่อมโยงโปรแกรมที่พัฒนากับโปรแกรม Mrdbms ที่มีมาก่อนหน้านี้เข้าด้วยกัน โดยให้ทำงานร่วมกันได้

1.2 ขอบเขตของงานวิจัย

1. ศึกษาและพัฒนาเฉพาะส่วนที่เป็นภาษาสอบถามฐานข้อมูลของภาษา SQL
2. วากยสัมพันธ์ที่ศึกษาเพื่อนำไปพัฒนาโปรแกรมใช้รูปแบบของ SQL มาตรฐานและ SQL ที่ใช้ใน ORACLE เพื่อให้ครอบคลุมถึงการดำเนินงานการสอบถามข้อมูล
3. ดำเนินงานในส่วนของการวิเคราะห์ศัพท์และวากยสัมพันธ์ ส่วนการตีความหมาย การดำเนินงานใช้ routine ที่มีอยู่แล้วของการดำเนินงานที่ชนิดสัมพันธ์และบางส่วนได้จากการพัฒนาเพิ่มเติม

ระยะเวลาในการดำเนินงาน

ใช้เวลาประมาณ 4 ปีตั้งรายละเอียดแสดงด้วยผังภาพต่อไปนี้

ขั้นตอน ที่	กพ.	มีค.	เมษ.	พค.	มิย.	กค.	สค.	กย.	ตค.	พย.	ธค.	มค.	กพ.	มีค.
	-2534-											2535-2538		
1	*													
2		*												
3			*											
4			*	*									*	
5				*	*	*	*	*	*	*	*			
6											*	*	*	*
7											*	*	*	*

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ทำให้ผู้วิจัยเข้าใจวิธีการพัฒนาส่วนหนึ่งของระบบจัดการฐานข้อมูลดีขึ้น
2. ได้โปรแกรมแปลภาษาสอบถามฐานข้อมูล SQL ซึ่งเป็นส่วนประกอบหนึ่งของระบบจัดการฐานข้อมูลเชิงสัมพันธ์
3. ทำให้งานพัฒนาระบบจัดการฐานข้อมูลเชิงสัมพันธ์ของงานวิทยานิพนธ์ที่มีมาก่อนหน้านี้มีความสมบูรณ์มากยิ่งขึ้น

1.5 สถานที่และอุปกรณ์ที่ใช้

สถานที่

ห้องปฏิบัติการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์ มหาวิทยาลัยสงขลานครินทร์
วิทยาเขตหาดใหญ่

อุปกรณ์

1. เครื่องคอมพิวเตอร์ VAX 11/785 ซึ่งเป็นมินิคอมพิวเตอร์ที่มีหน่วยความจำหลัก 12 MB ใช้ระบบปฏิบัติการ Ultrix-32
2. เทอมินัลชนิด VT 220 และเทอมินัลชนิดไมโครคอมพิวเตอร์
3. ตัวแปลภาษาซี (C Compiler) มาตรฐาน
4. โปรแกรมอรรถประโยชน์ตรวจสอบกลุ่มคำ (Token) ชื่อ Lex
5. โปรแกรมอรรถประโยชน์ช่วยการตรวจไวยากรณ์ของภาษา SQL ชื่อ Yacc
6. โปรแกรมอรรถประโยชน์ช่วยการตรวจและแก้ไขข้อผิดพลาดขณะทำงานของโปรแกรมภายใต้ระบบปฏิบัติการ Ultrix-32 ชื่อ Dbx
7. เครื่องพิมพ์

1.6 สรุปเนื้อหา

ในวิทยานิพนธ์เล่มนี้มีรายละเอียดสรุปเนื้อหาดังต่อไปนี้ บทที่ 1 กล่าวถึงความเป็นมาของปัญหา วัตถุประสงค์ ขอบเขต ขั้นตอนการดำเนินงานและระยะเวลาในการทำวิทยานิพนธ์ รวมถึงประโยชน์ที่คาดว่าจะได้รับในการทำวิทยานิพนธ์ครั้งนี้ บทที่ 2 กล่าวถึงการศึกษาแนวความคิด ประวัติความเป็นมา วากยสัมพันธ์และความหมายของคำสั่งสอบถามของภาษาสอบถามฐานข้อมูล SQL ตลอดจนการเปรียบเทียบลักษณะไวยากรณ์ของภาษาสอบถามฐานข้อมูล SQL มาตรฐานที่กำลังใช้อยู่ในระบบจัดการฐานข้อมูลบางตัว บทที่ 3 จะสรุปรายละเอียดโครงสร้างของระบบ Mrdbms เดิมและโครงสร้างระบบ Mrdbms ใหม่ ส่วนบทที่ 4 กล่าวถึงการพัฒนาตัวแปลภาษา SQL โดยจะสรุปโครงสร้างของตัวแปลภาษา ขั้นตอนการแปลภาษารวมถึงการดำเนินงานและขั้นตอนในการดำเนินงานกับข้อความสั่ง SQL และบทที่ 5 คือบทสรุป กล่าวถึงปัญหาและอุปสรรคในการทำวิทยานิพนธ์ครั้งนี้ รวมถึงข้อเสนอแนะแนวทางในการพัฒนาโปรแกรมส่วนอื่นของระบบจัดการฐานข้อมูล Mrdbms

ภาษาฐานข้อมูล SQL

ส่วนประกอบหลักของระบบจัดการฐานข้อมูล (Database Management -

System DBMS) ซึ่งเป็นด่านแรกสำหรับให้ผู้ใช้สามารถติดต่อดำเนินงานกับฐานข้อมูลได้ คือ ภาษาฐานข้อมูล ผู้ใช้สามารถใช้ภาษาฐานข้อมูลติดต่อกับฐานข้อมูลโดยผ่านทางระบบจัดการ

ฐานข้อมูลโดยผู้ใช้ไม่จำเป็นต้องทราบวิธีการจัดเก็บข้อมูลจริงๆ ว่าถูกจัดเก็บอยู่ในรูปแบบใด และไม่ต้องเรียนรู้การเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์ใดๆ ในการประมวลผลกับฐานข้อมูล ภาษาฐานข้อมูลจะประกอบด้วยข้อความสั่งดำเนินการที่คล้ายคลึงกับภาษาคอมพิวเตอร์ระดับสูงต่างๆ ไป แต่ถูกออกแบบลักษณะของภาษามาเป็นพิเศษสำหรับการทำงานกับข้อมูลในฐานข้อมูลโดยเฉพาะ เพื่อให้ง่ายและสะดวกต่อการใช้สำหรับผู้ใช้ทุกๆ ไป

ภาษา SQL อ่านออกเสียงอย่างเป็นทางการว่า "เอส-คิว-แอล" แต่หลายคนยังคงอ่านว่า "ซีเควล" ตามชื่อเก่า SEQUEL ซึ่งเป็นชื่อเดิมของ SQL เพื่อเป็นการระลึกถึงและให้เกิดรักแก่ภาษาต้นกำเนิด ภาษา SQL เป็นภาษาฐานข้อมูลเชิงสัมพันธ์ (Relational Database Language) มาตรฐาน ANSI ของ IOS (International Organization for Standardization) และเป็นภาษาฐานข้อมูลหลักของระบบจัดการฐานข้อมูลเชิงสัมพันธ์จำนวนมากซึ่งกำลังเป็นที่นิยมใช้กันอยู่ในปัจจุบันเช่น DB2, SQL/DS, ORACLE, INFORMIX และ UNIFY เป็นต้น แม้แต่ dBase IV ซึ่งเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ที่ใช้กันแพร่หลายมากบนเครื่องไมโครคอมพิวเตอร์ในปัจจุบันและมีภาษาฐานข้อมูลของตนเองโดยเฉพาะก็ยังคงต้องมี SQL เพื่อสนองความต้องการของผู้ใช้ที่ต้องการใช้ภาษาฐานข้อมูลมาตรฐานนี้

2.1 ประวัติความเป็นมา

Dr. E.F Codd นักวิจัยของห้องปฏิบัติการวิจัยบริษัท IBM ที่เมือง San Jose ภา
รัฐแคลิฟอร์เนียได้เสนอรูปแบบของฐานข้อมูลเชิงสัมพันธ์ (Relational Data Model)

ขึ้นมาในปี คศ. 1969 [Date (1969)] รูปแบบนี้ก็ได้อีกเป็นที่นิยมใช้กันอย่างแพร่หลายไป
ยังกลุ่มนักวิจัยตามห้องปฏิบัติการวิจัยต่างๆในมหาวิทยาลัย ในโรงงานอุตสาหกรรมและอื่นๆ
ในประเทศสหรัฐอเมริกา หรือที่นั่น ภาษาฐานข้อมูลสำหรับใช้กับรูปแบบข้อมูลเชิงสัมพันธ์ได้
ถูกคิดค้นขึ้นมามากมาย ภาษาที่เป็นที่นิยมใช้กันมากได้แก่ SEQUEL ซึ่งถูกคิดค้นโดย D.D

Chamberlin ร่วมกับคณะบุคคลในห้องปฏิบัติการวิจัยของบริษัท IBM [Chamberlin and
Boyce (1974)] โดยช่วงปี 1974-1975 ได้มีโครงการจัดทำระบบจัดการฐานข้อมูลต้นแบบ

ชื่อ "System R" และใช้ภาษาฐานข้อมูลชื่อ SEQUEL และเปลี่ยนชื่อใหม่เป็น SQL ในเวลา
ต่อมา ระบบจัดการฐานข้อมูลต้นแบบ System R ได้ชื่อว่าเป็นระบบต้นแบบที่ระบบจัดการฐาน
ข้อมูลเชิงสัมพันธ์อื่นๆใช้เป็นแบบในการพัฒนา

โครงการ System R ได้รับผลสำเร็จในขั้นทดสอบเป็นอย่างดีจึงทำให้ IBM
และบริษัทผู้ผลิตระบบจัดการฐานข้อมูลรายใหญ่ของโลกรายอื่นเห็นความสำคัญในการพัฒนา
ผลิตภัณฑ์ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ซึ่งใช้ภาษาฐานข้อมูล SQL (SQL-based) ออกสู่
ท้องตลาดกันอย่างมากมายในเวลาต่อมา เช่น ORACLE (ของ Oracle Corp.) SQL/DS
(สำหรับระบบปฏิบัติการ DOS/VSE และ VM/CMS ของ IBM) DB2 (สำหรับระบบปฏิบัติการ
MVS ของ IBM) DG/SQL (ของ Data General Corp) และ SYBASE (ของ
Sybase Inc.) เป็นต้น

2.2 ลักษณะภาษา

ในช่วงระยะต้นของเทคโนโลยีฐานข้อมูล ภาษาฐานข้อมูลจะประกอบด้วยรูปแบบภาษา
การดำเนินงานอยู่ 2 ลักษณะใหญ่ๆตามการแบ่งของ CODASYL DBTG [DBTG (1970)] คือ

- ภาษากำหนดโครงสร้าง/นิยาม (Data Definition Language - DDL)
เป็นรูปแบบของภาษาฐานข้อมูลที่ใช้ในการกำหนดชื่อ โครงสร้าง และรายละเอียด
ต่างๆ ของฐานข้อมูลและข้อมูลในฐานข้อมูล

- ภาษาคำเนิงาน (Data Manipulation Language - DML)

เป็นรูปแบบของภาษาฐานข้อมูลที่ใช้ในการประมวลผลกับฐานข้อมูลเพื่อให้ได้ผลลัพธ์ตามที่ต้องการ อันประกอบด้วย การดำเนินการเพิ่มข้อมูลเข้า (Insert) ลบข้อมูลออก (Delete) ปรับเปลี่ยนค่า (Update) และสอบถามข้อมูล (Query or Retrieve) จากฐานข้อมูล

ต่อมา ภาษาคำเนิงานในส่วนการสอบถามข้อมูลจากฐานข้อมูลมีชื่อเรียกเป็นพิเศษว่า

"ภาษาสอบถาม" (Query Language) ภาษา SQL มีรูปแบบภาษาเป็นไปตามโครงสร้างดังกล่าวข้างต้น คือ ประกอบด้วย

- ภาษากำหนดโครงสร้าง/นิยาม
- ภาษาคำเนิงานในส่วนการเพิ่มข้อมูลเข้า การลบข้อมูลออก และการปรับเปลี่ยนค่าข้อมูลเก่า
- ภาษาสอบถามข้อมูล

รูปที่ 2.1 ถึงรูปที่ 2.3 เป็นตัวอย่างตารางข้อมูลชื่อ S, P และ SP ของฐานข้อมูลเชิงสัมพันธ์ซึ่งใช้ในหนังสือของ Date [Date (1981)] ซึ่งจะใช้ในการแสดงตัวอย่างการใช้ภาษาฐานข้อมูลในหัวข้อต่างๆ ของบทนี้ โดย

- S แทนรายละเอียดข้อมูลของผู้ส่งของ (Supplier) มี SNUM เป็นคีย์หลัก
- P แทนรายละเอียดข้อมูลของชิ้นส่วนสินค้า (Part) มี PNUM เป็นคีย์หลัก
- SP แทนรายละเอียดข้อมูลของการส่งชิ้นส่วนสินค้าที่ผู้ส่งของส่ง (Shipment) มี SNUM และ PNUM ร่วมกันเป็นคีย์หลัก (Primary Key)

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

รูปที่ 2.1 ตารางข้อมูล S

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

รูปที่ 2.2 ตารางข้อมูล P

SP	<u>SNUM</u>	<u>PNUM</u>	QTY
	S1	P1	300
	S1	P2	200
	S1	P3	400
	S1	P4	200
	S1	P5	100
	S1	P6	100
	S2	P1	300
	S2	P2	400
	S3	P2	200
	S4	P2	200
	S4	P4	300
	S4	P5	400

รูปที่ 2.3 ตารางข้อมูล SP

2.2.1 การกำหนดโครงสร้าง/นิยาม

รูปแบบของภาษา SQL ที่ใช้ในการกำหนดชื่อ โครงสร้างและรายละเอียดต่างๆ ของตารางข้อมูลและรายละเอียดข้อมูลภายในตารางข้อมูล จะขอแสดงด้วยรูปแบบข้อความสั่งของ SQL มาตรฐาน ดังตัวอย่างต่างๆ ดังต่อไปนี้

1. การสร้างตารางข้อมูลหลัก (Base Relation) พร้อมกำหนดรายละเอียดของข้อมูลที่จะถูกเก็บในตาราง

ตัวอย่าง สร้างตารางข้อมูล S

```
CREATE TABLE S
(SNUM CHAR(5) NOT NULL,
SNAME CHAR(20),
STATUS DECIMAL(3),
CITY CHAR(15),
PRIMARY KEY (SNUM))
```

ตัวอย่าง สร้างตารางข้อมูล SP

```
CREATE TABLE SP
(SNUM CHAR(5) NOT NULL,
PNUM CHAR(6) NOT NULL,
QTY DECIMAL(5),
PRIMARY KEY (SNUM,PNUM),
FOREIGN KEY (SNUM) REFERENCE S,
FOREIGN KEY (PNUM) REFERENCE P )
```

2. การสร้างตารางข้อมูลจำลองโดยเลือกข้อมูลบางแถวหรือบางสัณฐานจากตารางข้อมูลที่มีอยู่แล้ว ตารางข้อมูลจำลองนี้เรียกว่า วิว (View)

ตัวอย่าง สร้างวิวโดยเลือกเอาข้อมูลจากสคมภ์ทั้งหมดจากตารางข้อมูล P และเลือกข้อมูลเฉพาะแถวที่ค่า COLOR เป็น 'Red' และตั้งชื่อวิวนี้ว่า "REDPARTS" และสคมภ์ชื่อ "WEIGHT" ที่เลือกมา ให้เปลี่ยนชื่อเป็น "WT"

```
CREATE VIEW REDPARTS (PNUM,PNAME,COLOR,WT,CITY)
AS SELECT PNUM, PNAME, COLOR, WEIGHT, CITY
FROM P
WHERE COLOR = 'Red'
```

ตัวอย่าง สร้างวิวโดยเลือกเอาข้อมูลเฉพาะสคมภ์ PNUM และ WT จากตาราง REDPARTS และเลือกข้อมูลเฉพาะแถวที่มีค่า CITY เป็น 'London' พร้อมทั้งตั้งชื่อวิวนี้ว่า "REDPARTS1"

```
CREATE VIEW REDPARTS1
AS SELECT REDPARTS.PNUM, REDPARTS.WT
FROM REDPARTS
WHERE REDPARTS.CITY = 'London'
```

2.2.2 ภาษาคำสั่งเนื้องาน

รูปแบบของภาษา SQL ที่ใช้ในการประมวลผลเพื่อให้ได้ผลลัพธ์จากข้อมูลในฐานข้อมูลตามที่ต้องการ เช่น การเพิ่มข้อมูล การลบข้อมูล และการเปลี่ยนแปลงค่าข้อมูลภายในฐานข้อมูล จะขอแสดงด้วยรูปแบบข้อความสั่งของ SQL มาตรฐานดังตัวอย่างต่อไปนี้

1. การเพิ่มข้อมูลหนึ่งแถวเข้าในตารางข้อมูล SP

ตัวอย่าง เพิ่มค่าหนึ่งแถวข้อมูลเข้าในตารางข้อมูล SP

```
INSERT INTO SP
VALUES ('S5', 'P1', 1000)
```

2. การปรับเปลี่ยนค่าบางสคมีของตารางข้อมูล S

ตัวอย่าง เปลี่ยนค่าข้อมูลของสคมีชื่อ status ให้เป็นสองเท่าของค่าเดิม เฉพาะแถวข้อมูลที่มีค่า city เป็น 'London'

```
UPDATE S
SET STATUS = 2 * S.STATUS
WHERE CITY = 'London'
```

3. ลบแถวข้อมูลซึ่งมีค่า weight มากกว่า 15 ออกจากตารางข้อมูล P

ตัวอย่าง

```
DELETE
FROM P
WHERE WEIGHT > 15
```

2.2.3 ภาษาสอบถามฐานข้อมูล

ในระบบจัดการฐานข้อมูลเชิงสัมพันธ์ ภาษาสอบถามฐานข้อมูลถูกจัดรูปแบบออกเป็น 2 แบบคือ

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

ภาษาพีรคณิตสัมพันธ์ มีลักษณะเป็นชุดของข้อความสั่งซึ่งประกอบด้วยการปฏิบัติการกับตารางข้อมูลในฐานข้อมูลอย่าง เป็นลำดับขั้นตอนจนกว่าจะได้คำตอบซึ่งเป็นตารางข้อมูลที่ต้องการ ข้อความสั่งปฏิบัติการพีรคณิตสัมพันธ์แบ่งตามลักษณะการดำเนินงานออกเป็น 2 กลุ่มคือ

- ข้อความสั่งปฏิบัติการมาตรฐานที่ใช้กับเซตในทางคณิตศาสตร์

(Set Operations) ประกอบด้วยข้อความสั่ง

- Union

- Intersect

- Minus (Difference) และ

- Product (Cross Product)

- ข้อความสั่งปฏิบัติการพิเศษ (Special Operations) ประกอบด้วย

ข้อความสั่ง

- Select

- Project

- Join และ

- Divide

สำหรับแต่ละข้อความสั่งในกลุ่มที่ต้องใช้ตารางข้อมูล 2 ตารางคือข้อความสั่ง Union, Intersect และ Minus ตารางข้อมูลทั้งสองที่ใช้จะต้องมีคุณสมบัติที่เรียกว่า "Union Compatible" [Date(1989)] ตัวอย่างการใช้ข้อความสั่งในภาษาพีชคณิตสัมพันธ์พร้อมผลลัพธ์ที่ได้ต่อไปนี้จะใช้ตัวอย่างตารางข้อมูล A และ B ในรูปที่ 2.4 และรูปที่ 2.5 ตามลำดับ โดยตารางข้อมูล A และตารางข้อมูล B นี้มีคุณสมบัติเป็น Union Compatible

A

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

รูปที่ 2.4 ตารางข้อมูล A

B

SNUM	SNAME	STATUS	CITY
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S7	Smith	10	London
S8	Blake	30	Paris

รูปที่ 2.5 ตารางข้อมูล B

ตัวอย่าง ข้อความสั่ง

C = A INTERSECT B

ผลลัพธ์ที่ได้ :

C	SNUM	SNAME	STATUS	CITY
	S2	Jones	10	Paris
	S3	Blake	30	Paris
	S4	Clark	20	London

ตัวอย่าง ข้อความสั่ง

D = A UNION B

ผลลัพธ์ที่ได้ :

D	SNUM	SNAME	STATUS	CITY
	S1	Smith	20	London
	S2	Jones	10	Paris
	S3	Blake	30	Paris
	S4	Clark	20	London
	S5	Adams	30	Athens
	S7	Smith	10	London
	S8	Blake	30	Paris

ภาษาแคลคูลัสสัมพันธ์ มีลักษณะเป็นข้อความสั่งที่บอกลักษณะของแถวข้อมูลในตารางคำตอบที่ต้องการ ข้อความสั่งในภาษาแคลคูลัสสัมพันธ์หนึ่งข้อความสามารถเขียนให้อยู่ในรูปข้อความสั่งของภาษาพีชคณิตสัมพันธ์ได้หลายๆ ข้อความสั่งที่สมมูลกัน อาจกล่าวได้ว่าภาษาแคลคูลัสสัมพันธ์จะบอกให้ทราบถึงความต้องการของผู้ใช้ว่า ต้องการข้อมูลที่มีเงื่อนไขหรือลักษณะอย่างไร ส่วนขั้นตอนการดำเนินงานที่กระทำเพื่อให้ได้มาซึ่งคำตอบที่ต้องการนั้น เป็นส่วนของการใช้ภาษาพีชคณิตสัมพันธ์

ภาษาสอบถามข้อมูล SQL จัดว่าเป็นภาษาที่มีลักษณะเป็นภาษาแคลคูลัสสัมพันธ์ แต่ละข้อความสั่งในภาษา SQL สามารถแทนได้ด้วยชุดของข้อความสั่งในภาษาพีชคณิตสัมพันธ์ที่สมมูลกันดังตัวอย่างต่อไปนี้ โดยข้อความสั่งของภาษาพีชคณิตสัมพันธ์ที่ใช้มีชื่อว่า IRA (Instructional Relational Algebra) ถูกออกแบบและใช้เพื่อการศึกษาในมหาวิทยาลัยโมนาช ประเทศออสเตรเลีย [Ken (1983)]

ตัวอย่าง

คำถาม : Get supplier numbers and cities for suppliers
who supply part P2

ข้อความสั่งในภาษา SQL อาจเขียนได้ว่า :

```
SELECT  SNAME, CITY
FROM    SP,S
WHERE   SP.SNUM = S.SNUM
AND     SP.PNUM = 'P2'
```

และชุดของข้อความสั่งในภาษาพีชคณิตสัมพันธ์ที่สมมูลกันคือ :

```
T1 = JOIN S [SNUM] SP
T2 = SELECT T1 [PNUM = 'P2']
T3 = PROJECT T2 [SNAME,CITY]
```

2.3 ภาษาสอบถาม SQL มาตรฐานและภาษาสอบถาม SQL ที่ใช้ในซอฟต์แวร์ ORACLE

ไวยากรณ์ของภาษาสอบถาม SQL มีทั้งที่เป็นมาตรฐานของ ANSI และที่เป็นลักษณะเฉพาะของระบบจัดการฐานข้อมูลระบบหนึ่งทำนองคล้ายคลึงกับภาษาคอมพิวเตอร์เช่น ภาษาปาสคาลซึ่งมีทั้งที่เป็นมาตรฐานของ ANSI และที่เป็นลักษณะเฉพาะ เช่น Microsoft Pascal หรือ Turbo Pascal เป็นต้น ในหัวข้อนี้จะกล่าวถึงข้อความสั่งของ SQL ที่เป็นภาษาสอบถามฐานข้อมูลมาตรฐานของ ANSI และที่ใช้ในระบบจัดการฐานข้อมูล ORACLE และจะเรียกข้อความสั่งนี้ว่า "ข้อความสอบถามฐานข้อมูล" หรือเรียกสั้นๆ ว่า "ข้อความสอบถาม"

ส่วนของข้อความสอบถามฐานข้อมูล SQL ทั้งมาตรฐานและของ ORACLE ประกอบด้วยประโยคต่างๆ ที่เหมือนกัน คือ

1. Select-clause
2. From-clause
3. Where-clause
4. Group-by-clause
5. Having-clause

ข้อความสอบถามฐานข้อมูล SQL ของ ORACLE จะมีอนุประโยคเพิ่มจาก SQL มาตรฐานอีกเล็กน้อย คือ

1. Connect-by-clause
2. Order-by-clause
3. For-update-of-clause

ข้อความสั่งในการสอบถามข้อมูลจากฐานข้อมูลของ SQL คือ ข้อความสั่งที่ขึ้นต้นด้วยคำว่า "SELECT" จึงมักเรียกว่าข้อความสั่ง "SELECT" รูปแบบข้อความสั่ง "SELECT" ของ SQL มาตรฐานของ ANSI และของซอฟต์แวร์ ORACLE มีดังต่อไปนี้ โดยคำที่เขียนด้วยอักษรภาษาอังกฤษตัวใหญ่ทั้งหมดเป็นคำสงวน (Reserved Word)

SQL มาตรฐาน [Date (1989)]

```
SELECT  [ALL|DISTINCT] selection table-expression
FROM    table-reference-comma-list
[ WHERE search-condition ]
[ GROUP BY column-reference-comma-list ]
[ HAVING search-condition ]
```

SQL ของซอฟต์แวร์ ORACLE [Sachs (1987)]

```
SELECT  [ALL|DISTINCT] { [table.]* |expr,expr,... }
FROM    table,table,...
[ WHERE condition ]
[ CONNECT BY condition [START WITH condition]]
[ GROUP BY expr,expr,... [HAVING condition]]
[ ORDER BY {expr|posn} [ASC|DESC],
            {expr|posn} [ASC|DESC],... ]
[ FOR UPDATE OF column,column,... [NOWAIT] ];
```

ต่อไปนี้เป็นตัวอย่างข้อความสั่งของ SQL มาตรฐานจากหนังสือของ Date

[Date (1989)]

- (1). SELECT SNUM
FROM S
WHERE STATUS IS NULL
- (2). SELECT PNUM
FROM P
WHERE WEIGHT NOT IN (12,16,17)

```
(3).  SELECT  SNAME
      FROM    S
      WHERE   EXISTS ( SELECT  *
                      FROM SP
                      WHERE  SP.SNUM = S.SNUM)
```

```
(4).  SELECT  DISTINCT SP.SNUM
      FROM    SP
      WHERE   QTY > ALL
            ( SELECT  QTY
              FROM    SP
              WHERE   SNUM = 'S1')
```

```
(5).  SELECT  SNUM, SNAME
      FROM    S
      GROUP BY SNUM
```

```
(6).  SELECT  PNUM
      FROM    SP
      GROUP BY PNUM
      HAVING  SUM(QTY) > 1000
```


ข้อความสอบตามฐานข้อมูล SQL มาตรฐานและของซอฟต์แวร์ ORACLE มีรูปแบบของภาษาในส่วนที่แตกต่างกัน คือ

1. SQL ของซอฟต์แวร์ ORACLE มีอนุประโยค Connect-by-clause และ Order-by-clause และ For-update-of-clause เพิ่มขึ้นจากภาษา

SQL มาตรฐาน

ตัวอย่าง SELECT *

 FROM S

 ORDER BY CITY DESC

2. SQL ของซอฟต์แวร์ ORACLE กำหนดให้มีการใช้ Set Operation ได้ ได้แก่ Union, Minus และ Intersect ส่วน SQL มาตรฐานนั้นกำหนดให้ใช้เฉพาะ Union เท่านั้น

ตัวอย่าง SELECT * FROM A

 MINUS

 SELECT * FROM B

3. SQL มาตรฐานกำหนดให้มีการใช้ Group Function มาตรฐานอันได้แก่ MAX, MIN, AVG, SUM และ COUNT เท่านั้น ส่วน SQL ของ ORACLE ได้เพิ่มส่วนที่เป็นฟังก์ชันขึ้นมาโดยมีรายละเอียดดังนี้

- Number Functions เช่น

ABS(n), FLOOR(n), POWER(m,n), SQRT(n) ฯลฯ

- Char Functions เช่น

ASCII(char), LOWER(char), SUBSTR(char,m[,n]) ฯลฯ

- Group Functions เช่น

COUNT(), MAX(), MIN(), SUM() และ AVG()

- Conversion Functions เช่น

TO_CHAR(n,[,fmt]), TO_DATE(char,[,fmt]) ฯลฯ

- Date Functions เช่น

NEXT_DAY(d,char), ADD_MONTHS(d,n) ฯลฯ

- Functions อื่นๆ เช่น

GREATEST(expr,expr,..), VSIZE(expr) ฯลฯ

4. SQL มาตรฐานไม่อนุญาตให้มีการใช้ Function ในระดับที่ซ้อนกันเกินกว่า 1 ระดับ แต่ SQL ของ ORACLE อนุญาตให้ใช้ได้

ตัวอย่าง SQL ของ ORACLE :

```
SELECT      AVG( SUM(QTY) )
FROM        SP
GROUP BY   PNUM
```

5. เงื่อนไขการค้นหา (Search Condition) ใน Where_clause ของ SQL มาตรฐานเริ่มต้นด้วยค่าคงที่ได้แต่ของ SQL ของ ORACLE ต้องเริ่มต้นด้วยชื่อสคมีตั้งแต่หนึ่งชื่อเป็นต้นไปเสมอ

ตัวอย่าง SQL มาตรฐาน :

```
SELECT      SNAME
FROM        S
WHERE       'P2' IN (SELECT PNUM
                    FROM SP
                    WHERE SP.SNUM = S.SNUM)
```

ตัวอย่าง SQL ของ ORACLE ที่สมันกัน :

SELECT SNAME

FROM S

WHERE SNUM IN

(SELECT SNUM

FROM SP

WHERE PNUM = 'P2')

2.4 การประมวลผลข้อความสอบถามฐานข้อมูลของ SQL

ในงานวิทยานิพนธ์นี้ ใช้รูปแบบข้อความสอบถามฐานข้อมูลของ SQL มาตรฐานและของซอฟต์แวร์ ORACLE ในการพัฒนาตัวแปลภาษา SQL โดยนำบางรูปแบบของ SQL ของซอฟต์แวร์ ORACLE ที่มีรูปแบบเข้าใจง่ายและเอื้ออำนวยความสะดวกต่อการสอบถามฐานข้อมูลได้ดีกว่าภาษา SQL มาตรฐานมาใช้ร่วมกับภาษา SQL มาตรฐาน และมีการเพิ่มเติมรูปแบบบางส่วนในการดำเนินงานกับเซตของข้อมูลซึ่งยังไม่มีทั้งในรูปแบบของทั้ง SQL มาตรฐาน และ SQL ของ ORACLE

ภาษาสอบถามฐานข้อมูล SQL จัดเป็นภาษาแคลคูลัสสัมพันธ์ที่แต่ละข้อความสั่งมีการอธิบายลักษณะของสมาชิกในตารางข้อมูลคำตอบที่ต้องการ โดยที่แต่ละข้อความสั่งในภาษาแคลคูลัสสัมพันธ์สามารถเขียนให้อยู่ในรูปชุดข้อความสั่งของภาษาพีชคณิตสัมพันธ์ งานวิทยานิพนธ์ที่ดำเนินงานมาก่อนหน้านี้ [ทวีศักดิ์(2534)] ได้พัฒนาข้อความสั่งปฏิบัติการพีชคณิตสัมพันธ์ที่ใช้กันทั่วไป 8 ข้อความสั่ง คือ

- กลุ่มข้อความสั่งสำหรับปฏิบัติการมาตรฐานที่ใช้กับเซต ได้แก่ Union Intersect Minus และ Product
- กลุ่มข้อความสั่งสำหรับปฏิบัติการพิเศษ ได้แก่ Select Project Join และ Divide

ในงานพัฒนาตัวแปลภาษา SQL ของงานวิทยานิพนธ์นี้ ได้นำการดำเนินงานของคำสั่งปฏิบัติการในภาษาพีชคณิตสัมพันธ์ของงานวิทยานิพนธ์ที่ดำเนินงานมาก่อนหน้านี้มาใช้ในการประมวลผลข้อความสั่งของภาษาสอบถามฐานข้อมูล SQL ที่พัฒนา ในหัวข้อนี้เป็นตัวอย่างการสอบถามฐานข้อมูลด้วย SQL และลำดับการเรียกใช้คำสั่งปฏิบัติการพีชคณิตสัมพันธ์ซึ่งแทนด้วยข้อความสั่งในภาษาพีชคณิตสัมพันธ์ที่สมนัยกับที่ผู้วิจัยได้ออกแบบและพัฒนา โดยชุดข้อความสั่งพีชคณิตสัมพันธ์ที่ใช้เป็นรูปแบบของโปรแกรม IRA [Ken (1983)] ก่อนที่จะเสนอตัวอย่างของรูปแบบของภาษาสอบถามฐานข้อมูล SQL ที่สมนัยกับขั้นตอนการดำเนินงานของชุดข้อความสั่งในภาษาพีชคณิตสัมพันธ์ดังแสดงในรูปที่ 2.6 โดยกำหนดชื่อตัวแปรที่ใช้ไว้ดังนี้

T, T1, T2, ...	แทนชื่อตารางข้อมูลชั่วคราวที่สร้างขึ้นใหม่สำหรับ ผลลัพธ์ที่ได้จากการดำเนินงาน
rel, rel1, rel2, ...	แทนชื่อตารางข้อมูลในฐานข้อมูล
att, att1, att2, ...	แทนชื่อของสคีม่าในตารางข้อมูล
cond, cond1, cond2, ...	แทนเงื่อนไขการกระทำ
val, val1, val2, val3, ...	แทนค่าคงที่ประเภทอักขระ หรือเลขจำนวน

ลำดับที่	SQL	ลำดับข้อความสั่งภาษาพีชคณิตสัมพันธ์
1	การแสดงค่าของบางสคมก้อย่างไม่มีเงื่อนไข	
	<pre>select att1,att2,... from rel;</pre>	$T = \text{PROJECT rel}[att1,att2,...]$
2	การค้นหาแถวของตารางข้อมูลตามเงื่อนไขที่กำหนดเงื่อนไขเดียว	
	<pre>select * from rel where att = val;</pre>	$T = \text{SELECT rel [att = val]}$
3	การค้นหาแถวของตารางข้อมูลตามเงื่อนไขมากกว่า 1 เงื่อนไขด้วยการใช้ "and" และ "or"	
3.1	<pre>select * from rel where cond1 and cond2;</pre>	$T1 = \text{SELECT rel [cond1]}$ $T2 = \text{SELECT rel [cond2]}$ $T = T1 \text{ INTERSECT } T2$

ตารางที่ 2.6 ข้อความสั่งของภาษาสอบถามฐานข้อมูล SQL และลำดับของข้อความสั่งของภาษาพีชคณิตสัมพันธ์ที่สมมูลกัน

ลำดับที่	SQL	ลำดับข้อความสั่งภาษาพีชคณิตสัมพันธ์
3.2	<pre>select * from rel where cond1 or cond2;</pre>	<pre>T1 = SELECT rel [cond1] T2 = SELECT rel [cond2] T = T1 UNION T2</pre>
4	การค้นหารายละเอียดจากตารางข้อมูลที่มีความสัมพันธ์กันกับตารางข้อมูลอื่น	
4.1	<p>รูปแบบที่ 1</p> <pre>select * from rel1, rel2 where rel1.att = rel2.att</pre>	<pre>T = JOIN rel1 [att] rel2</pre>
4.2	<p>รูปแบบที่ 2</p> <pre>select * from rel1 where att in (select att from rel2);</pre>	<pre>T1 = JOIN rel1 [att] rel2 T = PROJECT T1 [ทุก attribute ของ rel1]</pre>

ตารางที่ 2.6 ข้อความสั่งของภาษาสอบถามฐานข้อมูล SQL และลำดับของข้อความสั่งของภาษาพีชคณิตสัมพันธ์ที่สมมูลกัน (ต่อ)

ลำดับที่	SQL	ลำดับข้อความสั่งภาษาที่ชคณิตสัมพันธ์
5	การค้นหารายละเอียดจากตารางข้อมูล โดยเลือกตามเขตของค่าที่กำหนดให้	
5.1	รูปแบบที่ 1 (cond แบบที่ 1)	
5.2	<pre>select att1 from rel group by att1 having att contains [val1, val2,...,valn]</pre>	$T1 = [att:val1, val2, \dots, valn]$ $T2 = PROJECT \ rel[att1, att]$ $T = DIVIDE \ T2[(att) \supseteq (att)]T1$
5.2	<pre>select att1 from rel group by att1 having att contains (att from Sub-Query)</pre>	$T1 = [att:values \ from \ sub-query]$ $T2 = PROJECT \ rel[att1, att]$ $T = DIVIDE \ T2[(att) \supseteq (att)]T1$
5.3	รูปแบบที่ 3 (cond แบบที่ 3)	

หมายเหตุ

$T1 = [att : val1, val2, \dots, valn]$ เป็นการกำหนดให้ T1 เป็นตารางข้อมูล
ที่มีเฉพาะค่าของสคีม่า att อันประกอบด้วย val1, val2, ... , valn

ตารางที่ 2.6 ข้อความสั่งของภาษาสอบถามฐานข้อมูล SQL และลำดับของข้อความสั่ง
ของภาษาที่ชคณิตสัมพันธ์ที่สัมพันธ์กัน (ต่อ)

ตัวอย่าง การค้นหารายละเอียดค่าของทุกสมการของทุกแถวที่มีอยู่ในตารางข้อมูล
S โดยไม่มีเงื่อนไขใดๆ

ข้อความสั่งในภาษาแกลกคูล์สสัมพันธ์ :

```
SELECT * FROM S ;
```

ชุดของข้อความสั่งในภาษาพีรคณิศสัมพันธ์ที่สมนัยกัน :

```
DISPLAY S
```

ผลลัพธ์ที่ได้คือ

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

ตัวอย่าง การค้นหารายละเอียดค่าของบางสัณคณ์ของทุกแถวในตารางข้อมูล P โดยไม่มีเงื่อนไขใดๆ

ข้อความสั่งในภาษาแควลคลุ้สสัมพันธ์ :

```
SELECT  PNUM, PNAME
FROM    P ;
```

ชุดของข้อความสั่งในภาษาพีรคณิศสัมพันธ์ที่สมนัยกัน :

```
T1 = PROJECT P [PNUM,PNAME]
DISPLAY T1
```

ผลลัพธ์ที่ได้คือ

T1	PNUM	PNAME
	P1	Nut
	P2	Bolt
	P3	Screw
	P4	Screw
	P5	Cam
	P6	Cog

ตัวอย่าง การค้นหารายละเอียดค่าของบางสคมีก์ของบางแถวในการางข้อมูล SP
ตามเงื่อนไขที่กำหนดให้และแสดงผลลัพธ์เฉพาะค่าที่ไม่ซ้ำกัน

ข้อความสั่งในภาษาแกลคกุลัสสัมพันธ์ :

```
SELECT  DISTINCT QTY
FROM    SP
WHERE   PNUM = "P2" ;
```

ชุดของข้อความสั่งในภาษาพีชคณิตสัมพันธ์ที่สมนัยกัน :

```
T1 = SELECT SP [PNUM = "P2"]
T2 = PROJECT T1 [QTY]
DISPLAY T2
```

ผลลัพธ์ที่ได้คือ

T2	QTY
	200
	400

ตัวอย่าง การค้นหารายละเอียดค่าของทุกสัณฐานของบางแถวในตารางข้อมูล P ตามเงื่อนไขที่กำหนดให้ (1 เงื่อนไข)

ข้อความสั่งในภาษาแกลกคูลส์สัมพันธ์ :

```
SELECT *
FROM P
WHERE CITY = "London";
```

ชคของข้อความสั่งในภาษาที่ชคณิตสัมพันธ์ที่สมนัยกัน :

```
T1 = SELECT P [CITY = "London"]
DISPLAY T1
```

ผลลัพธ์ที่ได้คือ

T1	PNUM	PNAME	COLOR	WEIGHT	CITY
	P1	Nut	Red	12	London
	P4	Screw	Red	14	London
	P6	Cog	Red	19	London

ตัวอย่าง การค้นหารายละเอียดค่าของทุกสมรรถนะของบางแถวในตารางข้อมูล P ตามเงื่อนไขที่กำหนดให้ (มากกว่า 1 เงื่อนไข)

ข้อความสั่งในภาษาแกลกคูลัสสัมพันธ์ :

```
SELECT *
FROM P
WHERE CITY = "London"
AND PNUM > "P2" ;
```

ชุดของข้อความสั่งในภาษาพีรคณิตสัมพันธ์ที่สมมูลกัน :

```
T1 = SELECT P [CITY = "London"]
T2 = SELECT P [PNUM > "P2"]
T3 = T1 INTERSECT T2
DISPLAY T3
```

ผลลัพธ์ที่ได้คือ

T3	PNUM	PNAME	COLOR	WEIGHT	CITY
	P4	Screw	Red	14	London
	P6	Cog	Red	19	London

ตัวอย่าง การค้นหารายละเอียดค่าของทุกสัคมีก์ของบางแถวในตารางข้อมูล P ตามเงื่อนไขที่กำหนดให้ (มากกว่า 1 เงื่อนไข)

ข้อความสั่งในภาษาแกลคคูลัสสัมพันธ์ :

```
SELECT *
FROM P
WHERE CITY = "London"
OR PNUM > "P2" ;
```

ชุดของข้อความสั่งในภาษาพีชคณิตสัมพันธ์ที่สมมูลกัน :

```
T1 = SELECT P [CITY = "London"]
T2 = SELECT P [PNUM > "P2"]
T3 = T1 UNION T2
DISPLAY T3
```

ผลลัพธ์ที่ได้คือ

T3	PNUM	PNAME	COLOR	WEIGHT	CITY
	P1	Nut	Red	12	London
	P3	Screw	Blue	17	Rome
	P4	Screw	Red	14	London
	P5	Cam	Blue	12	Paris
	P6	Cog	Red	19	London

ตัวอย่าง การค้นหารายละเอียดข้อมูลซึ่งต้องใช้สองตารางข้อมูลที่มีความสัมพันธ์กัน โดยเชื่อมโยงตารางทั้งสองด้วยค่าสคีม่าที่ใช้โคโนเมนร่วมกัน

ข้อความสั่งในภาษานเนคคอลลัสสัมพันธ์ :

```
SELECT  SNAME
FROM    S,SP
WHERE   S.SNUM = SP.SNUM
AND    PNUM = "P2" ;
```

ชุดของข้อความสั่งในภาษาที่รคณิตสัมพันธ์ที่สมมูลกัน :

```
T1 = JOIN S [SNUM] SP
T2 = SELECT T1 [PNUM = "P2"]
T3 = PROJECT T2 [SNAME]
DISPLAY T3
```

ผลลัพธ์ที่ได้คือ

T3	SNAME
	Smith
	Jones
	Blake
	Clark

ตัวอย่าง การค้นหารายละเอียดข้อมูลซึ่งต้องใช้สองตารางข้อมูลที่มีความสัมพันธ์กัน โดยใช้ subquery

ข้อความสั่งในภาษาแควลคลัสส์สัมพันธ์ :

```

SELECT  SNAME
FROM    S
WHERE   SNUM IN
        (SELECT  SNUM
         FROM    SP
         WHERE   PNUM = "P2" ) ;

```

ชุดของข้อความสั่งในภาษาพีรคณิตสัมพันธ์ที่สมมูลกัน :

```

T1 = SELECT SP [PNUM = "P2"]
T2 = JOIN T1 [SNUM] S
T3 = PROJECT T2 [SNAME]
DISPLAY T3

```

ผลลัพธ์ที่ได้คือ

T3	SNAME
	Smith
	Jones
	Blake
	Clark

ตัวอย่าง การค้นหารายละเอียดข้อมูลซึ่งต้องใช้สองตารางข้อมูลที่มีความสัมพันธ์กันโดยใช้ subquery และ in-predicate

ข้อความสั่งในภาษาแกลกคูลส์สัมพันธ์ :

```
SELECT SNAME
FROM S
WHERE SNUM IN
      (SELECT SNUM
       FROM SP
       WHERE PNUM IN ["P1", "P2", "P3"]);
```

ชุดของข้อความสั่งในภาษาที่ซคณิตสัมพันธ์ที่สมนัยกัน :

```
T1 = SELECT SP [PNUM = "P1"]
T2 = SELECT SP [PNUM = "P2"]
T3 = T1 UNION T2
T4 = SELECT SP [PNUM = "P3"]
T5 = T3 UNION T4
T6 = JOIN T5 [SNUM] S
T7 = PROJECT T6 [SNAME]
DISPLAY T7
```

ผลพหุคูณ

T7

SNAME
Smith
Jones
Blake
Clark

ตัวอย่าง การค้นหารายละเอียดข้อมูล โดยมีการจัดกลุ่มข้อมูลเพื่อนำไปเปรียบเทียบ
เทียบกับข้อมูลในเซตตามที่กำหนดให้ (รูปแบบที่ 1)

ข้อความสั่งในภาษาแคลคูลัสสัมพันธ์ :

```
SELECT  SNUM
FROM    SP

GROUP  BY  SNUM

HAVING  PNUM CONTAINS ["P2","P1"] ;
```

ชุดของข้อความสั่งในภาษาที่ชนิดสัมพันธ์ที่สมนัยกัน :

```
T1 = SELECT SP [PNUM = "P2"]
T2 = SELECT SP [PNUM = "P1"]
T3 = T1 UNION T2
T4 = PROJECT T3 [PNUM]
T5 = PROJECT SP [SNUM, PNUM]
T6 = DIVIDE T5 [(PNUM) >= (PNUM)] T4
T7 = PROJECT T6 [SNUM]

DISPLAY T7
```

ผลลัพธ์ที่ได้คือ

T7	SNUM
	S1
	S2

ตัวอย่าง การค้นหารายละเอียดข้อมูล โดยมีการจัดกลุ่มข้อมูลเพื่อนำไปเปรียบเทียบเกี่ยวกับข้อมูลในเขตตามที่กำหนดให้ (แบบที่ 2)

ข้อความสั่งในภาษาแควลคลัสส์สัมพันธ์ :

```

SELECT  SNUM
FROM    SP
GROUP   BY  SNUM
HAVING  PNUM CONTAINS (SELECT  PNUM
                        FROM    SP
                        WHERE   SNUM = "S2") ;

```

ชุดของข้อความสั่งในภาษาที่ซคณิตสัมพันธ์ที่สมนัยกัน :

```

T1 = SELECT  SP [SNUM = "S2"]
T2 = PROJECT T1 [PNUM]
T3 = PROJECT SP [SNUM,PNUM]
T4 = DIVIDE  T3 [(PNUM)>=(PNUM)] T2
T5 = PROJECT T4 [SNUM]
DISPLAY  T5

```

ผลลัพธ์ที่ได้คือ

T5	SNUM
	S1
	S2

ตัวอย่าง การค้นหารายละเอียดข้อมูล โดยมีการจัดกลุ่มข้อมูลเพื่อนำไปเปรียบเทียบ
เทียบกับข้อมูลในเรคตามที่กำหนดให้ (แบบที่ 3)

ข้อความสั่งในภาษาแควลคลัสสิกส์สัมพันธ์ :

```
SELECT  SNUM
FROM    SP
GROUP  BY  SNUM
HAVING  PNUM = ["P2", "P1"] ;
```

ชุดของข้อความสั่งในภาษาพีชคณิตสัมพันธ์ที่สมมูลกัน :

```
T1 = SELECT SP [PNUM = "P2"]
T2 = SELECT SP [PNUM = "P1"]
T3 = T1 UNION T2
T4 = PROJECT T3 [PNUM]
T5 = PROJECT SP [SNUM, PNUM]
T6 = DIVIDE T5 [(PNUM) = (PNUM)] T4
T7 = PROJECT T6 [SNUM]
DISPLAY T7
```

ผลลัพธ์ที่ได้คือ

T7	SNUM
	S2

ตัวอย่าง การค้นหารายละเอียดข้อมูล โดยมีการจัดกลุ่มข้อมูลเพื่อนำไปเปรียบเทียบ
เทียบกับข้อมูลในเขตตามที่กำหนดให้ (แบบที่ 4)

ข้อความสั่งในภาษาแควลคลัสส์สัมพันธ์ :

```
SELECT  SNUM
FROM    SP

GROUP  BY  SNUM

HAVING PNUM IN ["P2","P1"] ;
```

ชุดของข้อความสั่งในภาษาพีชคณิตสัมพันธ์ที่สมมูลกัน :

```
T1 = SELECT SP [PNUM = "P2"]
T2 = SELECT SP [PNUM = "P1"]
T3 = T1 UNION T2
T4 = PROJECT T3 [PNUM]
T5 = PROJECT SP [SNUM,PNUM]
T6 = DIVIDE T5 [(PNUM) <= (PNUM)] T4
T7 = PROJECT T6 [SNUM]

DISPLAY T7
```

ผลลัพธ์ที่ได้คือ

T7	SNUM
	S2
	S3

ตัวอย่าง การค้นหารายละเอียดข้อมูล โดยเลือกบางแถวที่ไม่อยู่ในเงื่อนไขที่กำหนดให้

ข้อความสั่งในภาษาแควลคลัสสิกส์สัมพันธ์ :

```
SELECT *
FROM P
WHERE PNUM NOT IN ["P3","P1"] ;
```

ชุดของข้อความสั่งในภาษาพีรคณิตสัมพันธ์ที่สมมูลกัน :

T1 = SELECT P [PNUM = "P1"]

T2 = SELECT P [PNUM = "P3"]

T3 = T1 UNION T2

T4 = P MINUS T3

DISPLAY T4

ผลลัพธ์ที่ได้คือ

T4	PNUM	PNAME	COLOR	WEIGHT	CITY
	P2	Bolt	Green	17	Paris
	P4	Screw	Red	14	London
	P5	Cam	Blue	12	Paris
	P6	Cog	Red	19	London

ตัวอย่าง การค้นหารายละเอียดโดยใช้ QUANTIFIER "SOME"

ข้อความสั่งในภาษาแกลกคูล์สัสมัณฑ์ :

```
SELECT *
FROM S
WHERE SNUM > SOME ["S1","S3"];
```

ชุดของข้อความสั่งในภาษาที่ซกณิศัสมัณฑ์ที่สมนัยกัน :

```
T1 = SELECT S [SNUM > "S1"]
```

```
T2 = SELECT S [SNUM > "S3"]
```

```
T3 = T1 UNION T2
```

```
DISPLAY T3
```

ผลลัพธ์ที่ได้คือ

T3	SNUM	SNAME	STATUS	CITY
	S2	Jones	10	Paris
	S3	Blake	30	Paris
	S4	Clark	20	London
	S5	Adams	30	Athens

ตัวอย่าง การค้นหารายละเอียดโดยใช้ QUANTIFIER "ALL"

ข้อความสั่งในภาษาแกลกอลัสสัมพันธ์ :

```
SELECT *
FROM S
WHERE SNUM > ALL (SELECT SNUM
FROM SP);
```

ชุดของข้อความสั่งในภาษาที่ชนิดสัมพันธ์ที่สัมพันธ์กัน :

```
T1 = PROJECT SP [SNUM]
T2 = JOIN S [S.SNUM > T1.SNUM] T1
T3 = PROJECT T2 [ทุก attribute ใน S]
DISPLAY T3
```

ผลลัพธ์ที่ได้คือ

T3	SNUM	SNAME	STATUS	CITY
	S5	Adams	30	Athens

โดยทั่วไปในภาษาสอบถามฐานข้อมูล SQL มีฟังก์ชันให้เลือกใช้เพื่ออำนวยความสะดวกให้กับผู้ใช้งานยิ่งขึ้น เราเรียกฟังก์ชันเหล่านี้ว่า Built-in-Function ซึ่งมีดังต่อไปนี้

1. MAX() เป็นฟังก์ชันสำหรับหาค่าสูงสุดของข้อมูลในสคีมที่กำหนดให้ในวงเล็บ
2. MIN() เป็นฟังก์ชันสำหรับหาค่าต่ำสุดของข้อมูลในสคีมที่กำหนดให้ในวงเล็บ
3. SUM() เป็นฟังก์ชันสำหรับหาค่าผลรวมของข้อมูลในสคีมที่กำหนดให้ในวงเล็บ
4. COUNT() เป็นฟังก์ชันสำหรับนับจำนวนแถวที่มีอยู่ในตารางข้อมูล
5. AVG() เป็นฟังก์ชันสำหรับหาค่าเฉลี่ยของสคีมที่กำหนดให้ในวงเล็บ
6. VAR() เป็นฟังก์ชันสำหรับหาค่าความแปรปรวน(Variance) ของค่าในสคีมที่กำหนดให้ในวงเล็บ
7. SD() เป็นฟังก์ชันสำหรับหาค่าความเบี่ยงเบนมาตรฐาน (Standard Divlatlon) ของค่าในสคีมที่กำหนดให้ในวงเล็บ

เนื่องจากได้มีการจัดทำโครงงานของนักศึกษาในเรื่องของฟังก์ชันทั้ง 7 ชนิดที่กล่าวมาแล้วข้างต้น โดยจัดทำเป็นโมดูลที่แยกเป็นอิสระทำให้สามารถเรียกใช้เชื่อมโยงกับงานวิจัยนี้ได้ ดังนั้นในงานวิจัยนี้ได้ทำการเรียกโมดูลของฟังก์ชันทั้ง 7 ชนิดมาเชื่อมโยงเพื่อให้เป็นงานวิจัยที่สมบูรณ์มากยิ่งขึ้นดังตัวอย่างที่จะแสดงต่อไปนี้ โดยมีข้อกำหนดของตัวแปรเพื่อใช้ประกอบการยกตัวอย่างดังนี้

INT, INT1, INT2 แทนชื่อตัวแปรสำหรับค่าประเภทเลขจำนวนเต็ม
 REAL แทนชื่อตัวแปรสำหรับค่าประเภทเลขจำนวนจริง

ตัวอย่าง นับจำนวนของแถวทั้งหมดที่มีอยู่ในตารางข้อมูล S

ข้อความสั่งในภาษาแกลคูลัสสัมพันธ์ :

```
SELECT  COUNT(*)  
FROM    S ;
```

ชุดของข้อความสั่งในภาษาที่ชนิดสัมพันธ์ที่สมนัยกัน :

```
INT = COUNT S [ALL]  
DISPLAY INT
```

ผลลัพธ์ที่ได้คือ

```
NO OF RECORDS = 5
```

ตัวอย่าง นับจำนวนของแถวของตารางข้อมูล P ที่มีเงื่อนไขตรงตามที่กำหนดให้

ข้อความสั่งในภาษาแวลกวลล์สัมพันธ์ :

```
SELECT  COUNT(PNUM)
FROM    P
WHERE   COLOR = "Red" ;
```

ชุดของข้อความสั่งในภาษาที่ชกณิตสัมพันธ์ที่สมนัยกัน :

```
T1 = SELECT  P [COLOR = "Red"]
INT = COUNT  T1 [PNUM]
DISPLAY INT
```

ผลลัพธ์ที่ได้คือ

PNUM

COUNT(PNUM)

3

ตัวอย่าง นับจำนวนของแถวของตารางข้อมูล SP ตามเงื่อนไขที่กำหนดให้ โดย
เลือกเฉพาะแถวที่ค่าในสคีม่าที่ต้องการไม่ซ้ำกัน

ข้อความสั่งในภาษาแกลกคูลส์สัมพันธ์ :

```
SELECT  COUNT (DISTINCT PNUM)
FROM    SP
WHERE   SNUM > 'S1' ;
```

ชุดของข้อความสั่งในภาษาพีชคณิตสัมพันธ์ที่สมมูลกัน :

```
T1 = SELECT  SP [SNUM > "S1"]
INT1 = COUNT  T1 [DISTINCT PNUM]
DISPLAY INT1
```

ผลลัพธ์ที่ได้คือ

```
PNUM
COUNT(DISTINCT)
```

ตัวอย่าง หาค่าสูงสุดและค่าต่ำสุดของค่าในสภมภ์ที่กำหนดให้จากตารางข้อมูล SP

ข้อความสั่งในภาษาแกลกอลส์สัมพันธ์ :

```
SELECT MAX(QTY), MIN(QTY)
FROM SP ;
```

ชุดของข้อความสั่งในภาษาพีชคณิตสัมพันธ์ที่สมแนบกัน :

```
INT1 = MAX SP[QTY]
INT2 = MIN SP[QTY]
DISPLAY INT1 INT2
```

ผลลัพธ์ที่ได้คือ

QTY	QTY
(MAX)	(MIN)
<hr/>	
400	100

บทที่ 3

โครงสร้างของระบบ Mrdbms

เนื่องจากงานวิทยานิพนธ์นี้เป็นการพัฒนาโปรแกรมต่อเนื่องจากงานการพัฒนา
ระบบจัดการฐานข้อมูล Mrdbms ของวิทยานิพนธ์ที่มีมาก่อน ดังนั้น ปทานุกรมข้อมูล
โครงสร้างแฟ้มข้อมูล โครงสร้างข้อมูลและสารบบแฟ้มข้อมูลที่ใช้จึงเป็นระบบเดียวกัน โดยถูก
ปรับเปลี่ยนเล็กน้อยเพื่อใช้งานวิทยานิพนธ์นี้

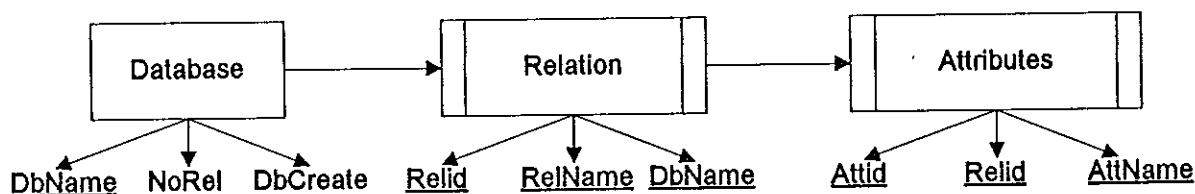
3.1 โครงสร้างของระบบ

ระบบงานเดิมประกอบด้วยองค์ประกอบที่สำคัญพอสรุปเป็น 4 หัวข้อใหญ่ คือ
ปทานุกรมข้อมูล (Data Dictionary) แฟ้มข้อมูล สารบบแฟ้มข้อมูล (File Directory)
โครงสร้างข้อมูล (Data Structure) และแผนภูมิระบบ (System Chart) จะขอกล่าว
สรุปพอสังเขป ผู้สนใจสามารถดูรายละเอียดเพิ่มเติมได้จากงานวิทยานิพนธ์ที่มีมาก่อนหน้านี้
[เพ็ชรพร (2534)]

ปทานุกรมข้อมูล

ปทานุกรมข้อมูล เปรียบเสมือนฐานข้อมูลของระบบจัดการฐานข้อมูลเพราะรายละเอียด
ต่าง ๆ ของข้อมูลที่ระบบจัดการฐานข้อมูลต้องการใช้ในการดำเนินงานต่างๆ จะถูกจัดเก็บใน
ปทานุกรมข้อมูลนี้ ปทานุกรมข้อมูลที่ใช้ในการพัฒนาตัวแปลภาษา SQL นี้ประกอบด้วยข้อมูลและ
ความสัมพันธ์ของข้อมูลแทนได้ด้วยแผนภาพโอ-อาร์ (Object-Relationship Diagram)
[Theerachetmongkol and Montgomery (1980)] ซึ่งสมนัยกับโครงสร้างตารางข้อมูล
ต่างๆ ดังแสดงในรูปที่ 3.1 โดยความสัมพันธ์หลักของข้อมูลในปทานุกรมข้อมูลมีว่า

- แต่ละฐานข้อมูลประกอบด้วยตารางข้อมูลจำนวนมาก ในขณะที่ตารางข้อมูลหนึ่งๆ ต้องอยู่ในฐานข้อมูลเดียวเท่านั้น
- แต่ละตารางข้อมูลประกอบด้วยหลายแอตทริบิวต์ ในขณะที่แต่ละแอตทริบิวต์ต้องอยู่ในตารางข้อมูลเดียวเท่านั้น



ตารางข้อมูลที่สัมพันธ์กัน :

Database(DbName, NoRel, DbCreate)

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

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

รูปที่ 3.1 แผนภาพโออาร์ของปทานุกรมข้อมูลและตารางข้อมูลที่สัมพันธ์กัน

โครงสร้างแฟ้มข้อมูลของปทานุกรมข้อมูล

จากปทานุกรมข้อมูลในรูปที่ 3.1 ในแต่ละสี่เหลี่ยมผืนผ้าหรือหนึ่งตารางข้อมูลจะถูกแทนด้วยหนึ่งแฟ้มข้อมูล โดยรายละเอียดของแต่ละแฟ้มข้อมูลมีดังนี้

Database คือแฟ้มข้อมูลสำหรับเก็บรายละเอียดข้อมูลของฐานข้อมูลทั้งหมดที่ถูกสร้างในระบบขณะเวลาหนึ่งๆ และหนึ่งเรคคอร์ด (Record) แทนข้อมูลของหนึ่งฐานข้อมูลรายละเอียดของโครงสร้างแฟ้มข้อมูลมีดังนี้

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

Relation คือแฟ้มข้อมูลสำหรับเก็บรายละเอียดของทุกตารางข้อมูลของแต่ละฐานข้อมูลที่ปรากฏในแฟ้มข้อมูล Database มีรายละเอียดดังนี้

<u>ชื่อฟิลด์</u>	<u>ชนิด</u>	<u>รูปแบบ</u>	<u>ขนาด</u>	<u>คำอธิบาย</u>
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 คือพื้นที่ข้อมูลสำหรับเก็บรายละเอียดของแอตทริบิวต์ที่มีในแต่ละตารางข้อมูลในฐานข้อมูลหนึ่งๆ มีรายละเอียดดังนี้

<u>ชื่อฟิลด์</u>	<u>ชนิด</u>	<u>รูปแบบ</u>	<u>ขนาด</u>	<u>คำอธิบาย</u>
AttrId	S	I	2	รหัสแอตทริบิวต์
AttName	P	C	15	ชื่อแอตทริบิวต์
RelId	N	I	3	รหัสตารางข้อมูลที่เป็นเจ้าของ
AttFstPos	N	I	3	ตำแหน่งที่เริ่มต้นของข้อมูลในเรคอร์ด
AttLstPos	N	I	3	ตำแหน่งที่สุดท้ายของข้อมูลในเรคอร์ด
AttLngth	N	I	3	ขนาดของข้อมูลในเรคอร์ด
AttDomPos	N	I	1	โดเมนของแอตทริบิวต์
AttStatus	N	C	1	สถานะของแอตทริบิวต์ [P:Primary Key *S:Secondary Key N:Non Key]

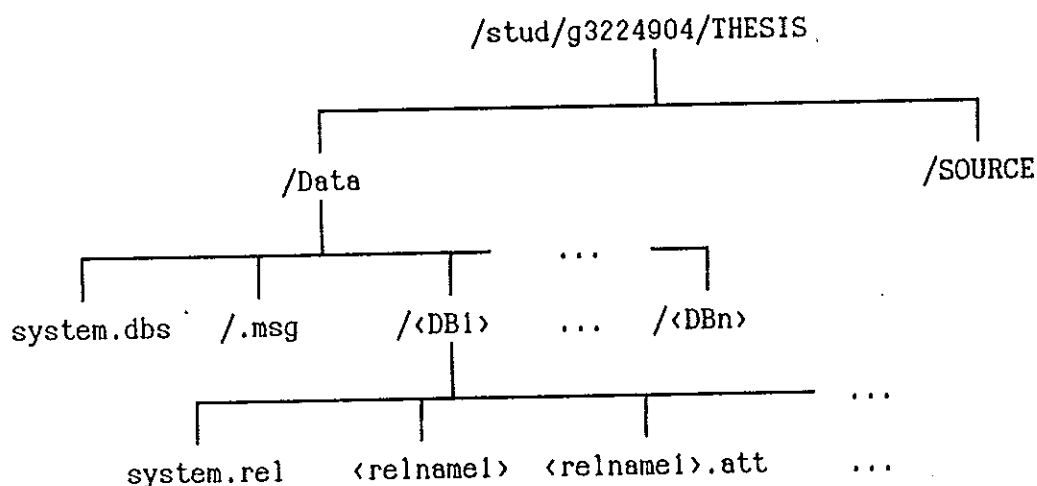
หมายเหตุ

- ชนิด คือข้อมูลบอกชนิดหรือประเภทของแอตทริบิวต์ว่าเป็น Primary Key (P) Secondary Key (S) หรือ Non Key (N)
- รูปแบบ คือประเภทของข้อมูลว่าเป็นตัวอักษร(C) เลขจำนวนเต็ม(I)หรือวันที่(D)

- ขนาด คือจำนวนอักขระหรือตัวเลขที่มากที่สุดที่สามารถแทนข้อมูลชิ้นหนึ่งๆ
- เครื่องหมาย * หมายถึง ยังไม่มีการพัฒนาในส่วนนี้

สารบบแฟ้มข้อมูล

สารบบแฟ้มข้อมูล เป็นที่จัดเก็บแฟ้มข้อมูลที่เกี่ยวข้องทั้งหมดของระบบ ถูกออกแบบโครงสร้างเป็นแผนผังต้นไม้ (Tree Diagram) ในงานวิทยานิพนธ์นี้ได้นำสารบบแฟ้มข้อมูลของงานวิทยานิพนธ์ที่มีมาก่อนหน้านี้มาดำเนินการต่อ โดยเปลี่ยนชื่อสารบบหลักจากสารบบเดิมคือ /stud/g302402 เป็น /stud/g3224904 โดยแสดงรายละเอียดของแต่ละสารบบดังผังภาพในรูปที่ 3.2



รูปที่ 3.2 การจัดเก็บสารบบแฟ้มข้อมูลของ Mrdbms

สารบบ "/stud/g3224904/THESIS/SOURCE" ใช้เก็บแฟ้มข้อมูลที่เป็นโปรแกรมที่พัฒนาทั้งหมด

สารบบ "/stud/g3224904/THESIS/Data" ใช้เก็บข้อมูลที่เกี่ยวข้องกับการทำงานทั้งหมดของระบบโดยมีการจำแนกเป็นสารบบย่อยดังนี้

- แฟ้มข้อมูล "system.dbs" ใช้เก็บรายละเอียดของแต่ละฐานข้อมูลที่ถูกสร้างขึ้นในระบบ

- สารบบ `"/stud/g3224904/THESIS/Data/.msg"` ใช้เก็บแฟ้มข้อมูลเกี่ยวกับคำแนะนำในการใช้คำสั่งปฏิบัติการต่างๆ

- สารบบ `"/stud/g3224904/THESIS/Data/<DBn>"` โดย $n=1,2,\dots$ ใช้เก็บแฟ้มข้อมูลบอกรายละเอียดโครงสร้างของแต่ละฐานข้อมูลพร้อมข้อมูลที่ถูกสร้างขึ้น

โดยแต่ละสารบบประกอบด้วยแฟ้มข้อมูลต่างๆ ของหนึ่งฐานข้อมูล คือ

- แฟ้มข้อมูล `"system.rel"` ใช้เก็บรายละเอียดของทุกตารางข้อมูลในฐานข้อมูลนั้นๆ

- แฟ้มข้อมูล `"<relnameN>"` โดย $N=1,2,\dots$ เป็นชื่อแฟ้มข้อมูลจริงของแต่ละตารางข้อมูลในฐานข้อมูล

- แฟ้มข้อมูล `"<relnameN>.att"` โดย $N=1,2,\dots$ เป็นชื่อแฟ้มข้อมูลที่จัดเก็บรายละเอียดของแอตทริบิวต์ของแต่ละตารางข้อมูลในฐานข้อมูล

โครงสร้างข้อมูล

รายละเอียดของโครงสร้างข้อมูลของปทานุกรมข้อมูลของระบบเป็นแบบรายการโยง (Linked List) และรูปต้นไม้แบบค้นหาทวิภาค (Binary Search Tree) ซึ่งมีรายละเอียดสำคัญแสดงดังรูปที่ 3.3 โดยมีรายละเอียดดังนี้

1. รายการโยงของฐานข้อมูล (Database Linked List)

มีโครงสร้างเป็นแบบหลายตัวโยง (Multilink) โดยแต่ละโหนดของรายการใช้แทนข้อมูลหนึ่งเรคอร์ดจากแฟ้มข้อมูล `system.dbs` ซึ่งแทนรายละเอียดต่างๆ ของหนึ่งฐานข้อมูลในระบบ

2. รายการโยงของรีเลชัน (Relation Linked List)

มีโครงสร้างเป็นแบบหลายตัวโยง โดยแต่ละโหนดของรายการใช้แทนข้อมูลหนึ่งเรคอร์ดจากแฟ้มข้อมูล `system.rel` ซึ่งแทนรายละเอียดต่างๆ ของหนึ่งตารางข้อมูล

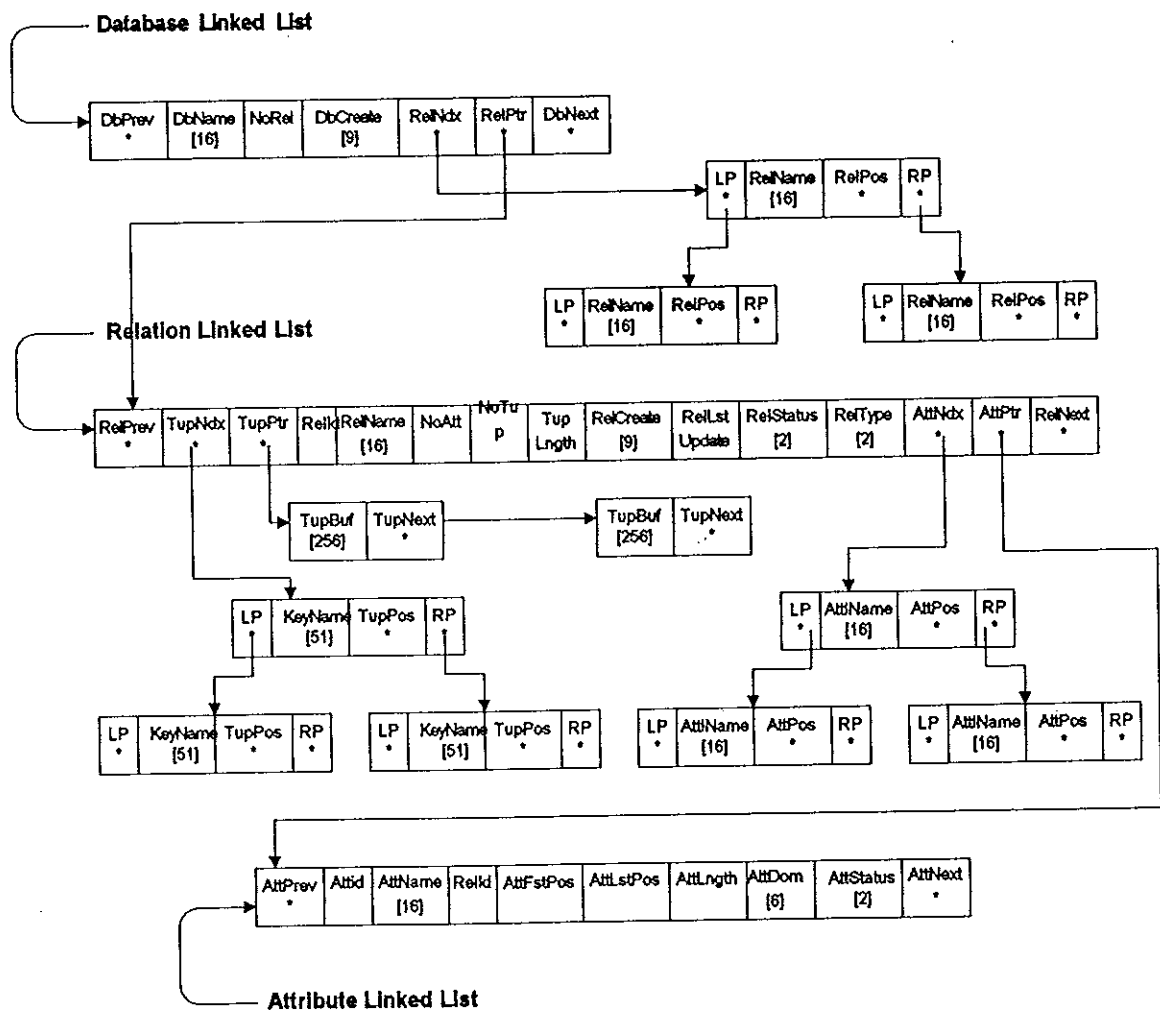
3. รายการโยงของแอตทริบิวต์ (Attribute Linked List)

มีโครงสร้างเป็นแบบรายการโยงคู่ (Doubly Linked List) โดยแต่ละโหนด

ของรายการใช้แทนข้อมูลหนึ่งเรกอร์ดจากแฟ้มข้อมูล <relnameN>.att ซึ่งแทนรายละเอียดต่างๆ ของหนึ่งแอคทริบิว

4. รายการโยงของทูเปิล (Tuple Linked List)

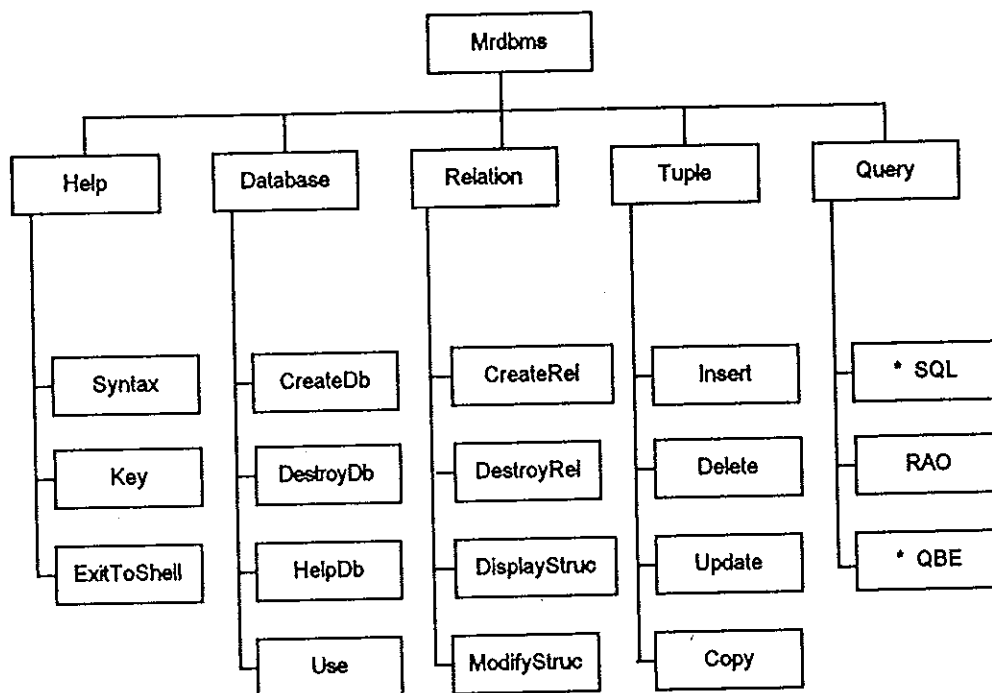
มีโครงสร้างเป็นแบบรายการโยงเดี่ยว (Singly Linked List) โดยที่แต่ละโหนดของรายการใช้แทนข้อมูลหนึ่งเรกอร์ดจากแฟ้มข้อมูล <relnameN> ซึ่งเป็นแฟ้มข้อมูลที่ใช้เก็บข้อมูลจริงๆ ของหนึ่งตารางข้อมูล โดยหนึ่งโหนดของรายการใช้แทนหนึ่งแถวข้อมูล



รูปที่ 3.3 โครงสร้างข้อมูลของปทานุกรมข้อมูลขณะที่อยู่ในหน่วยความจำหลัก

แผนภูมิระบบ

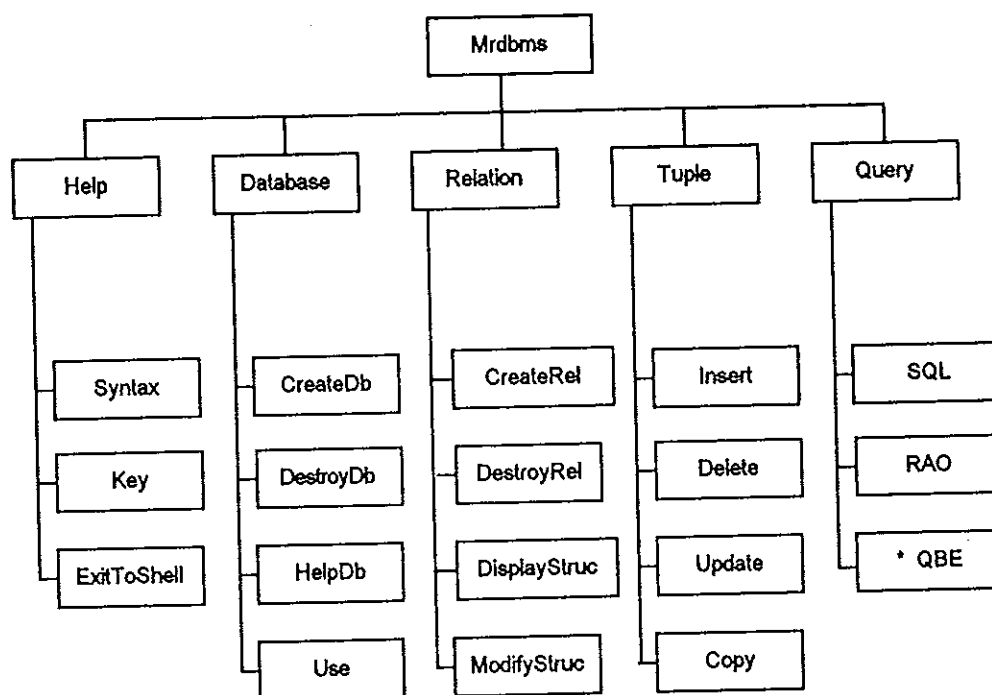
แผนภาพในรูปที่ 3.4 แสดงแผนภูมิระบบของระบบงานของวิทยานิพนธ์ที่มีมาก่อนหน้านี้ ซึ่งเรียกชื่อว่า Mrdbms เพื่อให้เห็นโครงสร้างการดำเนินงานทั้งหมด ส่วนงานวิทยานิพนธ์นี้ที่นำมาเชื่อมโยงกับระบบงานเดิมคือ ส่วนของรายการย่อชื่อ "SQL" ซึ่งอยู่ภายใต้รายการ "Query" โดยในผังภูมิระบบของระบบงานเดิมได้ใส่เครื่องหมาย "*" ไว้ข้างหน้าเพื่อบอกให้ทราบว่าผังยังไม่ได้มีการดำเนินการใดๆ กับส่วนของรายการย่อชื่อนี้



รูปที่ 3.4 แผนภูมิระบบ Mrdbms

3.2 โครงสร้างระบบ Mrdbms ใหม่

ในงานวิทยานิพนธ์นี้ได้เชื่อมโยงกับงานวิทยานิพนธ์ที่มีมาก่อนหน้านี้ในส่วนของรายการย่อยชื่อ "SQL" ซึ่งอยู่ภายใต้รายการ "Query" ในผังภูมิระบบ งานวิทยานิพนธ์นี้เป็นการพัฒนาตัวแปลภาษาสอบถามฐานข้อมูล SQL ซึ่งได้นำบางรูปแบบของ SQL ของซอฟต์แวร์ ORACLE มาใช้ร่วมกับภาษา SQL มาตรฐานและมีการเพิ่มเติมบางรูปแบบในการดำเนินงานกับเซตของข้อมูล ดังแสดงในรูปที่ 3.5



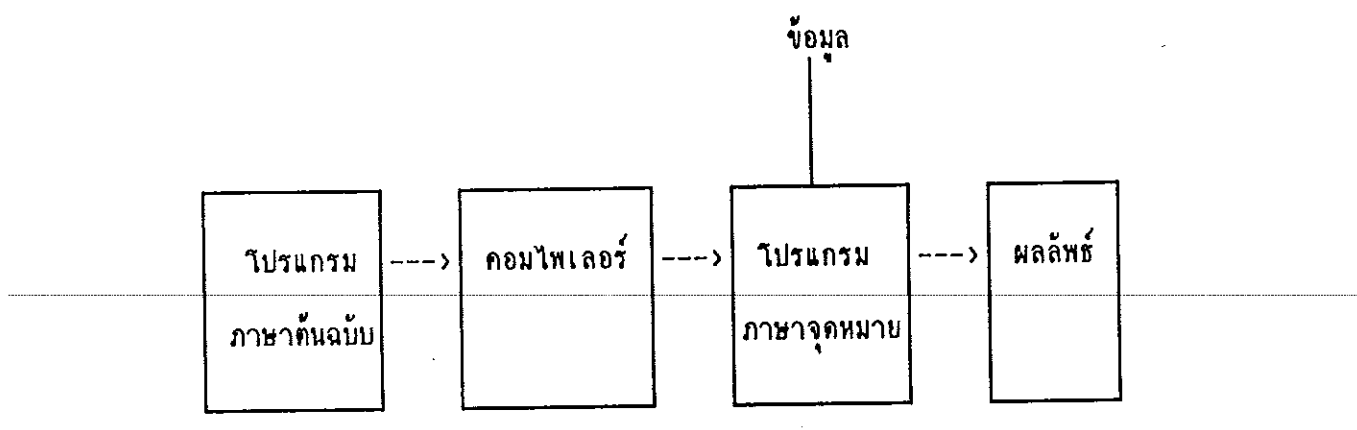
รูปที่ 3.5 แพนภูมิระบบ Mrdbms ใหม่

ตัวแปลภาษา SQL

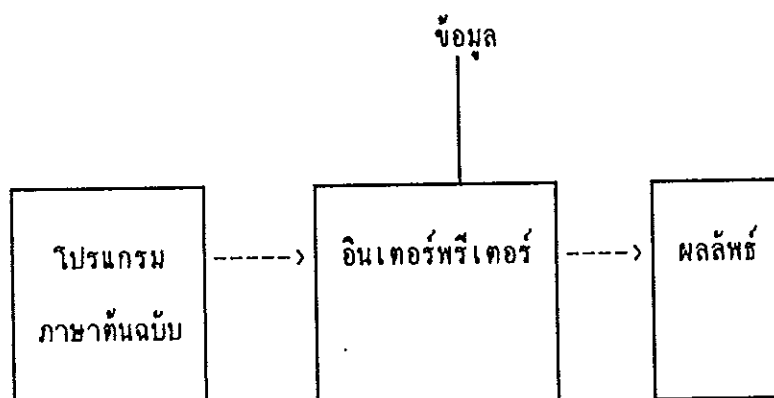
ตัวแปลภาษา (Translator) คือ โปรแกรมซึ่งทำหน้าที่แปลโปรแกรมจากภาษาหนึ่งไปเป็นอีกภาษาหนึ่งและดำเนินการตามความหมายของแต่ละข้อความสั่งของภาษา หรือตีความและดำเนินการตามข้อความสั่งของโปรแกรมที่กำลังถูกแปลโดยตรง ตัวแปลภาษามี 2 แบบ คือ ตัวแปลภาษาระดับต่ำ ได้แก่ แอสเซมบลี และตัวแปลภาษาระดับสูง ได้แก่ คอมไพเลอร์และอินเตอร์พรีเตอร์

4.1 ตัวแปลภาษาระดับสูง

ตัวแปลภาษาระดับสูง คือ ตัวแปลภาษาที่ใช้แปลภาษาคอมพิวเตอร์ระดับสูงไปเป็นภาษาเครื่องเรียกว่า คอมไพเลอร์ (Compiler) คอมไพเลอร์บางตัวแปลภาษาระดับสูงไปเป็นภาษาแอสเซมบลี (Assembly Language) ก่อน แล้วใช้ แอสเซมเบลอร์ (Assembler) แปลภาษาแอสเซมบลีไปเป็นภาษาเครื่องอีกทีหนึ่งแล้วจึงดำเนินการ เนื่องจากมี แอสเซมเบลอร์อยู่แล้ว ทำให้ช่วยลดขั้นตอนการพัฒนาตัวแปลภาษา ตัวแปลภาษาระดับสูงอีกประเภทหนึ่ง เรียกว่า อินเตอร์พรีเตอร์ (Interpreter) ทำหน้าที่ ตีความและดำเนินการตามข้อความสั่งโดยตรงหรือแปลภาษาระดับสูงไปเป็นรหัสระหว่างกลาง (Intermediate - Code) ก่อน แล้วดำเนินการตามความหมายของรหัสนี้ระหว่างกลางนั้นโดยตรง กระบวนการดำเนินการของคอมไพเลอร์และอินเตอร์พรีเตอร์แสดงในรูปที่ 4.1



ก. คอมไพเลอร์



ข. อินเทอร์พรีเตอร์

รูปที่ 4.1 ขั้นตอนการดำเนินงานทั่วไปของคอมไพเลอร์และอินเทอร์พรีเตอร์

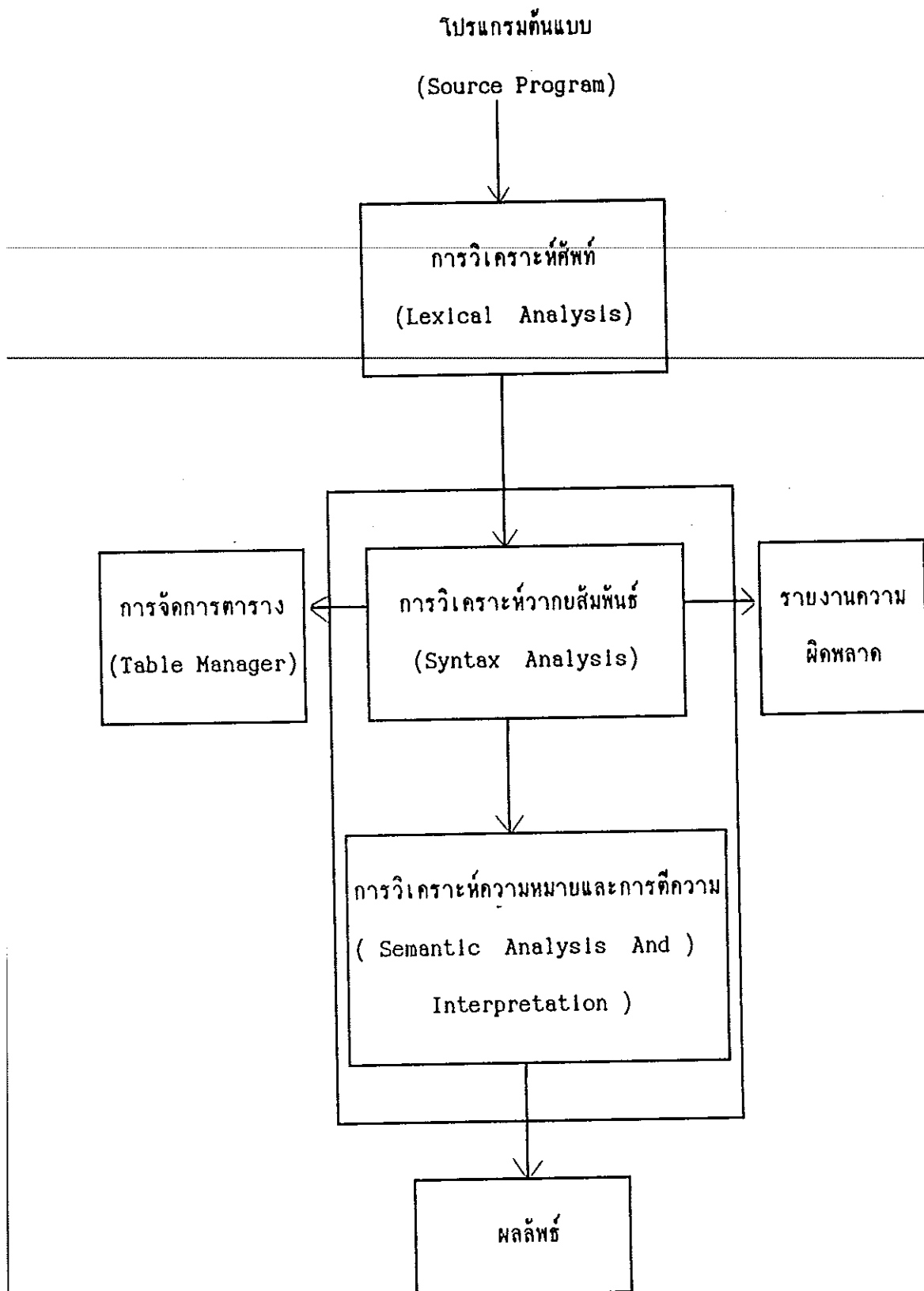
ตัวแปลภาษาประเภท คอมไพเลอร์ เริ่มกระบวนการดำเนินงานตั้งแต่การนำโปรแกรมต้นฉบับ (Source Program) มาผ่านขั้นตอนการแปลจะได้โปรแกรมภาษาจุดหมาย (Object Program) และเมื่อนำข้อมูลมาดำเนินงานกับโปรแกรมภาษาจุดหมายก็จะได้ผลลัพธ์ตามที่ต้องการ ตัวแปลภาษาประเภท คอมไพเลอร์ เหมาะสมกับการทำงานประเภทงานที่มีการพัฒนาโปรแกรมขนาดใหญ่มาก เนื่องจากมีการสร้างโปรแกรมภาษาจุดหมายให้ ทำให้สามารถนำโปรแกรมภาษาจุดหมายมาใช้งานอีกครั้งโดยไม่ต้องเสียเวลาทำการแปลใหม่

คอมไพเลอร์จึงเหมาะสมกับงานประเภทที่โปรแกรมต้นฉบับมีขนาดใหญ่ ถูกใช้บ่อย และไม่มีการเปลี่ยนแปลงในโปรแกรมต้นฉบับบ่อยครั้งนัก

ตัวแปลภาษาประเภท อินเตอร์พรีเตอร์ มีขั้นตอนการดำเนินงานสั้นกว่าขั้นตอนการดำเนินงานของตัวแปลภาษาประเภทคอมไพเลอร์เล็กน้อย คือ ไม่มีการสร้างโปรแกรมภาษาจุดหมายขึ้นมาทุกครั้งที่น่าโปรแกรมต้นฉบับมาผ่านขั้นตอนการแปล เราสามารถใส่ข้อมูลให้ดำเนินงานได้เลย ก็จะได้ผลลัพธ์ออกมา จะเห็นว่าไม่มีการเสียเนื้อที่ในการจัดเก็บโปรแกรมภาษาจุดหมาย และไม่เสียเวลาคำนวณดำเนินงานกับโปรแกรมภาษาจุดหมาย ตัวแปลภาษาอินเตอร์พรีเตอร์นี้เหมาะสมกับงานประเภทที่โปรแกรมต้นฉบับมีขนาดเล็กไม่ใหญ่ ถูกใช้ไม่บ่อย และมีการเปลี่ยนแปลงในโปรแกรมต้นฉบับบ่อย

ในงานวิทยานิพนธ์นี้ ได้เลือกพัฒนาตัวแปลภาษาประเภทอินเตอร์พรีเตอร์ เนื่องจากลักษณะงานที่ทำเป็นงานที่มีการสอบถามข้อมูลในฐานข้อมูลบ่อยครั้ง ในแต่ละครั้งเป็นคำถามที่มีลักษณะแตกต่างกันไปหลายรูปแบบ การสอบถามเป็นคำถามสั้นๆ ไม่ซับซ้อนมากนัก และต้องการคำตอบแบบทันที จึงไม่มีความจำเป็นต้องสร้างโปรแกรมจุดหมายขึ้นมา

ตัวแปลภาษาแบบอินเตอร์พรีเตอร์ที่ถูกพัฒนาในงานวิทยานิพนธ์นี้ มีขั้นตอนต่างๆ ดังแสดงในรูปที่ 4.2 โดยโปรแกรมต้นแบบก็คือ แต่ละข้อความสั่งของภาษาสอบถามฐานข้อมูล SQL นั่นเอง



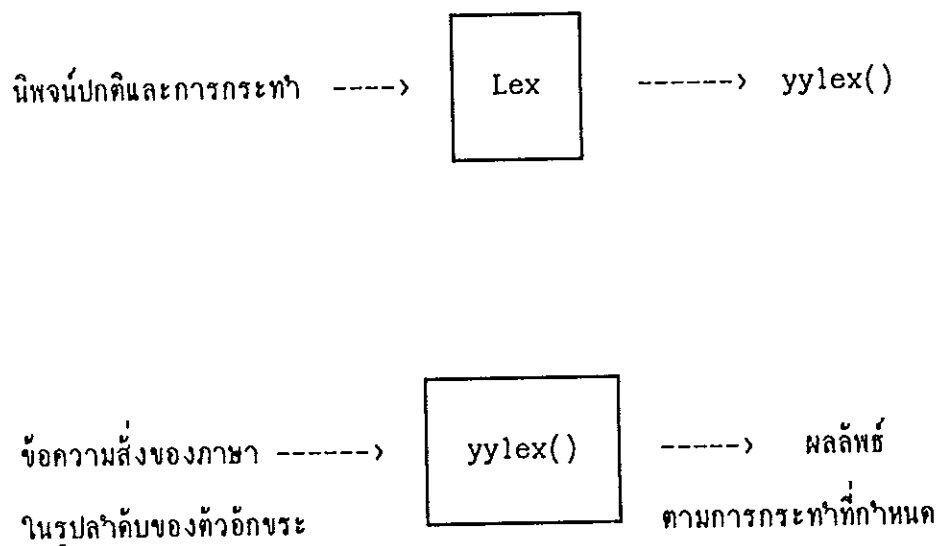
รูปที่ 4.2 ขั้นตอนต่างๆ ของอินเทอร์พรีเตอร์ที่ใช้ในงานวิทยานิพนธ์นี้

4.2 โปรแกรม Lex และ Yacc

ในงานวิทยานิพนธ์นี้ได้้นำโปรแกรมอรรถประโยชน์ Lex และ Yacc มาช่วยในขั้นตอนการดำเนินงานของตัวแปลภาษาที่ถูกพัฒนา โดยนำโปรแกรม Lex มาใช้ในขั้นตอนการวิเคราะห์คำศัพท์และนำโปรแกรม Yacc มาใช้ในขั้นตอนการวิเคราะห์ว่ากยสัมพันธ์ การวิเคราะห์ความหมาย และการตีความ

4.2.1 โปรแกรม Lex

Lex เป็นโปรแกรมผลิตโปรแกรมย่อย (Procedure) ภาษาซีชื่อ yylex() จากข้อมูลเข้าซึ่งเป็นรายละเอียดและรูปแบบไวยากรณ์ของภาษาที่จะถูกแปลในรูปแบบนิพจน์ปกติและการกระทำ โปรแกรมย่อยดังกล่าวสามารถนำไปตรวจจับรูปแบบเหมือน (Pattern Matching) จากข้อมูลในรูปแบบข้อความสิ่งของภาษาที่ถูกอ่านเข้ามาทีละอักขระ และเมื่อตรวจพบรูปแบบจะดำเนินงานตามการกระทำที่กำหนดไว้ ข้อมูลเข้า/ออกของโปรแกรม Lex และโปรแกรมย่อย yylex() แสดงในรูปที่ 4.3



รูปที่ 4.3 ข้อมูลเข้า/ออกของ Lex และ yylex()

ข้อมูลเข้าของโปรแกรม Lex คือรูปแบบของตัวแปรชื่อสงวน (Reserve Name) สัญลักษณ์ต่างๆ ที่ใช้ในภาษาและรูปแบบไวยากรณ์ของภาษาในรูปนิพจน์ปกติและการกระทำตามรูปแบบเฉพาะสำหรับการดำเนินงานของโปรแกรม Lex ซึ่งประกอบด้วย 3 ส่วนใหญ่ๆ แต่ละส่วนถูกคั่นด้วยเครื่องหมาย "%" ดังนี้

Definitions

%%

Rules

%%

User Subroutines

บรรทัดซึ่งมีสัญลักษณ์ "%" ในลำดับแรก (ต่อจากส่วน Definitions) จะต้องมีเสมอ ส่วนบรรทัดอื่นๆ จะมีหรือไม่มีก็ได้ ดังนั้นข้อมูลเข้าที่สั้นที่สุดของ Lex คือ "%%"

ส่วน Definitions เป็นส่วนของการกำหนดอักขระ สัญลักษณ์ และชื่อแฟ้มต่างๆ ที่จะใช้ในการแปลภาษา

ส่วน Rules เป็นส่วนของนิพจน์ปกติและการกระทำเพื่อกำหนดนิยามและกฎเกณฑ์ต่างๆ ตามไวยากรณ์ของภาษา โดยส่วนของนิพจน์ปกติและการกระทำจะต้องมีช่องว่างคั่นอย่างน้อยหนึ่งทีหรืออักขระตั้งระยะ (Tab Character) คั่นอย่างน้อยหนึ่งตัว และส่วนของนิพจน์ปกติต้องเริ่มที่สคัมภ์ที่หนึ่งของบรรทัดเสมอ

ในส่วน Definitions อาจมีข้อความสั่งภาษาซีที่ต้องการให้โปรแกรม Lex นำไปไว้ในบรรทัดก่อนถึงชื่อโปรแกรมย่อย yylex() ที่ถูกสร้างโดยข้อความสั่งเหล่านี้จะถูกเขียนภายในเครื่องหมาย "%{" "%}" ในส่วนการกระทำของส่วน Rules อาจมีข้อความสั่งภาษาซีที่ต้องการให้โปรแกรม Lex นำไปไว้ในโปรแกรมย่อย yylex() ต่อจากข้อความสั่งที่มีอยู่เดิมแล้ว โดยข้อความสั่งเหล่านี้จะถูกเขียนไว้ในเครื่องหมาย "{" และ "}" แต่ถ้ามีข้อความเดียวอาจไม่มีเครื่องหมาย "{" และ "}" ก็ได้

ส่วน User Subroutines เป็นส่วนบรรจุโปรแกรมย่อยต่างๆ ที่เขียนด้วยภาษาซี ที่ผู้พัฒนาตัวแปลภาษาเขียนขึ้นเองเพื่อใช้ในการวิเคราะห์ตัวแปลภาษาที่ถูกพัฒนาออกเหนือจากโปรแกรมย่อย yylex() ทุกข้อความสั่งที่เขียนด้วยภาษาซี จะถูกนำไปเขียนต่อท้ายโปรแกรมย่อย yylex() โดยที่ข้อความสั่งในส่วนนี้จะเริ่มเขียนที่สคอมภ์ใดก็ได้

สัญกรณ์ (Notation) ต่างๆ ที่ใช้ในนิพจน์ปกติมีดังนี้

x	หมายถึง	x
"x"	หมายถึง	x
[xy]	หมายถึง	x หรือ y
[x-z]	หมายถึง	x หรือ y หรือ z
[^x]	หมายถึง	อักขระใดๆ ยกเว้น x
.	หมายถึง	อักขระใดๆ ยกเว้นอักขระขึ้นบรรทัดใหม่
^x	หมายถึง	x อยู่ที่ตำแหน่งแรกของบรรทัด
<y>x	หมายถึง	x ขณะกำลังอยู่ในเงื่อนไข y
x\$	หมายถึง	x อยู่ที่ตำแหน่งสุดท้ายของบรรทัด
x?	หมายถึง	x หรือไม่มี x
x*	หมายถึง	ไม่มี x หรือมี x เรียงติดกันตั้งแต่ 1 ตัวขึ้นไป
x+	หมายถึง	x เรียงติดกันตั้งแต่ 1 ตัวขึ้นไป
x y	หมายถึง	x หรือ y
(x)	หมายถึง	x
x/y	หมายถึง	x และตามด้วย y

Lex จะสร้างตัวแปรและโปรแกรมย่อยให้สามารถถูกนำมาใช้ในส่วนการกระทำของ Rules ตัวอย่าง เช่น

- `yytext` เป็นตัวแปรชนิดแถวลำดับของอักขระ ใช้เก็บรูปแบบที่ตรวจพบ เช่น ถ้าต้องการพิมพ์รูปแบบที่ตรวจพบซึ่งตรงกับนิพจน์ปกติ `[a-z]+` เขียนเป็นนิพจน์ปกติ และการกระทำได้ดังนี้

```
[a-z]+      printf("%s",yytext);
```

- `yylen` เป็นตัวแปรชนิดเลขจำนวนเต็ม ใช้เก็บความยาวของรูปแบบที่ตรวจพบ เช่น ถ้าต้องการนับจำนวนของคำที่ประกอบด้วยอักษรภาษาอังกฤษและจำนวนอักษรของคำทั้งหมด เก็บไว้ในตัวแปรชื่อ `words` และ `chars` ตามลำดับ โดยที่ตัวแปรทั้งสองนี้ผู้ใช้ต้องกำหนดขึ้นเอง สามารถเขียนเป็นนิพจน์ปกติและการกระทำได้ดังนี้

```
[a-zA-Z]+   { words++; chars+=yylen; }
```

- `yylineno` เป็นตัวแปรชนิดเลขจำนวนเต็ม ใช้เก็บหมายเลขบรรทัดที่ตรวจพบรูปแบบเช่น ถ้าต้องการผลลัพธ์ที่ได้จากการใส่หมายเลขบรรทัด ตามด้วยอักขระตั้งระยะและตามด้วยข้อมูลเข้าของบรรทัดนั้นๆ สามารถเขียนเป็นนิพจน์ปกติและการกระทำได้ ดังนี้

```
^.*\n      printf("%d\t%s",yylineno-1,yytext);
```

- `yyomore()` เป็นโปรแกรมย่อยใช้บ่งบอกให้ทราบว่ารูปแบบที่ตรวจพบครั้งต่อไป จะต้องนำมาต่อท้ายรูปแบบในตัวแปร `yytext` เพราะถ้าไม่บอก รูปแบบเก่าที่เก็บใน `yytext` จะถูกแทนที่ด้วยรูปแบบที่ตรวจพบใหม่ทุกครั้ง
- `yyless(n)` เป็นโปรแกรมย่อย ทำหน้าที่เก็บค่าในตัวแปร `yytext` จำนวน `n` ตัว (นับจากตัวแรก) ที่เหลือให้ใส่กลับไปข้อมูลที่เข้าแล้วนำแบบรูปที่ตรวจพบใหม่มาต่อท้าย

โปรแกรม Lex มีวิธีใช้ดังต่อไปนี้

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

โดยค่าที่อยู่ใน "[]" จะมีหรือไม่มีก็ได้

`file` เป็นชื่อแฟ้มข้อมูลที่ได้รับข้อมูลเข้าของ Lex ถ้าไม่มีแสดงว่าแฟ้มข้อมูลเข้าคือ ข้อมูลจากการกดแป้นอักขระจากแป้นพิมพ์

`[tvfn]` เป็นพารามิเตอร์ให้เลือกใช้

- t ระบุการแสดงผลลัพธ์ที่ได้จากโปรแกรม Lex ออกทางจอภาพ ถ้าไม่ระบุผลลัพธ์ที่ได้จะเก็บอยู่ในแฟ้มข้อมูลชื่อ `lex.yy.c` หรือ `lex.yy.r` สำหรับตัวแปลภาษา C มาตรฐาน หรือตัวแปลภาษา Ratfor ตามลำดับ
- v พิมพ์สถิติของค่าบางค่า ถ้าไม่กำหนดจะไม่พิมพ์ให้
- n ยกเลิกค่า -v ไม่ต้องพิมพ์ค่าสถิติ
- f ไม่ต้องการให้ Lex ย่อขนาดของตาราง ปกติถ้าไม่มีค่านี้ Lex จะย่อตารางให้ทำให้ขนาดโปรแกรมเล็กลง

โปรแกรมย่อย `yylex()` ที่ได้จะถูกเก็บอยู่ในแฟ้มข้อมูล `lex.yy.c` และต้องถูกแปลด้วยตัวแปลภาษาซีพร้อมกับเชื่อมโยงกับโปรแกรมย่อยอื่นๆ ที่ Lex จัดไว้ให้ในห้องสมุด (Library) สำหรับระบบปฏิบัติการ Ultrix-32 ผู้ใช้จะต้องใช้พารามิเตอร์ `-ll` ต่อท้ายชื่อแฟ้มข้อมูล `lex.yy.c` เพื่ออ้างถึงโปรแกรมย่อยดังกล่าวดังตัวอย่างเช่น

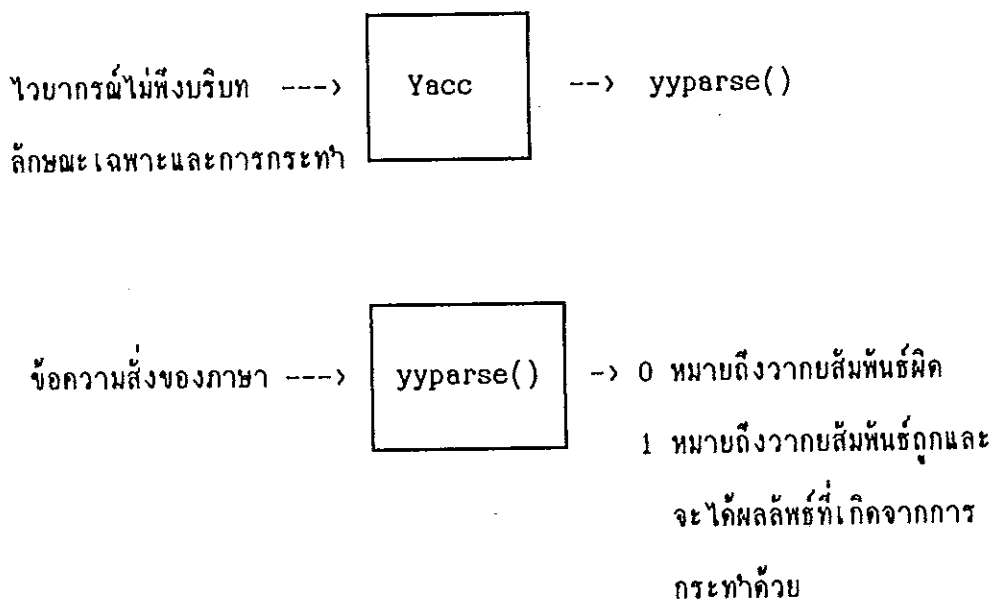
```
>cc lex.yy.c -ll
```

และจะถูกนำไปรวมไว้ในตัวแปลภาษาที่ถูกพัฒนา

4.2.2 โปรแกรม Yacc

Yacc เป็นโปรแกรมผลิตโปรแกรมย่อยภาษาซี ชื่อ yacc() สำหรับสร้างขั้นตอนในการวิเคราะห์วากยสัมพันธ์ วิเคราะห์ความหมาย และตีความ ในการสร้างตัวแปลภาษา SQL ข้อมูลเข้าของโปรแกรม Yacc เป็นไวยากรณ์ไม่หึ่งบริบท

(Context-Free-Grammar) และลักษณะเฉพาะ (Attribute) รวมกันเรียกว่า "ไวยากรณ์ลักษณะเฉพาะ" (Attributed Grammar) ของภาษาสอบถามฐานข้อมูล SQL ส่วนข้อมูลเข้าของโปรแกรมย่อย yacc() ก็คือแต่ละอักขระของข้อความสั่ง SQL ข้อมูลเข้า/ออกของ Yacc และ yacc() แสดงในรูปที่ 4.4



รูปที่ 4.4 ข้อมูลเข้า/ออกของ Yacc และ yacc()

ไวยากรณ์ไม่หึ่งบริบทเขียนเป็นกฎ (rule) ตั้งแต่หนึ่งกฎขึ้นไป โดยแต่ละกฎมีรูปแบบซึ่งใช้กับ Yacc ดังนี้

lhs : rhs ;

โดยที่ lhs เป็นสัญลักษณ์ไม่สิ้นสุด (Nonterminal Symbol)

rhs เป็นสายอักขระ (string) ซึ่งประกอบด้วยสัญลักษณ์ไม่สิ้นสุดหรือสัญลักษณ์สิ้นสุด (Terminal Symbol) หรือเป็นสายอักขระว่าง (Empty String)

":" เป็นสัญลักษณ์ใช้แบ่ง lhs และ rhs

";" เป็นสัญลักษณ์บอกการสิ้นสุดของแต่ละกฎ

และแต่ละกฎจะมีสัญลักษณ์พิเศษเรียกว่า สัญลักษณ์เริ่มต้น (Starting Symbol)

สำหรับบอกการเริ่มต้นของกฎ

ข้อมูลเข้าของโปรแกรม Yacc คือคำสั่งวน รูปแบบของไวยากรณ์ลักษณะเฉพาะของภาษา และการกระทำเฉพาะสำหรับการดำเนินงานกับโปรแกรม Yacc ซึ่งประกอบด้วย 3 ส่วนใหญ่ๆ แต่ละส่วนถูกกั้นด้วยเครื่องหมาย "%" ดังนี้

Declarations

%%

Rules

%%

Programs

บรรทัดซึ่งมีสัญลักษณ์ "%" ในลำดับแรก (ต่อจากส่วน Declarations) จะต้องมีเสมอ ส่วนบรรทัดอื่นๆ จะมีหรือไม่มีก็ได้ การเขียนข้อความในหมายเหตุให้เขียนอยู่ภายในเครื่องหมาย "/*" กับ "*/" เหมือนในภาษาซี

ส่วน Declarations เป็นส่วนของการกำหนดชื่อแฟ้มต่างๆ ที่ใช้ในการแปลภาษา และสัญลักษณ์สิ้นสุด โดยสัญลักษณ์สิ้นสุดสามารถระบุได้โดยคำสั่ง %token ตามด้วยชื่อสัญลักษณ์สิ้นสุดและระหว่างแต่ละคำสั่ง และแต่ละชื่อจะต้องมีช่องว่างหรืออักขระตั้งระยะ คั่นอย่างน้อยหนึ่งตัว

ส่วน Rules ประกอบด้วยกฎต่างๆในรูปไวยากรณ์ไม่พึงบริบท ลักษณะเฉพาะและการกระทำ

ส่วน Programs เป็นส่วนบรรจุโปรแกรมย่อยต่างๆ ที่เขียนด้วยภาษาซีที่ผู้พัฒนาตัวแปลภาษาเขียนขึ้นเอง โปรแกรมย่อยนี้จะถูกนำไปเขียนต่อท้ายโปรแกรมย่อย `yyparse()`

ในส่วน Declarations อาจมีข้อความสั่งภาษาซีที่ต้องการให้โปรแกรม Yacc นำไปไว้ในบรรทัดก่อนถึงชื่อโปรแกรมย่อย `yyparse()` ที่ถูกสร้างโดยข้อความสั่งเหล่านี้จะถูกเขียนภายในเครื่องหมาย `"%[" "%"]"` ในส่วนการกระทำของส่วน Rules อาจมีข้อความสั่งภาษาซีที่ต้องการให้โปรแกรม Yacc นำไปไว้ในโปรแกรมย่อย `yyparse()` ต่อจากข้อความสั่งที่มีอยู่เดิมแล้ว โดยข้อความสั่งเหล่านี้จะถูกเขียนไว้ในเครื่องหมาย `"{" "` และ `"}"` แต่ถ้ามีข้อความเดียวอาจไม่มีเครื่องหมาย `"{" "` และ `"}"` ก็ได้

โปรแกรม Yacc มีวิธีใช้ดังต่อไปนี้

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

โดยค่าที่อยู่ใน `"[" "` จะมีหรือไม่มีก็ได้

- `grammar` เป็นชื่อแฟ้มข้อมูลที่เก็บข้อมูลเข้าของ Yacc
- `v` พิมพ์สถิติของค่าบางค่าไว้ในแฟ้มข้อมูล `y.output` ถ้าไม่กำหนดจะไม่พิมพ์ให้
- `d` เก็บรหัสของสัญลักษณ์สิ้นสุดทั้งหมดไว้ในแฟ้มข้อมูล `y.tab.h`

โปรแกรมย่อย `yyparse()` ที่ได้จะถูกเก็บอยู่ในแฟ้มข้อมูล `y.tab.c` และต้องถูกแปลด้วยตัวแปลภาษาซีพร้อมกับเชื่อมโยงกับโปรแกรมย่อยอื่นๆ ที่ Yacc จัดไว้ให้ในห้องสมุด (Library) สำหรับระบบปฏิบัติการ Ultrix-32 ผู้ใช้จะต้องใช้พารามิเตอร์ `-ly` ต่อท้ายชื่อแฟ้มข้อมูล `y.tab.c` เพื่ออ้างถึงโปรแกรมย่อยดังกล่าวดังตัวอย่างเช่น

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

และจะถูกนำไปรวมไว้ในตัวแปลภาษาที่ถูกพัฒนา

4.3 องค์ประกอบของคิวแปลภาษา SQL

คิวแปลภาษา SQL ประกอบด้วยองค์ประกอบหลักคือ เรขของอักขระที่ใช้ คำสงวน คำแปร คำคงที่ และการปฏิบัติการเชิงสัมพันธ์ โดยมีรายละเอียดดังนี้

เรขของอักขระ

เรขของอักขระ (Character Sets) ใน SQL ประกอบด้วย ตัวอักษร ตัวเลข และ

อักขระพิเศษ ได้แก่

- ตัวอักษร (Letter) ประกอบด้วยตัวอักษรภาษาอังกฤษจำนวน 52 ตัวคือ อักษรภาษาอังกฤษตัวพิมพ์เล็ก (a-z) และอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ (A-Z)
- ตัวเลข (Digit) มีจำนวน 10 ตัว คือ 0-9
- อักขระพิเศษ (Special Characters) มีทั้งหมด 13 ตัว ได้แก่
= < > | . () [] " ; , *

คำสงวน

คำสงวน (Reserved Word) เป็นคำเฉพาะที่กำหนดไว้ในภาษา SQL เพื่อให้เกิดการสื่อความหมายเฉพาะอย่างตัวอักษรในคำสงวนสามารถเขียนได้ทั้งอักษรตัวพิมพ์ใหญ่ หรือ อักษรตัวพิมพ์เล็ก เราไม่สามารถตั้งชื่อตัวแปรให้ซ้ำกับคำสงวนได้ ตัวอย่างของคำสงวนทั้งหมดที่มีอยู่ในภาษา SQL มีดังนี้ คือ

ALL	AND	ASC	AVG	BY	CONTAINS
COUNT	DESC	DISTINCT	FROM	GROUP	HAVING
IN	MAX	MIN	NOT	OR	ORDER
SD	SELECT	SOME	SUM	UNIQUE	VAR
WHERE					

ตัวแปร

ตัวแปร (Variable) ในภาษา SQL มีข้อกำหนดว่าต้องขึ้นต้นด้วยตัวอักษรเสมอ และอาจตามด้วยตัวอักษรหรือตัวเลขก็ได้ โดยชื่อของตัวแปรต้องไม่ตรงกับคำสงวน ตัวแปรสามารถมีความยาวเท่าไรก็ได้ อักษรตัวพิมพ์เล็กและอักษรตัวพิมพ์ใหญ่จะถือว่าไม่เหมือนกัน ตัวอย่างตัวแปรที่ใช้ในภาษา SQL เช่น

snum sname pnum pname status weight เป็นต้น

ค่าคงที่

ค่าคงที่ (Constant) เป็นค่าที่กำหนดขึ้นมาเพื่อให้เข้ากับรูปแบบของตัวแปร ซึ่งแบ่งได้เป็น 2 รูปแบบ คือ

- ค่าคงที่ชนิดจำนวนเต็ม (Integer Constant) เป็นค่าคงที่ที่ประกอบด้วยตัวเลขตั้งแต่ 1 ตัวขึ้นไป ตัวอย่างค่าคงที่จำนวนเต็มที่ใช้ในภาษา SQL เช่น 30, 17
- ค่าคงที่ชนิดอักขระ (Character Constant) จะได้แก่อักขระใดๆ ตั้งแต่หนึ่งตัว ซึ่งจะต้องเป็นอักขระที่อยู่ในเซตอักขระของ SQL

การดำเนินการเชิงสัมพันธ์ (Relational Comparison)

การดำเนินการ หมายถึง การหาผลลัพธ์จากการนำข้อมูลหนึ่งหรือสองค่า มาดำเนินการตามเครื่องหมายหรือสัญลักษณ์ซึ่งแทนตัวกระทำหรือตัวดำเนินการ (operator) ตัวดำเนินการในภาษา SQL แบ่งได้เป็น 3 ประเภทคือ

- การดำเนินการเชิงเปรียบเทียบ เป็นการดำเนินการเปรียบเทียบค่าระหว่างค่าสองค่า ซึ่งเป็นชนิดเดียวกัน และผลลัพธ์ที่ได้ต้องมีค่าชนิดตรรกะซึ่งมีค่าได้ 2 ค่าเท่านั้นคือ "จริง" หรือ "เท็จ" ตัวดำเนินการเปรียบเทียบในภาษา SQL มี 6 ตัวคือ
- | | | |
|------|---------|--------------------|
| ">" | หมายถึง | มากกว่า |
| ">=" | หมายถึง | มากกว่าหรือเท่ากับ |
| "<" | หมายถึง | น้อยกว่า |

"<="	หมายถึง	น้อยกว่าหรือเท่ากับ
"="	หมายถึง	เท่ากับ
"<>", "!="	หมายถึง	ไม่เท่ากับ

- การคำนวณงานเชิงตรรกะ เป็นการคำนวณงานระหว่างค่า 2 ค่าซึ่งเป็นข้อมูลชนิดตรรกะ มีค่าได้ 2 ค่าเท่านั้น คือ "จริง" หรือ "เท็จ" ตัวคำนวณงานตรรกะมี

3 ตัวคือ

- "AND" เป็นตัวคำนวณงานที่ให้ผลลัพธ์เป็นค่าความจริงเป็น "จริง" ก็ต่อเมื่อค่า 2 ค่าที่นำมาคำนวณงานกันมีค่าความจริงเป็น "จริง" ทั้งคู่เท่านั้น นอกนั้นจะให้ผลลัพธ์เป็นค่าความจริง "เท็จ" หมด
- "OR" เป็นตัวคำนวณงานที่ให้ผลลัพธ์เป็นค่าความจริงเป็น "เท็จ" ก็ต่อเมื่อค่า 2 ค่าที่นำมาคำนวณงานกันมีค่าความจริงเป็น "เท็จ" ทั้งคู่เท่านั้น นอกนั้นจะให้ผลลัพธ์เป็น "จริง"
- "NOT" เป็นตัวคำนวณงานกับค่าความจริง 1 ค่า ผลลัพธ์ที่ได้เป็นค่าความจริงที่ตรงกันข้ามกับค่าความจริงของค่าที่ถูกคำนวณงาน
- การคำนวณงานในรูปของฟังก์ชัน เป็นการคำนวณงานที่ให้ผลลัพธ์ได้ทุกรูปแบบ อาจเป็นเซตหรือค่าเดี่ยวๆ ตัวคำนวณงานตรรกะมีดังนี้คือ
 - Built-in Function เป็นฟังก์ชันที่ใช้ในการคำนวณงานกับกลุ่มค่าจากแอตทริบิวของตารางข้อมูลใดๆ ประกอบด้วย ฟังก์ชัน MAX, MIN, COUNT, SUM, AVG, VAR และ SD ซึ่งดำเนินการต่อไปนี้คือ หาค่าสูงสุด หาค่าต่ำสุด นับจำนวนแถวข้อมูล หาค่าผลรวม หาค่าเฉลี่ย หาค่าส่วนเบี่ยงเบนมาตรฐาน และหาค่าความแปรปรวนของข้อมูล ตามลำดับ

- คำว่าเงื่อนไข UNIQUE, DISTINCT คำเนิการขจัดข้อมูลซ้ำซ้อน
- คำว่าเงื่อนไข * เป็นการบอกให้แสดงค่าทุกๆ แอททริบิวในตารางข้อมูล
- คำว่าเงื่อนไข IN, CONTAIN
- คำว่าเงื่อนไข SOME, ALL คำเนิการกับเซตของข้อมูล
- คำว่าเงื่อนไข ASC, DESC คำเนิการจัดเรียงข้อมูลจากน้อยไปมาก และ
มากไปน้อยตามลำดับ

วากยสัมพันธ์(Syntax)

รูปแบบของ SQL ที่ใช้ในงานวิทยานิพนธ์นี้ได้นำบางรูปแบบของ SQL มาตรฐานและของซอฟต์แวร์ ORACLE ที่มีรูปแบบสั้นและเข้าใจง่ายมาใช้และมีการออกแบบเพิ่มเติมรูปแบบบางส่วนในการดำเนินงานกับเซตของข้อมูลซึ่งยังไม่มีในรูปแบบของทั้ง SQL มาตรฐานและ SQL ของ ORACLE รูปแบบของข้อความสอบถาม SQL ที่ใช้แสดงคัวผ้งภาพในรูปที่ 4.5

```

<select_stmt> :
    SELECT      "*"
                ; [UNIQUE|DISTINCT] atname
                ; <atname_list> [ "," <atname_list>]
                ; <function_list> [ "," <function_list>]
    FROM        relname [ "," relname]

    [WHERE      <condition_clause> [OR|AND <condition_clause>] ]

    [GROUP BY  atname [HAVING <condition_clause>] ]
  
```

รูปที่ 4.5 รูปแบบของข้อความสอบถาม SQL


```
[ORDER BY atname [ASC|DESC]
      [ "," atname [ASC|DESC]] ]
```

```
<condition_clause> : <atname_list> <operator> <term_exp>
```

```
<atname_list> : atname [ "." atname ]
               | atname [ "," atname ]
```

```
<function_list> : MAX | MIN | SUM | AVG | COUNT | SD | VAR
                "(" "*" | atname | DISTINCT atname )"
```

```
<operator> : [NOT] GT | LT | GE | LE | EQ | NE | IN | CONTAINS
            [SOME|ALL]
```

```
<term_exp> : <single_term>
            | "[" single_term [ "," single_term ] "]"
            | "(" <select_stmt> )"
```

```
<single_term> : " atname " | constant | atname "." atname
```

รูปที่ 4.5 รูปแบบของข้อความสอบถาม SQL (ต่อ)

ความหมายของสัญลักษณ์ซึ่งใช้ในการเขียนรูปแบบข้อความสอบถาม

- [] หมายถึงข้อความภายในวงเล็บนี้จะเลือกใช้หรือไม่ใช้ก็ได้ เวลาใช้งานไม่ต้องพิมพ์วงเล็บนี้
- < > หมายถึงข้อความภายในเครื่องหมายนี้สามารถแยกเป็นกลุ่มคำย่อยได้อีก
-
- " " หมายถึงข้อความภายในเครื่องหมายนี้จะต้องเป็นสัญลักษณ์ หรือคำเช่นเดียวกับข้อความนั้น
-

คำที่เป็นอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ทั้งคำจะหมายถึงคำสงวน ในข้อความสอบถามจะต้องเขียนให้ตรงกัน

คำที่เป็นอักษรภาษาอังกฤษตัวพิมพ์เล็กทั้งคำจะหมายถึงคำของตัวแปรในข้อความสอบถามจะมีรายละเอียดดังนี้

attname แทนชื่อแอตทริบิวต์ในตารางข้อมูล

relname แทนชื่อตารางข้อมูล

constant แทนเลขจำนวนเต็ม

4.4 โครงสร้างของตัวแปลภาษา SQL

ตัวแปลภาษา SQL ซึ่งถูกพัฒนาในงานวิทยานิพนธ์นี้เป็นตัวแปลภาษาประเภท อินเทอร์พรีเตอร์ ซึ่งมีขั้นตอนในการแปล 3 ขั้นตอนคือ

- ขั้นตอนการวิเคราะห์สัหัท
- ขั้นตอนการวิเคราะห์วากยสัมพันธ์
- ขั้นตอนการวิเคราะห์ความหมาย คีความ และค่าเนิงงาน

ขั้นตอนการวิเคราะห์สัหัท

โดยทั่วไป ขั้นตอนการวิเคราะห์สัหัทจะเริ่มจากการทำงานของโปรแกรมที่เรียกว่า สแกนเนอร์ (Scanner) ทำหน้าที่อ่านข้อความสั่งของภาษาทีละอักขระ รวมกลุ่มของอักขระ ตามรูปแบบที่กำหนดในไวยากรณ์ของภาษา โดยแต่ละกลุ่มของอักขระนี้เรียกว่า โทเคน (Token) และถ้ามีข้อผิดพลาดเกิดขึ้นจะแสดงข้อความบอกข้อผิดพลาดด้วย

ในการดำเนินงานส่วนนี้ ได้ใช้โปรแกรมอรรถประโยชน์ Lex ช่วยสร้างโปรแกรม สแกนเนอร์จากข้อมูลเข้าซึ่งเป็นชุดอักขระ คำสงวนและกฎเกณฑ์การตั้งชื่อตัวแปรในรูปนิพจน์ ปกติและการกระทำ

ข้อมูลเข้าของโปรแกรม Lex ในงานวิทยานิพนธ์นี้ มีดังนี้

(ส่วน Definitions)

```
%[
#include "head.h"
#include "extrn.h"
%]
alpha      [a-zA-Z_]
alphanum   [a-zA-Z0-9_]
cnum       [1-9]
digit      [0-9]
```

digitSign [0-9_-]

%%

(ส่วน Rules)

```

\"          {   strcat(Cmd,yytext);


---


              if (FlagEdl==0) return  QUOTATION;
              else {   if (yytext[yyvaleng-1]=='\n') {


---


                      yytext[yyvaleng-1]='\0';
                      return STRING;   }
                      else yymore(); } }

", "       {   strcat(Cmd,yytext);
              if (FlagEdl==0) return  GT;
              else {   if (yytext[yyvaleng-1]=='\n') {
                      yytext[yyvaleng-1]='\0';
                      return STRING;   }
                      else yymore(); } }

"<"      {   strcat(Cmd,yytext);
              if (FlagEdl==0) return  LT;
              else {   if (yytext[yyvaleng-1]=='\n') {
                      yytext[yyvaleng-1]='\0';
                      return STRING;   }
                      else yymore(); } }

">="     {   strcat(Cmd,yytext);
              if (FlagEdl==0) return  GE;
              else {   if (yytext[yyvaleng-1]=='\n') {
                      yytext[yyvaleng-1]='\0';

```

```

        return STRING; }
    else ymore(); } }

"<="      {  strcat(Cmd,ytext);


---


            if (FlagEdi==0) return LE;
            else {  if (ytext[yyvaleng-1]=='\n') {


---


                    yytext[yyvaleng-1]='\0';
                    return STRING; }
                    else ymore(); } }

"="        {  strcat(Cmd,ytext);
            if (FlagEdi==0) return EQ;
            else {  if (ytext[yyvaleng-1]=='\n') {
                    yytext[yyvaleng-1]='\0';
                    return STRING; }
                    else ymore(); } }

"<>"      {  strcat(Cmd,ytext);
            if (FlagEdi==0) return NE;
            else {  if (ytext[yyvaleng-1]=='\n') {
                    yytext[yyvaleng-1]='\0';
                    return STRING; }
                    else ymore(); } }

"|="       {  strcat(Cmd,ytext);
            if (FlagEdi==0) return NE;
            else {  if (ytext[yyvaleng-1]=='\n') {
                    yytext[yyvaleng-1]='\0';

```

```

        return STRING; }
        else ymore(); } }
    "." { strcat(Cmd,yytext);
        if (FlagEdl==0) return DOT;
        else { if (yytext[yyvaleng-1]=='\n') {
                yytext[yyvaleng-1]='\0';
                return STRING; }
                else ymore(); } }
    "*" { strcat(Cmd,yytext);
        if (FlagEdl==0) return STAR;
        else { if (yytext[yyvaleng-1]=='\n') {
                yytext[yyvaleng-1]='\0';
                return STRING; }
                else ymore(); } }
    "," { strcat(Cmd,yytext);
        if (FlagEdl==0) return COMMA;
        else { if (yytext[yyvaleng-1]=='\n') {
                yytext[yyvaleng-1]='\0';
                return STRING; }
                else ymore(); } }
    ";" { strcat(Cmd,yytext);
        if (FlagEdl==0) return SEMI;
        else { if (yytext[yyvaleng-1]=='\n') {
                yytext[yyvaleng-1]='\0';
                return STRING; }

```

```

else ymore(); } }

{cnum}{digit}*      {  strcat(Cmd,ytext);
                      if (FlagEdi==0) return CONSTANT;
                      else { if (ytext[yyval-1]=='\n') {


---


yytext[yyval-1]='\0';
return STRING; }


---


else ymore(); } }

{digitSign}*       {  strcat(Cmd,ytext);
                      if (FlagEdi==0) return lookup(IDENT);
                      else { if (ytext[yyval-1]=='\n') {
yytext[yyval-1]='\0';
return STRING; }
else ymore(); } }

{alpha}{alphanum}* {  strcat(Cmd,ytext);
                      if (FlagFilename==1) {
FlagFilename = 0;
return FILENAME; }
if (FlagEdi==0) return lookup(IDENT);
else { if (ytext[yyval-1]=='\n') {
yytext[yyval-1]='\0';
return STRING; }
else ymore(); } }

[ \t]*             {  if (ytext[0] != '\0')
                      strcat(Cmd,ytext);
                      if (FlagEdi) {

```

```

        if (yytext[yyvaleng-1]=='\n') {
            yytext[yyvaleng-1]='\0';
            return STRING; }
        else ymore(); } }

```

```

\n          {  strcat(Cmd,yytext);
              if (FlagEd1) {

```

```

                    if (yytext[yyvaleng-1]=='\n')
                        yytext[yyvaleng-1]='\0';
                    return STRING; }
                return yytext[0]; }

```

```

"]"        {  strcat(Cmd,yytext);
              if (FlagEd1==0)
                  return RBCK;
              else if (FlagEd1) {
                  if (yytext[yyvaleng-1]=='\n') {
                      yytext[yyvaleng-1]='\0';
                      return STRING; }
                  else ymore(); } }

```

```

"["        {  strcat(Cmd,yytext);
              if (FlagEd1==0) return LBCK;
              else if (FlagEd1) {
                  if (yytext[yyvaleng-1]=='\n') {
                      yytext[yyvaleng-1]='\0';
                      return STRING; }
                  else ymore(); } }

```



```

"("      {  strcat(Cmd,yytext);
          if (FlagEdl==0) return  LPA;
          else if (FlagEdl) {
              if (yytext[yyvaleng-1]=='\n') {


---


                  yytext[yyvaleng-1]='\0';


---


                  return STRING; }

```

```

          else ymore(); } }

```

```

)"      {  strcat(Cmd,yytext);
          if (FlagEdl==0)
              return  RPA;
          else if (FlagEdl) {
              if (yytext[yyvaleng-1]=='\n') {
                  yytext[yyvaleng-1]='\0';
                  return STRING; }
              else ymore(); } }

```

```

[#]      {  if (FlagEdl)
              return  TERMINATE;
          else return  yytext[0]; }
          if (FlagEdl==0) return  yytext[0];
          else if (FlagEdl) {
              if (yytext[yyvaleng-1]=='\n') {
                  yytext[yyvaleng-1]='\0';
                  return STRING; }
              else ymore(); } }

```

ขั้นตอนการวิเคราะห์วากยสัมพันธ์

ตัววิเคราะห์วากยสัมพันธ์เป็นโปรแกรมที่ทำหน้าที่สองอย่างต่อไปนี้

1. รับรหัสของแต่ละโทกเคนที่ได้จากขั้นตอนการวิเคราะห์ศัพท์
2. ตรวจสอบความถูกต้องของโครงสร้างของแต่ละข้อความสั่งของภาษา

ในการดำเนินงานส่วนนี้ได้ใช้โปรแกรมอรรถประโยชน์ Yacc ช่วยสร้างโปรแกรมพาสเซอร์ (Parser) เพื่อทำหน้าที่ในการวิเคราะห์การกระจายจากข้อมูลเข้าซึ่งในงานวิทยานิพนธ์นี้เป็นรูปแบบไวยากรณ์ไม่หึ่งบริบท และ ลักษณะเฉพาะของภาษาสอบถาม SQL

การวิเคราะห์การกระจายของ Yacc จะเป็นแบบ LALR(1) ซึ่งมีหลักการวิเคราะห์การกระจายแบบล่างขึ้นบน (Bottom-up Parsing) ในที่นี้ขอกล่าวถึงรายละเอียดของไวยากรณ์ไม่หึ่งบริบทและลักษณะเฉพาะที่ใช้กับโปรแกรม Yacc แต่โดยย่อ ดังนี้

ส่วนประกอบของแต่ละข้อความสั่งของไวยากรณ์ไม่หึ่งบริบท ประกอบด้วย 4 ส่วน คือ

- สัญลักษณ์เทอร์มินอล คือ โทกเคนที่ไม่สามารถกระจายต่อไปได้อีก เช่น คำสงวน ตัวแปร เป็นต้น
- สัญลักษณ์นอนเทอร์มินอล คือ กลุ่มของโทกเคนตั้งแต่หนึ่งตัวขึ้นไปมารวมกัน กลุ่มของโทกเคนเหล่านี้ยังสามารถกระจายต่อไปเป็นสัญลักษณ์เทอร์มินอลหรือสัญลักษณ์นอนเทอร์มินอลได้อีก
- สัญลักษณ์เริ่มต้น คือ สัญลักษณ์นอนเทอร์มินอลที่ถูกเลือกขึ้นมาเพื่อใช้เป็นตัวแรกสุดที่จะพิจารณาทำการกระจายต่อไป
- กฎ คือ การกำหนดวิธีที่สัญลักษณ์นอนเทอร์มินอลสามารถเปลี่ยนแปลงไปเป็นสัญลักษณ์เทอร์มินอล หรือสัญลักษณ์นอนเทอร์มินอลอื่นๆ กฎจะประกอบด้วย สัญลักษณ์นอนเทอร์มินอลอยู่ทางด้านซ้ายตามด้วยเครื่องหมาย ":" และตามด้วยสัญลักษณ์นอนเทอร์มินอล และ/หรือสัญลักษณ์เทอร์มินอล ซึ่งอยู่ทางขวา

กฎการเปลี่ยนแปลงในไวยากรณ์ลักษณะเฉพาะมีรูปแบบดังนี้

สัญลักษณ์นอนเทอร์มินอล :

- ' สัญลักษณ์เทอร์มินอลกลุ่มที่ 1 ' สัญลักษณ์นอนเทอร์มินอลกลุ่มที่ 1
- ! ' สัญลักษณ์เทอร์มินอลกลุ่มที่ 2 ' สัญลักษณ์นอนเทอร์มินอลกลุ่มที่ 2
- ! ' สัญลักษณ์เทอร์มินอลกลุ่มที่ 3 ' สัญลักษณ์นอนเทอร์มินอลกลุ่มที่ 3

- ! ' สัญลักษณ์เทอร์มินอลกลุ่มที่ n ' สัญลักษณ์นอนเทอร์มินอลกลุ่มที่ n ;

หมายเหตุ

- ! เป็นสัญลักษณ์ใช้แทนคำว่า "หรือ"
- ' ' เป็นสัญลักษณ์ที่บอกถึงโทกเคนภายในเครื่องหมายเป็นสัญลักษณ์เทอร์มินอล
- : ทำหน้าที่แบ่งแยกสัญลักษณ์ทางซ้ายและขวา สัญลักษณ์ทางซ้ายจะเป็นสัญลักษณ์นอนเทอร์มินอลหนึ่งตัวเท่านั้น สัญลักษณ์ทางขวานั้นประกอบด้วยสัญลักษณ์เทอร์มินอลและ/หรือสัญลักษณ์นอนเทอร์มินอล ส่วนอื่น ๆ นั้นถือว่าเป็นสัญลักษณ์ถูกแทนที่ไม่ได้
- ; เป็นสัญลักษณ์บอกการสิ้นสุดของกฎการเปลี่ยนแปลง

สัญลักษณ์นอนเทอร์มินอลมักจะเขียนด้วยตัวอักษรภาษาอังกฤษตัวเล็ก ส่วนสัญลักษณ์เทอร์มินอลมักจะเขียนด้วยตัวอักษรภาษาอังกฤษตัวใหญ่

ตัวอย่าง

```

program      :   PROGRAM  statements  END ;           (1)
statements   :   statement statements                (2)
              ;
statement    :   STOP  ' ; ' ;                       (3)

```

ตัวอย่างข้างต้นนี้เป็นกฎที่ใช้ในการกระจายไวยากรณ์ลักษณะเฉพาะของภาษาหนึ่ง โดยมี สัญลักษณ์เทอร์มินอล ได้แก่ program statements และ statement และ สัญลักษณ์เทอร์มินอล ได้แก่ PROGRAM END STOP และ ';' โดยมี program เป็น สัญลักษณ์เทอร์มินอลที่ถูกเลือกให้เป็นสัญลักษณ์เริ่มต้นในการใช้กฎการกระจาย

วากยสัมพันธ์ของภาษาสอบถาม SQL ในงานวิทยานิพนธ์นี้ได้เขียนอยู่ในรูปแบบแผนภาพ วากยสัมพันธ์ดังกล่าวมาแล้วในหัวข้อ 4.3 โดยมีรายละเอียดของแผนภาพวากยสัมพันธ์ของ SQL นี้อยู่ในภาคผนวก ก. ต่อไปนี้จะแสดงให้เห็นการกระจายไวยากรณ์ของภาษา SQL ในงานวิทยานิพนธ์นี้ โดยเขียนอยู่ในรูปแบบตามข้อกำหนดของโปรแกรม Yacc . โดยจะแสดง เพียงบางส่วนเท่านั้น ผู้สนใจสามารถดูได้จากโปรแกรมต้นฉบับ

```
/* keyword */
```

```
%token ALL AND APPEND ASC BY CHANGE CONTAINS CLOSE CONSTANT DEL
```

```
%token DOT FROM GROUP DESC DISTINCT HELP
```

```
%token HAVING IDENT INSERT LIST LISTREL LISTDB OPEN OR ORDER PRN
```

```
%token RUN QUIT SAVE SELECT SOME STRING SYNTAX UNIQUE WHERE GETF
```

```
%token MAX MIN SUM COUNT AVG SD VAR
```

```
/* operator token */
```

```
%token EQ GT LT GE LE NE IN NOT RPA LPA LBCK RBCK
```

```
%token SEMI COMMA STAR QUATION TERMINATE RAOCOMM FILENAME
```

```
%%
```

```
line :
```

```
    | line lineCommand;
```

```
lineCommand : query_command
```

```
            | editor_command enter
```



```

select_stmt  : select_clause
              where_clause
              group_clause
              having_clause
              order_clause
              | error;

```

```

select_clause : SELECT  set101 sel_list  set103
              FROM    rel_list  set104
              | error;

```

```

commline     : STRING    set181;

```

```

where_clause : WHERE    set131 condition_clause  set132
              ;;

```

```

condition_clause : operator_cond  set133  set005  set017
                  boolean_cond_list ;

```

```

boolean_cond_list : OR  set006  condition_clause
                  | AND set007  condition_clause
                  ;;

```

```

set_definition : SOME      set174
               | ALL      set175
               ;;

```

```

:
```

```

:
```

```

:
```

ขั้นตอนการวิเคราะห์ความหมายและตีความ

การวิเคราะห์ความหมายและตีความ โดยทั่วไปแล้วจะเป็นขั้นตอนที่กระทำไปพร้อมๆ กับขั้นตอนวิเคราะห์วากยสัมพันธ์ เรียกการทำงานในลักษณะนี้ว่า การทำงานที่วากยสัมพันธ์เป็นตัวกำหนด (Syntax Directed) ขั้นตอนการวิเคราะห์การกระจายที่ใช้โปรแกรม Yacc จะมีการตรวจสอบวากยสัมพันธ์ควบคู่ไปกับการวิเคราะห์ความหมายและตีความ ซึ่งโปรแกรม Yacc กำหนดให้มีการกระทำอื่นๆ เมื่อมีการลดทอนได้โดยการเขียนคำสั่งต่างๆ ให้อยู่ภายในเครื่องหมาย "{" และ "}" โดยเขียนไว้ทางขวามือของเครื่องหมาย ":"

งานวิธานพจน์นี้ได้เขียนการกระทำ (Action) ในโปรแกรม Yacc แทนด้วยรหัสในรูปแบบ setXXX โดยที่ XXX เป็นเลขจำนวนเต็มแทนการกระทำที่ใช้เพื่อดำเนินการค้นหาข้อมูลมีรายละเอียดดังนี้

การกระทำ	การดำเนินงาน
set003	กำหนดว่าเป็นการดำเนินงานเลือกแถวข้อมูลที่มีค่าไม่ซ้ำกัน
set004	กำหนดว่าเป็นการดำเนินงานเลือกทุกๆ แอตทริบิวต์ในตารางข้อมูล
set006	กำหนดว่าเป็นการดำเนินงานเชิงตรรกะโดยใช้ตัวดำเนินงาน "OR"
set007	กำหนดว่าเป็นการดำเนินงานเชิงตรรกะโดยใช้ตัวดำเนินงาน "AND"
set011	กำหนดว่าเป็นการดำเนินงานเชิงตรรกะโดยใช้ตัวดำเนินงาน "NOT"
set015	กำหนดว่ามีการจัดเรียงลำดับข้อมูลตามแอตทริบิวต์ที่กำหนดให้
set032	กำหนดค่าเริ่มต้นให้กับตัวแปรทั้งหมดที่เกี่ยวข้องกับคำสั่งสอบถาม
set033	กรณียังไม่มีการเปิดใช้ฐานข้อมูล จะแสดงข้อความบอกความผิดพลาด
set101	กำหนดสถานะของโทกเคนตัวล่าสุดว่าเป็นตัวแปรชื่อแอตทริบิวต์
set103	กำหนดสถานะของโทกเคนตัวล่าสุดว่าเป็นตัวแปรชื่อตารางข้อมูล

ตารางที่ 4.6 หมายเลขรหัสแทนการดำเนินงานของตัวแปลภาษา SQL

การกระทำ	การดำเนินงาน
set105	กำหนดว่าเป็นการดำเนินการคำสั่งปฏิบัติการ JOIN
set106	สำเนาโทกเคนที่เกี่ยวข้องทั้งหมดในตาราง JOINTAB เพื่อนำไปใช้ในคำสั่งปฏิบัติการ JOIN
set131	กำหนดค่าเริ่มต้นให้กับตัวแปรทั้งหมดที่ใช้ในส่วนเงื่อนไขการกระทำ
set133	การตรวจสอบประเภทและค่าของ operand ที่นำมาเปรียบเทียบว่าเป็นค่าประเภทเดียวกันหรือไม่
set139	กำหนดประเภทของนิพจน์ทางขวามือในเงื่อนไขการกระทำเป็นประเภทอักขระ
set140	กำหนดประเภทของนิพจน์ทางขวามือในเงื่อนไขการกระทำประเภทค่าคงที่
set144	กำหนดสถานะของโทกเคนตัวล่าสุดว่าเป็นตัวแปรชื่อออตทริบิวที่ใช้ในคำสั่งปฏิบัติการ JOIN
set146	กำหนดให้ตัวแปรโอเปอเรเตอร์เชิงเปรียบเทียบมีค่าเท่ากับ ">"
set147	กำหนดให้ตัวแปรโอเปอเรเตอร์เชิงเปรียบเทียบมีค่าเท่ากับ "<"
set148	กำหนดให้ตัวแปรโอเปอเรเตอร์เชิงเปรียบเทียบมีค่าเท่ากับ ">="
set149	กำหนดให้ตัวแปรโอเปอเรเตอร์เชิงเปรียบเทียบมีค่าเท่ากับ "<="
set150	กำหนดให้ตัวแปรโอเปอเรเตอร์เชิงเปรียบเทียบมีค่าเท่ากับ "="
set151	กำหนดให้ตัวแปรโอเปอเรเตอร์เชิงเปรียบเทียบมีค่าเท่ากับ "!="
set152	กำหนดให้ตัวแปรโอเปอเรเตอร์เชิงเปรียบเทียบมีค่าเท่ากับ "IN"
set153	กำหนดให้ตัวแปรโอเปอเรเตอร์เชิงเปรียบเทียบมีค่าเท่ากับ "CONTAINS"
set156	กำหนดสถานะของโทกเคนตัวล่าสุดว่าเป็นตัวแปรชื่อออตทริบิวที่ใช้ในการดำเนินงานจัดกลุ่มย่อยของแถวของตารางข้อมูล

ตารางที่ 4.6 หมายเลขรหัสแทนการดำเนินงานของตัวแปลภาษา SQL (ต่อ)

การกระทำ	การดำเนินงาน
set161	กำหนดสถานะของโทกเคนตัวล่าสุดว่าเป็นตัวแปรชื่อแอตทริบิวต์ที่ใช้ในการดำเนินงานจัดเรียงลำดับแถวข้อมูล
set163	กำหนดการจัดเรียงลำดับแถวข้อมูลตามแอตทริบิวต์ที่กำหนดแบบ Ascending
set164	กำหนดการจัดเรียงลำดับแถวข้อมูลตามแอตทริบิวต์ที่กำหนดแบบ Descending
set171	กำหนดค่าเริ่มต้นให้กับตัวแปรทั้งหมดที่ใช้ในส่วนเงื่อนไขการกระทำใน Having Clause
set174	กำหนดว่าเป็นการดำเนินงานกับเซตของข้อมูล โดยใช้ตัวดำเนินงาน SOME
set175	กำหนดว่าเป็นการดำเนินงานกับเซตของข้อมูล โดยใช้ตัวดำเนินงาน ALL
set182	กำหนดประเภทการดำเนินงานในรูปของฟังก์ชัน COUNT ดำเนินการนับจำนวนแถวข้อมูลที่มีค่าแอตทริบิวต์ตามที่กำหนด
set183	กำหนดประเภทการดำเนินงานในรูปของฟังก์ชัน COUNT ดำเนินการนับจำนวนแถวข้อมูลที่ไม่ซ้ำกันของค่าแอตทริบิวต์ตามที่กำหนด
set184	กำหนดประเภทการดำเนินงานในรูปของฟังก์ชัน COUNT ดำเนินการนับจำนวนแถวทั้งหมดของตารางข้อมูล
set185	กำหนดให้สถิติที่ต้องการคำนวณเท่ากับ MAX
set186	กำหนดให้สถิติที่ต้องการคำนวณเท่ากับ MIN
set187	กำหนดให้สถิติที่ต้องการคำนวณเท่ากับ SUM
set188	กำหนดให้สถิติที่ต้องการคำนวณเท่ากับ AVG
set189	กำหนดให้สถิติที่ต้องการคำนวณเท่ากับ COUNT
set190	กำหนดให้สถิติที่ต้องการคำนวณเท่ากับ SD
set191	กำหนดให้สถิติที่ต้องการคำนวณเท่ากับ VAR

ตารางที่ 4.6 หมายเลขรหัสแทนการดำเนินงานของตัวแปลภาษา SQL (ต่อ)

4.5 การดำเนินงานของระบบ

จากโครงสร้างของระบบ Mrdbms ในรูปที่ 3.5 การดำเนินงานของตัวแปลภาษา SQL ที่ถูกพัฒนาในงานวิทยานิพนธ์จะเริ่มต้นเมื่อผู้ใช้เลือกใช้เมนูย่อย "SQL" ในเมนูหลัก "Query" โดยก่อนเรียกใช้เมนูย่อย "SQL" ผู้ใช้จะต้องผ่านการเรียกใช้ระบบ Mrdbms และต้องเปิดฐานข้อมูลที่ต้องการใช้เสมอ

รูปที่ 4.7 ถึงรูปที่ 4.10 เป็นแผนภาพกระแสข้อมูล (Data Flow Diagram) แสดงขั้นตอนการดำเนินงานของตัวแปลภาษา SQL อย่างสังเขป โดยเริ่มต้นที่กระบวนการหมายเลข 7 ของรูปที่ 4.7 ซึ่งเป็นกระบวนการดำเนินงานเมื่อผู้ใช้เลือกใช้เมนูหลัก "Query" รายละเอียดการดำเนินงานอย่างสังเขปของแต่ละกระบวนการในแต่ละผังภาพกระแสข้อมูล มีดังนี้

จากรูปที่ 4.7

กระบวนการที่	1
ชื่อกระบวนการ	Mrdbms
คำอธิบาย	เป็นการเข้าไปดำเนินงานในระบบ Mrdbms มีการรับข้อมูลจากผู้ใช้โดยให้เลือกการแสดงผลการทำงานกับจอภาพได้ 2 ชนิดคือ จอภาพ VT ซึ่งเป็นเทอร์มินอลของเครื่อง VAX 11/785 และจอภาพของเครื่องไมโครคอมพิวเตอร์ที่ถูกจำลองเป็นเทอร์มินอลเช่นเดียวกัน
กระบวนการที่	2
ชื่อกระบวนการ	System_Menu
คำอธิบาย	เป็นการดำเนินงานในส่วนของการแสดงและควบคุมการเลือกรายการหลักของระบบ Mrdbms ซึ่งมีลักษณะแบบ Pull down Menu
กระบวนการที่	3-6
ชื่อกระบวนการ	Help Database Relation Tuple

คำอธิบาย เป็นการดำเนินงานในส่วนต่างๆ ของระบบ Mrdbms ซึ่งผู้สนใจสามารถดูรายละเอียดเพิ่มเติมได้จากงานวิทยานิพนธ์ที่มีมาก่อนหน้านี้ [เพ็ชรพร (2534)]

กระบวนการที่ 7

ชื่อกระบวนการ Query

คำอธิบาย เป็นส่วนการดำเนินงานให้ผู้ใช้เลือกใช้ภาษาสอบถามฐานข้อมูลซึ่งแบ่งออกได้เป็น 3 ภาษา คือ

- ภาษา RAO (Relational Algebra Operation)
- ภาษา SQL (Structured Query Language)
- ภาษา QBE (Query By Example)

กระบวนการที่ 7.1

ชื่อกระบวนการ SQL

คำอธิบาย เป็นส่วนการดำเนินงานสอบถามข้อมูล SQL

จากรูปที่ 4.8

กระบวนการที่ 7.1.1

ชื่อกระบวนการ Build_Data_Dictionary

คำอธิบาย เป็นการสร้างปทานุกรมข้อมูลสำหรับการปฏิบัติการของส่วนภาษาสอบถาม SQL

กระบวนการที่ 7.1.2

ชื่อกระบวนการ Query_Process

คำอธิบาย เป็นการเตรียมการดำเนินงานเริ่มต้นของตัวแปลภาษา โดยให้ผู้ใช้ป้อนข้อความสอบถาม SQL ทางแป้นพิมพ์

กระบวนการที่ 7.1.3

ชื่อกระบวนการ Select_Statement

คำอธิบาย เป็นกระบวนการดำเนินงานในส่วนของการประมวลผลข้อความสอบถาม SQL

กระบวนการที่ 7.1.4

ชื่อกระบวนการ Additional_Command

คำอธิบาย เป็นกระบวนการดำเนินงานในส่วนของการประมวลผลเพื่อเปลี่ยนแปลงแก้ไขข้อความในคอมมานด์บัฟเฟอร์ (command buffer) ซึ่งเป็นส่วนของหน่วยความจำหลักที่ใช้เก็บข้อความสอบถาม SQL ล่าสุดที่ถูกดำเนินงาน เพื่อนำไปใช้ใหม่

จากรูปที่ 4.9

กระบวนการที่ 7.1.3.1

ชื่อกระบวนการ Load_Tuple

คำอธิบาย เป็นการดำเนินงานบรรจุข้อมูลจากตารางข้อมูลที่กำลังใช้จากความจำสำรองลงบนหน่วยความจำหลัก

กระบวนการที่ 7.1.3.2

ชื่อกระบวนการ Select

คำอธิบาย เป็นการเรียกใช้การดำเนินงานของคำสั่งปฏิบัติการ Select ของ RAO

กระบวนการที่ 7.1.3.3

ชื่อกระบวนการ Join

คำอธิบาย เป็นการเรียกใช้การดำเนินงานของคำสั่งปฏิบัติการ Join ของ RAO

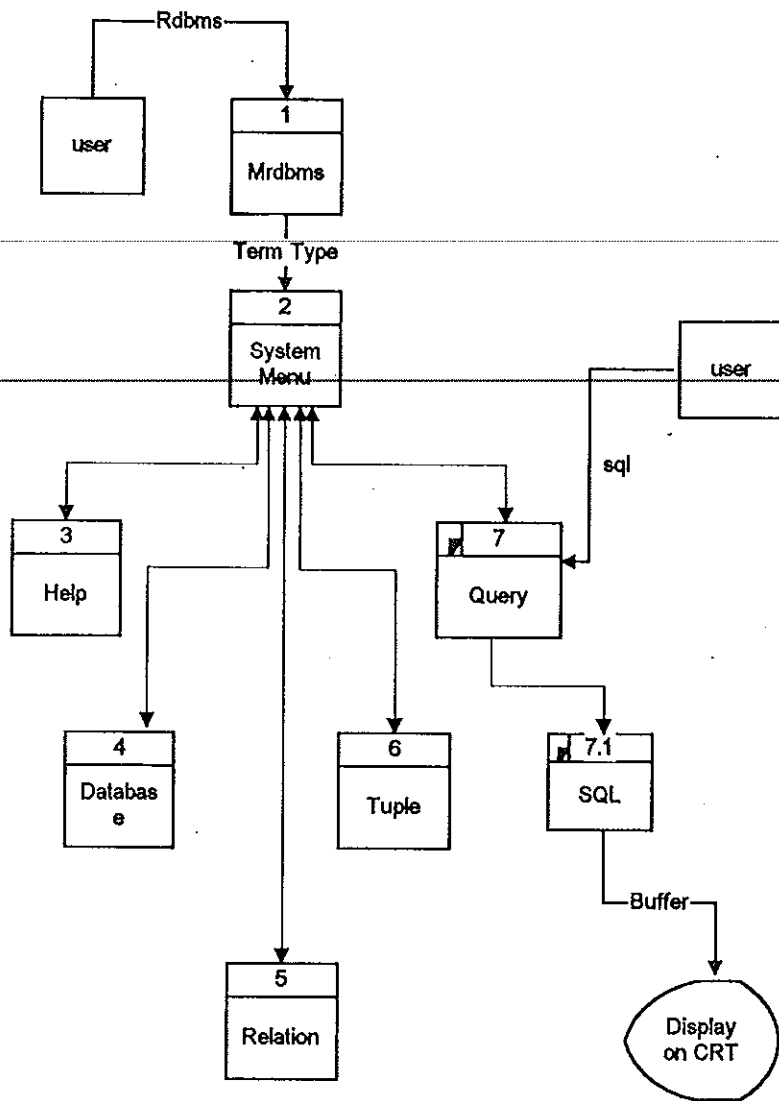
กระบวนการที่ 7.1.3.4

ชื่อกระบวนการ Minus

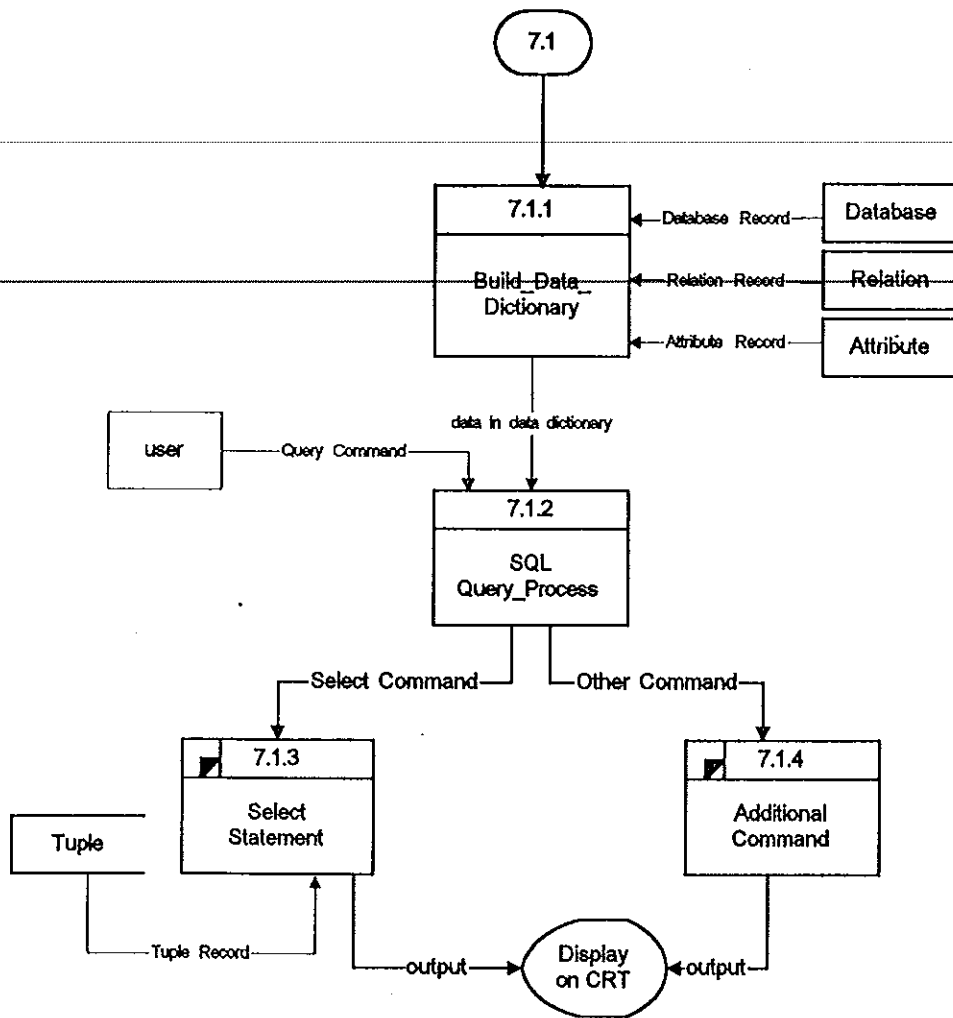
คำอธิบาย เป็นการเรียกใช้การดำเนินงานของคำสั่งปฏิบัติการ Minus ของ RAO

กระบวนการที่	7.1.3.5
ชื่อกระบวนการ	Divide
คำอธิบาย	เป็นการเรียกใช้การดำเนินงานของคำสั่งปฏิบัติการ Divide ของ RAO
กระบวนการที่	7.1.3.6
ชื่อกระบวนการ	Project
คำอธิบาย	เป็นการเรียกใช้การดำเนินงานของคำสั่งปฏิบัติการ Project ของ RAO
กระบวนการที่	7.1.3.7
ชื่อกระบวนการ	Group_Data
คำอธิบาย	เป็นการดำเนินงานจัดกลุ่มข้อมูลที่อยู่ในหน่วยความจำหลัก
จากรูปที่ 4.10	
กระบวนการที่	7.1.4.1
ชื่อกระบวนการ	Start_And_Stop_Additional_Command
คำอธิบาย	เป็นการดำเนินงานควบคุมกระบวนการในการประมวลผล ประโยคคำสั่งอื่นๆ ที่ไม่ใช่คำสั่งในการสอบถามฐานข้อมูล
กระบวนการที่	7.1.4.2
ชื่อกระบวนการ	Open_Database
คำอธิบาย	เป็นกระบวนการดำเนินงานเปิดฐานข้อมูล
กระบวนการที่	7.1.4.3
ชื่อกระบวนการ	Close_Database
คำอธิบาย	เป็นกระบวนการดำเนินงานปิดฐานข้อมูล เมื่อมีการยกเลิก การใช้ฐานข้อมูลนั้นๆ

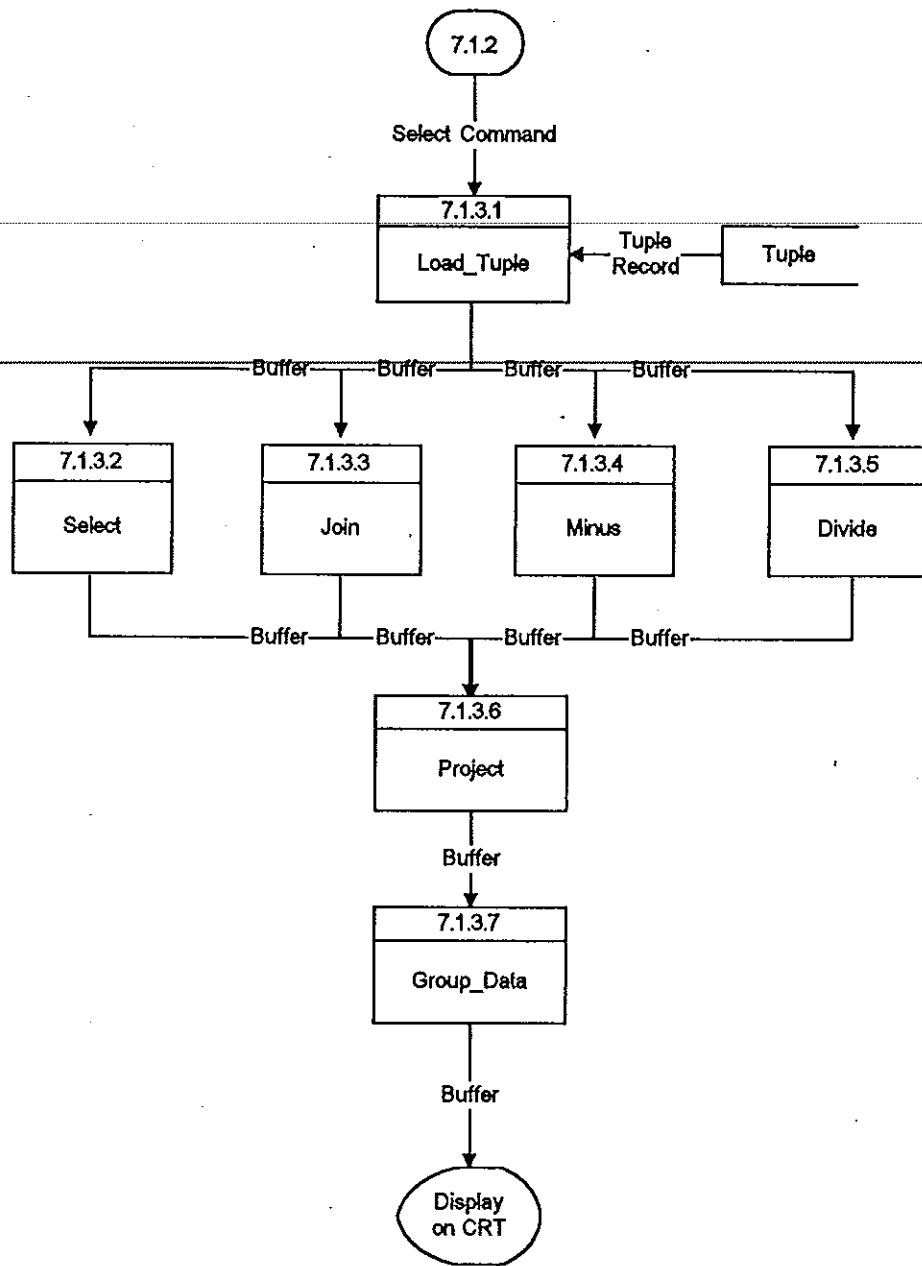
กระบวนกรที่	7.1.4.4
ชื่อกระบวนกร	Run_Query_In_Buffer
คำอธิบาย	เป็นการดำเนินการประมวลผลข้อความสอบถามจาก command buffer
กระบวนกรที่	7.1.4.5
ชื่อกระบวนกร	Getf
คำอธิบาย	เป็นการดำเนินการประมวลผลข้อความสอบถาม โดยเรียกใช้ข้อความสอบถามจากแฟ้มข้อมูลที่ระบุไว้
กระบวนกรที่	7.1.4.6
ชื่อกระบวนกร	Listrel
คำอธิบาย	เป็นการแสดงรายละเอียดของตารางข้อมูลทั้งหมดที่กำลังใช้งานอยู่
กระบวนกรที่	7.1.4.7
ชื่อกระบวนกร	Listdb
คำอธิบาย	เป็นการแสดงรายละเอียดของฐานข้อมูลทั้งหมดที่มีอยู่ในระบบ
กระบวนกรที่	7.1.4.8
ชื่อกระบวนกร	Help
คำอธิบาย	เป็นกระบวนกรแสดงคำสั่งการดำเนินงานทั้งหมดที่มีใช้ใน ตัวแปลภาษา
กระบวนกรที่	7.1.4.9
ชื่อกระบวนกร	Editor_Command
คำอธิบาย	เป็นการดำเนินการเพื่ออำนวยความสะดวกต่างๆ ในการใช้ คำสั่งปฏิบัติการ SQL



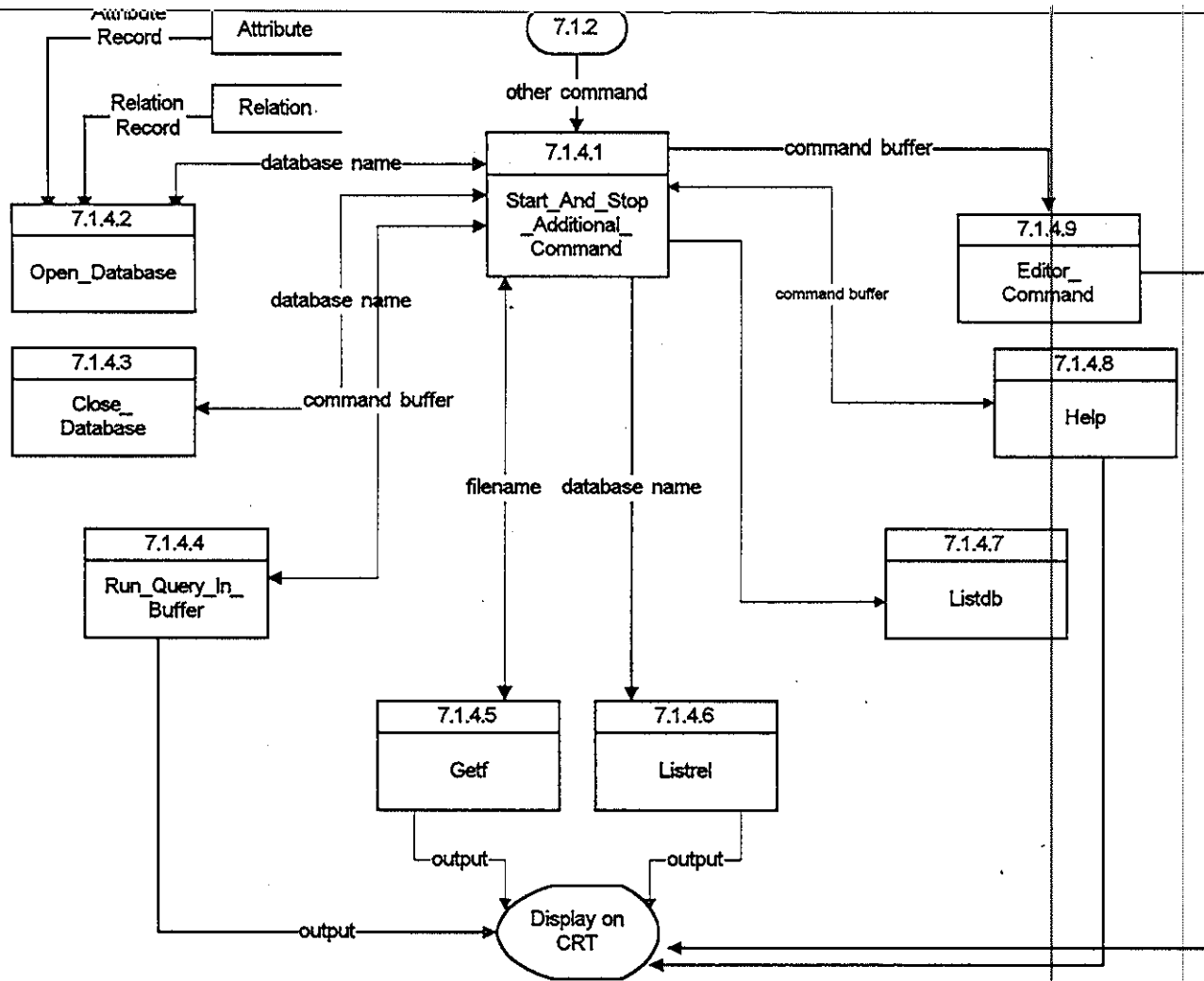
รูปที่ 4.7 แผนภาพกระแสข้อมูลระบบ Mrdbms



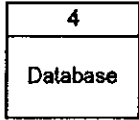
รูปที่ 4.8 แผนภาพกระแสข้อมูลของระบบ SQL ในรายการหลัก



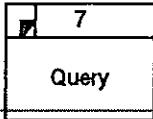
รูปที่ 4.9 แผนภาพกระแสข้อมูลของกระบวนการ Select_Statement



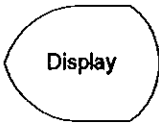
4.10 แผนภาพกระแสน้ำของกระบวนการ Additional_Command



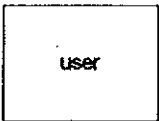
แสดงหมายเลขกระบวนการและชื่อกระบวนการดำเนินงาน



แสดงหมายเลขกระบวนการและชื่อกระบวนการที่ยังมีกระบวนการย่อยอีก



แสดงผลลัพธ์ที่ได้จากทางจอภาพ



แสดงการป้อนข้อมูลเข้าทางจอภาพ



แทนข้อมูลที่อยู่ในหน่วยความจำ

4.6 การดำเนินงานเชื่อมโยงกับคำสั่งปฏิบัติการของภาษาพีรคณิตสัมพันธ์

ดังได้กล่าวมาแล้วว่า งานวิธานิพนธ์นี้เป็นงานวิธานิพนธ์ต่อเนื่องจากงานวิจัยเดิมที่มีมาก่อนหน้านี้ โดยมีการเรียกใช้การดำเนินงานต่างๆ มาประกอบเป็นการดำเนินงานที่ต่อเนื่องกันเพื่อให้เป็นงานวิธานิพนธ์ที่สมบูรณ์ยิ่งขึ้น ส่วนใหญ่เป็นการเรียกใช้โปรแกรมย่อยของคำสั่งปฏิบัติการในโปรแกรม RAO ของงานวิจัยที่มีมาก่อนหน้านี้ และเรียกใช้คำสั่งปฏิบัติการฟังก์ชันต่างๆ ดังนั้นในหัวข้อนี้ จะขอกล่าวถึงรายละเอียดของรูปแบบการเรียกโปรแกรมย่อยเหล่านี้

JOIN

TUPNODE *joinprocess(rel1,rel2,Jcnd)

RELNODE *rel1, *rel2;

Int Jcnd;

โดยที่ Jcnd คือ EQ, NE, GT, GE, LT และ LE

MINUS

TUPNDXNODE *diffprocess(rel1,rel2,cnt)

RELNODE *rel1, *rel2;

Int cnt;

DIVIDE

divided(rel1,rel2,rslname)

RELNODE *rel1, *rel2

char *rslname

โดยที่ rslname คือชื่อตารางข้อมูลที่เป็นผลลัพธ์ที่ได้จากการ divide

PROJECT

TOUITAB *project (pt,fst,len)

TOUITAB *pt

INT fst, len

SELECT

TOUITAB *SQL_select (tab,pt)

WHERE TAB *tab;

TOUITAB *pt;

UNION

TOUITAB *UnionProcess (pt,pt1)

TOUITAB *pt,pt1;

INTERSECT

TOUITAB *IntersectProcess (pt,pt1)

TOUITAB *pt,pt1;

COUNT

INT DoCount(rel,att,type)

RELNODE *rel;

ATTNODE *att;

int type;

โดยที่ type คือรูปแบบของการนับข้อมูล มีรายละเอียดดังนี้คือ

1 เป็นการนับข้อมูลของทุกแถว โดยมีระบุชื่อสคีม่าเฉพาะ

2 เป็นการนับข้อมูลของทุกแถวที่ไม่มีค่าซ้ำกัน

3 เป็นการนับข้อมูลทุกสดมภ์ของทุกแถว

SUM

INT DoSum (rel, att)

RELNODE *rel;

ATTNODE *att;

MAX

CHAR *DoMax (rel, att)

RELNODE *rel;

ATTNODE *att;

MIN

CHAR *DoMin (rel,att)

RELNODE *rel;

ATTNODE *att;

SD

FLOAT DoSd (rel,att)

RELNODE *rel;

ATTNODE *att;

VAR

FLOAT DoVar (rel,att)

RELNODE *rel;

ATTNODE *att;

AVG

FLOAT DoAver (rel,att)

RELNODE *rel;

ATTNODE *att;

ต่อไปนี้เป็นคำสั่งที่มีการพัฒนาเพิ่มเติมในงานวิทยานิพนธ์ เพื่อให้เป็นข้อความสอบถามที่สมบูรณ์มากยิ่งขึ้น โดยได้นำบางคำสั่งปฏิบัติการ RAO มาแก้ไขและปรับปรุงเพื่อให้ครอบคลุมขอบเขตของการสอบถามมากยิ่งขึ้น มีรายละเอียดดังนี้

- โปรแกรมย่อย Union , Intersection, Project และ Select ถูกนำมาใช้ โดยปรับปรุงเพียงเล็กน้อย
- โปรแกรมย่อย Divide ปรับปรุงและสร้างเงื่อนไขของการ Divide เพิ่มเติมเพื่อให้สอดคล้องกับการใช้ข้อความสอบถามข้อมูล

GROUP

TOUTTAB *DoIntoGroup(node)

FUNCTAB *node

ORDER

SORT (str,n,fst,lst)

CHAR *str;

INT n,fst,lst;

โดยที่ str เป็นข้อมูลที่ต้องการจัดเรียง
 n เป็นจำนวนเรคอร์ดของข้อมูล
 fst, lst เป็นตำแหน่งแรกและสุดท้ายของข้อมูลที่ต้องการจัดเรียง

UNIQUE

TOUITAB *CheckDup (pt)

TOUITAB *pt

EDITOR COMMAND

DelRoutine(no) โดยที่ no เป็นหมายเลขคำสั่งที่ต้องการจะลบข้อความ

ChangeRoutine (ts,ss) โดยที่ ss เป็นข้อความเดิมในหน่วยความจำสำรองที่
 ต้องการจะเปลี่ยนและ ts เป็นข้อความใหม่

InsertRoutine (no) โดยที่ no เป็นหมายเลขของคำสั่งที่ต้องการจะเพิ่มข้อความ

AppendRoutine (str) โดยที่ str เป็นข้อความที่เพิ่มต่อท้ายในคำสั่งนั้น

QUERY COMMAND คำสั่งที่พัฒนามีดังนี้ คือ

OPEN, CLOSE, RUN, GETF, LISTREL, LISTDB และ HELP สามารถ

ดูรายละเอียดเพิ่มเติมได้ในภาคผนวก ข.

ต่อไปนี้เป็นตัวอย่างของข้อความสอบถาม SQL ชุดคำสั่งปฏิบัติการที่สัมพันธ์กันที่สัมพันธ์กัน และขั้นตอนการเรียกใช้โปรแกรมย่อยของชุดคำสั่งปฏิบัติการที่สัมพันธ์กัน

ตัวอย่าง

ข้อความสั่งในภาษาแคลคูลัสสัมพันธ์ :

```
SELECT  SNAME
FROM    S,SP
WHERE   S.SNUM = SP.SNUM
AND     PNUM = "P2"
```

ชุดของข้อความสั่งในภาษาที่สัมพันธ์กันที่สัมพันธ์กัน :

```
T1 = JOIN S[SNUM]SP
T2 = SELECT T1[PNUM = "P2"]
T3 = PROJECT T2[SNAME]
DISPLAY T3
```

ขั้นตอนการดำเนินงาน :

```
t1 = *joinprocess("s", "sp", "=")
t2 = *SQL_select ([pnum="p2"], "t1")
t3 = *project ("t2", fst, len)
```

หมายเหตุ

fst = first position of attribute SNAME

len = length of attribute SNAME

ตัวอย่าง การค้นหารายละเอียดข้อมูล โดยมีการจัดกลุ่มข้อมูลเพื่อนำไปเปรียบเทียบกับข้อมูล
ในเขตตามที่กำหนดให้

ประโยคคำสั่งในภาษานกกลศาสตร์สัมพันธ์ :

SELECT *

FROM SP

GROUP BY SNUM

HAVING PNUM = ["P2", "P1"];

ชุดของข้อความสั่งในภาษาพีชคณิตสัมพันธ์ที่สมมูลกัน :

T1 = SELECT SP [PNUM = "P2"]

T2 = SELECT SP [PNUM = "P1"]

T3 = T1 UNION T2

T4 = PROJECT T3 [PNUM]

T5 = PROJECT SP [SNUM PNUM]

T6 = DIVIDE T5 [(PNUM)=(PNUM)] T4

DISPLAY T6

ขั้นตอนการดำเนินงาน :

t1 = *SQL_select ([pnum="P2"], "sp")

t2 = *SQL_select ([pnum="P1"], "sp")

t3 = *UnionProcess ("t1", "t2")

t4 = PROJECT ("t3", fst, len)

t5 = PROJECT ("SP", fst, len)

t6 = DIVIDE ("t5", "t4", "t6")

fst = first position of attribute PNUM

len = length of attribute PNUM

ตัวอย่าง การค้นหารายละเอียดข้อมูล โดยมีการจัดกลุ่มข้อมูลเพื่อนำไปเปรียบเทียบกับข้อมูล
ในเซตตามที่กำหนดให้

ประโยคคำสั่งในภาษานกกลศาสตร์สัมพันธ์ :

```
SELECT *
```

```
FROM P
```

```
WHERE PNUM NOT IN ["P3", "P1"];
```

ชุดของข้อความสั่งในภาษาพีชคณิตสัมพันธ์ที่สมมูลกัน :

```
T1 = SELECT P [PNUM = "P1"]
```

```
T2 = SELECT P [PNUM = "P3"]
```

```
T3 = T1 UNION T2
```

```
T4 = P MINUS T3
```

```
DISPLAY T4
```

ขั้นตอนการดำเนินงาน :

```
t1 = *SQL_select ([pnum="P1"], "p")
```

```
t2 = *SQL_select ([pnum="p3"], "p")
```

```
t3 = *UnionProcess ("t1", "t2")
```

```
t4 = *diff_process ("p", "t3")
```

4.7 การตรวจสอบความผิดพลาด

ลักษณะของความผิดพลาดที่อาจเกิดได้กำหนดไว้ในการทำงานของตัวแปลภาษา SQL มี 2 รูปแบบคือ ความผิดพลาดในการตรวจสอบรูปแบบของภาษา (Syntax Error) และความผิดพลาดในขณะตรวจสอบความหมาย (Semantic Error)

ข้อความผิดพลาดในการตรวจสอบรูปแบบของภาษา

ความผิดพลาดนี้เกิดขึ้นหลังจากที่โปรแกรม scanner อ่านข้อความสอบถามเข้ามาและส่งรหัสคำกลับไปให้โปรแกรม parser ตรวจสอบว่ารหัสคำที่ Scanner ส่งคามานั้นตรงกับรูปแบบไวยากรณ์ที่กำหนดไว้หรือไม่ ถ้ารหัสคำไม่ตรงกันจะเกิดความผิดพลาดในขณะทำการแปลภาษาขึ้น

ตัวอย่าง select * form s;

คำอธิบาย

ความผิดพลาดข้างต้นเกิดขึ้นที่คำว่า "form" ที่ถูกต้องเป็นคำว่า "from"

ข้อความผิดพลาดในขณะตรวจสอบความหมาย

ความผิดพลาดนี้เกิดขึ้นหลังจากที่ผ่านการตรวจสอบรูปแบบของภาษาเรียบร้อยแล้ว เป็นความผิดพลาดในส่วนของความหมาย

ตัวอย่าง select * from name;

คำอธิบาย

สมมติว่าไม่มีตารางชื่อ "name" ในฐานข้อมูล ความผิดพลาดข้างต้นเกิดขึ้นที่คำว่า "name" ซึ่งเป็นตัวแปรชื่อตารางข้อมูลที่มีอยู่ในฐานข้อมูลที่กำลังมีการใช้งานอยู่ เมื่อมีการตรวจสอบชื่อตารางข้อมูลจากฐานข้อมูลที่กำลังใช้งานอยู่ ไม่ปรากฏว่ามีชื่อตารางข้อมูลชื่อ "name" ดังนั้น จะปรากฏข้อความผิดพลาดว่า

"Error : Relname ** xxx ** not found"

ในงานวิทยานิพนธ์นี้ ได้กำหนดรูปแบบของข้อความแสดงข่าวสารความผิดพลาดเป็น
ข้อความภาษาอังกฤษ รายละเอียดดังแสดงในตารางที่ 4.11

ข้อความ	ความหมาย
Attribute ** xxx ** not found	ไม่พบแอตทริบิวต์ xxx ในตารางข้อมูล
Attribute join error	แอตทริบิวต์ที่ใช้ดำเนินการ JOIN ไม่ถูกต้อง
Attribute in subquery not related with attribute in query	แอตทริบิวต์ที่ใช้ดำเนินการใน query ไม่สัมพันธ์ กับแอตทริบิวต์ใน subquery
Database ** xxx ** does not open	ฐานข้อมูลชื่อ xxx ยังไม่ได้ถูกเรียกใช้
Database ** xxx ** not found	ไม่พบฐานข้อมูลชื่อ xxx ในระบบ
Database not open	ไม่มีฐานข้อมูลใดถูกเรียกใช้
Database ** xxx *** open already	ฐานข้อมูลชื่อ xxx ถูกเรียกใช้เรียบร้อยแล้ว
Duplicate attribute ** xxx **	มีการเรียกใช้แอตทริบิวต์ชื่อ xxx ซ้ำกัน
Duplicate relation's name	มีการเรียกใช้ชื่อตารางข้อมูลที่ซ้ำกัน
Wrong data type	ไม่สามารถนำชนิดของแอตทริบิวต์ไปดำเนินการ ฟังก์ชันได้
Data type not compatible	ไม่สามารถดำเนินการได้เนื่องจากมีความผิด พลาดในเกี่ยวกับแอตทริบิวต์
Function ** xxx *** not found	ไม่มีการดำเนินการสำหรับฟังก์ชันนี้
syntax in having clause	รูปแบบการใช้ Having clause ไม่ถูกต้อง
Attribute in having clause	แอตทริบิวต์ใน Having clause ไม่ถูกต้อง
Missing group_by_clause	ต้องมีการใช้ Group_by_clause ด้วย

ตารางที่ 4.11 ข้อความแสดงข่าวสารความผิดพลาดและสาเหตุของความผิดพลาด

ข้อความ	สาเหตุ
Missing relname	ไม่ได้ใส่ชื่อตารางข้อมูล
No line in buffer	ไม่มีข้อมูลใน command buffer
No. of relname not related	จำนวนของตารางข้อมูลไม่สัมพันธ์กัน
No. of relname more than one not open database yet	จำนวนของตารางข้อมูลมีมากกว่าหนึ่ง ต้องมีการเรียกใช้ฐานข้อมูลก่อน
Open more than one database	มีการเรียกใช้ฐานข้อมูลมากกว่าหนึ่ง
Operator function error	ตัวดำเนินการในฟังก์ชันไม่ถูกต้อง
Relname ** xxx ** not found	ไม่พบตารางข้อมูลที่ต้องการในฐานข้อมูล
String not match	ค้นหาไม่พบสายอักขระที่กำหนดไว้
Data type not match, expect INTEGER	ค่าที่ใช้ต้องเป็นเลขจำนวนเต็ม
Data type not match, expect STRING	ค่าที่ใช้ต้องเป็นสายอักขระ
Unable to create or open file	ไม่สามารถสร้างหรือเปิดแฟ้มข้อมูลที่กำหนดไว้ได้
Unable to open file	ไม่สามารถเปิดแฟ้มข้อมูลที่กำหนดให้ได้

ตารางที่ 4.11 ข้อความแสดงข่าวสารความผิดพลาดและสาเหตุของความผิดพลาด (ต่อ)

บทที่ 5

สรุปและข้อเสนอแนะ

เนื่องจากงานวิทยานิพนธ์นี้เป็นงานต่อเนื่องจากงานวิทยานิพนธ์ที่มีมาก่อนหน้านี้ ซึ่งประกอบด้วยส่วนการใช้ภาษานิยามฐานข้อมูลและภาษาคำเนิงานกับฐานข้อมูลอันมีการจัดการด้านการเพิ่ม ลบ และเปลี่ยนแปลงข้อมูล และส่วนของการสอบถามฐานข้อมูลนั้นได้มีการพัฒนาโปรแกรมการประมวลผลภาษาฐานข้อมูลเชิงสัมพันธ์แบบพีชคณิตที่เรียกว่า RAO ขึ้นมา โดยโปรแกรมทั้ง 2 ส่วนนี้มีรูปแบบเป็นการจัดการแบบหน้าต่าง (Menu Pulldown) ซึ่งอำนวยความสะดวกให้กับผู้ใช้ทั่วไปที่ไม่จำเป็นต้องศึกษารูปแบบของไวยากรณ์คำสั่งมาก่อน เป็นการประมวลผลทีละขั้นตอนและได้ผลลัพธ์ออกมาทันที

ในส่วนของภาษาสอบถามฐานข้อมูลของงานวิทยานิพนธ์นี้ เป็นการประมวลผลผ่านทางแป้นพิมพ์และทางจอภาพ โดยผู้ใช้จะต้องศึกษาลักษณะรูปแบบไวยากรณ์ของภาษาฐานข้อมูล SQL และเขียนการสอบถามข้อมูลอยู่ในรูปของข้อความสั่งด้วยภาษาฐานข้อมูล SQL ผ่านทางแป้นพิมพ์ เพื่อคำเนิงานกับฐานข้อมูลและแสดงผลพร้อมทั้งที่ทางจอภาพ

การพัฒนาโปรแกรมตัวแปลภาษา SQL ในงานวิทยานิพนธ์นี้ เป็นการศึกษาและออกแบบข้อความสั่ง SQL ให้ครอบคลุมการสอบถามฐานข้อมูลได้อย่างทั่วถึง โดยมีการกำหนดลักษณะของอักขระที่ใช้ คำสงวน ชนิดของนิพจน์การกระทำ ชนิดและลักษณะการใช้งานของฟังก์ชันต่างๆ

การคำเนิงานของตัวแปลภาษา SQL นี้ เป็นลักษณะของงานประเภทโต้ตอบกับผู้ใช้ตลอดเวลา จึงนำตัวแปลภาษาประเภทอินเทอร์พรีเตอร์มาใช้ในการตรวจสอบรูปแบบของข้อมูลเข้าซึ่งมีการรอรับข้อมูลเข้าทางแป้นพิมพ์ พร้อมกับแสดงผลพร้อมทั้งที่ทางจอภาพหรือเครื่องพิมพ์ โดยการคำเนิงานประมวลได้ใช้ปทานุกรมข้อมูลร่วมกับงานที่ดำเนินการมาก่อนหน้านี้ และเรียกใช้โปรแกรมการคำเนิงานคำสั่งปฏิบัติการฐานข้อมูลเชิงสัมพันธ์แบบพีชคณิตของ RAO ที่สมันักกันมาทำงานร่วมกันได้

5.1 ปัญหาและอุปสรรค

ในส่วนของการดำเนินงานพัฒนาโปรแกรมตัวแปลภาษานี้ เนื่องจากเป็นงานวิทยานิพนธ์ ต่อเนื่องจากงานวิทยานิพนธ์ที่มีมาก่อนหน้านี้ ซึ่งมีการพัฒนาระบบการดำเนินงานยูนิกส์ บน เครื่องมินิคอมพิวเตอร์ VAX 11/785 มาก่อน ซึ่งมีข้อจำกัดในด้านการให้บริการใช้เครื่อง และระบบมัลติยูเซอร์ ซึ่งเป็นการทำงานพร้อมกันที่หลายๆ คน ทำให้การดำเนินงานล่าช้า ไป และในเรื่องของภาษาที่ใช้ในการพัฒนา เนื่องจากผู้วิจัยไม่มีประสบการณ์ในการเขียน โปรแกรมภาษาซีมาก่อน ในเรื่องของพอยเตอร์ การกำหนดเนื้อที่ให้หน่วยความจำ ฯลฯ ซึ่งเป็นอีกปัญหาหนึ่งในการดำเนินงานพัฒนาครั้งนี้

ในส่วนของการเขียนข้อความสอบถาม SQL และลำดับคำสั่งภาษาที่ซคณิตสัมพันธ์ที่สมนัยกันนั้น ในบางข้อความสอบถาม SQL ไม่สามารถแปลงให้อยู่ในรูปลำดับคำสั่งภาษาที่ซคณิตสัมพันธ์ที่สมนัยกันได้ เนื่องจากภาษาที่ซคณิตสัมพันธ์ที่เรียกมาใช้จากงานที่มีมาก่อนหน้านี้ไม่เอื้ออำนวยให้กับการทำงานบางรูปแบบ

5.2 ข้อเสนอแนะ

การดำเนินงานประมวลผลของตัวแปลภาษา SQL นี้จะสามารถสอบถามข้อมูลได้หลากหลายและทั่วถึงได้ จำเป็นต้องมีคำสั่งปฏิบัติการใน RAO ที่สมนัยกันมากกว่านี้ ดังนั้นจึงควรมีการเพิ่มเติมคำสั่งปฏิบัติการใน RAO และเพิ่มส่วนสอบถามข้อมูลให้สามารถทำได้ในกรณีสอบถามข้อมูลที่ซับซ้อน ดังตัวอย่างต่อไปนี้

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

รูปแบบคำสั่ง

```
select sname
from s
where snum <> "S2"
and snum in ( select snum
              from sp
              group by snum
              having pnum contains (select pnum
                                    from sp
                                    where snum = "S2"));
```

ชุดของข้อความสั่งในภาษาที่ซกคณิตสัมพันธ์ที่สมนัยกันควรเป็นดังนี้ :

```
T1 = SELECT SP [SNUM = "S2"]
T2 = PROJECT T1 [ PNUM]
T3 = SELECT SP [ SNUM <> "S2"]
T4 = PROJECT T3 [SNUM, PNUM]
T5 = DIVIDE T4 [(PNUM) >= (PNUM)] T2
T6 = JOIN S [SNUM] T5
T7 = PROJECT T6 [SNAME]
DISPLAY T7
```

บรรณานุกรม

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

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

อิว ไอบรากาญจนกุล (2534), Yacc : Yet Another Compiler-Compiler,
ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์

อิว ไอบรากาญจนกุล (2534), Lex : A Lexical Analyzer Generator,
ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์

อิว ไอบรากาญจนกุล(2534), การเขียนคอมไพเลอร์สำหรับ IBM PC,
บริษัท ซีเอ็ดยูเอ็คชั่น จำกัด

Alfred V. Aho, Jeffrey D. Ullman(1979), Principle of Compiler Design,
3rd edition, Addison-Wesley Publishing Company.

Arthur B. Pyster (1988), Compiler Design and Construction, Tools
and Techniques (with C and Pascal), 2nd edition,
Van Nostrand Reinhold Company.

Date C.J. (1989), An introduction to database system, 5th edition,
Addison-Wesley Publishing Company.

Date C.J. (1989), A Guide to the SQL standard, 2nd edition,
Addison-Wesley Publishing Company.

Chamberlin D.D., Reisner P. and Boyce R.F. (1975),
Human factors Evaluation of Two Data Base Query Languages :
SQUARE and SEQUEL, Proc. AFIPS National Computer Conference,
Anaheim : CA.

Chamberlin, Astrahan, Eswaran, Griffiths, Lorie, Mehl, Reisner and
Wade (1976),
SEQUEL 2 : A Unified Approach to Data Definition,
Manipulation and Control,
IBM Research Laboratory SanJose, CA.

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

Horspool R.N. (1986), C-programming in the Berkeley UNIX
environment, Prentice-Hall Publishing Company.

Rinchart H. and Winston (1983), Unix Programmer's Manual,
Vol.1, Bell Laboratories.

Tare Ramknishna S. (1988), Unix Utilities, McGraw-Hill Book Company.

Theerachetmongkol A. and Montgomery A.Y. (1981),

The Variant Relational Model Research Report,

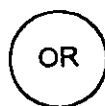
Monash University.

ח. חזקת רב.

วากยสัมพันธ์ของข้อความสอบถาม SQL ในรูปผังภาพวากยสัมพันธ์ (Syntax Diagram)

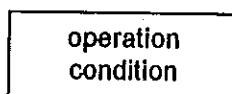
ในบทที่ 4 ได้กล่าวถึง รูปแบบวากยสัมพันธ์ของภาษา SQL ที่ใช้ในงานวิชานี้ และ การเขียนวากยสัมพันธ์ของภาษา SQL ในโปรแกรม Yacc มาแล้วตามลำดับ ในบทนี้จะขออธิบายวากยสัมพันธ์ของภาษา SQL ที่ใช้ จะใช้สัญลักษณ์ของภาพแทนในลักษณะที่เรียกว่า ผังภาพวากยสัมพันธ์ โดยเขียนแยกเป็นรูปภาพของของสัญลักษณ์ที่ถูกต้องแทนที่ได้ ผังภาพวากยสัมพันธ์มีกฎเกณฑ์การเขียนดังนี้

1. อักษรใดๆ ที่ปรากฏในวงกลม หรือ รูปสัญลักษณ์ต่อไปนี้

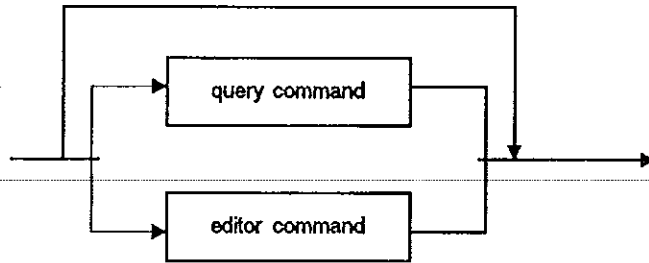


จะต้องเขียนในโปรแกรมเช่นเดียวกับที่ปรากฏในผังภาพ

2. ทุกๆ ข้อความที่ปรากฏในรูปสี่เหลี่ยมผืนผ้า แทนด้วยสัญลักษณ์ที่ถูกต้องแทนที่ได้ เขียนผังภาพวากยสัมพันธ์ของภาษา SQL ที่ใช้ในงานวิชานี้ ได้ดังนี้

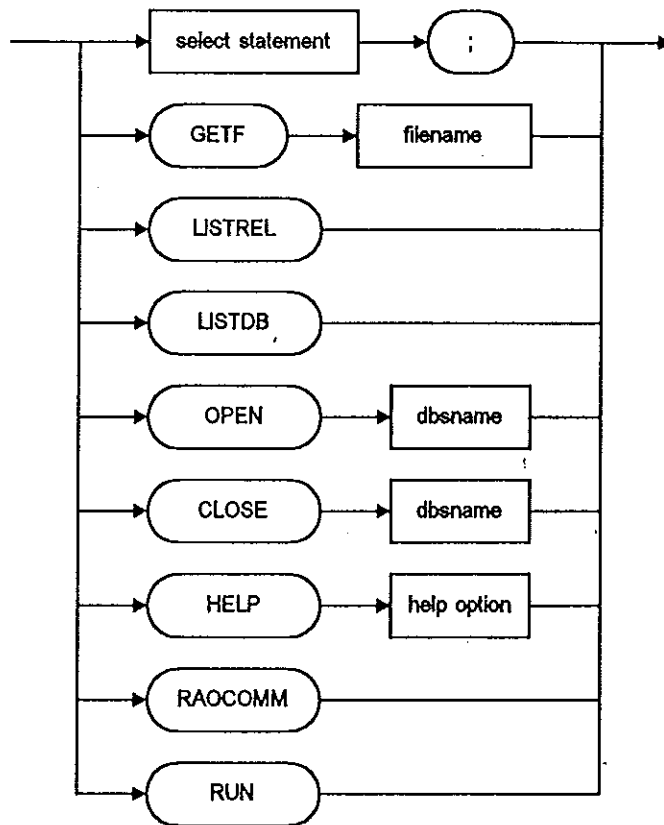


SQL Command



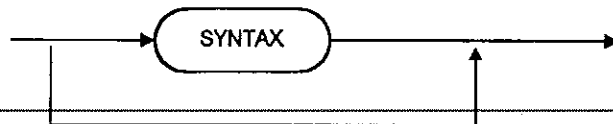
รูปที่ 1. SQL Command

Query Command



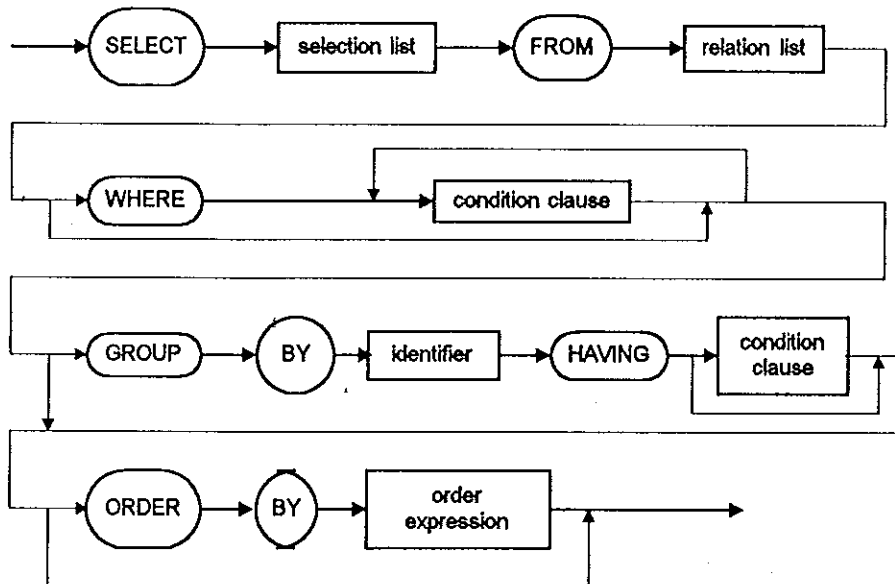
รูปที่ 2. Query Command

help option



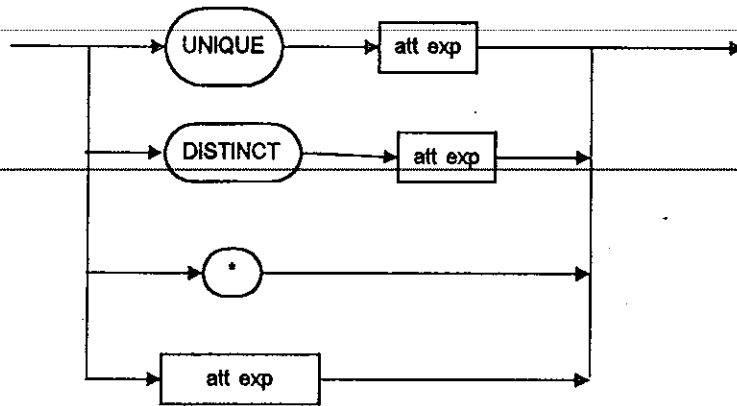
รูปที่ 3. help option

select stmt



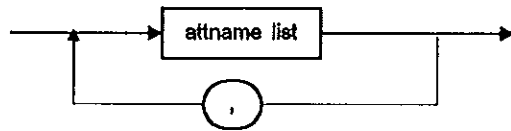
รูปที่ 4. select stmt

selection list

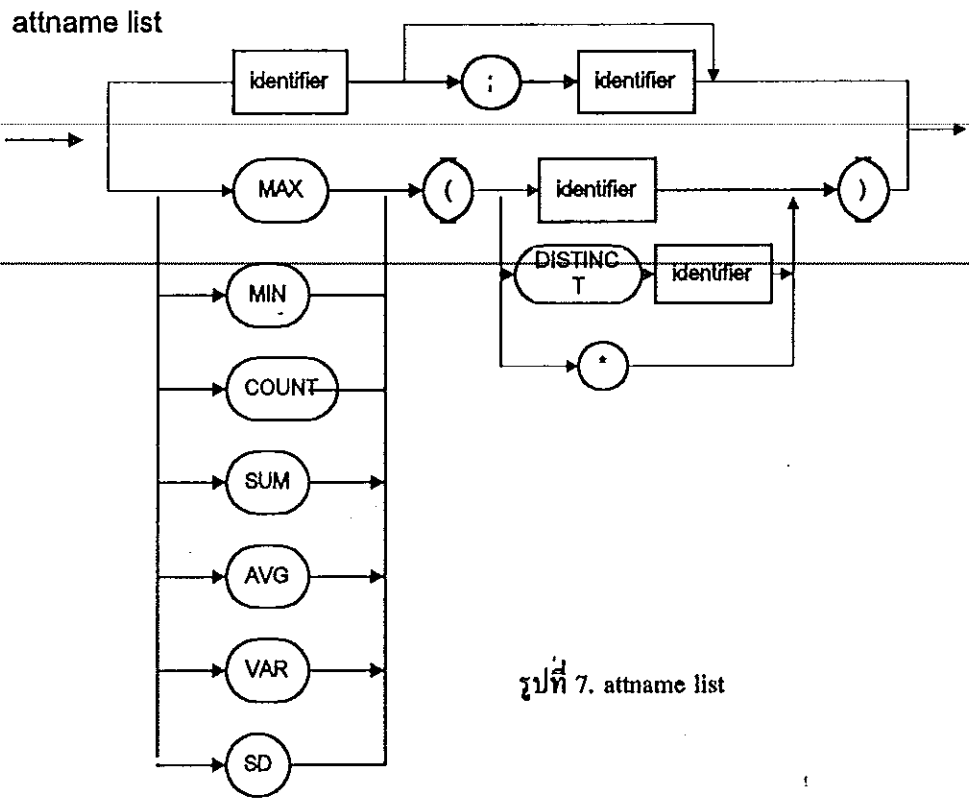


รูปที่ 5. selection list

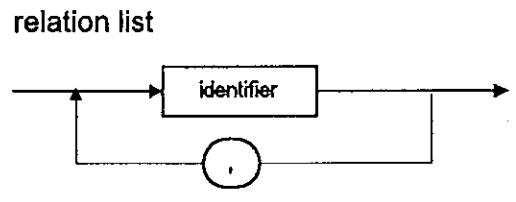
att exp



รูปที่ 6. att exp

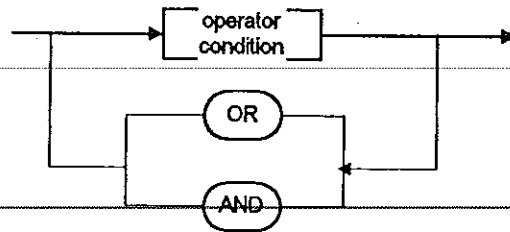


รูปที่ 7. atname list



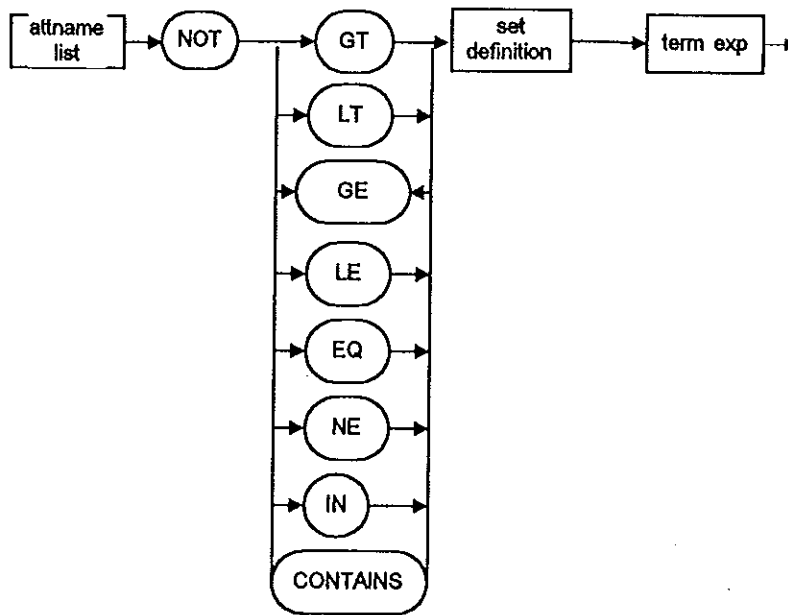
รูปที่ 8. select stmt

condition clause



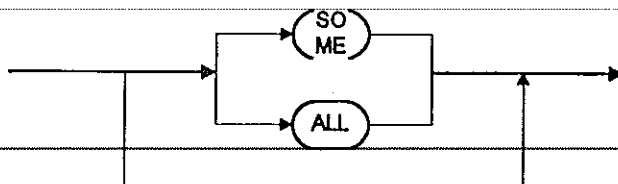
รูปที่ 9. condition clause

operator condition



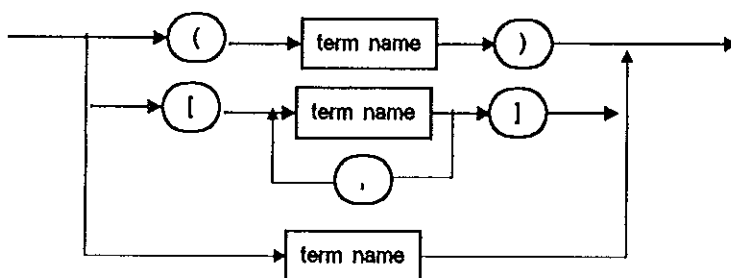
รูปที่ 10. operator condition

set definition



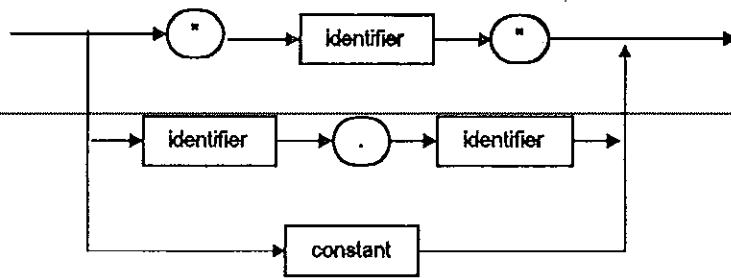
รูปที่ 11. set definition

term exp



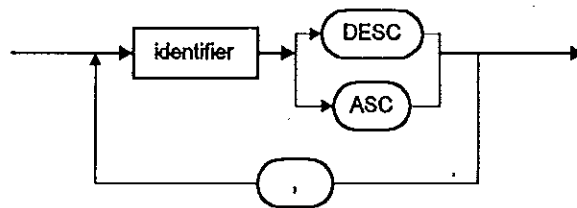
รูปที่ 12. term exp

term name



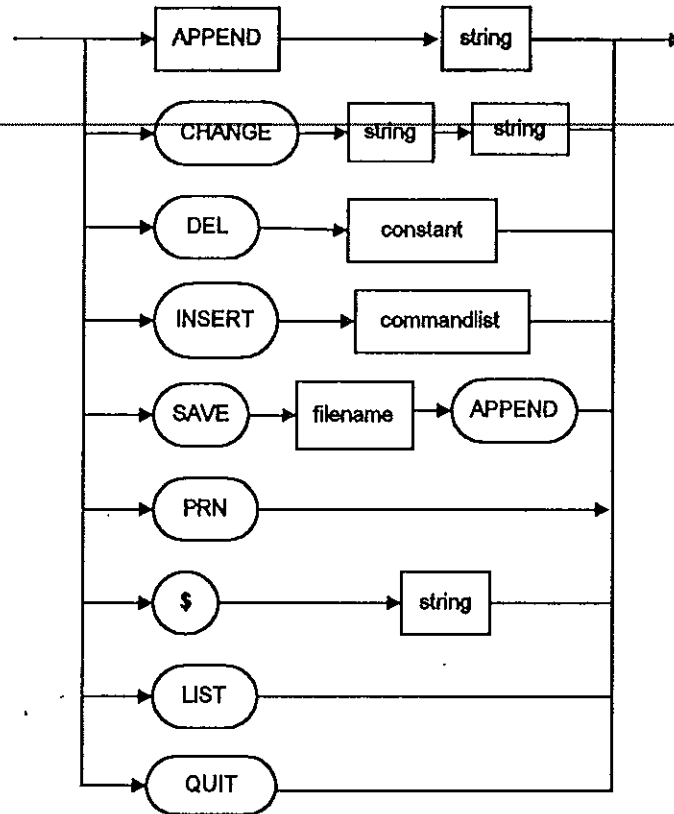
รูปที่ 13. term name

order expression



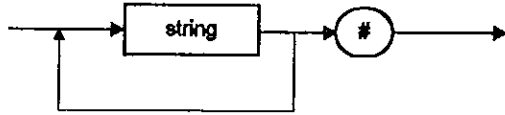
รูปที่ 14. order expression

editor command



รูปที่ 15. editor command

commandlist



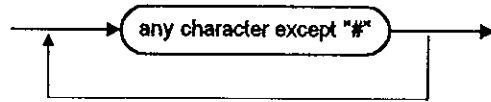
รูปที่ 16. commandlist

identifier



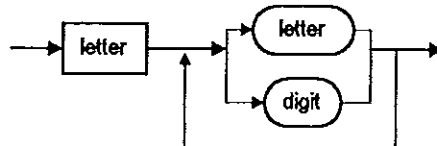
รูปที่ 17. dbname, filename

identifier



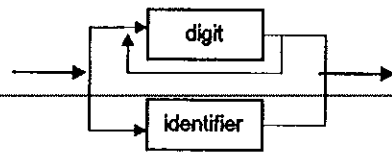
รูปที่ 18. string

identifier



รูปที่ 19. identifier

constant



รูปที่ 20. constant

หมายเหตุ digit หมายถึงตัวเลข 0..9

letter หมายถึงตัวอักษรภาษาอังกฤษ a..z หรือ A..Z

any character หมายถึงอักขระใดๆ

ד. חזמארת

เอกสารสำหรับผู้ใ้

โปรแกรมตัวแปลภาษา SQL นี้ เป็นงานวิทยานิพนธ์ต่อเนื่องจากงานวิทยานิพนธ์ที่มีมาก่อนหน้านี้คือ ระบบ Mrdbms โดยเลือกใช้ตัวแปลภาษาประเภทอินเตอร์พรีเตอร์มาทำการแปลและดำเนินการ โดยใช้โปรแกรม Lex และ Yacc ช่วยในขั้นตอนการวิเคราะห์ศัพท์ การวิเคราะห์วากยสัมพันธ์ และการวิเคราะห์ความหมายและตีความ โปรแกรมตัวแปลภาษา SQL นี้จะต้องเรียกใช้ผ่านทางโปรแกรมระบบเดิมซึ่งมีการจัดการแบบหน้าต่าง โดยเลือกรายการย่อย "SQL" จากรายการหลัก "Query" ของระบบ Mrdbms จะขอกล่าวถึงขั้นตอนการเรียกใช้โปรแกรมตามลำดับต่อไปนี้

1. การเรียกใช้โปรแกรมระบบ Mrdbms

เนื่องจากโปรแกรมตัวแปลภาษา SQL นี้จะต้องเรียกใช้ผ่านทางโปรแกรมระบบเดิมคือระบบ Mrdbms ดังนั้นขอกล่าวถึงวิธีการเรียกใช้โปรแกรม Mrdbms และการเริ่มใช้งานอย่างย่อๆ ดังนี้

1.1 การเรียกใช้โปรแกรม Mrdbms

หลังจากที่เข้าไปอยู่ในการดำเนินการของระบบ Unix เรียบร้อยแล้ว ให้พิมพ์คำว่า "Rdbms" ทางแป้นพิมพ์ ดังนี้

```
sriltrang> Rdbms
```

1.2 เมื่อเริ่มใช้งานโปรแกรมจะถามผู้ใ้ถึงชนิดของเครื่องที่ใช้งาน โดยจะแสดงข้อความ

WHAT KIND OF TERMINAL?

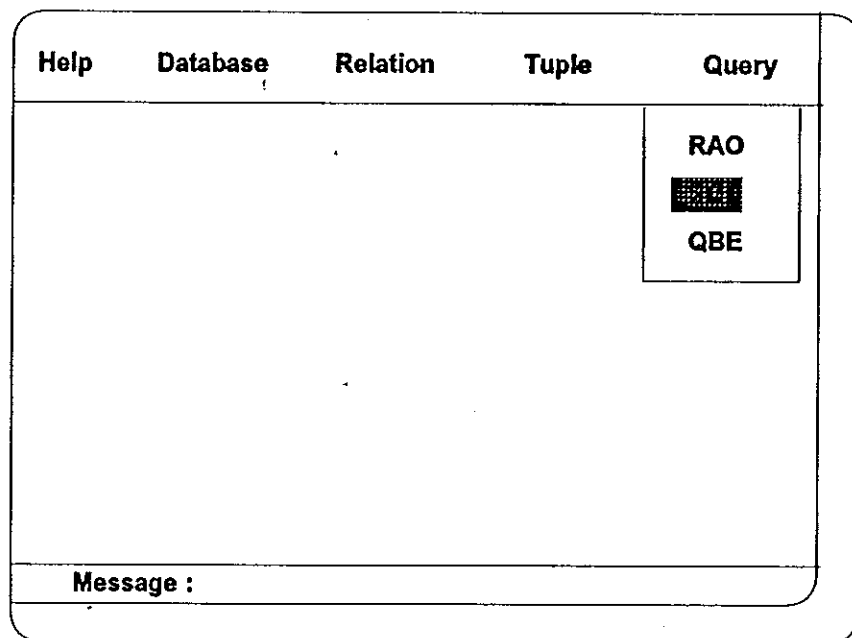
VT terminal

Micro terminal

SELECT [_]

ให้ผู้ใช้ระบุความต้องการโดยกดอักษร 'V' หรือ 'H' เพื่อจะได้ปรับจอภาพสำหรับการแสดงผลที่ได้จากการทำงาน เมื่อการทำงานเรียบร้อยแล้วจะปรากฏจอภาพ รายการหลัก Query ดังรูปที่ ข.1 รายละเอียดของรายการหลัก Help Database Relation และ Tuple ไม่ขอกล่าวในที่นี้ ผู้สนใจสามารถดูรายละเอียดเพิ่มเติมได้จากงานวิทยานิพนธ์ที่มีมาก่อนหน้านี้ [(เพ็ชรพร (2534)] ส่วนรายการหลัก Query เป็นการทำงานของส่วนภาษารฐานข้อมูล ซึ่งมีรายการย่อย 3 รายการคือ รายการย่อย RAO, SQL และ QBE

รายการหลัก Query



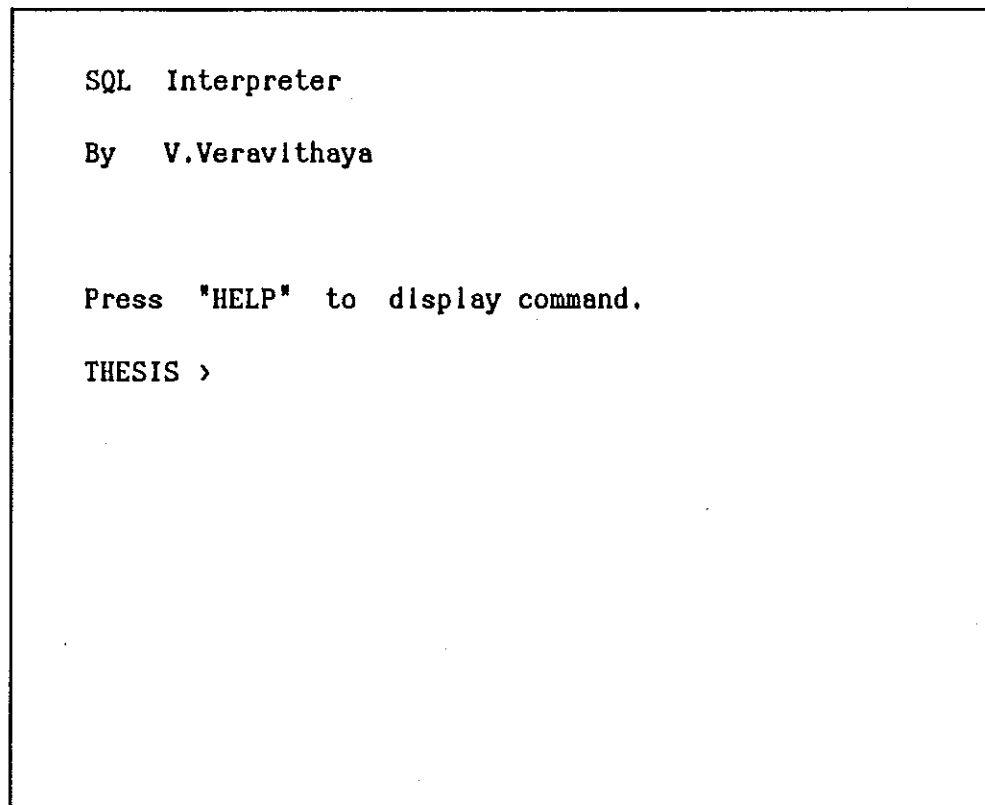
รูปที่ ข.1 จอภาพรายการหลัก Query ของระบบ Mrdbms

2. การเรียกใช้โปรแกรมตัวแปลภาษา SQL

เนื่องจากรูปแบบการทำงานของโปรแกรมตัวแปลภาษา SQL นี้เป็นแบบ Line Editor ดังนั้นเพื่อให้การใช้งานเป็นไปได้อย่างรวดเร็วและถูกต้อง ขอแนะนำให้ผู้ศึกษารูปแบบไวยากรณ์ภาษา SQL ในงานวิทยานิพนธ์นี้ ซึ่งได้กล่าวรายละเอียดไว้ในบทที่ 4 แล้ว ส่วนวิธีการเรียกใช้โปรแกรม วิธีการสร้างข้อความสั่งของภาษา คำสั่งการใช้ Editor Command และการออกจากโปรแกรมจะได้กล่าวรายละเอียดดังนี้ ตามลำดับ

2.1 การเรียกใช้โปรแกรมตัวแปลภาษา SQL

การเรียกใช้โปรแกรมตัวแปลภาษา SQL นี้จะต้องเรียกใช้ผ่านโปรแกรม Mrdbms โดยผู้ใช้เลือกรายการย่อย "SQL" จากรายการหลัก "Query" ในรูปที่ ข.1 หลังจากนั้นจะเป็นการเริ่มต้นเข้าสู่โปรแกรมตัวแปลภาษา SQL ดังแสดงรายละเอียดในรูปที่ ข.2



รูปที่ ข.2 การใช้โปรแกรมตัวแปลภาษา

- จากรูปที่ ข.2 เมื่อผู้ใช้พิมพ์คำว่า "HELP" ทางแป้นพิมพ์ จะแสดงรายละเอียดสรุปคำสั่งทั้งหมดที่ใช้ในตัวแปลภาษา SQL ทางจอภาพดังรูปที่ ข.3

***** SUMMARY OF SQL INTERPRETER COMMANDS *****	
H E L P	
COMMAND	MEANING
HELP	Display SQL Interpreter command.
HELP SYNTAX	Display SQL language syntax.
D A T A B A S E	
COMMAND	MEANING
OPEN filename	Open database name "filename".
CLOSE filename	Close database name "filename".
LISTREL	List all relations in current database.
LISTDB	List all databases in data dictionary.
more ..	

รูปที่ ข.3 รายการ HELP

LINE EDITOR	
COMMAND	MEANING
INSERT	Insert command line at current line.
DEL linenum	Delete command line of line no. "linenum"
CHANGE	Change command line at current line.
APPEND	Append command line at current line.
SAVE filename	Save current result in "filename.lst" at current directory.
SAVE filename APPEND	Save and append current result in "filename.lst" at current directory.
LIST linenum	List command line buffer no linenum.
LIST	List all command line buffer.
EXECUTE	
COMMAND	MEANING
RUN	Rerun command in command line buffer.
GETF filename	Run command from text file name "filename.scp".
more ...	

```
PRN          | Print current result of command line buffer.  
RAOCOMM     | Print RAO operation of current result.  
$  command  | Use shell command on unix system.
```

E X I T

```
QUIT        |   Quit SQL Interpreter.
```

รูปที่ ข.3 รายการ HELP (ต่อ)

- จากรูปที่ ข.2 เมื่อผู้ใช้พิมพ์คำว่า "HELP SYNTAX" ทางแป้นพิมพ์ จะแสดงรายละเอียดสรุปรูปแบบข้อความสั่ง SELECT ที่ใช้ในตัวแปลภาษา SQL ทางจอภาพดังแสดงในรูปที่ ข.4

***** SUMMARY SYNTAX OF SQL LANGUAGE *****

SELECT STATEMENT

=====

```

SELECT      "*"
            | [UNIQUE|DISTINCT] atname
            | <atname_list> ["," <atname_list>]
            | <function_list> ["," <function_list>]

FROM        relname [ "," relname]

[ WHERE     <condition_clause> [OR|AND <condition_clause>]]

[ GROUP BY atname [ HAVING <condition_clause> ] ]

[ ORDER BY atname [ASC|DESC] [ "," atname [ASC|DESC]] ]

```

รูปที่ ข.4 รูปแบบข้อความสอบถาม SELECT

***** SUMMARY SYNTAX OF SQL LANGUAGE *****

<condition_clause> : <attname_list> <operator> <term_exp>

<attname_list> : attname ["." attname]
 | attname ["," attname]

<function_list> : MAX | MIN | SUM | AVG | COUNT | SD | VAR
 "(" "*" | attname | DISTINCT attname ")"

<operator> : [NOT] GT | LT | GE | LE | EQ | NE | IN | CONTAINS
 [SOME|ALL]

<term_exp> : <single_term>
 | "[" single_term ["," single_term]]"
 | "(" <select_stmt> ")"

<single_list> : "" attname "" | constant | attname "."
 attname

รูปที่ ข.4 รูปแบบข้อความสอบถาม SELECT (ต่อ)

2.2 การรับและประมวลผลข้อความสอบถาม SQL

ในงานวิทยานิพนธ์นี้กำหนดให้มีวิธีการประมวลผลข้อความสอบถาม SQL ได้ 3 แบบตามความต้องการของผู้ใช้ แสดงรายละเอียดดังนี้

- โดยการป้อนข้อความสอบถาม SQL ทางแป้นพิมพ์โดยตรง

ตัวอย่าง `select * from p;`

- โดยการเรียกใช้ข้อความสอบถาม SQL จากแฟ้มข้อมูล

ตัวอย่าง

`getf filename` เป็นการเรียกใช้ข้อความสอบถาม SQL จากแฟ้มข้อมูลชื่อ `filename`

- โดยการเรียกใช้ข้อความสอบถาม SQL จากหน่วยความจำหลัก

ตัวอย่าง

`list` เป็นการแสดงข้อความสอบถาม SQL ที่ใช้ประมวลผลครั้งล่าสุดจากหน่วยความจำหลัก

`run` เป็นการนำข้อความสอบถาม SQL ที่ได้จากคำสั่ง `list` มาประมวลผล

เพื่ออำนวยความสะดวกในการใช้ข้อความสอบถาม SQL มากขึ้น จึงมีคำสั่งเพื่อดำเนินงานเพิ่มเติม โดยการดำเนินงานนี้จะไม่ทำให้ได้ตารางข้อมูลผลลัพธ์ คำสั่งเพื่ออำนวยความสะดวกในการใช้ข้อความสอบถามมีดังนี้

รูปแบบ `OPEN <database>`

ความหมาย เป็นคำสั่งเพื่อเรียกใช้ฐานข้อมูล `<database>`

ตัวอย่าง `SQL> open base`

เป็นการเรียกใช้ฐานข้อมูลชื่อ `base`

รูปแบบ CLOSE <database>
ความหมาย เป็นคำสั่งเพื่อบกเลิกการเรียกใช้ฐานข้อมูล <database>
ตัวอย่าง SQL> close base
 เป็นการยกเลิกการเรียกใช้ฐานข้อมูลชื่อ base

รูปแบบ LISTDB
ความหมาย เป็นคำสั่งเพื่อแสดงชื่อฐานข้อมูลทั้งหมดในระบบ

รูปแบบ LISTREL
ความหมาย เป็นคำสั่งเพื่อแสดงชื่อตารางข้อมูลทั้งหมดในฐานข้อมูลที่เรียกใช้

รูปแบบ SAVE
ความหมาย เป็นคำสั่งเพื่อเก็บผลลัพธ์ที่ได้จากการดำเนินงานครั้งล่าสุดไว้ในแฟ้มข้อมูล

ตัวอย่าง SQL> save name
 เป็นการเก็บผลลัพธ์ที่ได้จากการดำเนินงานครั้งล่าสุดไว้ในแฟ้มข้อมูลชื่อ name.lst

ตัวอย่าง SQL> save name append
 เป็นการเก็บผลลัพธ์ที่ได้จากการดำเนินงานครั้งล่าสุด ต่อท้ายแฟ้มข้อมูล ชื่อ name.lst

รูปแบบ LIST
ความหมาย เป็นคำสั่งเพื่อแสดงข้อความสอบถามที่เก็บอยู่ในหน่วยความจำหลัก
ตัวอย่าง SQL> list
 เป็นการแสดงข้อความสอบถามที่เก็บอยู่ในหน่วยความจำหลัก

ตัวอย่าง SQL> list 2
เป็นการแสดงข้อความสอบถามในบรรทัดที่ 2 ในหน่วยความจำหลัก

รูปแบบ PRN

ความหมาย เป็นคำสั่งเพื่อพิมพ์ผลลัพธ์ที่ได้จากการดำเนินการคำสั่งล่าสุดทางเครื่องพิมพ์

รูปแบบ RAOCOMM

ความหมาย เป็นคำสั่งเพื่อแสดงลำดับคำสั่งปฏิบัติ RAO ที่สัมพันธ์กับข้อความสอบถาม SQL ที่ดำเนินการอยู่

รูปแบบ CHANGE

ความหมาย เป็นคำสั่งการดำเนินการเปลี่ยนแปลง ของข้อความบางส่วนของข้อความสอบถาม SQL ที่อยู่ในหน่วยความจำหลัก

ตัวอย่าง SQL> list 2

SQL> change

Enter old string : *

Enter new string : snum

เป็นการเปลี่ยนแปลงสายอักขระในบรรทัดที่ 2 ของข้อความสอบถามข้อมูล SQL ครั้งล่าสุด จากสายอักขระ "*" เป็น "snum"

รูปแบบ APPEND

ความหมาย เป็นคำสั่งการดำเนินการเพิ่มต่อท้าย ข้อความบางส่วนของข้อความสอบถาม SQL ที่อยู่ในหน่วยความจำหลัก

ตัวอย่าง SQL> list 2
 SQL> append
 Enter string to append : ,sname
 เป็นการเพิ่มต่อท้ายสายอักขระ ",sname" ในบรรทัดที่ 2 ของ
 ข้อความสอบถามข้อมูล SQL ครั้งล่าสุด

รูปแบบ DEL
ความหมาย เป็นคำสั่งการดำเนินงาน ลบ ข้อความบางส่วนของข้อความสอบ
 ถาม SQL ที่อยู่ในหน่วยความจำหลัก
ตัวอย่าง SQL> del 3
 เป็นการลบข้อความในบรรทัดที่ 3 ของข้อความสอบถามข้อมูล
 SQL ครั้งล่าสุด

รูปแบบ INSERT
ความหมาย เป็นคำสั่งการดำเนินงาน เพิ่ม ข้อความในข้อความสอบถาม
 SQL ที่อยู่ในหน่วยความจำหลัก
ตัวอย่าง SQL> list
 SQL> insert
 [3] where city = "London";
 [4] #
 เป็นการเพิ่มสายอักขระ where city = "London" ในบรร
 ทัดหลังสุด(ในที่นี้คือบรรทัดที่ 2)ของข้อความสอบถามข้อมูล SQL
 ครั้งล่าสุด โดยที่เครื่องหมาย "#" หมายถึงสิ้นสุดข้อความที่จะ
 เพิ่ม

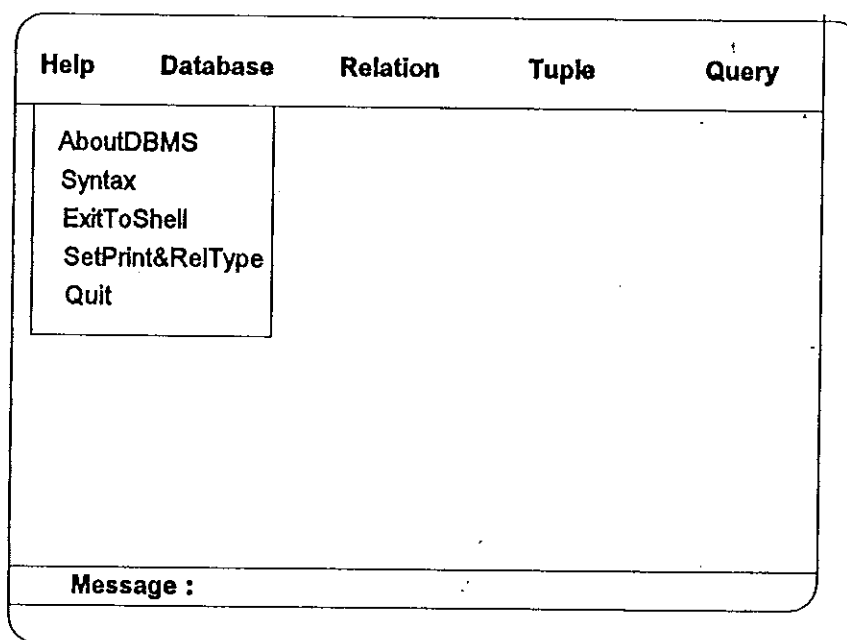
รูปแบบ QUIT

ความหมาย เป็นคำสั่งเพื่อยกเลิกการทำงานในส่วนตัวแปลภาษา SQL ออก
ไปสู่ระบบงานเดิม

- จากรูปที่ ข.2 เมื่อผู้ใช้พิมพ์คำสั่ง "QUIT" จะปรากฏจอภาพรายการหลักของ
โปรแกรม Mrdbms ดังรูปที่ ข.1 เป็นการกลับคืนสู่ระบบ Mrdbms

3. วิธีออกจากโปรแกรม Mrdbms

การออกจากโปรแกรม Mrdbms ให้ผู้ใช้เลือกรายการย่อย Quit จากรายการหลัก
Database ซึ่งเป็นการออกจากโปรแกรม Mrdbms และกลับไปยัง Shell Command ของ
ระบบยูนิกซ์ แสดงดังภาพที่ ข.5



รูปที่ ข.5 จอภาพขณะใช้งานของรายการย่อย Quit

ประวัติผู้เขียน

ชื่อ นางสาววราภรณ์ วีระวิทยา
วัน เดือน ปี เกิด วันที่ 7 เดือนตุลาคม พ.ศ. 2510

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วิทยาศาสตรบัณฑิต (คณิตศาสตร์)	มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่	2531

ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

ทุนมูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสาร

ตำแหน่งและสถานที่ทำงาน

ตำแหน่ง Database Analyst
บริษัท คอนโทรลคาสต้า (ประเทศไทย) จำกัด
CDG HOUSE
202 ถ.รัชดา นางลิ้นจี่
ช่องนนทรี ยานนาวา
กรุงเทพฯ 10120