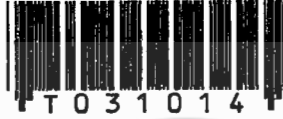


การขยายการใช้ระบบฐานข้อมูลแบบสัมพันธ์เพื่อสนับสนุนระบบผู้เชี่ยวชาญ  
EXTENSIONS OF RELATIONAL DATABASE SYSTEMS TO SUPPORT  
AN EXPERT SYSTEM



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษิตตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต  
สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ  
บัณฑิตวิทยาลัย  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
พ.ศ.2541

ISBN 974-622-138-8

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

๓ ๐ ๑

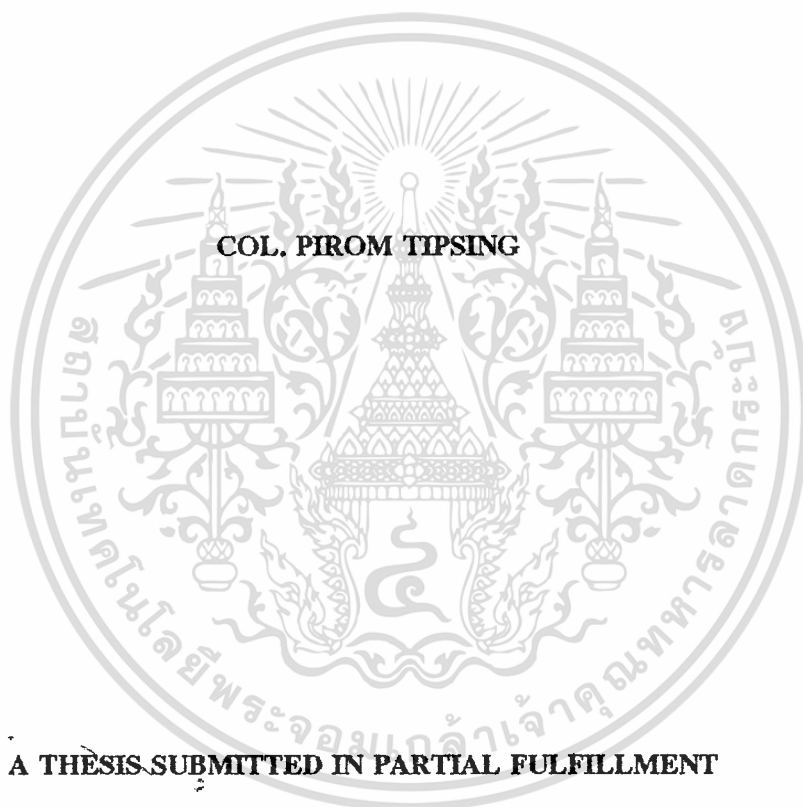
เลขหน้.....

เลขทะเบียน..... 31014

วัน, เดือน, ปี ๒๕๔๑

ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**EXTENSIONS OF RELATIONAL DATABASE SYSTEMS TO SUPPORT  
AN EXPERT SYSTEM**



**COL. PIROM TIPSING**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE  
MASTER OF SCIENCE IN COMPUTER SCIENCE AND INFORMATION  
TECHNOLOGY  
SCHOOL OF OGRADUATE STUDIES  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**1998**

**ISBN 974-622-138-8**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์                      การขยายการใช้ระบบฐานข้อมูลแบบสัมพันธ์เพื่อ  
สนับสนุนระบบผู้เชี่ยวชาญ  
นักศึกษา                                      พันเอก ภิรมย์ ทิพย์สิงห์  
อาจารย์ผู้ควบคุมวิทยานิพนธ์        ดร. เอ็น ปิ่นเงิน  
ระดับการศึกษา                            วิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์  
และเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้า  
เจ้าคุณทหารลาดกระบัง  
พ.ศ.    2541

บทคัดย่อ

ปัจจุบันการเรียกใช้ข้อมูลจากฐานข้อมูลจะใช้ข้อมูลที่ชัดเจน การกำหนดเงื่อนไขจะต้องชัดเจนและครบถ้วน แต่ปกติมนุษย์จะคิดหาเหตุผลโดยใช้ข้อมูลข่าวสารที่ไม่ค่อยมีความชัดเจน ไม่สมบูรณ์และเป็นข้อมูลที่ไม่เฉพาะเจาะจง งานวิจัยนี้ได้นำเสนอวิธีการเพิ่มความสามารถของฐานข้อมูลแบบสัมพันธ์โดยใช้แนวคิดพีชซีเซตและความเป็นไปได้ในหัวข้อ “การขยายการใช้ระบบฐานข้อมูลแบบสัมพันธ์เพื่อสนับสนุนระบบผู้เชี่ยวชาญ” ซึ่งจะช่วยให้สามารถแทนและดำเนินการกับข้อมูลข่าวสารที่ไม่ชัดเจนได้

**Thesis Title** Extensions of Relational Database Systems  
to Support an Expert System

**Student** Col. Pirom Tipping

**Thesis Advisor** Dr. Owen Pin-ngern

**Level of Study** Master of Science in Computer Science  
and Information Technology

**Department** Mathematic and Computer Science Department  
Faculty of Science  
King Mongkut's Institute of Technology  
Ladkrabang

**Year** 1998

### ABSTRACT

At the present, we retrieve precise information from databases. Satisfied conditions must be precise and complete. But human being usually reason with imprecise, incomplete, or vague information. This research deals with an extension of relational database by applying concepts of fuzzy sets and possibility, in “**Extensions of Relational Database Systems to Support an Expert System**” to allow representation and manipulation of imprecise information.

## กิตติกรรมประกาศ

ขอขอบคุณ พลเอก มนตรี สุภาพร อคิดเจ้ากรมกรมการสนเทศทหาร กองบัญชาการทหารสูงสุด ที่กรุณาสับสนุนการก่อตั้งโครงการการศึกษาระดับปริญญาโทระหว่างกองบัญชาการทหารสูงสุดกับสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ขอขอบคุณ พลตรี โกญจนาท จุณณะภาค เจ้ากรมกรมการสนเทศทหาร กองบัญชาการทหารสูงสุด ที่กรุณาสสนับสนุนและให้โอกาสข้าราชการทหารทุกท่านศึกษาหาความรู้เพิ่มเติมในทุกระดับการศึกษา

ขอขอบคุณ นาวาอากาศเอกหญิง นาดเจลิยว ทวีเพชร รองผู้อำนวยการสถาบันคอมพิวเตอร์ทหาร กรมการสนเทศทหาร ที่กรุณาให้ข้อเสนอแนะพร้อมทั้งมอบเอกสารประกอบการค้นคว้า

ขอขอบคุณ พันเอกหญิง ชมพูนุท จุฑะพุทธิ ผู้อำนวยการกองปฏิบัติการ กรมการสนเทศทหาร ที่กรุณาให้คำปรึกษาและมอบเอกสารประกอบการค้นคว้า

ขอขอบคุณ นาวาเอกหญิง วิมลวรรณ วิจิตรานันท์ นายทหารเทคนิค กรมการสนเทศทหาร ที่กรุณาคอยให้กำลังใจให้เวลาให้คำปรึกษาและให้ยืมอุปกรณ์คอมพิวเตอร์เพื่อทำวิทยานิพนธ์

ขอขอบคุณ ดร. เอื้อน ปิ่นเงิน อาจารย์ผู้ควบคุมวิทยานิพนธ์ ที่กรุณาให้คำแนะนำและให้คำปรึกษาเป็นอย่างดีตลอดเวลา ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูง

พันเอก ภิรมย์ ทิพย์สิงห์

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญภาพ.....	X
<b>บทที่</b>	
1. บทนำ.....	1
1.1 วัตถุประสงค์ของการวิจัย.....	1
1.2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.4 โครงสร้างของวิทยานิพนธ์.....	2
2. ฟังก์ชันเซตฐานข้อมูลและความสัมพันธ์.....	5
2.1 ทฤษฎีฟังก์ชันเซต.....	5
2.2 ฐานข้อมูล.....	7
2.3 ภาษาธรรมชาติและการใช้ภาษาธรรมชาติเป็นคำถาม.....	8
2.3.1 คำถามภาษา SQL และคำถามภาษาธรรมชาติ.....	8
2.3.2 การวิเคราะห์ภาษาธรรมชาติเพื่อสร้างภาษา SQL.....	9
2.3.3 การสร้างประโยคคำถามภาษา SQL จากภาษาธรรมชาติ.....	9
2.3.4 การแปลภาษาธรรมชาติเป็นภาษา SQL.....	10
2.3.5 รูปแบบของคำถามภาษาธรรมชาติที่นำมาใช้.....	12
2.3.6 หลักเกณฑ์การใช้เงื่อนไขสำหรับความสัมพันธ์.....	13
2.3.7 ขอบเขตของคำถามภาษาธรรมชาติที่นำมาใช้.....	15
2.4 ฟังก์ชันสมาชิก.....	15

สารบัญ (ต่อ)

บทที่

หน้า

2.5 ความสัมพันธ์แบบพีชชี.....	15
2.5.1 นิยามความสัมพันธ์ .....	15
2.5.2 ความสัมพันธ์ที่นำมาใช้ในระบบ.....	15
2.5.2.1 ความสัมพันธ์ PROXIMITY.....	15
2.5.2.2 ความสัมพันธ์ YOUNG.....	17
2.5.2.3 ความสัมพันธ์ OLD.....	18
2.5.2.4 ความสัมพันธ์ TALL.....	19
2.5.3 การจัดการเกี่ยวกับความสัมพันธ์.....	20
2.5.3.1 การเพิ่มความสัมพันธ์.....	20
2.5.3.1 การเก็บความสัมพันธ์.....	20
2.5.3.2 การนำความสัมพันธ์มาใช้ในระบบ.....	20
2.5.4 การประยุกต์ใช้ความสัมพันธ์ที่เป็นจำนวนเลข.....	20
2.5.4.1 การขยายค่าความสัมพันธ์.....	20
2.5.4.2 การขยายซ้ำค่าความสัมพันธ์ .....	22
2.5.4.3 เปรียบเทียบผลการหาค่าความสัมพันธ์ และผลการประยุกต์ใช้ความสัมพันธ์.....	25
2.5.4.4 การจำกัดค่าคำตอบ.....	26
<b>8. การสร้างและออกแบบระบบ FDB.....</b>	<b>28</b>
3.1 การสร้างและออกแบบฐานข้อมูล.....	28
3.1.1 ฐานข้อมูล PERSON.....	28
3.1.2 ฐานข้อมูล TEMPFDBX .....	29
3.2 การสร้างและออกแบบโปรแกรมประมวลผล.....	29
3.2.1 โปรแกรมประมวลผลหลัก.....	29
3.2.1.1 การสร้างเสริมและปรับปรุงฐานข้อมูลเสมือน.....	29
3.2.1.2 การอ่านข้อมูล.....	33
3.2.1.3 การแจกจ่ายงาน.....	33
3.2.1.4 การจำกัดค่าคำตอบ.....	33

## สารบัญ (ต่อ)

บทที่

หน้า

3.2.1.5 การออกรายงาน.....	34
3.2.2 โปรแกรมประมวลผลภาษาธรรมชาติ.....	34
3.2.2.1 การแปลคำตามภาษาธรรมชาติ ประเภทที่มีฟิชชีเป็นจำนวนเลข.....	34
3.2.2.2 การแปลคำตามภาษาธรรมชาติ ประเภทที่มีฟิชชีเป็นข้อความ.....	38
3.2.3 โปรแกรมประมวลผลหาค่าตัวเลข.....	39
3.2.3.1 การหาค่าส่วนเงื่อนไขความสัมพันธ์ปกติ.....	40
3.2.3.2 การหาค่าส่วนเงื่อนไขขยายความสัมพันธ์.....	40
3.2.3.3 การหาค่าส่วนเงื่อนไขขยายซ้ำความสัมพันธ์.....	43
3.2.3.4 การตรวจสอบเงื่อนไขการจำกัดค่าคำตอบ.....	47
3.2.4 โปรแกรมประมวลผลข้อความ.....	47
3.2.4.1 การหาค่าส่วนเงื่อนไขความสัมพันธ์ข้อความ ที่ค่าแอดตริบิวต์มีข้อความเดียว.....	48
3.2.4.2 การหาค่าส่วนเงื่อนไขความสัมพันธ์ข้อความ ที่ค่าแอดตริบิวต์มีหลายข้อความต่อกัน.....	48
3.3 Header File.....	48
4. การทดลองและผลการทดลอง.....	49
4.1 การทดลองหาค่าความสัมพันธ์เกี่ยวกับจำนวนเลข.....	49
4.1.1 การทดลองหาค่าความสัมพันธ์ในเงื่อนไขปกติ.....	49
4.1.2 การทดลองหาค่าความสัมพันธ์ในเงื่อนไขขยาย.....	52
4.1.3 การทดลองหาค่าความสัมพันธ์ในเงื่อนไขขยายซ้ำ.....	56
4.1.4 การทดลองการจำกัดค่าคำตอบ.....	59
4.2 การทดลองหาค่าความสัมพันธ์เกี่ยวกับข้อความ.....	63
4.2.1 การทดลองหาค่าความสัมพันธ์ข้อความที่แอดตริบิวต์มี ข้อความเดียว.....	63

## สารบัญ (ต่อ)

บทที่

หน้า

4.2.2 การทดลองหาค่าความสัมพันธ์ข้อความที่แอดรีบีวดีมี หลายข้อความต่อกัน.....	64
<b>5. สรุปผลการทดลองและข้อเสนอแนะ.....</b>	<b>65</b>
5.1 บทสรุป.....	65
5.2 ข้อจำกัดของระบบ.....	65
5.3 ข้อเสนอแนะ.....	66
<b>เอกสารอ้างอิง.....</b>	<b>67</b>
<b>ภาคผนวก .....</b>	<b>68</b>
<b>ภาคผนวก ก. ตัวอย่าง โปรแกรมในระบบ Fuzzy Data Base (FDB).....</b>	<b>69</b>
ก.1 โปรแกรมประมวลผลหลัก.....	70
ก.2 HEADER FILE.....	137
<b>ภาคผนวก ข. สำเนาบทความวิจัยที่ได้รับการตีพิมพ์.....</b>	<b>146</b>
<b>ประวัติผู้เขียน.....</b>	<b>160</b>

## สารบัญตาราง

ตารางที่	หน้า
1 แสดงความสัมพันธ์ของ PERSON ใน BDB.....	7
2 แสดงความสัมพันธ์ของ YOUNG ใน VDB.....	8
3 แสดงเงื่อนไขสำหรับความสัมพันธ์จำนวนเลขที่สมบูรณ์ในตัวเอง.....	13
4 แสดงเงื่อนไขสำหรับความสัมพันธ์จำนวนเลขที่ไม่สมบูรณ์ในตัวเอง.....	14
5 แสดงเงื่อนไขสำหรับความสัมพันธ์ข้อความที่ค่าแอดทริบิวต์มีข้อความเดียว.....	14
6 แสดงเงื่อนไขสำหรับความสัมพันธ์ข้อความที่ค่าแอดทริบิวต์ มีข้อความหลายข้อความต่อกัน.....	15
7 แสดงรูปแบบการขยายความสัมพันธ์ที่เป็นจำนวนเลข .....	20
8 แสดงรูปแบบการขยายข้อความความสัมพันธ์ที่เป็นจำนวนเลข .....	22
9 แสดงระดับการเป็นสมาชิกที่ได้จากการหาค่าความสัมพันธ์ การขยายค่าความสัมพันธ์ และการขยายข้อความความสัมพันธ์ YOUNG.....	25
10 แสดงรูปแบบการจำกัดค่าคำตอบที่เป็นจำนวนเลข .....	26
11 แสดงความสัมพันธ์ YOUNG ใน VDB.....	30
12 แสดงความสัมพันธ์ OLD ใน VDB.....	30
13 แสดงความสัมพันธ์ TALL ใน VDB.....	31
14 แสดงความสัมพันธ์ PROXIMITY ของ age ใน VDB.....	31
15 แสดงความสัมพันธ์ PROXIMITY ของ height ใน VDB.....	32
16 แสดงความสัมพันธ์ Name ใน VDB.....	32
17 แสดงความสัมพันธ์ Province ใน VDB.....	33
18 แสดงคำตอบจากคำสั่ง "list the man from person whose age is proximity 30".....	49
19 แสดงคำตอบจากคำสั่ง "list the men from person whose height is tall".....	50
20 แสดงคำตอบจากคำสั่ง "list the man from person whose age is young".....	51
21 แสดงคำตอบจากคำสั่ง "list the man from person whose age is old".....	51
22 แสดงคำตอบจากคำสั่ง "find the people from person who is young".....	52
23 แสดงคำตอบจากคำสั่ง "find the people from person that age is not young".....	53
24 แสดงคำตอบจากคำสั่ง "find the people from person that age is very young".....	53
25 แสดงคำตอบจากคำสั่ง "find the people from person that age is not old".....	54

## สารบัญตาราง (ต่อ)

ตารางที่		หน้า
26	แสดงคำตอบจากคำสั่ง "find the people from person that age is very old".....	55
27	แสดงคำตอบจากคำสั่ง "find the people from person that height is not tall".....	55
28	แสดงคำตอบจากคำสั่ง "find the people from person that height is very tall".....	56
29	แสดงคำตอบจากคำสั่ง "find the people from person that age is very very young".....	57
30	แสดงคำตอบจากคำสั่ง "find the people from person that age is very not young".....	57
31	แสดงคำตอบจากคำสั่ง "find the people from person that age is not very young".....	58
32	แสดงคำตอบจากคำสั่ง "find the people from person that age is not not young".....	59
33	แสดงคำตอบจากคำสั่ง "list the man from person whose age is young with 0.5".....	59
34	แสดงคำตอบจากคำสั่ง "find the man from person whose age is very young with 0.5" ...	60
35	แสดงคำตอบจากคำสั่ง "find the man from person whose age is not young with 0.3".....	60
36	แสดงคำตอบจากคำสั่ง "find man from person whose age is very very young with 0.01".....	61
37	แสดงคำตอบจากคำสั่ง "find people from person that age is very not young with 0.6".....	61
38	แสดงคำตอบจากคำสั่ง "find man from person that age is not very young with 0.5".....	62
39	แสดงคำตอบจากคำสั่ง "find people from person that age is not not young with 0.5".....	62
40	แสดงคำตอบจากคำสั่ง "list the people from person whose name is 'มงคล'".....	63
41	แสดงคำตอบจากคำสั่ง "list the people from person whose name is 'กนก'".....	63
42	แสดงคำตอบจากคำสั่ง "select name from person who lives in 'กรุงเทพ'".....	64
43	แสดงคำตอบจากคำสั่ง "select name from person who lives in 'อ่างทอง'".....	64

## สารบัญญภาพ

ภาพที่	หน้า
1 แสดงกราฟระดับการเป็นสมาชิกของฟิชชีเซ็ด.....	6
2 แสดงกราฟของความสัมพันธ์ PROXIMITY.....	16
3 แสดงกราฟของความสัมพันธ์ YOUNG.....	17
4 แสดงกราฟของความสัมพันธ์ OLD.....	18
5 แสดงกราฟของความสัมพันธ์ TALL.....	19
6 แสดงกราฟของการขยายความสัมพันธ์ YOUNG ด้วย VERY.....	21
7 แสดงกราฟของการขยายความสัมพันธ์ YOUNG ด้วย NOT.....	21
8 แสดงกราฟของการขยายซ้ำ VERY YOUNG ด้วย VERY.....	22
9 แสดงกราฟของการขยายซ้ำ NOT YOUNG ด้วย VERY.....	23
10 แสดงกราฟของการขยายซ้ำ VERY YOUNG ด้วย NOT.....	24
11 แสดงกราฟของการขยายซ้ำ NOT YOUNG ด้วย NOT.....	24
12 แสดงกราฟระดับการเป็นสมาชิกที่ได้จากการหาค่าความสัมพันธ์ การขยายค่าความสัมพันธ์และการขยายซ้ำค่าความสัมพันธ์ YOUNG.....	26
13 แสดงกราฟของการจำกัดค่าคำตอบความสัมพันธ์ YOUNG ด้วย WITH.....	27
14 โครงสร้างพื้นฐานของระบบ FDB เพื่อสนับสนุนระบบผู้เชี่ยวชาญ.....	28
15 แสดงกราฟของความสัมพันธ์ YOUNG ที่อายุ 28 ปี.....	40
16 แสดงกราฟของ VERY YOUNG ที่อายุ 28 ปี.....	41
17 แสดงกราฟของ NOT YOUNG ที่อายุ 28 ปี.....	42
18 แสดงกราฟของ VERY VERY YOUNG ที่อายุ 28 ปี.....	43
19 แสดงกราฟของ VERY NOT YOUNG ที่อายุ 28 ปี.....	44
20 แสดงกราฟของ NOT VERY YOUNG ที่อายุ 28 ปี.....	45
21 แสดงกราฟของ NOT NOT YOUNG ที่อายุ 28 ปี.....	46
22 แสดงกราฟของ YOUNG WITH 0.5 ที่อายุ 28 ปี.....	47

# บทที่ 1

## บทนำ

### 1. บทนำ

การวิจัยในด้านวิทยาการคอมพิวเตอร์ เช่น การสร้างระบบผู้เชี่ยวชาญ จำเป็นต้องใช้สารสนเทศที่มีข้อเท็จจริงอย่างถูกต้องครบถ้วนเพื่อจำลองการหาเหตุผลของมนุษย์ แต่เมื่อต้องนำสารสนเทศที่เกี่ยวข้องมากมายมาอ้างอิงในการประมวลผล ย่อมทำให้มีสารสนเทศบางส่วนมีความไม่สมบูรณ์หรือมีความคลุมเครือเกิดขึ้น เช่น ข้อสนเทศที่ว่า "ยุคเศรษฐกิจชะลอตัวทำให้ประชาชนจนลง" คำว่า "จน" เป็นคำที่คลุมเครือขาดความชัดเจนว่าจนอะไร จนเท่าไร และจนอย่างไร ทำให้ไม่สามารถประมวลผลได้ชัดเจนตามต้องการ

การวิจัยเรื่อง "การขยายการใช้ระบบฐานข้อมูลแบบสัมพันธ์เพื่อสนับสนุนระบบผู้เชี่ยวชาญ" จึงได้จำลอง Fuzzy Data Base (FDB) ขึ้นมาเป็นเครื่องมือในการแก้ปัญหาดังกล่าว

FDB ออกแบบมาเพื่อให้สามารถแสดงและกระทำต่อสารสนเทศที่ไม่สมบูรณ์หรือคลุมเครือ โดยสร้างจากพื้นฐานของระบบฐานข้อมูลแบบสัมพันธ์ผนวกกับทฤษฎีฟัซซีเซตและการกระจายที่เป็นไปได้ จึงทำให้สามารถประมวลผลสารสนเทศที่มีความคลุมเครือตามความต้องการ

FDB เป็นระบบสนับสนุนผู้เชี่ยวชาญที่สามารถใช้ภาษาคำสั่งภาษารธรรมชาติ (natural-language-like) เหมือนภาษาอังกฤษในการถามเพื่อค้นหาคำตอบจากฐานข้อมูลเพื่อให้สามารถแสดงและกระทำต่อสารสนเทศที่ไม่สมบูรณ์หรือมีความคลุมเครือ ทำให้การใช้งานง่ายขึ้น

#### 1.1 วัตถุประสงค์ของการวิจัย

การวิจัยเรื่อง "การขยายการใช้ระบบฐานข้อมูลแบบสัมพันธ์เพื่อสนับสนุนระบบผู้เชี่ยวชาญ" มีวัตถุประสงค์ดังนี้ (1) เพื่อแก้ปัญหาการประมวลผลข้อมูลที่ไม่สมบูรณ์หรือมีความคลุมเครือซึ่งไม่สามารถประมวลผลโดยระบบปกติได้ (2) เพื่อพัฒนาเทคนิคในการนำเอาภาษารธรรมชาติมาใช้สั่งงานคอมพิวเตอร์ (3) เพื่อศึกษาเทคนิคในการเขียนโปรแกรมให้สามารถรับสมการแทนระดับการเป็นสมาชิกเข้าไปใช้ในระบบโดยไม่ต้องเก็บสมการไว้ใน logic ของโปรแกรม (4) เพื่อแสวงหาแนวทางนำเอาทฤษฎีฟัซซีเซตมาใช้ให้เป็นประโยชน์และเพื่อขยายให้ระบบฐานข้อมูลที่มีอยู่ให้มีประโยชน์มากขึ้น

#### 1.2 ทฤษฎีและหลักการที่เกี่ยวข้อง

**ทฤษฎีฟัซซีเซต** ทฤษฎีฟัซซีเซตกล่าวถึงเซตของสิ่งใด ๆ ในยูนิเวอร์สที่อ้างถึง ถ้าสมาชิกอยู่ในเซตนั้นและมีค่าถูกต้องสมบูรณ์เราให้ระดับค่าการเป็นสมาชิกเป็น 1 ถ้าสมาชิกไม่เอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่ในเซตนั้นเราให้ระดับค่าการเป็นสมาชิกเป็น 0 และสมาชิกอื่น ๆ ของเซตนั้นเราให้ระดับค่าการเป็นสมาชิกอยู่ในช่วง [0,1]

**ความสัมพันธ์และคุณสมบัติของความสัมพันธ์แบบต่าง ๆ คือ**

**ความสัมพันธ์ PROXIMITY หรือความสัมพันธ์ใกล้เคียง** ใช้คำนวณสิ่งที่นับเป็นจำนวนเลขได้ว่ามีระดับค่าใกล้เคียงกับจำนวนเลขที่กำหนดมากน้อยเท่าใด

**ความสัมพันธ์ YOUNG หรือความสัมพันธ์ปฐมวัย** เป็นการหาว่าอายุเท่าใดจึงจะอยู่ในระดับหรือเกณฑ์หนุ่ม (YOUNG) หนุ่มมาก ไม่หนุ่ม หนุ่มมากๆ ไม่หนุ่มมาก หนุ่มไม่มาก (VERY YOUNG, NOT YOUNG, VERY VERY YOUNG, NOT VERY YOUNG, VERY NOT YOUNG) เป็นต้น

**ความสัมพันธ์ OLD หรือความสัมพันธ์ปัจฉิมวัย** เป็นการหาว่าอายุเท่าใดจึงจะอยู่ในระดับหรือเกณฑ์แก่ (OLD) แก่มาก ไม่แก่ แก่มากๆ ไม่แก่มาก แก่ไม่มาก (VERY OLD, NOT OLD, VERY VERY OLD, NOT VERY OLD, VERY NOT OLD) เป็นต้น

**ความสัมพันธ์ TALL หรือความสัมพันธ์สูง** เป็นการหาว่าความสูงเท่าใดจึงจะอยู่ในระดับหรือเกณฑ์สูง (TALL) สูงมาก ไม่สูง สูงมากๆ ไม่สูงมาก สูงไม่มาก (VERY TALL, NOT TALL, VERY VERY TALL, NOT VERY TALL, VERY NOT TALL) เป็นต้น

### 1.3 ประโยชน์ที่คาดว่าจะได้รับ

ระบบนี้ทำให้ใช้คอมพิวเตอร์ได้ง่ายขึ้นเพราะสามารถใช้ภาษาธรรมชาติที่มนุษย์ใช้กันอยู่สั่งให้คอมพิวเตอร์ทำงานได้ ทำให้คอมพิวเตอร์สามารถประมวลผลต่อข้อมูลที่มีความคลุมเครือและไม่ชัดเจนได้ ตามมุมมองของผู้ใช้ ตามข้อมูลที่เปลี่ยนแปลงไปตามขอบเขตใหม่ที่ต้องการโดยไม่ต้องเขียนโปรแกรมเพิ่มเติมเมื่อสิ่งเหล่านี้เปลี่ยนแปลงไปจากเดิม

ประโยชน์ทางอ้อมคือระบบนี้จะช่วยเพิ่มความคุ้มค่าให้แก่ข้อมูลจำนวนมากที่มีอยู่ในระบบฐานข้อมูลแบบสัมพันธ์ในหน่วยงานปกติให้สามารถใช้งานได้เพิ่มขึ้นและฐานข้อมูลที่สร้างขึ้นในระบบนี้สามารถนำไปใช้กับโปรแกรมปกติในงานอื่น ๆ ได้

ประโยชน์อีกอย่างหนึ่งคือจะได้ระบบใหม่ขึ้นมาอีกระบบหนึ่งที่สามารถเป็นต้นแบบเพื่อปรับปรุงไปใช้งานในสาขาต่าง ๆ ได้

### 1.4 โครงสร้างของวิทยานิพนธ์

**บทที่ 2 พืชซีเซ็ตฐานข้อมูลและความสัมพันธ์** กล่าวถึงทฤษฎีและองค์ประกอบต่าง ๆ ที่นำมาใช้ในระบบ FDB อันได้แก่

**ทฤษฎีพืชซีเซ็ต** พูดถึงที่มาและแนวความคิดอย่างย่อ ๆ ของพืชซีเซ็ตและทฤษฎีพืชซีเซ็ต รวมทั้งการกำหนดแนวทางที่จะนำพืชซีเซ็ตมาใช้ในระบบ FDB ม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ฐานข้อมูล** การนำระบบฐานข้อมูลแบบสัมพันธ์มาใช้ในระบบ FDB โดยใช้ฐานข้อมูลเดิมที่มีอยู่แล้วในหน่วยงานเป็นฐานข้อมูลหลักและสร้างฐานข้อมูลเสริมขึ้นมาประกอบการทำงาน

**ภาษาธรรมชาติและการใช้ภาษาธรรมชาติเป็นคำถาม** การที่จะใช้ภาษาธรรมชาติเป็นคำถามเพื่อค้นหาคำตอบจากฐานข้อมูลนั้น ต้องมีขั้นตอนต่าง ๆ เช่น การวิเคราะห์ภาษาธรรมชาติและการแปลภาษาธรรมชาติเป็นชุดภาษา SQL (Structured Query Language) เพื่อนำเอาภาษา SQL ไปกระทำต่อข้อมูลและฐานข้อมูลอีกที่หนึ่งเพื่อให้ได้คำตอบที่ต้องการ

**ฟังก์ชันสมาชิก** อันเป็นตัวกำหนดความสัมพันธ์แบบต่าง ๆ ของข้อมูลโดยนำแนวความคิดมาจากพีชชีเซต ผลลัพธ์ที่ได้จากการคำนวณจากฟังก์ชันสมาชิกเรียกว่าระดับค่าการเป็นสมาชิก

**ความสัมพันธ์** ความสัมพันธ์แบบต่าง ๆ ที่นำมาใช้ในระบบอันมีความสัมพันธ์ PROXIMITY, YOUNG, OLD และ TALL พร้อมรายละเอียดของแต่ละความสัมพันธ์ รวมทั้งการเก็บความสัมพันธ์และการนำความสัมพันธ์มาใช้ระบบ

**บทที่ 3 การสร้างและการออกแบบระบบ FDB** การสร้างและการออกแบบระบบ FDB มีแนวทางในการสร้างและออกแบบดังนี้

**การสร้างและการออกแบบฐานข้อมูล** ฐานข้อมูลแบ่งออกเป็น 2 ชนิด คือฐานข้อมูลหลักและฐานข้อมูลเสริม ฐานข้อมูลทั้งสองชนิดใช้ฐานข้อมูลแบบสัมพันธ์ (ORACLE) โดยฐานข้อมูลหลักใช้ฐานข้อมูลเดิมที่ใช้ในภารกิจต่าง ๆ ของหน่วยงานตามปกติ ในการประมวลผลเพื่อหาคำตอบจะไม่มีเปลี่ยนแปลงใด ๆ ในฐานข้อมูลหลัก สำหรับฐานข้อมูลเสริมโปรแกรมจะสร้างเสริมขึ้นมาชั่วคราวเพื่อเก็บความสัมพันธ์ต่าง ๆ ตามความต้องการตอบคำถาม ฐานข้อมูลเสริมสามารถเก็บไว้ใช้ในภารกิจอื่น ๆ ได้หากต้องการ

**การสร้างและการออกแบบโปรแกรม** โปรแกรมแบ่งออกเป็น 4 โปรแกรม คือโปรแกรมประมวลผลหลักทำหน้าที่เกี่ยวกับฐานข้อมูลและการออกรายงาน โปรแกรมประมวลผลภาษาธรรมชาติทำหน้าที่เกี่ยวกับรับคำถามภาษาธรรมชาติแล้วแปลเป็นภาษา SQL โปรแกรมประมวลผลตัวเลขทำหน้าที่ประมวลผลสิ่งที่นับเป็นจำนวนเลขได้และโปรแกรมประมวลผลข้อความทำหน้าที่ประมวลผลสิ่งที่นับเป็นจำนวนเลขไม่ได้ เนื่องจากโปรแกรมไม่ได้เก็บชื่อฐานข้อมูลหลักฐานข้อมูลเสริมหรือแอตทริบิวต์ของฐานข้อมูลไว้ในโปรแกรมจึงทำให้มีความยืดหยุ่นในการใช้งานสูง

**บทที่ 4 การทดลองและผลการทดลอง** การทดลองและผลการทดลองจากระบบที่สร้างขึ้น การทดลองแบ่งออกเป็น 2 ส่วนใหญ่ ๆ คือ การทดลองหาค่าความสัมพันธ์เกี่ยวกับ

เอ็กซิทเป็นเอ็กซิทพลังงานไฟฟ้าที่รับกระแสไฟฟ้าจากเสาอากาศ เมื่อผู้พูดเห็นไฟเขียวจะขึ้นด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนเลขและการทดลองหาค่าความสัมพันธ์เกี่ยวกับข้อความ

**การทดลองหาค่าความสัมพันธ์เกี่ยวกับจำนวนเลข** คือการทดลองหาค่าสิ่งที่นับเป็นจำนวนเลขได้ เช่น การทดลองหาค่าความสัมพันธ์ในเงื่อนไขปกติของความสัมพันธ์ PROXIMITY, YOUNG, OLD และ TALL การทดลองหาค่าความสัมพันธ์ในเงื่อนไขขยายโดยใช้คำ "NOT" และ "VERY" การทดลองหาค่าความสัมพันธ์ในเงื่อนไขขยายซ้ำโดยใช้คำ "NOT" และ "VERY" การทดลองจำกัดคำตอบโดยใช้คำ "WITH" เพื่อให้ได้คำตอบค่าความสัมพันธ์บางค่าที่อยู่ในขอบเขตที่ต้องการ

**การทดลองหาค่าความสัมพันธ์เกี่ยวกับข้อความ** คือการทดลองหาค่าสิ่งที่นับเป็นจำนวนเลขไม่ได้ เช่น การทดลองหาค่าความสัมพันธ์ข้อความที่แอดตริบิวต์มีเพียง "ข้อความเดียว" และการทดลองหาค่าความสัมพันธ์ข้อความที่แอดตริบิวต์มี "หลายข้อความต่อกัน"

**บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ** สรุปถึงความสำเร็จของวิทยานิพนธ์ที่สามารถนำเอาภาษารวมชาติและพีซีซีทีไปใช้ในการสนับสนุนระบบผู้เชี่ยวชาญที่มีฐานข้อมูลแบบสัมพันธ์และสามารถประมวลผลสารสนเทศที่ไม่สมบูรณ์หรือคลุมเครือได้ นอกจากนี้ยังได้พูดถึงขีดจำกัดของระบบพร้อมข้อเสนอแนะเพื่อเป็นแนวทางในการพัฒนาต่อไป

## บทที่ 2

### ฟัซซีเซต ฐานข้อมูล และความสัมพันธ์

#### 2. ฟัซซีเซต

ทฤษฎีและองค์ประกอบ ที่นำมาประกอบการพัฒนาาระบบ คือ ทฤษฎีฟัซซีเซต ฐานข้อมูล ภาษาธรรมชาติและความสัมพันธ์ต่าง ๆ มีรายละเอียดดังต่อไปนี้

#### 2.1 ทฤษฎีฟัซซีเซต

##### 2.1.1 ความเป็นมาและความหมายของทฤษฎีฟัซซีเซต

ในปี ค.ศ. 1962 ศาสตราจารย์ ลอตฟี เอ. ซาเดห์ (Lotfi A. Zadeh) ได้นำเสนอบทความเกี่ยวกับฟัซซีเซตชื่อ "mathematics of fuzzy or cloudy quantities which are not describable in terms of probability distributions" และต่อมาในปี ค.ศ. 1965 ศาสตราจารย์ ซาเดห์ ได้เสนอบทความเพิ่มเติมเป็นทฤษฎีฟัซซีเซต (The Theory of FUZZY SET) ทั้งฟัซซีเซตและทฤษฎีฟัซซีเซตกล่าวพอสรุปได้ดังนี้

2.1.1.1 ฟัซซีเซต เป็นเซตของความถูกต้องไม่แน่นอน (imprecision) ชนิดหนึ่งที่ได้จากการรวบรวมจากบรรดาสมาชิกแล้วจัดเข้าไว้เป็นพวก ๆ (class) แต่ละพวกมีขอบเขตไม่แน่นอน เราเรียกพวกเหล่านี้ว่า "ฟัซซีเซต"

2.1.1.2 ทฤษฎีฟัซซีเซต ให้สับเซต A ของยูนิเวิร์ส X ที่กล่าวถึง เป็นระดับค่า (gradual) ต่าง ๆ ที่เชื่อมระหว่าง full membership และ no membership และระดับค่าเหล่านี้มีการกำหนดกฎเกณฑ์ที่แน่นอน หรือ Fuzzy Relation R from X เป็น fuzzy sets อาจนิยามได้ดังนี้

$$A = \{ x \mid x \in X \} , \quad \mu_A : X \rightarrow [0, 1]$$

2.1.2 การนำฟัซซีมาพัฒนาาระบบ FDB ในชีวิตประจำวันของมนุษย์เรามักจะพบเห็นประโยคที่มีลักษณะคล้ายคลึงกับประโยคต่อไปนี้ "A เป็นเซตของคนหนุ่มในเมือง X"

จากประโยคดังกล่าวนี้ทำให้เกิดแนวทางในการนำฟัซซีมาพัฒนาาระบบ FDB คือ

2.1.2.1 กำหนดกฎเกณฑ์ฟัซซี ใช้ความสัมพันธ์และฟังก์ชันทางคณิตศาสตร์เป็นตัวกำหนดกฎเกณฑ์

2.1.2.2 กำหนดขอบเขตของฟัซซี ฟัซซีเซตมีการกำหนดขอบเขตไม่แน่นอน เช่น คนหนุ่ม (young) การกำหนดว่าอายุเท่าใดเป็นคนหนุ่มไม่แน่นอนมีความคลุมเครือ (fuzzy) ในขณะที่ยูนิเวิร์สที่กล่าวถึงคือเมือง X มีการกำหนดขอบเขตที่แน่นอนว่าต้องเป็นเมือง X เท่านั้น จึงจะอยู่ในกฎเกณฑ์

2.1.2.3 กำหนดค่าฟัซซีเซต เพื่อให้เห็นความแตกต่างระหว่างเซตธรรมดาเกี่ยวกับฟัซซีเซต จึงนำการกำหนดค่าความจริงของเซตธรรมดามาก้าวในที่นี่ด้วยคือ ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.2.3.1 การกำหนดค่าความจริงในเซตธรรมดา ในเซตธรรมดา

กำหนดค่าความจริงไว้ 2 ค่าคือ

2.1.2.3.1.1 สิ่งที่มีค่าเป็นจริง(True)ให้ค่าความจริงเป็น T หรือ 1

2.1.2.3.1.2 สิ่งที่มีค่าเป็นเท็จ(Fault)ให้ค่าความจริงเป็น F หรือ 0

### 2.1.2.3.2 การกำหนดค่าความจริงในฟัซซีเซต ในฟัซซีเซตกำหนด

ค่าความจริงไว้ 3 อย่าง คือ

2.1.2.3.2.1 สิ่งที่ไม่เป็นสมาชิกของเซต ให้ระดับค่าความจริงเป็น F (Fault) หรือ 0

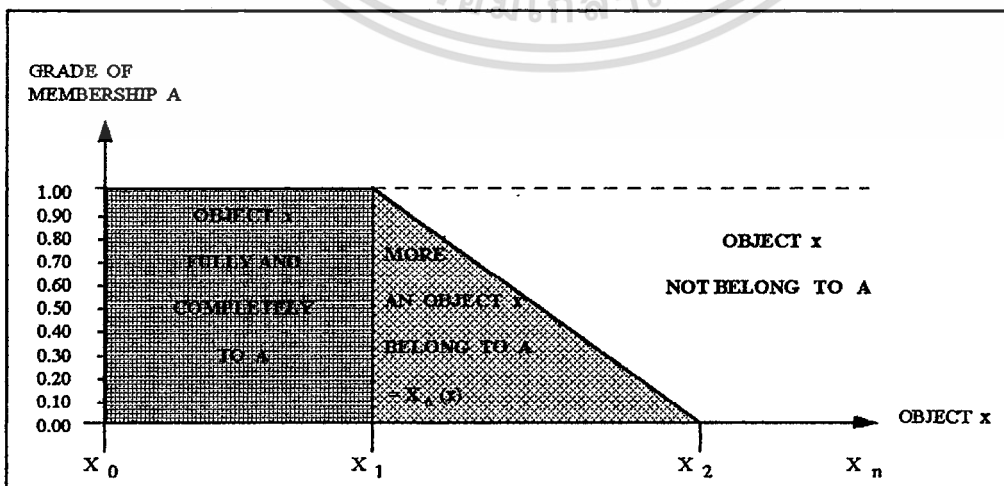
2.1.2.3.2.2 สิ่งที่เป็นสมาชิกของเซตถูกต้องสมบูรณ์ ให้ระดับค่าความจริงเป็นจริง T (True) หรือ 1

2.1.2.3.2.3 สิ่งที่เป็นสมาชิกของเซตถูกต้องไม่สมบูรณ์ ให้ระดับค่าความจริงเป็นความคลุมเครือหรือฟัซซีหรือให้ระดับค่าความจริงอยู่ระหว่าง 0 กับ 1

2.1.2.4 กำหนดฟังก์ชันระดับค่าการเป็นสมาชิก ในการหาค่าในฟัซซีเราให้บรรดาสมาชิกที่มีค่าถูกต้องสมบูรณ์ในเซต A มีระดับค่าการเป็นสมาชิกเป็น 1 ในขณะที่ สมาชิกที่ไม่อยู่ในเซต A เราให้ระดับค่าการเป็นสมาชิกเป็น 0 และให้ค่าสมาชิกอื่น ๆ ภายใต้เซต A ให้มีค่าระหว่าง 0 ถึง 1 โดยกำหนดระดับค่าการเป็นสมาชิกเป็น  $X_A(x)$  การกำหนดระดับค่าการเป็นสมาชิกของสมาชิกทั้งหมดแสดงได้ดังภาพที่ 1

ภาพที่ 1

แสดงกราฟระดับการเป็นสมาชิกของฟัซซีเซต



ภาพที่ 1 แสดงให้เห็นว่าสมาชิกของเซต A ที่ถูกต้องสมบูรณ์ (ตั้งแต่  $X_0$  ถึง  $X_1$ ) จะมีเอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาด้านนี้ เมื่ออยู่ภายใต้เงื่อนไขประการด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับค่าการเป็นสมาชิกเท่ากับ 1 สมาชิกของเซต A ที่ถูกต้องไม่สมบูรณ์ (มากกว่า  $X_1$  ถึง  $X_2$ ) จะมีระดับค่าการเป็นสมาชิกเท่ากับ  $X_A(x)$  สมาชิกอื่น ๆ (มากกว่า  $X_2$  ถึง  $X_n$ ) ที่ไม่เป็นสมาชิกของ A จะมีระดับค่าการเป็นสมาชิกเท่ากับ 0

## 2.2 ฐานข้อมูล

ฐานข้อมูลในระบบที่นำเสนอแบ่งออกเป็น 2 ชนิดคือ ฐานข้อมูลหลักและฐานข้อมูลเสมือน ฐานข้อมูลทั้งสองชนิดทำหน้าที่ดังนี้

**2.2.1 ฐานข้อมูลหลัก (Basic Data Base หรือ BDB)** คือฐานข้อมูลแบบสัมพันธ์เดิมที่ใช้อยู่ในภารกิจของหน่วยงานตามปกติ ข้อมูลเหล่านี้จะไม่มีการเปลี่ยนแปลงตลอดเวลาการใช้งานของผู้ใช้ ดังนั้นการกระทำใด ๆ เพื่อหาผลลัพธ์สุดท้ายในระบบ FDB ทำได้โดยไม่เปลี่ยนแปลงข้อเท็จจริงในฐานข้อมูลหลัก ตัวอย่างฐานข้อมูลหลักสามารถแสดงได้ดังตารางที่ 1

แสดงความสัมพันธ์ของ PERSON ใน BDB

Name	Age	Height	Province	Hair Color	Smart
กนก	25	152	ลพบุรี, ตรัง	0.8/brown + 0.6/black	0.6
ปัญญา	28	155	อ่างทอง	0.5/brown + 0.4/red	0.7
มงคล	30	160	-	0.3/blond + 0.7/red	0.4
ราตรี	45	167	นนทบุรี, สงขลา	1/brown	0.9
อนันต์	50	178	กรุงเทพฯ, ชลบุรี, ตาก	0.4/brown + 0.9/red	0.2

ตารางที่ 1 แสดงการเก็บค่าต่าง ๆ ในฐานข้อมูล PERSON เช่น Name เก็บเป็นข้อความเดียว Age และ Height เก็บเป็นเลขจำนวนเต็มสำหรับ Province เก็บเป็นข้อความหลายข้อความต่อกัน Hair Color เก็บค่ากระจายที่เป็นไปได้ของสีผม มีความหมายว่าใครมีผมสีอะไรบ้างก็สีแต่ละสีมีค่าอยู่ในช่วง [0,1] Smart เก็บค่าความฉลาดซึ่งมีค่าอยู่ในช่วง [0,1] ทั้ง Hair Color และ Smart เป็นข้อมูล BDB เดิมที่ไม่ได้นำมาใช้ในการวิจัย

**2.2.2 ฐานข้อมูลเสมือน (Virtual Data Base หรือ VDB)** ใช้เก็บรวบรวมความสัมพันธ์หรือฟังก์ชันต่าง ๆ (เช่น ความใกล้เคียง) ซึ่งสิ่งเหล่านี้ "อธิบาย" ถึงวิธีการคำนวณระดับของการใช้ค่าข้อมูลเพื่อตอบคำถามของผู้ใช้ นิยามของฐานข้อมูลส่วนนี้สามารถใช้เพื่อสะท้อนให้เห็นความต้องการคำตอบของผู้ใช้แต่ละคน ตัวอย่างฐานข้อมูลเสมือนแสดงไว้ในตารางที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับผูกขาดเห็นแก่ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2  
แสดงความสัมพันธ์ของ YOUNG ใน VDB

Name	Age	m <sub>YOUNG</sub>
กนก	25	1.00
ปัญญา	28	0.74
มงคล	30	0.50
ราตรี	45	0.06
อนันต์	50	0.04

ตารางที่ 2 แสดงการเก็บค่าต่าง ๆ ในฐานข้อมูล VDB เช่น Name และ Age เป็นข้อมูลเดิมที่นำมาจากฐานข้อมูลหลัก ส่วน m<sub>YOUNG</sub> เป็นข้อมูลที่ไม่มีอยู่ในฐานข้อมูลหลักแต่เกิดจากการคำนวณของระบบ ค่าของ m<sub>YOUNG</sub> จะเปลี่ยนแปลงตลอดเวลาการใช้งานขึ้นอยู่กับความต้องการของผู้ใช้ ถ้าผู้ใช้มีความเห็นไม่สอดคล้องกับข้อมูลที่มีอยู่ก็สามารถเปลี่ยนได้ตามความเห็นของผู้ใช้เอง จึงถือได้ว่า m<sub>YOUNG</sub> เป็นข้อมูลที่มีความคลุมเครือหรือไม่สมบูรณ์

### 2.3 ภาษาธรรมชาติและการใช้ภาษาธรรมชาติเป็นคำถาม

ตามปกติแล้วการเข้าถึงระบบฐานข้อมูลแบบสัมพันธ์ ต้องใช้ภาษาฐานข้อมูลภาษาใดภาษาหนึ่ง เช่น ใช้ภาษา SQL จึงจะเข้าถึงระบบฐานข้อมูลแบบสัมพันธ์ได้ แต่การใช้ภาษา SQL ในบางครั้งก็ไม่ได้สิ่งที่เราต้องการ ถ้าสิ่งที่เราต้องการมีความสลับซับซ้อนและข้อมูลที่มีความคลุมเครือหรือไม่สมบูรณ์แล้วภาษา SQL ช่วยเราไม่ได้เลย

**2.3.1 คำถามภาษา SQL และคำถามภาษาธรรมชาติ** คำถามเป็นสิ่งที่ผู้ใช้ใช้เพื่อค้นหาคำตอบจากฐานข้อมูล คำถามจะใช้ภาษาอะไรก็ได้ที่สามารถติดต่อกับฐานข้อมูลได้ เช่น ภาษา SQL

**2.3.1.1 คำถามภาษา SQL** รูปแบบของภาษา SQL เช่น คำถามมีรูปแบบที่จำกัด คือ SELECT-FROM-WHERE และสิ่งที่นำมาถามนั้นจะต้องเป็นชื่อแอตทริบิวต์

**2.3.1.2 คำถามภาษาธรรมชาติ** ในระบบ FDB คำถามสามารถใช้ภาษาธรรมชาติเหมือนภาษาอังกฤษได้ เช่น "find a man from person who is young" โดยรูปแบบคำถามสามารถใช้เทอมที่เป็นพีชชีนออกเหนือจากชื่อแอตทริบิวต์ตามปกติได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3.2 การวิเคราะห์ภาษาธรรมชาติเพื่อสร้างภาษา SQL

2.3.2.1 รูปแบบของประโยคคำถามที่ใช้ภาษาธรรมชาติ ในภาษาธรรมชาติ ถ้าต้องการทราบว่าใครเป็นคนหนุ่มจากฐานข้อมูลอาจถามได้หลายอย่าง เช่น

"find a man from person who is young"

"list men from person that is young"

"look the people from person whose age is young"

2.3.2.2 แนวทางการวิเคราะห์ภาษาธรรมชาติ ถ้าวิเคราะห์ถึงความหมาย จะเห็นได้ว่าคำถามเหล่านี้ต้องการสิ่งเดียวกันคือ "คนหนุ่ม" ถ้าวิเคราะห์ถึงโครงสร้างของประโยคเหล่านี้แล้วจะเห็นว่าโครงสร้างของประโยคมี 3 ส่วนหลัก คือ ต้องการเลือกอะไร (find, list, look) จากฐานข้อมูลไหน (from) ด้วยเงื่อนไขอย่างไร (who, that, whose) ซึ่งเป็นโครงสร้างแบบเดียวกับ SELECT-FROM-WHERE ในภาษา SQL จึงควรจะใช้ทดแทนกันได้ถ้า นำคำถามเหล่านี้มาปรับปรุงเสียใหม่

2.3.3 การสร้างประโยคคำถามภาษา SQL จากภาษาธรรมชาติ เพื่อประกอบการอธิบายจึงขอยกตัวอย่างประโยคคำถามและชื่อฐานข้อมูลดังนี้ คือ

ตัวอย่างประโยคคำถามภาษาธรรมชาติ

"find the people from a person whose age is young"

ตัวอย่างชื่อฐานข้อมูลเสมือน

"tempfdbx"

2.3.3.1 แฉงประโยคคำถามออกเป็นคำ จากตัวอย่างประโยคคำถามภาษาธรรมชาติสามารถแยกหรือแฉงประโยคออกเป็นคำต่าง ๆ ได้ ดังนี้

find	the	people	from	a	person	whose	age	is	young
------	-----	--------	------	---	--------	-------	-----	----	-------

2.3.3.2 ตัดคำที่ไม่ต้องการออก คำบางคำในภาษาธรรมชาติไม่ได้นำมาใช้ให้เป็นประโยชน์ในการประมวลผลทางด้านคอมพิวเตอร์ คำเหล่านี้สามารถตัดออกได้โดยความหมายไม่เปลี่ยนแปลง เช่น พวก article อันมี a, an, the เป็นต้น เมื่อตัดคำที่ไม่ต้องการเหล่านี้ออกจะเหลือกลุ่มคำต่าง ๆ ดังนี้

find		people	from		person	whose	age	is	young
------	--	--------	------	--	--------	-------	-----	----	-------

### 2.3.3.3 คำคำที่มีความหมายเหมือนกันมาแทน คำบางคำระบบ FDB

ได้เก็บคำศัพท์ที่มีความหมายเหมือนกันเอาไว้ ถ้าคำถามมีคำที่มีความหมายเหมือนกันก็นำคำศัพท์ที่เหมือนกันมาแทน คำที่มีความหมายเหมือนกันจริงๆ หรือความหมายเหมือนกันโดยอนุโลม เช่น people, person, citizen, man, men และ name เป็นต้น

จากประโยคข้างต้น คำที่มีความหมายเหมือนกันเช่น คำว่า find มีความหมายเหมือนกับคำว่า select และคำว่า people มีความหมายเหมือนกับคำว่า name เมื่อนำคำที่มีความหมายเหมือนกันมาแทนจะได้กลุ่มคำต่าง ๆ ดังนี้

select		name	from	person	whose	age	is	young
--------	--	------	------	--------	-------	-----	----	-------

### 2.3.3.4 ตัดส่วนเงื่อนไขของประโยคออก ส่วนที่เป็นเงื่อนไขคือส่วนของประโยคตั้งแต่คำ who, that, whose เป็นต้นไปจนจบประโยค ได้แก่กลุ่มคำต่อไปนี้

whose	age	is	young
-------	-----	----	-------

เมื่อตัดส่วนที่เป็นเงื่อนไขของประโยคออกแล้วจะเหลือกลุ่มคำต่าง ๆ ดังนี้

select	name	from	person
--------	------	------	--------

### 2.3.3.5 คำคำที่ขาดมาเพิ่มเติมแล้วจัดรูปแบบประโยคใหม่

เนื่องจากประโยคคำถามยังไม่สมบูรณ์ ดังนั้นจึงต้องหาสิ่งที่ขาดไปมาเพิ่มเติมให้เป็นคำสั่งที่สมบูรณ์ เช่น แทนชื่อฐานข้อมูลหลัก "person" ด้วยชื่อฐานข้อมูลเสมือน "tempfdbx" เพิ่มเครื่องหมายจุลภาค (,) จัดวางคำบางคำ เช่น age และ young ให้ถูกตำแหน่ง หลังจากนั้นก็จัดเรียงประโยคเสียใหม่ให้สมบูรณ์จะได้ประโยคคำถามภาษา SQL ดังนี้

select	name	,	age	,	young	from	tempfdbx
--------	------	---	-----	---	-------	------	----------

### 2.3.4 การแปลภาษาธรรมชาติเป็นภาษา SQL

การแปลภาษาธรรมชาติเป็นภาษา SQL ใช้หลักการในทำนองเดียวกันกับการสร้างประโยคคำถามดังกล่าวมาแล้ว ระบบนี้ได้แปลภาษาธรรมชาติจากแต่ละคำถามเป็นภาษา SQL รูปแบบต่าง ๆ รวมทั้งส่วนเงื่อนไข เช่น จากตัวอย่างคำถามและชื่อฐานข้อมูลเสมือนที่กล่าวมาแล้วคือ "find the people from a person whose age is young" และ "tempfdbx" สามารถแปลเป็นชุดคำสั่ง SQL รูปแบบต่าง ๆ ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 2.3.4.1 SELECT-FORM คือ

```
select name, age, young
from tempfdbx
```

จุดประสงค์ของ SELECT-FORM เพื่อใช้ในการอ่านข้อมูลจากฐานข้อมูลเสมือน

#### 2.3.4.2 CREATE-FORM คือ

```
create table
tempfdbx (name , age , young )
as select name , age , age from person
```

จุดประสงค์ของ CREATE-FORM เพื่อใช้ในการสร้างฐานข้อมูลเสมือน แอตตริบิวต์ที่ใช้นำมาจากฐานข้อมูลหลักและพีชชีเทอม รูปแบบนี้จะใช้เมื่อผู้ใช้ต้องการสร้างฐานข้อมูลเสมือนให้มีแอตตริบิวต์ที่ต้องการใช้หรือเมื่อข้อมูลในฐานข้อมูลหลักมีการเปลี่ยนแปลง ถ้าในคำถามมีข้อมูลที่ต้องการใช้เป็นชื่อเดิมและข้อมูลในฐานข้อมูลหลักไม่มีการเปลี่ยนแปลงก็ไม่จำเป็นต้องสร้างใหม่ทุกครั้งที่ใช้งาน

#### 2.3.4.3 UPDATE-FORM คือ

```
update tempfdbx
set young = 1.0
where ((age >= 0) and (age <= 25))
และ
update tempfdbx
set young = power(1+power(((age-25)/5),2),-1)
where ((age > 25) and (age <=130))
```

จุดประสงค์ของ UPDATE-FORM เพื่อใช้ในการปรับสภาพข้อมูลเสมือนให้ทันสมัยตามสภาพที่ใช้อยู่ในขณะนั้น เช่น หลังจากใช้คำสั่ง create table หรือหลังจากกำหนดความสัมพันธ์ใหม่หรือหลังจากทราบว่าข้อมูลมีการเปลี่ยนแปลง การใช้ UPDATE-FORM จะคำนวณค่าของความสัมพันธ์ใหม่ตามสภาพของข้อมูลที่ใช้อยู่ในขณะนั้น

#### 2.3.4.4 ALTER-FORM คือ

```
alter table tempfdbx
modify young number (4, 3)
```

จุดประสงค์ของ ALTER-FORM เพื่อใช้ในการปรับปรุงชนิดของแอตตริบิวต์ (data type) ที่เป็นเลขจำนวนเต็มให้เป็นเลขจำนวนทศนิยมเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการคำนวณ ไม่ว่าจะเป็นการคูณหารหรือการบวกหรือการลบ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.4.5 DROP-FORM คือ

drop table tempfdbx

จุดประสงค์ของ DROP-FORM เพื่อใช้ในการลบทิ้งฐานข้อมูลเสมือนเมื่อไม่ต้องการใช้

### 2.3.4.6 CONDITION-FORM ส่วนที่เป็นเงื่อนไขคือส่วนของคำถามภาษาธรรมชาติ

หลังคำ who, that, whose ซึ่งเมื่อนำมาแจกแจงแล้วจะได้เงื่อนไข คือ

age	is	young
-----	----	-------

จุดประสงค์ของ CONDITION-FORM เพื่อใช้ในการกำหนดเงื่อนไขของข้อมูลที่ต้องการ เงื่อนไขที่ได้นี้จะนำไปหาค่าผลลัพธ์ในการตอบคำถาม

การแปลจะกระทำทุกครั้งที่มีคำถาม สำหรับการเลือกรูปแบบต่าง ๆ ขึ้นอยู่กับ การตอบคำถามของผู้ใช้ระหว่างการใช้งาน

**2.3.5 รูปแบบของคำถามภาษาธรรมชาติที่นำมาใช้** เนื่องจากคำถามในภาษาธรรมชาติที่มีความหมายเดียวกันมีหลากหลายรูปแบบ จึงต้องกำหนดขอบเขตของภาษาธรรมชาติให้อยู่ในรูปแบบที่จำกัด ดังนั้นรูปแบบภาษาธรรมชาติที่นำมาใช้เป็นคำถามจึงต้องยึดถือโครงสร้าง SELECT-FROM-WHERE ตามภาษา SQL

กำหนดให้ R เป็นความสัมพันธ์ใน BDB ซึ่งมีแอตทริบิวต์ A, B, ... แล้ว select operation ใน FDB มีรูปแบบดังนี้

select look tell peek list see etc .	} A, B from R	where that whose who which etc .	} < comparison clause >
--	---------------	---	-------------------------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.6 **หลักเกณฑ์การใช้เงื่อนไขสำหรับความสัมพันธ์ (comparison clause)**  
 เงื่อนไขสำหรับความสัมพันธ์แบ่งออกเป็น 2 ชนิดคือ เงื่อนไขความสัมพันธ์เกี่ยวกับ "จำนวนเลข  
 หรือสิ่งที่นับเป็นจำนวนเลขได้" และเงื่อนไขความสัมพันธ์เกี่ยวกับ "ข้อความหรือสิ่งที่นับเป็น  
 จำนวนเลขไม่ได้" มีรายละเอียดดังนี้

2.3.6.1 **เงื่อนไขสำหรับความสัมพันธ์จำนวนเลข** แบ่งออกเป็น 2 แบบ  
 คือ เงื่อนไขของความสัมพันธ์จำนวนเลขที่สมบูรณ์ในตัวเอง เงื่อนไขของความสัมพันธ์จำนวน  
 เลขที่ไม่สมบูรณ์ในตัวเอง

2.3.6.1.1 **เงื่อนไขของความสัมพันธ์จำนวนเลขที่สมบูรณ์ในตัวเอง**  
 เงื่อนไขประเภทนี้ไม่ต้องอ้างอิงจำนวนเลขอื่นก็จะหาคำตอบได้ เช่น ความสัมพันธ์  
 YOUNG, OLD และ TALL เป็นต้น เงื่อนไขดังกล่าวนี้มีรูปแบบดังตารางที่ 3

ตารางที่ 3

แสดงเงื่อนไขสำหรับความสัมพันธ์จำนวนเลขที่สมบูรณ์ในตัวเอง

แอตทริบิวต์ A,B	VERB TO BE	คำขยาย (modifier)	คำขยาย (modifier)	ฟัซซีเทอม (fuzzy term)
age	is	not	not	young
	are	very	very	old
height	is	not	not	tall
	are	very	very	

2.3.6.1.2 **เงื่อนไขของความสัมพันธ์จำนวนเลขที่ไม่สมบูรณ์ในตัวเอง**  
 เงื่อนไขประเภทนี้ต้องอ้างอิงจำนวนเลขอื่นจึงจะหาคำตอบได้ เช่น ความสัมพันธ์  
 PROXIMITY เป็นต้น โดยเมื่อถามว่าอายุใกล้เคียง 30 ปี หรือความสูงใกล้เคียง 160 เซนติเมตร  
 จำเป็นต้องอ้างอิงจำนวนเลขอื่นคือ 30 หรือ 160 เป็นต้น

เนื่องจากเงื่อนไข PROXIMITY ใช้ได้กับแอตทริบิวต์ทั่ว ๆ ไปที่เป็นจำนวนเลข ดังนั้นคำ  
 ถามต้องระบุว่าเป็นความสัมพันธ์ PROXIMITY ของแอตทริบิวต์ตัวใด เช่น PROXIMITY ของ  
 age คือ "age is proximity 30" หรือ PROXIMITY ของ height คือ "height is proximity 160"

เงื่อนไขดังกล่าวนี้มีรูปแบบดังตารางที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตารางที่ 4

แสดงเงื่อนไขสำหรับความสัมพันธ์จำนวนเลขที่ไม่สมบูรณ์ในตัวเอง

แอตตริบิวต์	VERB TO	คำขยาย	คำขยาย	ฟัซซีเทอม	จำนวนเลข
A,B	BE	(modifier)	(modifier)	(fuzzy term)	อื่นที่อ้างถึง
age	is	not	not	proximity	1 ปี
	are	very	very		ถึง
height	is	not	not	proximity	1 ซม.
	are	very	very		ถึง
					250 ซม.

2.3.6.2 เงื่อนไขสำหรับความสัมพันธ์ข้อความ แบ่งออกเป็น 2 แบบ คือ เงื่อนไขสำหรับความสัมพันธ์ข้อความที่ "ค่าแอตตริบิวต์มีข้อความเดียว" และเงื่อนไขสำหรับความสัมพันธ์ข้อความที่ "ค่าแอตตริบิวต์มีข้อความหลายข้อความต่อกัน"

2.3.6.2.1 เงื่อนไขสำหรับความสัมพันธ์ข้อความที่ค่าแอตตริบิวต์มีข้อความเดียว แอตตริบิวต์ประเภทนี้ เช่น Name ดังกล่าวมาแล้ว เงื่อนไขชนิดนี้มีรูปแบบดังตารางที่ 5

## ตารางที่ 5

แสดงเงื่อนไขสำหรับความสัมพันธ์ข้อความที่ค่าแอตตริบิวต์มีข้อความเดียว

แอตตริบิวต์	VERB TO	ฟัซซีเทอม
A,B	BE	(ชื่อบุคคลากร)
name	is	'กนก'
	are	'อนันต์'

2.3.6.2.2 เงื่อนไขสำหรับความสัมพันธ์ข้อความที่ค่าแอตตริบิวต์มีข้อความหลายข้อความต่อกัน แอตตริบิวต์ประเภทนี้ เช่น Province ดังกล่าวมาแล้ว เงื่อนไขชนิดนี้มีรูปแบบดังตารางที่ 6 ได้รับความเห็นชอบจากสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตารางที่ 6

แสดงเงื่อนไขสำหรับความสัมพันธ์ข้อความที่ค่าแอดตริบิวต์มีข้อความหลายข้อความต่อกัน

แอดตริบิวต์ A,B	VERB TO BE	พีชซีเทอม (ชื่อจังหวัด)
province	is	'กรุงเทพ'
	are	'อ่างทอง'

**2.3.7 ขอบเขตของคำถามภาษารธรรมชาติที่นำมาใช้** รูปแบบภาษารธรรมชาติที่นำมาใช้เป็นคำถามนอกจากต้องยึดถือโครงสร้าง SELECT-FROM-WHERE แล้วยังมีข้อจำกัดอื่น ๆ คือจบประโยคคำถามด้วยเครื่องหมายอัฒภาค (;) ชื่อเฉพาะต้องอยู่ภายในเครื่องหมายฝนทอง (') เช่น 'กรุงเทพ' เป็นต้น

**2.4 ฟังก์ชันสมาชิก (Membership Functions)** เป็นตัวกำหนดระดับค่าการเป็นสมาชิกของข้อมูล เช่น MYOUNG และ MOLD (จะอธิบายรายละเอียดในหัวข้อต่อไป) ผลลัพธ์ที่ได้จากการคำนวณจากฟังก์ชันสมาชิกเรียกว่าระดับการเป็นสมาชิก (grade หรือ degree of membership)

## 2.5 ความสัมพันธ์แบบพีชซี (Fuzzy Relations)

**2.5.1 นิยามความสัมพันธ์** ถ้า  $A_i$  เป็นแอดตริบิวต์ที่คลุมเครือภายใต้โดเมน  $D_i$  แล้ว ค่าแอดตริบิวต์จะเป็นค่าที่กระจายที่เป็นไปได้บน  $D_i$  เขียนแทนโดย  $\Pi_{A_i}$  [1]

กำหนดให้  $D_1, \dots, D_n$  เป็นโดเมนเซตของสิ่งที่มีเป็นจำนวนเลขได้หรือสิ่งที่มีเป็นจำนวนเลขไม่ได้หรือ unit interval และให้  $R$  เป็นความสัมพันธ์ที่เป็นไปได้ของ  $n$  เซต นั่นคือ  $R$  เป็นเซตของ  $n$ -tuple  $(\Pi_{A_1}, \dots, \Pi_{A_n})$  เมื่อ  $\Pi_{A_i}$  ( $i=1, \dots, n$ ) เป็นค่ากระจายที่เป็นไปได้ของแอดตริบิวต์  $A_i$  ซึ่งกำหนดเหนือโดเมนเซต  $D_i$  [1]

หรือ Fuzzy Relation  $R$  from  $x$  to  $y$  เป็น fuzzy sets อาจนิยามได้ดังนี้

$$x * y = \{ (x,y) \mid x \in X, y \in Y \}, \quad \mu_R : X * Y \rightarrow [0, 1]$$

**2.5.2 ความสัมพันธ์ที่นำมาใช้ในระบบ** ความสัมพันธ์มีหลายรูปแบบขึ้นอยู่กับผู้เชี่ยวชาญจะสร้างขึ้นและกำหนดชื่อขึ้นมา ความสัมพันธ์ที่นำมาใช้ในการวิจัยมี 4 รูปแบบ

**2.5.2.1 ความสัมพันธ์ PROXIMITY** ให้  $D_i$  เป็นโดเมนของสิ่งที่มีเป็นจำนวนเลขได้ (numerical domain) และ  $x, y, z \in D_i$  แล้ว  $P(x,y) \in [0,1]$  เป็นความสัมพันธ์ใกล้เคียง

ตามคุณลักษณะที่สังเกตเห็นที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• **คุณสมบัติการสะท้อนกลับ (Reflexive)** เป็นความสัมพันธ์ระหว่างตัวเอง เช่น ความสัมพันธ์ระหว่าง  $x$  กับ  $x$  จะมีค่าความจริงเป็น 1 นั่นคือ  $P(x,x) = 1$

• **คุณสมบัติสมมาตร (Symmetric)** เป็นความสัมพันธ์ระหว่าง  $x$  กับ  $y$  จะมีค่าความจริงเท่ากับความสัมพันธ์ระหว่าง  $y$  กับ  $x$  นั่นคือ  $P(x,y) = P(y,x)$

• **คุณสมบัติการถ่ายทอด (Transitive)** เป็นความสัมพันธ์ระหว่าง  $x$  กับ  $z$  จะมีค่าความจริงเท่ากับหรือมากกว่าความสัมพันธ์ระหว่าง  $x$  กับ  $y$  คูณกับความสัมพันธ์ระหว่าง  $y$  กับ  $z$  นั่นคือ  $P(x,z) \geq \max_y P(x,y) * P(y,z)$  [1] หรือ  $P(x,z) \geq \bigvee_y \{ P(x,y) \wedge P(y,z) \}$   $y \in D_i$

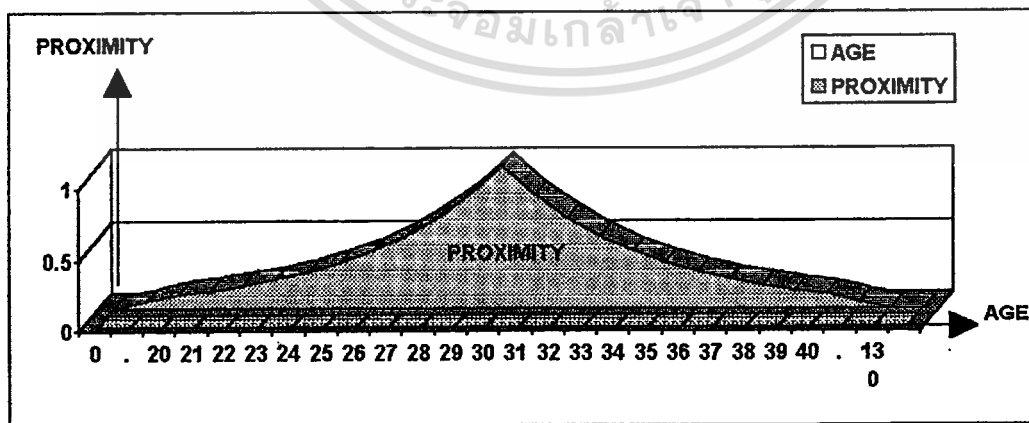
ความสัมพันธ์ใกล้เคียงได้มีผู้เชี่ยวชาญนำเสนอไว้หลายรูปแบบด้วยกัน ในบทวิจัยนี้ได้เลือกใช้ความสัมพันธ์ใกล้เคียงเพื่อนำมาอ้างอิงคือ

$$m_{\text{PROXIMITY}}(x,y) = e^{-c|x-y|} \quad \text{เมื่อ } c > 0 \quad [2] \quad \dots \text{สมการที่ (1)}$$

ค่า  $c$  จะกำหนดเท่าใดก็ได้แต่ต้องมากกว่า 0 โดยคำนึงถึงความเหมาะสมกับค่าของข้อมูลและความต้องการของผู้ใช้ เช่น ข้อมูลอายุของมนุษย์กับข้อมูลของเงินเดือนค่าของข้อมูลอยู่ในช่วงต่างกันค่า  $c$  ย่อมไม่เท่ากัน ในที่นี้กำหนดค่า  $c = 2.23 \times 10^{-1}$  และค่า  $e$  คือค่า exponential มีค่า  $\approx 2.71828$  [2] เมื่อ  $x$  เป็นเลขจำนวนเต็มที่กำหนด  $y$  เป็นค่าแอดตริบิวต์ที่นับเป็นจำนวนเลขได้ กราฟของความสัมพันธ์ PROXIMITY แสดงได้ดังภาพที่ 2

ภาพที่ 2

แสดงกราฟของความสัมพันธ์ PROXIMITY



ภาพที่ 2 แสดงให้เห็นค่าของความสัมพันธ์ PROXIMITY  $(x,y)$  จากข้อความ “AGE IS PROXIMITY 30” หรือ “อายุใกล้เคียง 30” เมื่อ  $x$  แทนจำนวนเลขที่กำหนดคือ 30 และ  $y$  แทนค่าของแอดตริบิวต์ AGE จากกราฟจะเห็นว่าค่า AGE ที่ถูกต้องสมบูรณ์หรืออายุเท่ากับ 30 ปี

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

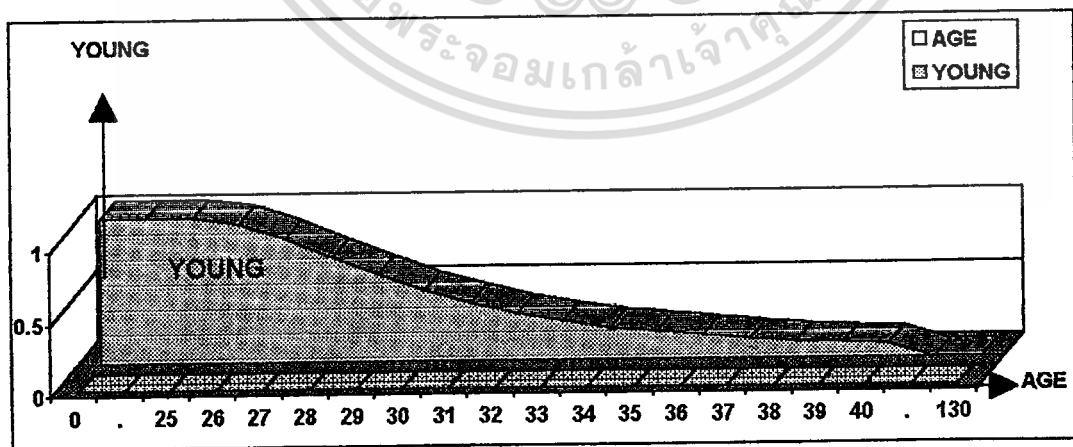
ระดับค่าการเป็นสมาชิกของ PROXIMITY จะมีค่าเท่ากับ 1 และค่า AGE ที่ถูกต้องไม่สมบูรณ์หรืออายุไม่เท่ากับ 30 ปีระดับค่าการเป็นสมาชิกของ PROXIMITY จะมีค่าระหว่าง 1 ถึง 0 โดยระดับค่าการเป็นสมาชิกจะเพิ่มขึ้นเข้าใกล้ 1 ตามค่าอายุที่วิ่งเข้าใกล้ 30 หรือถ้าความสัมพันธ์ใกล้เคียงเป็นศูนย์หรือ  $P(x,y) = 0$  แล้ว  $x$  กับ  $y$  จะไม่มีความใกล้เคียงกันเลย และถ้าความสัมพันธ์ใกล้เคียงเป็นหนึ่งหรือ  $P(x,y) = 1$  แล้ว  $x$  กับ  $y$  จะเป็นจำนวนเลขเดียวกัน

2.5.2.2 ความสัมพันธ์ YOUNG ให้เอกภพสัมพัทธ์ (relative universe) อยู่ในช่วงอายุ  $[0,130]$  ปี และให้  $x$  แทนค่าอายุมีหน่วยเป็นปีแล้ว YOUNG เป็นชื่อฟังก์ชันเซตที่นิยามระดับการเป็นสมาชิก  $x$  ต่อ YOUNG ดังนี้

$$m_{\text{YOUNG}}(x) = \begin{cases} 1 & \text{เมื่อ } 0 < x \leq 25 \\ \left[ 1 + \left( \frac{x-25}{5} \right)^2 \right]^{-1} & \text{เมื่อ } 25 < x \leq 130 \end{cases} \quad \dots \text{สมการที่ (2)}$$

จากความสัมพันธ์ YOUNG หมายความว่า คนที่มีอายุ 0 ถึง 25 ปี  $m_{\text{YOUNG}}$  จะมีค่าเท่ากับ 1 คืออายุยังอ่อนวัยหรือเยาว์วัยแน่นอน แต่คนที่มีอายุมากกว่า 25 ปีจนถึงอายุ 130 ปี  $m_{\text{YOUNG}}$  จะมีค่าค่อย ๆ ลดลง คือความเยาว์วัยจะลดลงตามลำดับอายุ กราฟของความสัมพันธ์ YOUNG แสดงได้ดังภาพที่ 3

ภาพที่ 3  
แสดงกราฟของความสัมพันธ์ YOUNG



ภาพที่ 3 แสดงให้เห็นคุณสมบัติของ  $m_{\text{YOUNG}}$  จากกราฟจะเห็นว่าค่า AGE ที่ถูกต้องสมบูรณ์หรืออายุระหว่าง 0 - 25 ปีระดับค่าการเป็นสมาชิกของ  $m_{\text{YOUNG}}$  จะมีค่าเท่ากับ 1 และค่า AGE ที่ถูกต้องไม่สมบูรณ์หรืออายุระหว่าง 26 - 130 ปี ระดับการเป็นสมาชิกของ  $m_{\text{YOUNG}}$  การคำนวณค่าไม่ต่างกันโดยทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

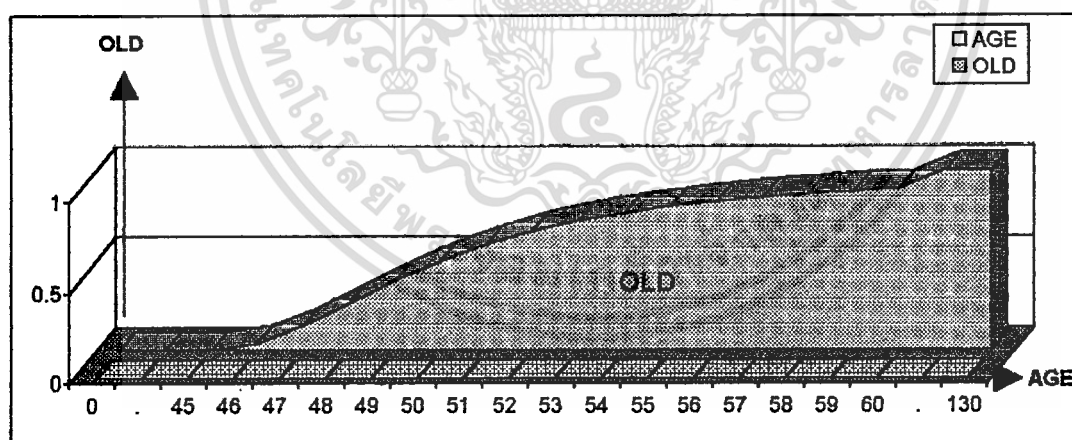
จะมีค่าระหว่าง 1 ถึง 0 โดยระดับการเป็นสมาชิกจะเพิ่มขึ้นเข้าใกล้ 1 ตามค่าอายุที่ลดน้อยลง หรือถ้าอายุมีค่ามาก  $m_{YOUNG}$  จะมีค่าน้อยและถ้าอายุมีค่าน้อย  $m_{YOUNG}$  จะมีค่ามากกลับกัน

**2.5.2.3 ความสัมพันธ์ OLD** ให้เอกภพสัมพัทธ์อยู่ในช่วงอายุ  $[0,130]$  ปี และให้  $x$  แทนค่าอายุมีหน่วยเป็นปีแล้ว OLD เป็นชื่อฟังก์ชันที่ จะนิยามระดับการเป็นสมาชิก  $x$  ต่อ OLD ดังนี้

$$m_{OLD}(x) = \begin{cases} 0 & \text{เมื่อ } 0 \leq x \leq 45 \\ \left[ 1 + ((x - 45) / 5)^{-2} \right]^{-1} & \text{เมื่อ } 45 < x \leq 130 \end{cases} \quad \dots \text{สมการที่ (3)}$$

จากความสัมพันธ์ OLD หมายความว่า คนที่มีอายุ 0 ถึง 45 ปี  $m_{OLD}$  จะมีค่าเท่ากับ 0 คืออายุยังไม่แก่แน่นอน แต่คนที่มีอายุมากกว่า 45 ปีจนถึงอายุ 130 ปี  $m_{OLD}$  จะมีค่าค่อย ๆ เพิ่มขึ้นคือความแก่เพิ่มขึ้นตามลำดับอายุ กราฟของความสัมพันธ์ OLD แสดงได้ดังภาพที่ 4

ภาพที่ 4  
แสดงกราฟของความสัมพันธ์ OLD



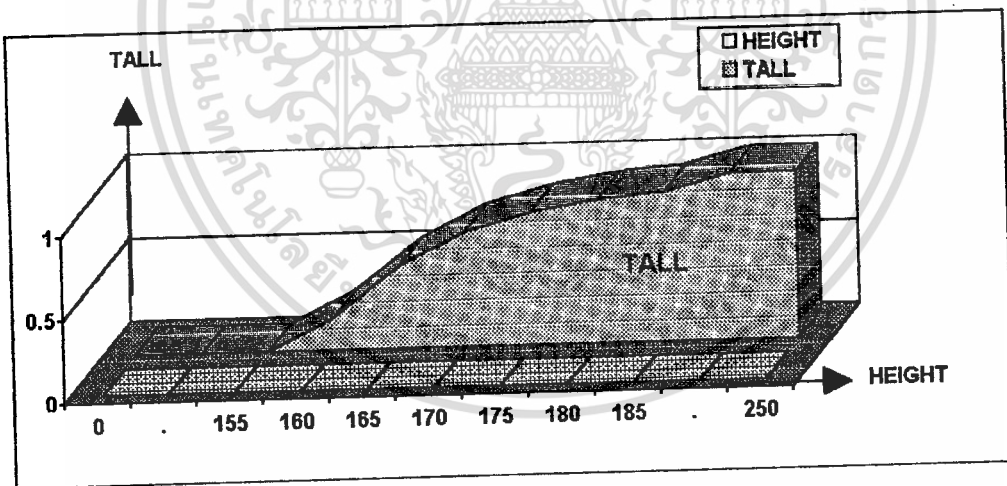
ภาพที่ 4 แสดงให้เห็นคุณสมบัติของ  $m_{OLD}$  จากกราฟจะเห็นว่าค่า AGE หรืออายุระหว่าง 0 ถึง 45 ปีไม่เป็นสมาชิกของ  $m_{OLD}$  ดังนั้นระดับค่าการเป็นสมาชิกของ  $m_{OLD}$  จะมีค่าเท่ากับ 0 และค่า AGE ที่ถูกต้องไม่สมบูรณ์หรืออายุระหว่าง 46 - 130 ปีระดับค่าการเป็นสมาชิกของ  $m_{OLD}$  จะมีค่าระหว่าง 0 ถึง 1 โดยระดับค่าการเป็นสมาชิกจะเพิ่มขึ้นเข้าใกล้ 1 ตามอายุที่มากขึ้น หรือถ้าอายุมีค่าน้อย  $m_{OLD}$  จะมีค่าน้อยและถ้าอายุมีค่ามากแล้ว  $m_{OLD}$  จะมีค่ามากตามไปด้วย

2.5.2.4 ความสัมพันธ์ TALL ให้เอกภพสัมพันธ์อยู่ในช่วงความสูง (0,250) เซ็นติเมตร และให้  $x$  แทนค่าความสูงมีหน่วยเป็นเซ็นติเมตรแล้ว TALL เป็นชื่อฟังก์ชันที่นิยามระดับการเป็นสมาชิก  $x$  ต่อ TALL ดังนี้

$$m_{TALL}(x) = \begin{cases} 0 & \text{เมื่อ } 0 \leq x \leq 155 \\ \left[ 1 + ((x - 155) / 10)^{-2} \right]^{-1} & \text{เมื่อ } 155 < x \leq 250 \end{cases} \quad \dots \text{สมการที่ (4)}$$

จากความสัมพันธ์ TALL คนที่มีความสูง 0 ถึง 155 เซ็นติเมตร  $m_{TALL}$  จะมีค่าเท่ากับ 0 หมายความว่าจัดอยู่ในกลุ่มเป็นคนไม่สูง แต่คนที่มีความสูงมากกว่า 155 เซ็นติเมตรถึง 250 เซ็นติเมตร  $m_{TALL}$  จะมีค่าค่อย ๆ เพิ่มขึ้น หมายความว่าสูงเพิ่มขึ้นตามลำดับความสูง กราฟของความสัมพันธ์ TALL แสดงได้ดังภาพที่ 5

ภาพที่ 5  
แสดงกราฟของความสัมพันธ์ TALL



ภาพที่ 5 แสดงให้เห็นคุณสมบัติของ  $m_{TALL}$  จากกราฟแสดงให้เห็นว่าค่า HEIGHT หรือความสูงระหว่าง 0 - 155 เซ็นติเมตรไม่เป็นสมาชิกของ  $m_{TALL}$  ดังนั้นระดับค่าการเป็นสมาชิกของ  $m_{TALL}$  จะมีค่าเท่ากับ 0 และค่า HEIGHT ที่ถูกต้องไม่สมบูรณ์หรือความสูงระหว่าง 156 - 250 เซ็นติเมตรระดับค่าการเป็นสมาชิกของ  $m_{TALL}$  จะมีค่าระหว่าง 0 ถึง 1 โดยระดับค่าการเป็นสมาชิกจะเพิ่มขึ้นเข้าใกล้ 1 ตามความสูงที่มากขึ้น หรือถ้าความสูงมีค่าน้อย  $m_{TALL}$  จะมีค่าน้อยและถ้าความสูงมีค่ามากแล้ว  $m_{TALL}$  จะมีค่ามากตามไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะโดยทางทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.3 การจัดการเกี่ยวกับความสัมพันธ์

**2.5.3.1 การเพิ่มความสัมพันธ์** ถ้าผู้ใช้ต้องการใช้ความสัมพันธ์เพียงชั่วคราว สามารถพิมพ์ความสัมพันธ์ผ่านทาง keyboard แต่ถ้าผู้ใช้ต้องการใช้ถาวรสามารถพิมพ์ความสัมพันธ์เก็บไว้ใน header file

**2.5.3.2 การเก็บความสัมพันธ์** ความสัมพันธ์เก็บไว้ที่ header file (ในการวิจัยนี้ไม่ได้เก็บไว้ที่ BDB) สำหรับการพิมพ์ความสัมพันธ์ผ่านทาง keyboard ชั่วคราวจะไม่เก็บไว้

**2.5.3.3 การนำความสัมพันธ์มาใช้ในระบบ** โปรแกรมจะใช้ความสัมพันธ์ต่าง ๆ ผ่านทาง SQL-FORM โดยในการแปลภาษาธรรมชาติเป็นภาษา SQL ระบบจะไปอ่านความสัมพันธ์มาสร้างเป็น SQL-FORM เมื่อใช้คำสั่งใน SQL-FORM ก็จะใช้ความสัมพันธ์ที่ฝังอยู่ใน SQL-FORM โดยอัตโนมัติ ดังนั้นจึงไม่จำเป็นต้องเก็บความสัมพันธ์ไว้ในโปรแกรม

**2.5.4 การประยุกต์ใช้ความสัมพันธ์ที่เป็นจำนวนเลข** นอกจากการใช้ความสัมพันธ์ตามปกติแล้วเรายังสามารถขยายความสัมพันธ์และสามารถจำกัดค่าคำตอบเพื่อกำหนดขอบเขตของคำตอบให้อยู่ในความต้องการได้ดังต่อไปนี้

**2.5.4.1 การขยายค่าความสัมพันธ์** การขยายค่าความสัมพันธ์ทำได้โดยใช้ขยายค่าความสัมพันธ์ที่เป็นจำนวนเลข ค่าขยายที่ใช้ เช่น VERY และ NOT เป็นต้น

**การขยายโดย VERY** ใช้วิธียกกำลังสอง การขยายโดย VERY จะเพิ่มความเข้มข้นให้กับความสัมพันธ์ เช่น จาก YOUNG (หนุ่ม) เป็น VERY YOUNG (หนุ่มมาก)

**การขยายโดย NOT** ใช้วิธีหา complement การขยายโดย NOT จะเพิ่มความหมายเชิงปฏิเสธให้ความสัมพันธ์ เช่น จาก YOUNG (หนุ่ม) เป็น NOT YOUNG (ไม่หนุ่ม) เป็นต้น

รูปแบบการขยายค่าความสัมพันธ์มีรูปแบบดังตารางที่ 7 และกราฟการขยายค่าความสัมพันธ์ เช่น ความสัมพันธ์ YOUNG แสดงได้ดังภาพที่ 6 และ 7

ตารางที่ 7

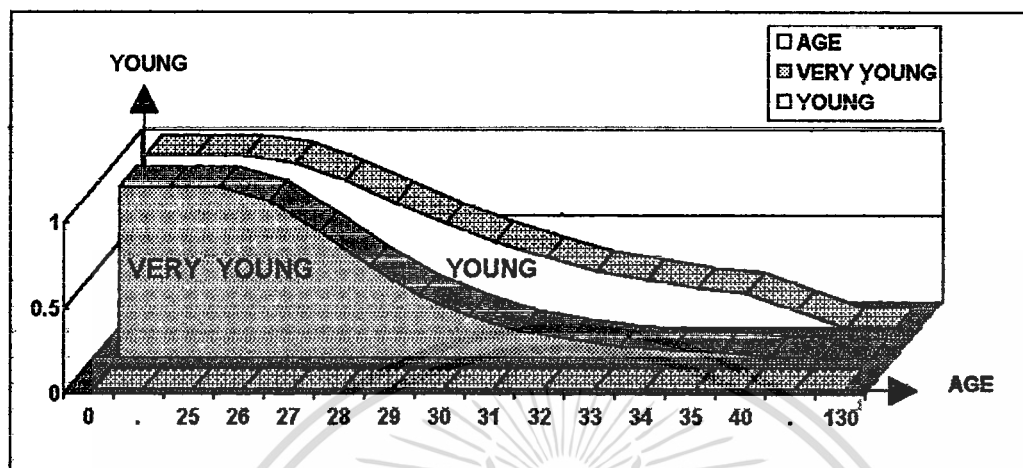
แสดงรูปแบบการขยายความสัมพันธ์ที่เป็น "จำนวนเลข"

ค่าขยาย	พืซซีเทอม	สมาชิกฟังก์ชัน
very	young	$\text{very}(\text{young}) = (\text{young})^2$
	old	$\text{very}(\text{old}) = (\text{old})^2$
	tall	$\text{very}(\text{tall}) = (\text{tall})^2$
not	young	$\text{not}(\text{young}) = 1 - \text{young}$
	old	$\text{not}(\text{old}) = 1 - \text{old}$
	tall	$\text{not}(\text{tall}) = 1 - \text{tall}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น มิใช่เพื่อให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 6

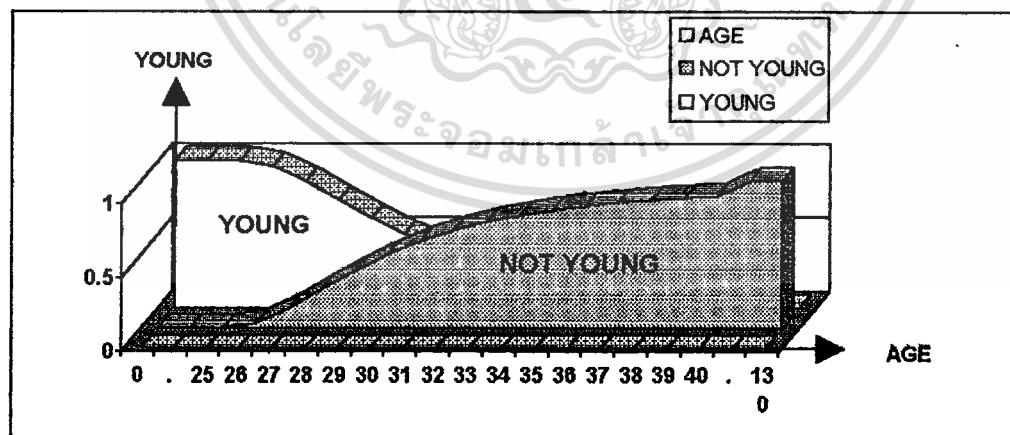
แสดงกราฟของการขยายความสัมพันธ์ YOUNG ด้วย VERY



ภาพที่ 6 แสดงให้เห็นกราฟของ YOUNG และ VERY YOUNG เปรียบเทียบกับบนแกน AGE จะเห็นว่ากราฟของ VERY YOUNG ที่อยู่ด้านหน้าทุก ๆ ค่าของ AGE จะมีค่าเท่ากับ  $(YOUNG)^2$  โดยอายุ 0 - 25 ปีค่าของกราฟ VERY YOUNG จะมีค่าเท่ากับกราฟของ YOUNG และอายุ 26 - 130 ปีค่าของกราฟ VERY YOUNG จะมีค่าลดต่ำลงกว่ากราฟของ YOUNG ที่อยู่ด้านหลังตามลำดับอายุหรือระดับค่าของกราฟ VERY YOUNG จะมีค่าเป็นกึ่งกลางสองของกราฟ YOUNG ที่อยู่ด้านหลัง

ภาพที่ 7

แสดงกราฟของการขยายความสัมพันธ์ YOUNG ด้วย NOT



ภาพที่ 7 แสดงให้เห็นกราฟของ YOUNG และ NOT YOUNG เปรียบเทียบกับบนแกน AGE จะเห็นว่ากราฟของ NOT YOUNG ที่อยู่ด้านหน้าทุก ๆ ค่าของ AGE จะมีค่าเท่ากับ  $1 - YOUNG$  โดยอายุ 0 - 25 ปีระดับค่าของกราฟ NOT YOUNG จะมีค่าเท่ากับ 0 และอายุ 26 - 130 ปีระดับค่าของกราฟ NOT YOUNG จะมีค่าสูงขึ้นตามลำดับอายุหรือระดับค่าของกราฟ NOT YOUNG จะมีค่าตรงข้ามกับกราฟ YOUNG

ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

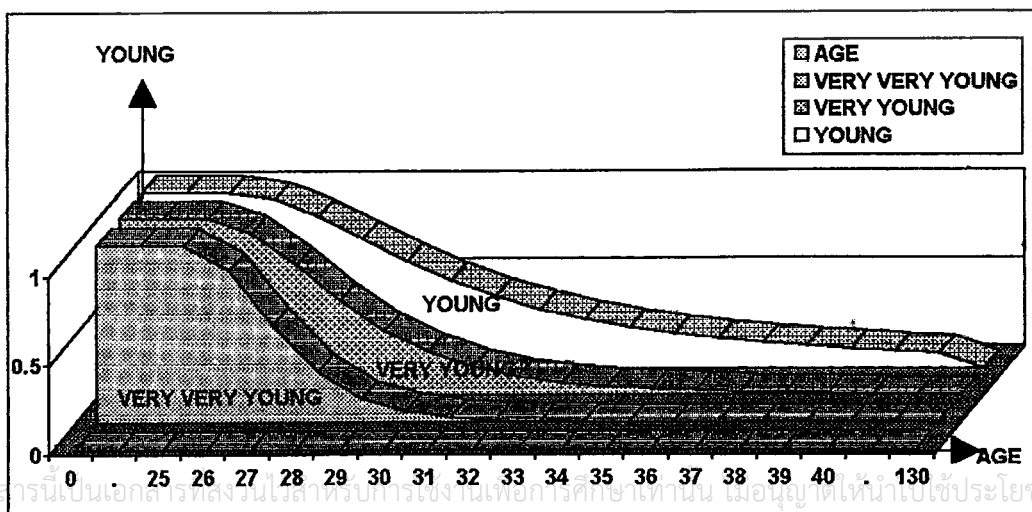
2.5.4.2 การขยายซ้ำค่าความสัมพันธ์ การขยายซ้ำค่าความสัมพันธ์ทำได้โดยทำได้โดยใช้ขยายค่าความสัมพันธ์ที่เป็นจำนวนเลข คำขยายที่ใช้ เช่น VERY และ NOT เหมือนการขยายค่าความสัมพันธ์ในข้อ 2.5.4.1 การขยายซ้ำความสัมพันธ์มีรูปแบบดังตารางที่ 8 และกราฟการขยายซ้ำค่าความสัมพันธ์ เช่น ความสัมพันธ์ YOUNG แสดงได้ดังภาพที่ 8 9 10 และ 11

ตารางที่ 8 แสดงรูปแบบการขยายซ้ำค่าความสัมพันธ์ที่เป็น "จำนวนเลข"

คำขยาย	พีชชีเทอม	สมาชิกฟังก์ชัน
very	very young	$\text{very}(\text{very young}) = (\text{very young})^2$
	very old	$\text{very}(\text{very old}) = (\text{very old})^2$
	very tall	$\text{very}(\text{very tall}) = (\text{very tall})^2$
very	not young	$\text{very}(\text{not young}) = (\text{not young})^2$
	not old	$\text{very}(\text{not old}) = (\text{not old})^2$
	not tall	$\text{very}(\text{not tall}) = (\text{not tall})^2$
not	very young	$\text{not}(\text{very young}) = 1 - (\text{very young})$
	very old	$\text{not}(\text{very old}) = 1 - (\text{very old})$
	very tall	$\text{not}(\text{very tall}) = 1 - (\text{very tall})$
not	not young	$\text{not}(\text{not young}) = 1 - (\text{not young})$
	not old	$\text{not}(\text{not old}) = 1 - (\text{not old})$
	not tall	$\text{not}(\text{not tall}) = 1 - (\text{not tall})$

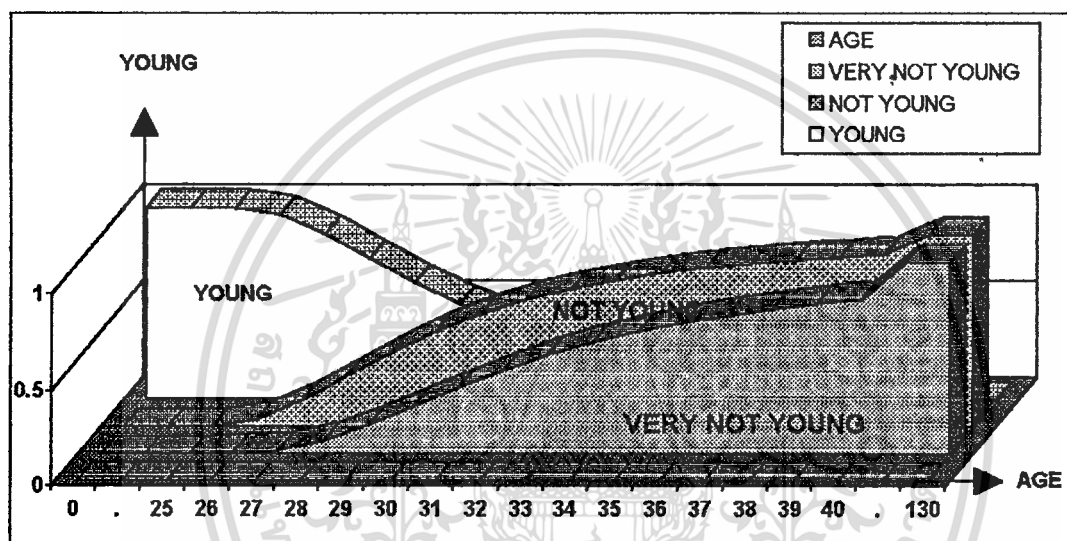
ภาพที่ 8

แสดงกราฟของการขยายซ้ำ VERY YOUNG ด้วย VERY



ภาพที่ 8 แสดงให้เห็นกราฟของ VERY VERY YOUNG, VERY YOUNG และ YOUNG เปรียบเทียบกันบนแกน AGE จะเห็นว่ากราฟของ VERY VERY YOUNG ที่อยู่ด้านหน้าทุก ๆ ค่าของ AGE จะมีค่าเท่ากับ (VERY YOUNG) <sup>2</sup> หรือระดับค่าของกราฟ VERY VERY YOUNG จะมีค่าเป็นกำลังสองของกราฟ VERY YOUNG ที่อยู่ตรงกลาง

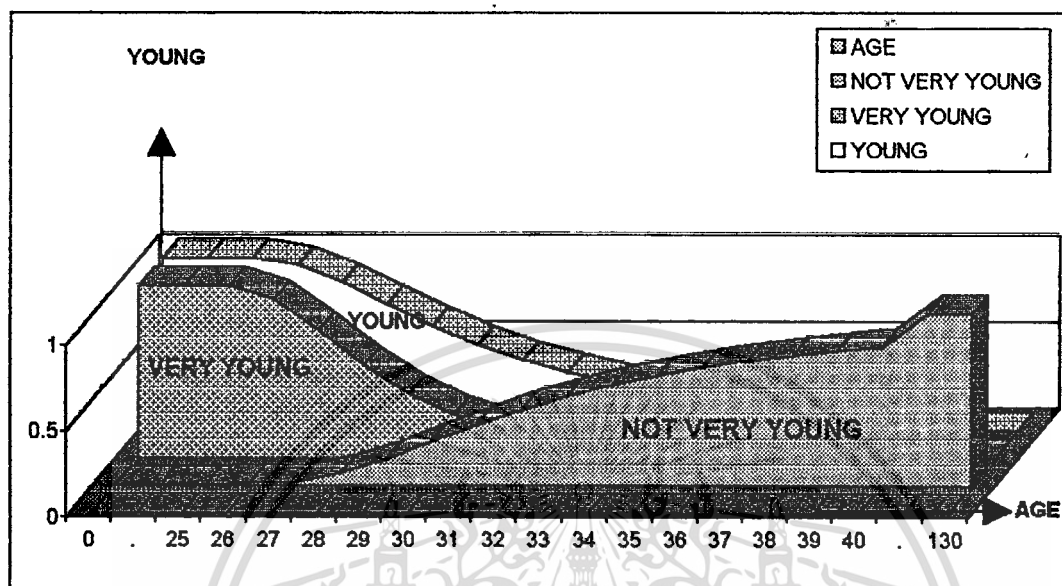
ภาพที่ 9  
แสดงกราฟของการขยายซ้ำ NOT YOUNG ด้วย VERY



ภาพที่ 9 แสดงให้เห็นกราฟของ VERY NOT YOUNG, NOT YOUNG และ YOUNG เปรียบเทียบกันบนแกน AGE จะเห็นว่ากราฟของ VERY NOT YOUNG ที่อยู่ด้านหน้าทุก ๆ ค่าของ AGE จะมีค่าเท่ากับ (NOT YOUNG) <sup>2</sup> หรือระดับค่าของกราฟ VERY NOT YOUNG จะมีค่าเท่ากับ กำลังสองของกราฟ NOT YOUNG ที่อยู่ตรงกลาง

ภาพที่ 10

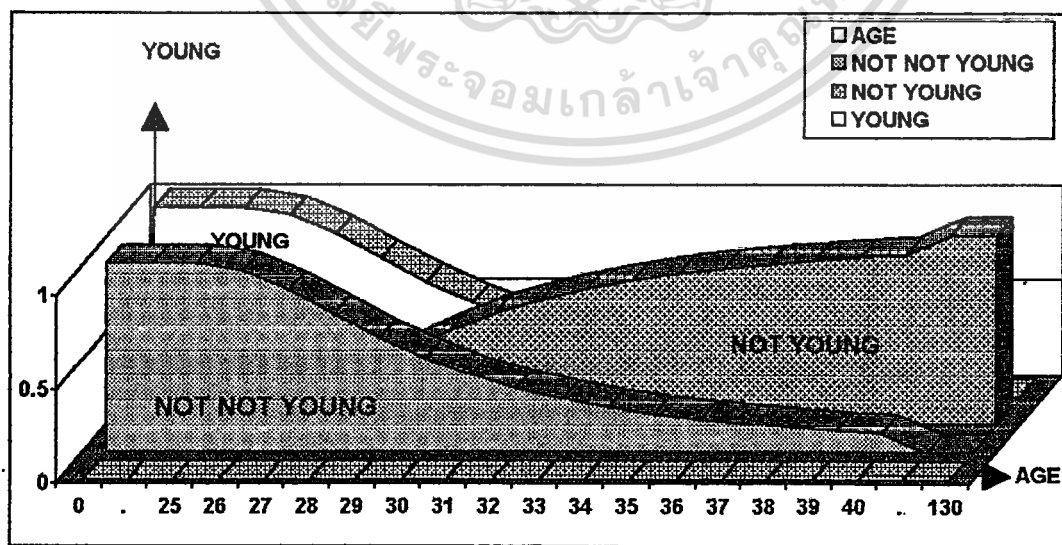
แสดงกราฟของการขยายซ้ำ VERY YOUNG ด้วย NOT



ภาพที่ 10 แสดงให้เห็นกราฟของ NOT VERY YOUNG, VERY YOUNG และ YOUNG เปรียบเทียบกันบนแกน AGE จะเห็นว่ากราฟของ NOT VERY YOUNG ที่อยู่ด้านหน้าทุก ๆ ค่าของ AGE จะมีค่าเท่ากับ 1 - (VERY YOUNG) หรือระดับค่าของกราฟ NOT VERY YOUNG จะมีค่าตรงข้ามกับกราฟของ VERY YOUNG ที่อยู่ตรงกลาง

ภาพที่ 11

แสดงกราฟของการขยายซ้ำ NOT YOUNG ด้วย NOT



ภาพที่ 11 แสดงให้เห็นกราฟของ NOT NOT YOUNG, NOT YOUNG และ YOUNG เปรียบเทียบกันบนแกน AGE จะเห็นว่ากราฟของ NOT NOT YOUNG ที่อยู่ด้านหน้าทุก ๆ ค่าของ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AGE จะมีระดับค่าเท่ากับ 1 - NOT YOUNG หรือระดับค่าของกราฟ NOT NOT YOUNG จะมีค่าตรงข้ามกับกราฟของ NOT YOUNG ที่อยู่ตรงกลางและจะมีค่าเท่ากับกราฟของ YOUNG ที่อยู่ด้านหลังสุด

**2.5.4.3 เปรียบเทียบผลการหาค่าความสัมพันธ์และผลการประยุกต์ใช้ความสัมพันธ์** จากอายุระหว่าง 0 ถึง 130 เมื่อหาความสัมพันธ์ เช่น YOUNG จากสมการที่ 2 การขยายความสัมพันธ์ตามตารางที่ 7 และการขยายซ้ำความสัมพันธ์ตามตารางที่ 8 จะได้ระดับการเป็นสมาชิกต่าง ๆ กัน สามารถเปรียบเทียบได้ดังตารางที่ 9 และภาพที่ 12

ตารางที่ 9

แสดงระดับการเป็นสมาชิกที่ได้จากการหาค่าความสัมพันธ์ การขยายค่าความสัมพันธ์ และการขยายซ้ำค่าความสัมพันธ์ YOUNG

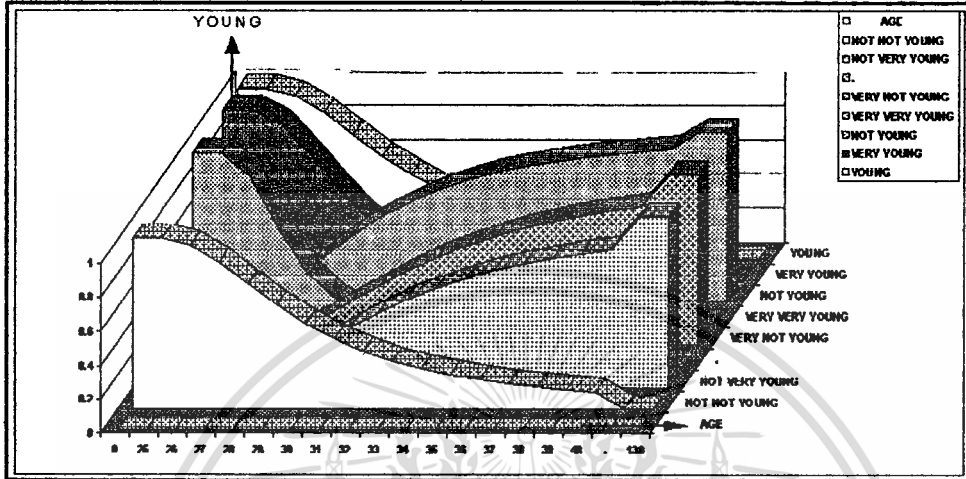
Age	YOUNG	VERY YOUNG	NOT YOUNG	VERY VERY YOUNG	VERY NOT YOUNG	NOT VERY YOUNG	NOT NOT YOUNG
00	1.000	1.000	0.000	1.000	0.000	0.000	1.000
25	1.000	1.000	0.000	1.000	0.000	0.000	1.000
26	0.962	0.925	0.038	0.586	0.001	0.074	0.962
27	0.862	0.743	0.138	0.552	0.019	0.256	0.862
28	0.735	0.540	0.265	0.291	0.070	0.459	0.735
29	0.610	0.372	0.390	0.138	0.152	0.627	0.610
30	0.500	0.250	0.500	0.062	0.250	0.750	0.500
31	0.410	0.168	0.590	0.028	0.348	0.831	0.410
32	0.338	0.114	0.662	0.013	0.438	0.885	0.338
33	0.281	0.078	0.719	0.006	0.516	0.921	0.281
34	0.236	0.055	0.764	0.003	0.583	0.944	0.236
35	0.200	0.040	0.800	0.001	0.640	0.960	0.200
36	0.171	0.029	0.829	0.000	0.687	0.970	0.171
37	0.148	0.021	0.852	0.000	0.725	0.978	0.148
38	0.129	0.016	0.871	0.000	0.758	0.983	0.129
39	0.113	0.012	0.887	0.000	0.786	0.987	0.113
40	0.100	0.010	0.900	0.000	0.810	0.990	0.100
130	0.000	0.000	1.000	0.000	1.000	1.000	0.000

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้เพื่อใช้ในการวิจัยและพัฒนาผลงานวิจัยที่เกี่ยวข้องกับการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 12

แสดงกราฟระดับการเป็นสมาชิกที่ได้จากการหาค่าความสัมพันธ์ การขยายค่าความสัมพันธ์ และการขยายซ้ำค่าความสัมพันธ์ YOUNG



จากภาพที่ 12 แสดงกราฟความสัมพันธ์ ดูจากหลังสุดมาด้านหน้าคือ YOUNG VERY\_YOUNG NOT\_YOUNG VERY\_VERY\_YOUNG VERY\_NOT\_YOUNG NOT\_VERY\_YOUNG NOT\_NOT\_YOUNG ตามลำดับ

2.5.4.4 การจำกัดค่าคำตอบ เพื่อคัดเลือกคำตอบให้ได้อยู่ในขอบเขตที่ต้องการอีกชั้นหนึ่ง การจำกัดค่าคำตอบทำได้โดยใช้ค่า "with" ตามด้วยค่า threshold (จำนวนเลขจุดทศนิยม) อันมีความหมายว่าต้องการคำตอบที่มีค่าเท่ากับหรือมากกว่าค่า threshold นั้น หรือ

$$\text{คำตอบ} = \text{ผลลัพธ์} \text{ เมื่อ } \text{ผลลัพธ์} \geq \text{threshold}$$

การจำกัดค่าคำตอบมีรูปแบบดังตารางที่ 10 และกราฟการจำกัดค่าคำตอบ เช่น ความสัมพันธ์ YOUNG แสดงได้ดังภาพที่ 13

ตารางที่ 10

แสดงรูปแบบการจำกัดค่าคำตอบความสัมพันธ์ที่เป็น "จำนวนเลข"

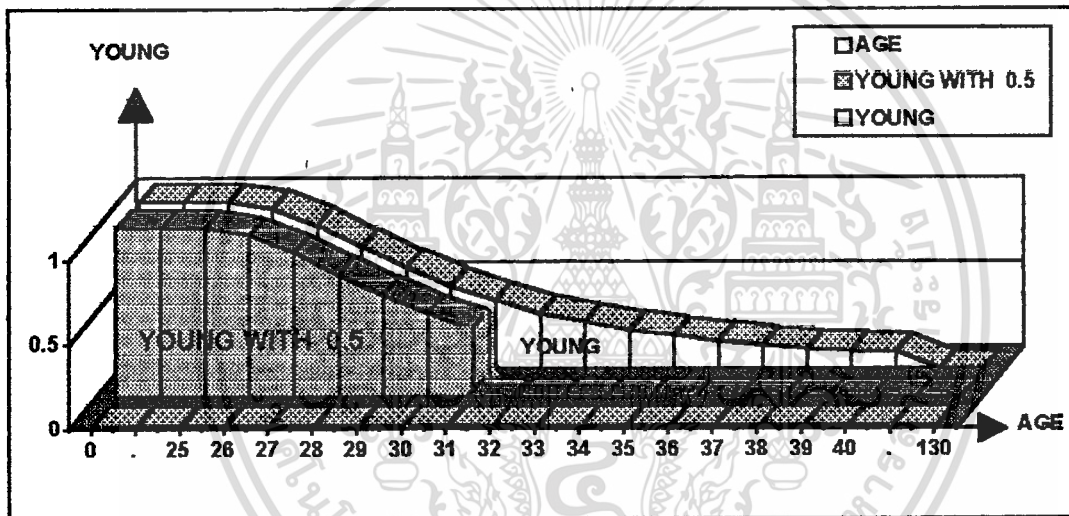
คำขยาย	พืชซีเทอม	with	threshold	สมาชิกฟังก์ชัน
	young		0.000	young $\geq$ threshold
	old	with	ถึง	old $\geq$ threshold
	tall		0.999	tall $\geq$ threshold
very	young		0.000	(young) <sup>2</sup> $\geq$ threshold
	old	with	ถึง	(old) <sup>2</sup> $\geq$ threshold
	tall		0.999	(tall) <sup>2</sup> $\geq$ threshold

## ตารางที่ 10 (ต่อ)

คำขยาย	พีชชีเทอม	with	threshold	สมาชิกฟังก์ชัน
	young		0.000	$(1 - \text{young}) \geq \text{threshold}$
not	old	with	ถึง	$(1 - \text{old}) \geq \text{threshold}$
	tall		0.999	$(1 - \text{tall}) \geq \text{threshold}$

ภาพที่ 13

แสดงกราฟของการจำกัดค่าคำตอบความสัมพันธ์ YOUNG ด้วยค่า WITH



ภาพที่ 13 แสดงให้เห็นกราฟของ YOUNG WITH '0.5' และ YOUNG เปรียบเทียบกันบน แกน AGE จะเห็นว่ากราฟของ YOUNG WITH '0.5' จะอยู่ในช่วงอายุ 0 ถึง 30 ปี

## บทที่ 3

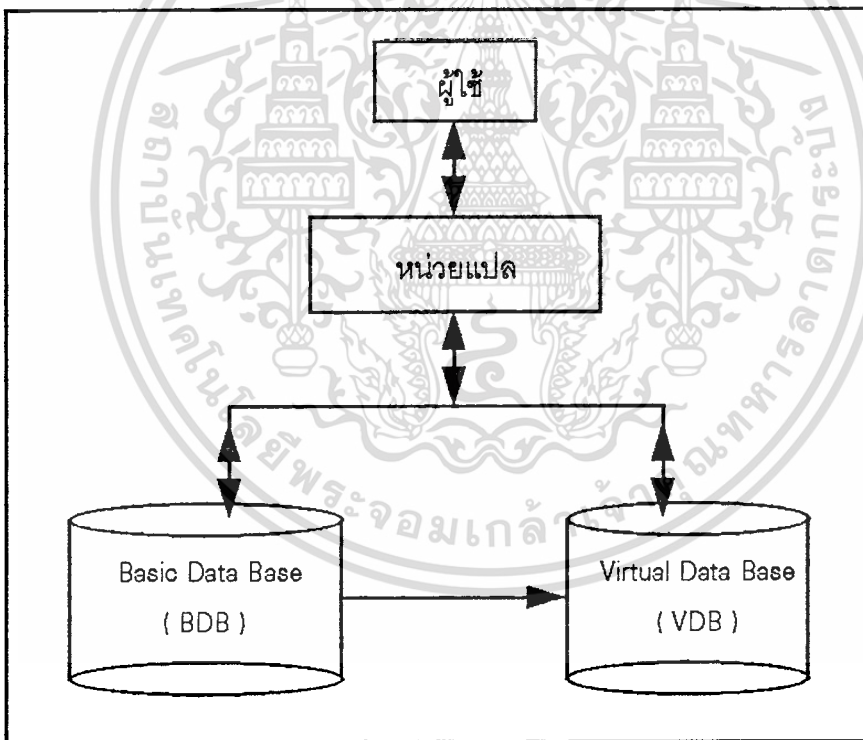
### การสร้างและการออกแบบระบบ FDB

#### 3. การสร้างและการออกแบบระบบ FDB

ระบบ FDB นั้นได้ออกแบบไว้เพื่อให้ใช้ได้ทั่ว ๆ ไป จึงไม่ได้เก็บชื่อฐานข้อมูลหลักและชื่อแอตทริบิวต์ของฐานข้อมูลหลักไว้ใน logic ของโปรแกรม จึงทำให้สามารถใช้ได้กับฐานข้อมูลหลักต่าง ๆ ที่มีในหน่วยงานได้

ภาพที่ 14

โครงสร้างพื้นฐานของระบบ FDB เพื่อสนับสนุนระบบผู้เชี่ยวชาญ



3.1 การสร้างและการออกแบบฐานข้อมูล ฐานข้อมูลมี 2 แบบ คือ ฐานข้อมูลหลักและฐานข้อมูลเสมือน ฐานข้อมูลทั้ง 2 แบบจะต้องเป็นระบบฐานข้อมูลแบบสัมพันธ์; ในที่นี้ใช้ระบบฐานข้อมูลแบบสัมพันธ์ ORACLE version 6

3.1.1 ฐานข้อมูล PERSON เป็นฐานข้อมูลหลักสำหรับกิจการอื่นใด ๆ ก็ได้แต่ต้องมีแอตทริบิวต์ที่เกี่ยวข้องกับความสัมพันธ์ที่ต้องการ เช่น ความสัมพันธ์ YOUNG ความสัมพันธ์เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OLD จะต้องมีแอตทริบิวต์ชื่อ Age และความสัมพันธ์ IALL จะต้องมีแอตทริบิวต์ชื่อ Height เป็นต้น ตัวอย่างฐานข้อมูล PERSON ได้แสดงไว้แล้วตามตารางที่ 1

**3.1.2 ฐานข้อมูล TEMPFDBX** เป็นฐานข้อมูลเสมือนที่ผู้ใช้กำหนดขึ้นมาระหว่างการใช้งานเพื่อให้ระบบ FDB สร้างขึ้นชั่วคราวสำหรับเก็บข้อมูลความสัมพันธ์ที่ผู้ใช้ต้องการ ฐานข้อมูล TEMPFDBX จะมีแอตทริบิวต์ไม่แน่นอน เพราะระบบจะสร้างขึ้นตามความความสัมพันธ์และคำสั่งของผู้ใช้ จะลบทิ้งเมื่อไม่ใช้ แต่สามารถเก็บไว้ใช้งานต่อไปได้ถ้าต้องการ

**3.2 การสร้างและการออกแบบโปรแกรมประมวล** การออกแบบโปรแกรมที่พัฒนาขึ้นมาไม่เก็บชื่อฐานข้อมูลหลักและแอตทริบิวต์ของฐานข้อมูลหลักไว้ใน logic ของโปรแกรมดังกล่าวมาแล้ว จึงทำให้สามารถใช้งานได้เกือบทุกฐานข้อมูล ทำให้เกิดความยืดหยุ่นในการใช้งานสูง โปรแกรมประกอบด้วยชุดโปรแกรมต่าง ๆ ดังนี้

**3.2.1 โปรแกรมประมวลผลหลัก** ทำหน้าที่ติดต่อกับระบบฐานข้อมูลแบบสัมพันธ์ เช่น การติดต่อขอเข้าไปทำงานในระบบฐานข้อมูลแบบสัมพันธ์ การจองหน่วยความจำเพื่อทำงาน การค้นหาข้อมูล การสร้างปรับปรุงแก้ไขลบทิ้งฐานข้อมูล การแจกจ่ายงานให้โปรแกรมย่อยรวมทั้งการออกรายงาน

**การทำงานของโปรแกรมประมวลผลหลัก** โดยติดต่อกับระบบฐานข้อมูลแบบสัมพันธ์ ถ้าติดต่อไม่ได้โปรแกรมจะเลิกทำงาน ถ้าติดต่อได้ก็แจกจ่ายงานให้โปรแกรมอื่น ๆ ในระบบทำงาน การจองพื้นที่สำหรับทำงาน เปิดปิดเพิ่มข้อมูล สร้างและปรับปรุงแก้ไขฐานข้อมูลเสมือนให้ทันสมัย จัดรูปร่างหน้าตารายงาน อ่านข้อมูลจากฐานข้อมูลหลักและฐานข้อมูลเสมือน การตรวจสอบเงื่อนไขตามที่ต้องการ การทำงานของโปรแกรมประมวลผลหลักแบ่งออกเป็นข้อ ๆ ดังนี้

**3.2.1.1 การสร้างเสริมและปรับปรุงฐานข้อมูลเสมือน** จากคำสั่ง create ที่ได้จากการแปลโดยโปรแกรมประมวลผลภาษาธรรมชาติ โปรแกรมประมวลผลหลักจะสร้างฐานข้อมูลเสมือนและปรับปรุงข้อมูลให้ทันสมัย ตัวอย่างคำสั่งและผลการสร้างฐานข้อมูลเสมือน เช่น

3.2.1.1.1 คำสั่ง create table tempfdbx (name , age, young) หลัง  
จากสร้างและปรับปรุงข้อมูลแล้วจะได้ฐานข้อมูลเหมือนแสดงได้ดังตารางที่ 11

ตารางที่ 11

แสดงความสัมพันธ์ YOUNG ใน VDB

Name	Age	YOUNG *
กนก	25	1.00
ปัญญา	28	0.74
มงคล	30	0.50
ราตรี	45	0.06
อนันต์	50	0.04

\*คำนวณจากสมการที่ (2)

ตารางที่ 11 แสดงการเก็บค่า Name Age และ YOUNG ที่โปรแกรมสร้างขึ้น โดยค่า  
ความสัมพันธ์ YOUNG ไม่มีในฐานข้อมูลหลักแต่ได้จากการคำนวณจากรูปร่างความสัมพันธ์ myOUNG

3.2.1.1.2 คำสั่ง create table tempfdbx (name , age , old) หลัง  
จากสร้างและปรับปรุงข้อมูลแล้วจะได้ฐานข้อมูลเหมือนแสดงได้ดังตารางที่ 12

ตารางที่ 12

แสดงความสัมพันธ์ OLD ใน VDB

Name	Age	OLD *
กนก	25	0.00
ปัญญา	28	0.00
มงคล	30	0.00
ราตรี	45	0.00
อนันต์	50	0.50

\*คำนวณจากสมการที่ (3)

ตารางที่ 12 แสดงการเก็บค่า Name Age และ OLD ที่โปรแกรมสร้างขึ้น โดยค่า  
ความสัมพันธ์ OLD ไม่มีในฐานข้อมูลหลักแต่ได้จากการคำนวณจากรูปร่างความสัมพันธ์ mOLD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.1.3 คำสั่ง create table tempfdbx (name, height, tall) หลัง  
จากสร้างและปรับปรุงข้อมูลแล้วจะได้ฐานข้อมูลเหมือนแสดงได้ดังตารางที่ 13

ตารางที่ 13  
แสดงความสัมพันธ์ TALL ใน VDB

Name	Height	TALL*
กนก	152	0.00
ปัญญา	155	0.00
มงคล	160	0.20
ราตรี	167	0.50
อนันต์	178	0.84

\*คำนวณจากสมการที่ (4)

ตารางที่ 13 แสดงการเก็บค่า Name Height และ TALL ที่โปรแกรมสร้างขึ้น โดยค่า  
ความสัมพันธ์ TALL ไม่มีในฐานข้อมูลหลักแต่ได้จากการคำนวณจากความสัมพันธ์ mTALL

3.2.1.1.4 คำสั่ง create table tempfdbx (name, age, proximity)  
หลังจากสร้างและปรับปรุงข้อมูลแล้วจะได้ฐานข้อมูลเหมือนแสดงได้ดังตารางที่ 14

ตารางที่ 14  
แสดงความสัมพันธ์ PROXIMITY ของ Age ใน VDB

Name	Age	PROXIMITY *
กนก	25	0.328
ปัญญา	28	0.640
มงคล	30	1.000
ราตรี	45	0.035
อนันต์	50	0.012

\*คำนวณจากสมการที่ (1)

ตารางที่ 14 แสดงการเก็บค่า Name Age และ PROXIMITY ที่โปรแกรมสร้างขึ้น โดยค่า  
ความสัมพันธ์ PROXIMITY ไม่มีในฐานข้อมูลหลักแต่ได้จากการคำนวณจากความสัมพันธ์

emPROXIMITYสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.1.5 คำสั่ง create table tempfdbx (name, height, proximity) หลังจากสร้างและปรับปรุงข้อมูลแล้วจะได้ฐานข้อมูลเหมือนแสดงได้ดังตารางที่ 15

ตารางที่ 15

แสดงความสัมพันธ์ PROXIMITY ของ Height ใน VDB

Name	Height	PROXIMITY *
กนก	152	0.168
ปัญญา	155	0.328
มงคล	160	1.000
ราตรี	167	0.328
อนันต์	178	0.018

\*คำนวณจากสมการที่ (1)

ตารางที่ 15 แสดงการเก็บค่า Name Height และ PROXIMITY ที่โปรแกรมสร้างขึ้น โดยค่าความสัมพันธ์ PROXIMITY ไม่มีในฐานข้อมูลหลักแต่ได้จากการคำนวณจากความสัมพันธ์ PROXIMITY

3.2.1.1.6 คำสั่ง create table tempfdbx (name) หลังจากสร้างและปรับปรุงข้อมูลแล้วจะได้ฐานข้อมูลเหมือนแสดงได้ดังตารางที่ 16

ตารางที่ 16

แสดงความสัมพันธ์ Name ใน VDB

Name
กนก
ปัญญา
มงคล
ราตรี
อนันต์

ตารางที่ 16 แสดงการเก็บค่า Name ที่โปรแกรมสร้างขึ้น โดย Name เป็นแอตทริบิวต์ที่

มีข้อความเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.1.7 คำสั่ง `create table tempfdbx (name , province)` หลังจากสร้างและปรับปรุงข้อมูลแล้วจะได้ฐานข้อมูลเหมือนแสดงได้ดังตารางที่ 17

ตารางที่ 17  
แสดงความสัมพันธ์ Province ใน VDB

Name	Province
กนก	ลพบุรี,ตรัง
บัญญัติ	อ่างทอง
มงคล	-
ราตรี	นนทบุรี,สงขลา
อนันต์	กรุงเทพ,ชลบุรี,ตาก

ตารางที่ 17 แสดงการเก็บค่า Name และ Province ที่โปรแกรมสร้างขึ้น โดย Name เป็นแอตทริบิวต์ที่มีข้อความเดียวและ Province เป็นแอตทริบิวต์ที่มีข้อความหลายข้อความต่อกัน

3.2.1.2. การอ่านข้อมูล มี 2 กรณีคือ อ่านฐานข้อมูลหลักเพื่อนำมาสร้างฐานข้อมูลเสมือนและอ่านฐานข้อมูลเสมือนเพื่อนำมาออกรายงาน

3.2.1.3 การแจกจ่ายงาน มี 2 กรณีคือ

3.2.1.3.1 ถ้ามีเงื่อนไขเป็น “ความสัมพันธ์จำนวนเลขและต้องขยายหรือขยายซ้ำเงื่อนไข” คือมีคำขยาย NOT หรือ VERY เช่น “is not young” หรือ “is very young” ก็ส่งไปให้โปรแกรมประมวลผลหาค่าตัวเลขคำนวณผลลัพธ์แล้วรับคำตอบกลับมา

3.2.1.3.2 ถ้ามีเงื่อนไขเป็น “ข้อความและต้องเลือกคำตอบ” เช่น “live in 'กรุงเทพ' ” หรือ “name is 'อนันต์' ” ก็ส่งให้โปรแกรมประมวลผลข้อความคำนวณผลลัพธ์แล้วรับคำตอบกลับมา

3.2.1.4 การจำกัดคำตอบ มี 2 กรณีคือ

3.2.1.4.1 เงื่อนไขความสัมพันธ์จำนวนเลขมีค่า “with” ในกรณีนี้ที่เงื่อนไขมีค่า “with” ตามด้วย threshold อันเป็นเงื่อนไขการจำกัดคำตอบ เช่น “young with 0.5” ก็นำคำตอบครั้งสุดท้ายไปตรวจสอบเงื่อนไขการจำกัดคำตอบ

3.2.1.4.2 เงื่อนไขความสัมพันธ์ข้อความมี “การเปรียบเทียบ” ในกรณีนี้ที่เงื่อนไขมีการเปรียบเทียบ เช่น “name is 'อนันต์' ” ก็นำคำตอบที่ได้จากโปรแกรมประมวลผลข้อความไปตรวจสอบเงื่อนไขการจำกัดคำตอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.5 การออกรายงาน เมื่อตรวจสอบเงื่อนไขรวมทั้งเรคคอร์ดปรากฏว่า ถ้ายอมรับได้ก็ส่งพิมพ์รายงานทางจอภาพ ถ้ายอมรับไม่ได้ก็จะลบส่วนที่เก็บไว้รอส่งพิมพ์ทั้งหมดทิ้ง

3.2.2 โปรแกรมประมวลผลภาษาธรรมชาติ ทำหน้าที่เป็นตัวรับคำสั่งจากผู้ใช้เป็นภาษาธรรมชาติ นำมาวิเคราะห์ความต้องการและเงื่อนไข แล้วแปลคำสั่งภาษาธรรมชาติให้เป็นภาษา SQL เพื่อใช้สำหรับกระทำกับข้อมูลและฐานข้อมูลเมื่อแปลเรียบร้อยแล้วก็ส่งให้โปรแกรมประมวลผลหลัก

การทำงานของโปรแกรมประมวลผลภาษาธรรมชาติ โดยรับคำสั่งภาษาธรรมชาติจากผู้ใช้แล้วนำมาแจกแจง วิเคราะห์ ตัดทิ้ง เพิ่มเติม หาค่าที่มีความหมายเหมือนกันมาแทน แล้วจัดเรียงประโยคใหม่เป็นชุดคำสั่ง SQL ตามความต้องการของคำถาม

หลักการแปลภาษาธรรมชาติเป็นภาษา SQL แบ่งออกเป็น 2 ประเภทคือ ถ้าคำถามเกี่ยวกับพีชชีที่เป็นจำนวนเลขจะแปลเป็นชุดคำสั่ง SQL จำนวน 6 รูปแบบและถ้าคำถามเกี่ยวกับพีชชีที่เป็นข้อความจะแปลเป็นชุดคำสั่ง SQL จำนวน 4 รูปแบบ เพื่อประกอบการอธิบายจึงสมมติชื่อฐานข้อมูลเสมือนเพียงชื่อเดียวคือ "tempfdbx" การแปลภาษาธรรมชาติจะแปลดังต่อไปนี้

3.2.2.1 การแปลคำถามภาษาธรรมชาติประเภทที่มีพีชชีเป็นจำนวนเลข จะรับคำถามภาษาธรรมชาติแล้วแปลเป็นชุดคำสั่งภาษา SQL จำนวน 6 รูปแบบ เช่น

3.2.2.1.1 คำถาม "list the men from person who age is young" แปลแล้วจะได้ข้อความดังนี้

#### 3.2.2.1.1.1 SELECT-FORM

```
select name, age, young
from tempfdbx
```

#### 3.2.2.1.1.2 CREATE-FORM

```
create table
tempfdbx (name , age , young )
as select name , age , age
from person
```

#### 3.2.2.1.1.3 UPDATE-FORM

```
update tempfdbx
set young = NULL
update tempfdbx
set young = 1.0
```

```

where ((age >= 0) and (age <= 25))
update tempfdbx
set    young = power(1+power(((age-25)/5),2),-1)
where ((age > 25) and (age <=130))

```

#### 3.2.2.1.1.4 ALTER-FORM

```

alter table    tempfdbx
modify        young number(4,3)

```

#### 3.2.2.1.1.5 DROP-FORM

```

drop table    tempfdbx

```

#### 3.2.2.1.1.6 CONDITION-FORM

```

age >? !? young

```

#### 3.2.2.1.2 คำถาม "list the men from person who is tall" แปล

แล้วจะได้ข้อความดังนี้

##### 3.2.2.1.2.1 SELECT-FORM

```

select name, tall
from    tempfdbx

```

##### 3.2.2.1.2.2 CREATE-FORM

```

create table
tempfdbx    (name , height , tall )
as select   name , height , height
from        person

```

##### 3.2.2.1.2.3 UPDATE-FORM

```

update tempfdbx
set    tall = NULL
update tempfdbx
set    tall = 0.0
where  ((height >= 0) and (height <= 155))
update tempfdbx
set    tall = power(1+power(((height-155)/10),-2),-1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ where ((height > 155) and (height <=250)) ระเบียบข้อดำเนินการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2.2.1.2.4 ALTER-FORM

```
alter table    tempfdbx
modify        tall number(4,3)
```

## 3.2.2.1.2.5 DROP-FORM

```
drop table    tempfdbx
```

## 3.2.2.1.2.6 CONDITION-FORM

```
?!> tall
```

3.2.2.1.3 คำถาม "list the men from person that is tall with .5"

แปลแล้วจะได้ข้อความดังนี้

## 3.2.2.1.3.1 SELECT-FORM

```
select name, tall
from    tempfdbx
```

## 3.2.2.1.3.2 CREATE-FORM

```
create table
tempfdbx    (name , height , tall )
as select   name , height , height
from        person
```

## 3.2.2.1.3.3 UPDATE-FORM

```
update tempfdbx
set    tall = NULL

update tempfdbx

set    tall = 0.0

where ((height >= 0) and (height <= 155))

update tempfdbx

set    tall = power(1+power(((height-155)/10),-2),-1)

where ((height > 155) and (height <=250))
```

## 3.2.2.1.3.4 ALTER-FORM

```
alter table    tempfdbx
modify        tall number(4,3)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2.2.1.3.5 DROP-FORM

```
drop table tempfdbx
```

## 3.2.2.1.3.6 CONDITION-FORM

```
?!? tall , >= 0.5
```

3.2.2.1.4 คำถาม "list the man from person whose height is not tall" แปลแล้วจะได้ข้อความดังนี้

## 3.2.2.1.4.1 SELECT-FORM

```
select name, height , tall
from tempfdbx
```

## 3.2.2.1.4.2 CREATE-FORM

```
create table
tempfdbx (name , height , tall )
as select name , height , height
from person
```

## 3.2.2.1.4.3 UPDATE-FORM

```
update tempfdbx
set tall = NULL
update tempfdbx
set tall = 0.0
where ((height >= 0) and (height <= 155))
update tempfdbx
set tall = power(1+power(((height-155)/10),-2),-1)
where ((height > 155) and (height <=250))
```

## 3.2.2.1.4.4 ALTER-FORM

```
alter table tempfdbx
modify tall number(4,3)
```

## 3.2.2.1.4.5 DROP-FORM

```
drop table tempfdbx
```

## 3.2.2.1.4.6 CONDITION-FORM

```
height?!?not?tall
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.1.5 คำถาม "list the men from person whose height is not tall with .5" แปลแล้วจะได้ข้อความดังนี้

#### 3.2.2.1.5.1 SELECT-FORM

```
select name, height , tall
from tempfdbx
```

#### 3.2.2.1.5.2 CREATE-FORM

```
create table
tempfdbx (name , height , tall )
as select name , height , height
from person
```

#### 3.2.2.1.5.3 UPDATE-FORM

```
update tempfdbx
set tall = NULL
update tempfdbx
set tall = 0.0
where ((height >= 0) and (height <= 155))
update tempfdbx
set tall = power(1+power(((height-155)/10),-2),-1)
where ((height > 155) and (height <=250))
```

#### 3.2.2.1.5.4 ALTER-FORM

```
alter table tempfdbx
modify tall number(4,3)
```

#### 3.2.2.1.5.5 DROP-FORM

```
drop table tempfdbx
```

#### 3.2.2.1.5.6 CONDITION-FORM

```
height?1?not?tall , >= 0.5
```

**3.2.2.2 การแปลคำถามภาษาธรรมชาติประเภทที่มีพีชคณิตเป็นข้อความ จะรับคำถามภาษาธรรมชาติแล้วแปลเป็นชุดคำสั่งภาษา SQL จำนวน 4 รูปแบบ เช่น**

3.2.2.2.1 คำถาม "list the people from person whose name is 'มงคล'" **แปลแล้วจะได้ข้อความดังนี้**

เอกสารนี้เป็นทรัพย์สินส่วนราชการสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2.2.2.1.1 SELECT-FORM

```
select name
from tempfdbx
```

## 3.2.2.2.1.2 CREATE-FORM

```
create table
tempfdbx (name )
as select name
from person
```

## 3.2.2.2.1.2 DROP-FORM

```
drop table tempfdbx
```

## 3.2.2.2.1.3 CONDITION-FORM

```
name?!?'มงคล'
```

3.2.2.2.2 คำถาม "select name from person who live in 'กรุงเทพ' "

แปลแล้วจะได้ข้อความดังนี้

## 3.2.2.2.2.1 SELECT-FORM

```
select name , province
from tempfdbx
```

## 3.2.2.2.2.2 CREATE-FORM

```
create table
tempfdbx (name ,province )
as select name , province
from person
```

## 3.2.2.2.2.2 DROP-FORM

```
drop table tempfdbx
```

## 3.2.2.2.2.3 CONDITION-FORM

```
province?!?'กรุงเทพ'
```

### 3.2.3 โปรแกรมประมวลผลหาค่าตัวเลข ทำหน้าที่ประมวลผลหาค่าตอบพีซีที่เป็น

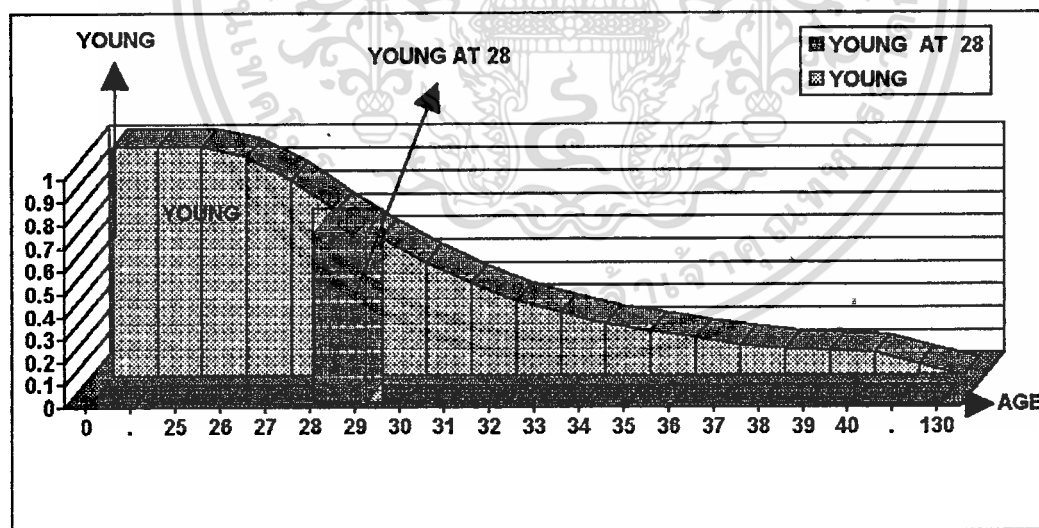
จำนวนเลขและมีเงื่อนไขขยาย เช่น จากความสัมพันธ์ที่มี NOT หรือ VERY มาขยาย โดยรับค่าที่ต้องคำนวณจากโปรแกรมประมวลผลหลักมาคำนวณค่าที่ต้องการ เมื่อบริการเรียบร้อยแล้วส่งผลลัพธ์กลับคืนไปยังโปรแกรมประมวลผลหลัก

ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรมประมวลผลหาค่าตัวเลข มี 2 แบบคือ การหาค่าเงื่อนไข การขยายและเงื่อนไขการขยายซ้ำและการประมวลผลจะประมวลผลจากทางขวาไปทางซ้ายของ ส่วนเงื่อนไข เพื่อประกอบการอธิบายจึงขอใช้อายุ 28 ปีและความสัมพันธ์ YOUNG เป็นตัวอย่าง พร้อมทั้งขออธิบายการหาค่าส่วนเงื่อนไขความสัมพันธ์ปกติที่อายุ 28 ปี (ข้อ 3.2.3.1) รวมทั้งการ ตรวจสอบเงื่อนไขการจำกัดค่าคำตอบที่อายุ 28 ปี (ข้อ 3.2.3.5) เนื่องจากมีความเกี่ยวข้องกัน อย่างมากไว้ในหัวข้อนี้ด้วย คือ

3.2.3.1 การหาค่าส่วนเงื่อนไขความสัมพันธ์ปกติ (ทำโดยโปรแกรม ประมวลผลหลัก) เช่น "is young" โดยโปรแกรมประมวลผลหลักอ่านค่าความสัมพันธ์ YOUNG อันเนื่องมาจากอายุ 28 ปีจาก tempfdbx ได้ค่า YOUNG = 0.735 จะไม่ส่งไปโปรแกรมประมวล ผลหาค่าตัวเลข แต่จะเก็บไว้เป็นคำตอบ กราฟความสัมพันธ์ YOUNG ที่อายุ 28 ปี แสดงได้ ดังภาพที่ 15

ภาพที่ 15  
แสดงกราฟของความสัมพันธ์ YOUNG ที่อายุ 28 ปี



ภาพที่ 15 แสดงให้เห็นค่ากราฟแท่งของ YOUNG ที่อายุ 28 และกราฟพื้นผิวของ YOUNG เปรียบเทียบกันบนแกน AGE ค่าของ YOUNG ที่อายุ 28 ได้มาจากการคำนวณและ อ่านโดยโปรแกรมประมวลผลหลัก

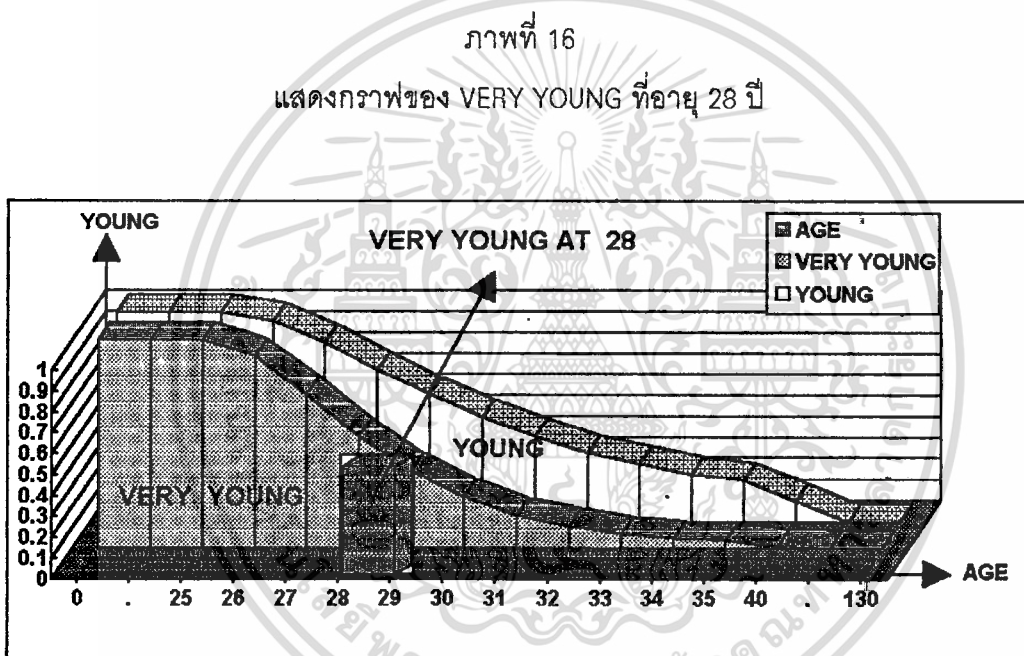
3.2.3.2 การหาค่าส่วนเงื่อนไขขยายความสัมพันธ์ คือส่วนเงื่อนไขที่มี NOT หรือ VERY มาขยายค่าความสัมพันธ์จากเงื่อนไขปกติ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.2.1 การหาค่าเมื่อขยายความสัมพันธ์ด้วย VERY เช่น "is very young" โปรแกรมประมวลผลหาค่าตัวเลขรับค่า YOUNG = 0.735 จากโปรแกรมประมวลผลหลัก (ข้อ 3.2.3.1) มาคำนวณหาค่าผลลัพธ์ดังนี้

$$\text{very}(\text{young}) = (\text{young})^2 \text{ หรือ}$$

$$\text{very}(0.735) = (0.735)^2$$

ได้ผลลัพธ์เท่ากับ 0.540 แล้วส่งผลลัพธ์กลับคืนโปรแกรมประมวลผลหลัก ผลลัพธ์การขยายความสัมพันธ์ YOUNG ด้วย VERY ที่อายุ 28 ปี แสดงได้ดังภาพที่ 16



ภาพที่ 16 แสดงให้เห็นค่ากราฟแท่งของ VERY YOUNG ที่อายุ 28 กราฟพื้นผิวของ VERY YOUNG และกราฟพื้นผิวของ YOUNG เปรียบเทียบกันบนแกน AGE ค่าของ VERY YOUNG ที่อายุ 28 ได้มาจากการคำนวณโดยโปรแกรมประมวลผลหาค่าตัวเลข

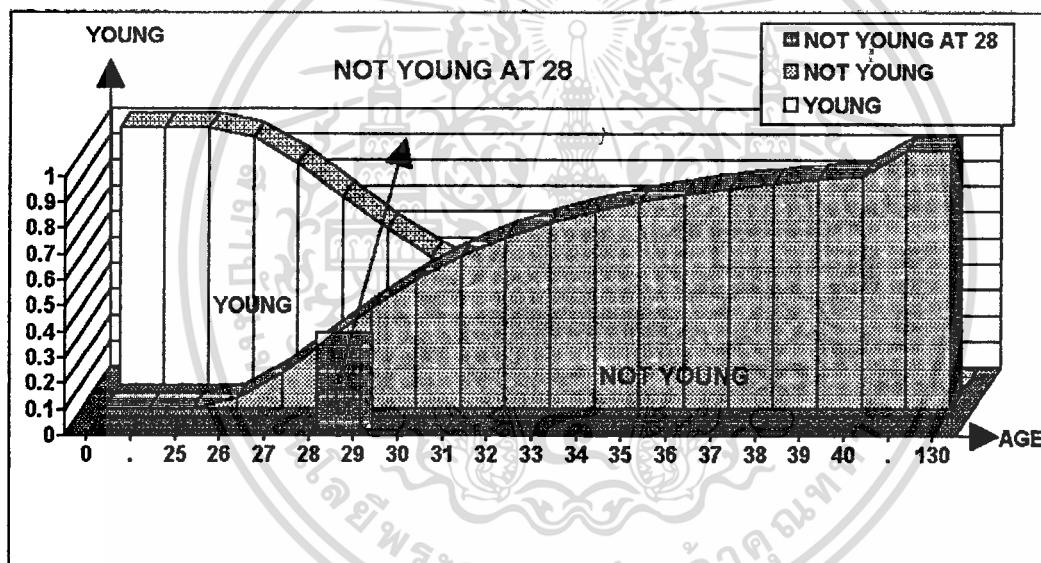
3.2.3.2.2 การหาค่าเมื่อขยายความสัมพันธ์ด้วย NOT เช่น "is not young" โปรแกรมประมวลผลค่าตัวเลขรับค่า YOUNG = 0.735 จากโปรแกรมประมวลผลหลัก (ข้อ 3.2.3.1) มาคำนวณหาค่าผลลัพธ์ดังนี้

$$\begin{aligned} \text{not(young)} &= (1 - \text{young}) \text{ หรือ} \\ \text{not}(0.735) &= (1 - 0.735) \end{aligned}$$

ได้ผลลัพธ์เท่ากับ 0.265 แล้วส่งผลลัพธ์กลับคืนโปรแกรมประมวลผลหลัก ผลลัพธ์การขยายความสัมพันธ์ YOUNG ด้วย NOT ที่อายุ 28 ปี แสดงได้ดังภาพที่ 17

ภาพที่ 17

แสดงกราฟของ NOT YOUNG ที่อายุ 28 ปี



ภาพที่ 17 แสดงให้เห็นค่ากราฟแท่งของ NOT YOUNG ที่อายุ 28 กราฟพื้นผิวของ NOT YOUNG และกราฟพื้นผิวของ YOUNG เปรียบเทียบกันบนแกน AGE ค่าของ NOT YOUNG ที่อายุ 28 ได้มาจากการคำนวณโดยโปรแกรมประมวลผลค่าตัวเลข

3.2.3.3 การหาค่าส่วนเงินไขขยายเข้าความสัมพันธ์ คือส่วนเงินไขที่มี NOT หรือ VERY มาขยายเงินไขขยายอีกทีหนึ่ง มีรายละเอียดดังต่อไปนี้

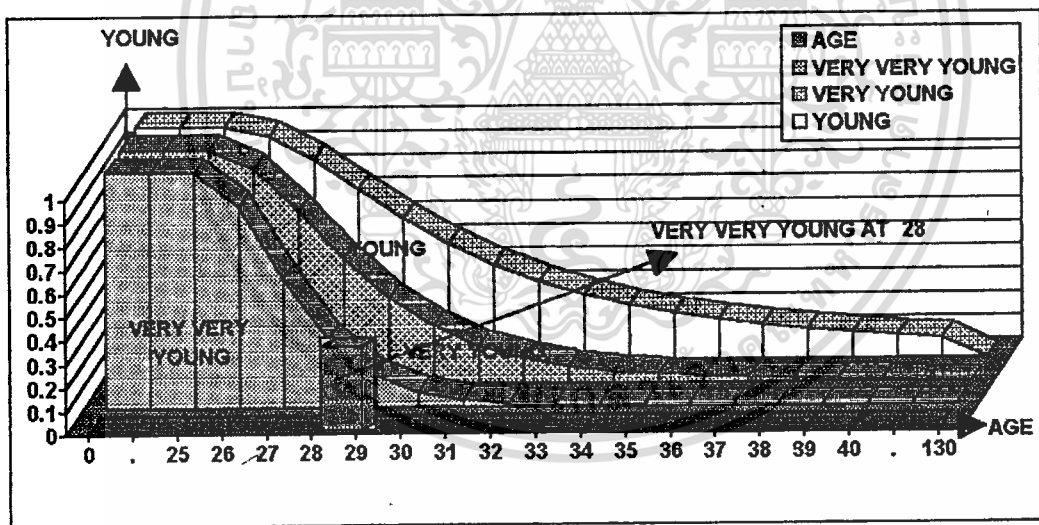
3.2.3.3.1 การหาค่าเมื่อขยายเข้า VERY ด้วย VERY เช่น 'is very very young' โปรแกรมประมวลหาค่าตัวเลขรับค่า VERY YOUNG = 0.540 จากโปรแกรมประมวลผลหลัก (ข้อ 3.2.3.2.1) มาคำนวณหาค่าผลลัพธ์ดังนี้

$$\text{very(very young)} = (\text{very young})^2 \text{ หรือ}$$

$$\text{very}(0.540) = (0.540)^2$$

ได้ผลลัพธ์เท่ากับ 0.291 แล้วส่งผลลัพธ์กลับคืนโปรแกรมประมวลผลหลัก ผลลัพธ์การขยายความสัมพันธ์ VERY YOUNG ด้วย VERY ที่อายุ 28 ปี แสดงได้ดังภาพที่ 18

ภาพที่ 18  
แสดงกราฟของ VERY VERY YOUNG ที่อายุ 28 ปี



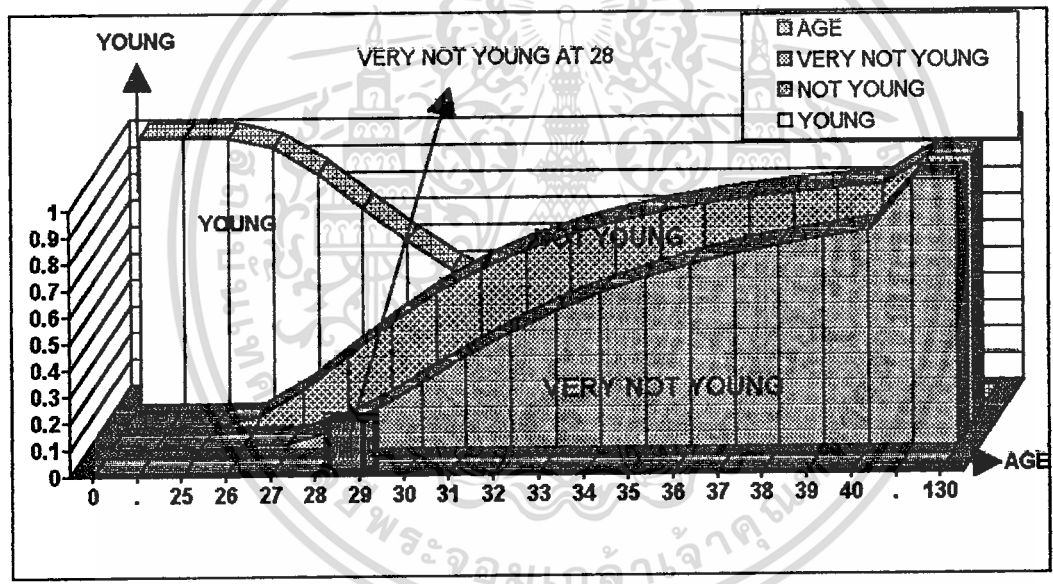
ภาพที่ 18 แสดงให้เห็นค่ากราฟแท่งของ VERY VERY YOUNG ที่อายุ 28 กราฟพื้นผิวของ VERY VERY YOUNG, VERY YOUNG และกราฟพื้นผิวของ YOUNG เปรียบเทียบกันบนแกน AGE ค่าของ VERY VERY YOUNG ที่อายุ 28 ได้มาจากการคำนวณโดยโปรแกรมประมวลผลหาค่าตัวเลข

3.2.3.3.2 การหาค่าเมื่อขยายซ้ำ NOT ด้วย VERY เช่น "is very not young" โปรแกรมประมวลผลาค่าตัวเลขรับค่า NOT YOUNG = 0.265 จากโปรแกรมประมวลผลหลัก (ข้อ 3.2.3.2.2) มาคำนวณหาค่าผลลัพธ์ดังนี้

$$\begin{aligned} \text{very(not young)} &= (\text{not young})^2 \text{ หรือ} \\ \text{very}(0.265) &= (0.265)^2 \end{aligned}$$

ได้ผลลัพธ์เท่ากับ 0.070 แล้วส่งผลลัพธ์กลับคืนโปรแกรมประมวลผลหลัก ผลลัพธ์การขยายความสัมพันธ์ NOT YOUNG ด้วย VERY ที่อายุ 28 ปี แสดงได้ดังภาพที่ 19

ภาพที่ 19  
แสดงกราฟของ VERY NOT YOUNG ที่อายุ 28 ปี



ภาพที่ 19 แสดงให้เห็นค่ากราฟแท่งของ VERY NOT YOUNG ที่อายุ 28 กราฟพื้นผิวของ VERY NOT YOUNG, NOT YOUNG และกราฟพื้นผิวของ YOUNG เปรียบเทียบกันบนแกน AGE ค่าของ VERY NOT YOUNG ที่อายุ 28 ได้มาจากการคำนวณโดยโปรแกรมประมวลผลาค่าตัวเลข

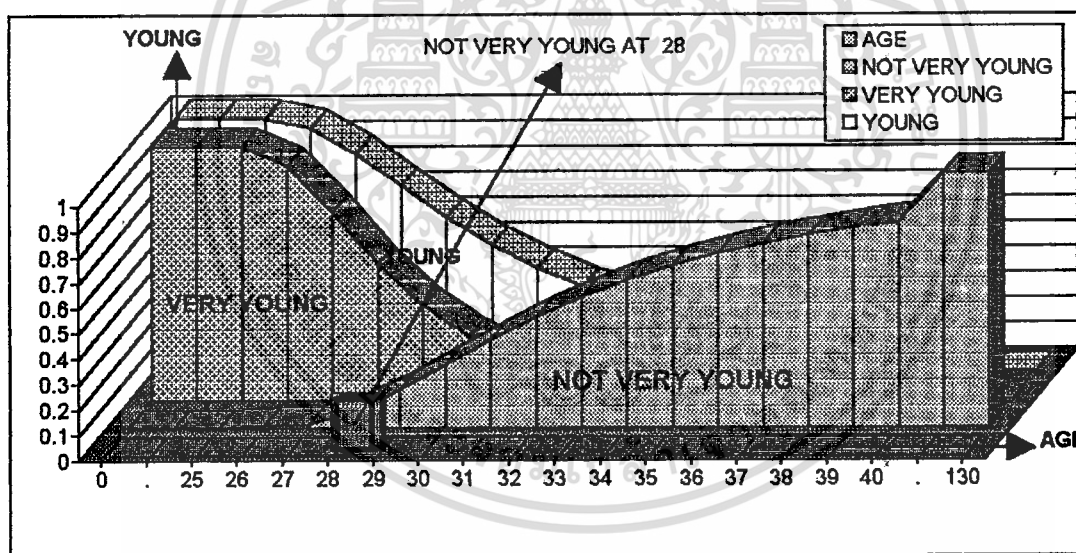
3.2.3.3.3 การหาค่าเมื่อขยายซ้ำ VERY ด้วย NOT เช่น "is not very young" โปรแกรมประมวลผลาค่าตัวเลขรับค่า VERY YOUNG = 0.540 จากโปรแกรมประมวลผลหลัก (ข้อ 3.2.3.2.1) มาคำนวณหาค่าผลลัพธ์ดังนี้

$$\begin{aligned} \text{not(very young)} &= (1 - \text{very young}) \quad \text{หรือ} \\ \text{not(0.540)} &= (1 - 0.540) \end{aligned}$$

ได้ผลลัพธ์เท่ากับ 0.460 แล้วส่งผลลัพธ์กลับคืนโปรแกรมประมวลผลหลัก ผลลัพธ์การขยายความสัมพันธ์ VERY YOUNG ด้วย NOT ที่อายุ 28 ปี แสดงได้ดังภาพที่ 20

หมายเหตุ เครื่องคำนวณได้ 0.459 เพราะเกิดจากการปัดจำนวนเลขหลังจุดทศนิยม

ภาพที่ 20  
แสดงกราฟของ NOT VERY YOUNG ที่อายุ 28 ปี



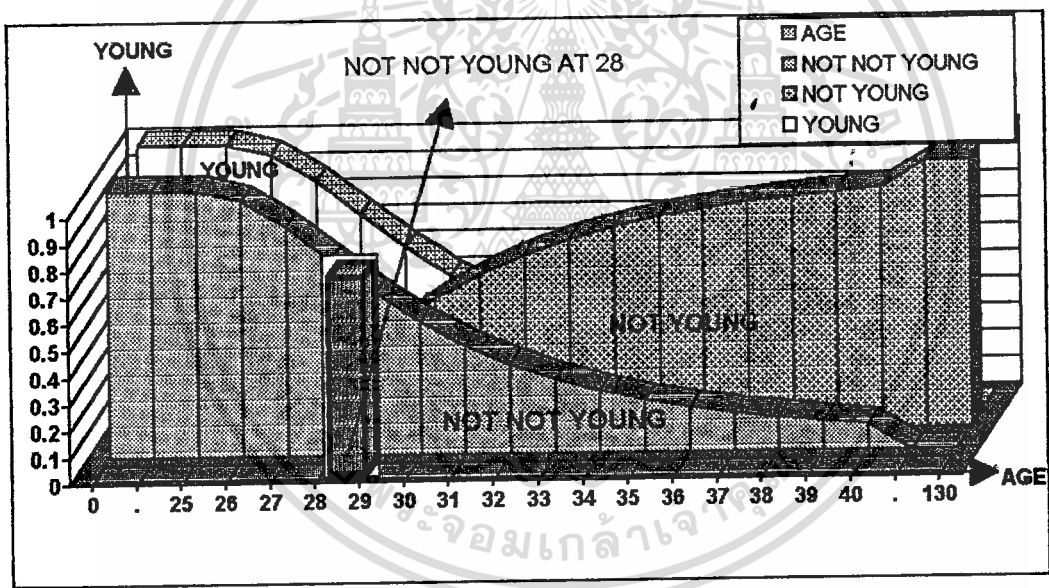
ภาพที่ 20 แสดงให้เห็นค่ากราฟแท่งของ NOT VERY YOUNG ที่อายุ 28 กราฟพื้นผิวของ NOT VERY YOUNG, VERY YOUNG และกราฟพื้นผิวของ YOUNG เปรียบเทียบกันบนแกน AGE ค่าของ NOT VERY YOUNG ที่อายุ 28 ได้มาจากการคำนวณโดยโปรแกรมประมวลผลาค่าตัวเลข

3.2.3.3.4 การหาค่าเมื่อขยายซ้ำ NOT ด้วย NOT เช่น "is not not young" โปรแกรมประมวลผลาค่าตัวเลขรับค่า NOT YOUNG = 0.265 จากโปรแกรมประมวลผลหลัก (ข้อ 3.2.3.2.2) มาคำนวณหาค่าผลลัพธ์ดังนี้

$$\begin{aligned} \text{not(not young)} &= (1 - \text{not young}) \quad \text{หรือ} \\ \text{not}(0.265) &= (1 - 0.265) \end{aligned}$$

ได้ผลลัพธ์เท่ากับ 0.735 แล้วส่งผลลัพธ์กลับคืนโปรแกรมประมวลผลหลัก ผลลัพธ์การขยายความสัมพันธ์ NOT YOUNG ด้วย NOT ที่อายุ 28 ปี แสดงได้ดังภาพที่ 21

ภาพที่ 21  
แสดงกราฟของ NOT NOT YOUNG ที่อายุ 28 ปี



ภาพที่ 21 แสดงให้เห็นค่ากราฟแท่งของ NOT NOT YOUNG ที่อายุ 28 กราฟพื้นผิวของ NOT NOT YOUNG, NOT YOUNG และกราฟพื้นผิวของ YOUNG เปรียบเทียบกับบนแกน AGE ค่าของ NOT NOT YOUNG ที่อายุ 28 ได้มาจากการคำนวณโดยโปรแกรมประมวลผลาค่าตัวเลข

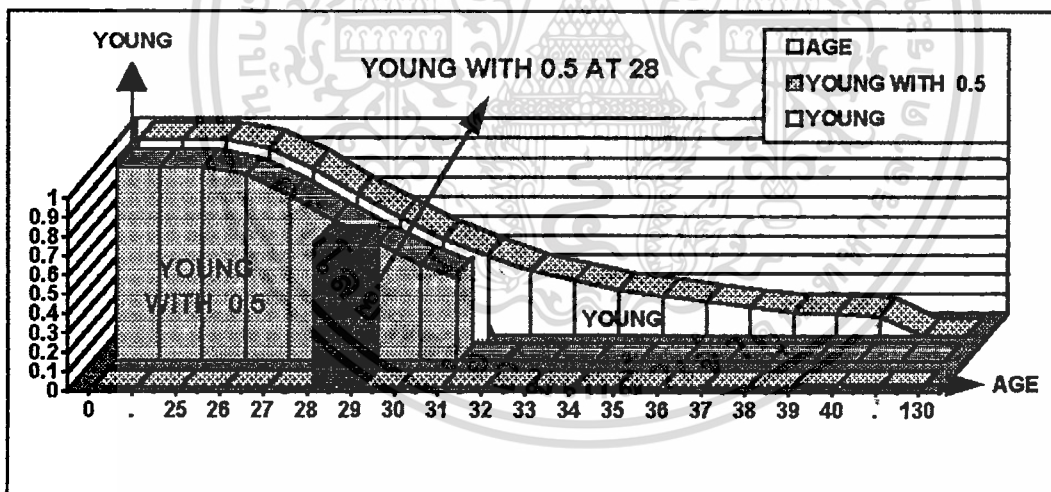
3.2.3.4 การตรวจสอบเงื่อนไขการจำกัดค่าคำตอบ (ทำโดยโปรแกรมประมวลผลหลัก) ถ้าส่วนเงื่อนไขมี "with" ตามด้วย threshold เช่น ส่วนเงื่อนไขคือ "is young with 0.5" โปรแกรมประมวลผลหลักนำค่าที่คำนวณได้ไปตรวจสอบค่าว่าเป็นความจริงหรือไม่

ที่อายุ 28 ปี ได้ค่าความสัมพันธ์ YOUNG = 0.735 (ข้อ 3.2.3.1) และค่า threshold = 0.5 มาคำนวณหาค่าผลลัพธ์ดังนี้

$$\begin{array}{lcl} \text{ผลลัพธ์} & \geq & \text{threshold} \\ 0.735 & \geq & 0.5 \end{array}$$

เมื่อเปรียบเทียบกันแล้วปรากฏว่าเป็นจริงยอมรับได้ จึงได้คำตอบ = 0.735 กราฟของการตรวจสอบเงื่อนไขการจำกัดค่าคำตอบของความสัมพันธ์ YOUNG with 0.5 ที่อายุ 28 ปี แสดงได้ดังภาพที่ 22

ภาพที่ 22  
แสดงกราฟของ YOUNG with 0.5 ที่อายุ 28 ปี



ภาพที่ 22 แสดงให้เห็นค่ากราฟแท่งของ YOUNG with 0.5 ที่อายุ 28 กราฟพื้นผิวของ YOUNG และกราฟพื้นผิวของ YOUNG with 0.5 เปรียบเทียบกันบนแกน AGE ค่าของ YOUNG with 0.5 ที่อายุ 28 ได้มาจากการคำนวณโดยโปรแกรมประมวลผลหลัก

3.2.4 โปรแกรมประมวลผลข้อความ ทำหน้าที่ประมวลผลหาคำตอบที่เป็นข้อความและมีเงื่อนไขเปรียบเทียบ โดยรับค่าและเงื่อนไขจากโปรแกรมประมวลผลหลักมา ตรวจสอบว่าตรงตามเงื่อนไขหรือไม่ เมื่อประมวลผลเรียบร้อยแล้วส่งผลลัพธ์คืนกลับไปยังโปรแกรมประมวลผลหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรมประมวลผลข้อความ มี 2 แบบคือ การหาค่าเงื่อนไขความสัมพันธ์ข้อความที่ค่าแอดตริบิวต์มีข้อความเดียวและเงื่อนไขความสัมพันธ์ข้อความที่ค่าแอดตริบิวต์มีข้อความหลายข้อความต่อกัน

**3.2.4.1 การหาค่าส่วนเงื่อนไขความสัมพันธ์ข้อความที่ค่าแอดตริบิวต์มีข้อความเดียว** แอดตริบิวต์ชนิดนี้ เช่น Name ดังกล่าวมาแล้ว ส่วนที่เป็นเงื่อนไข เช่น "name is 'มงคล' " โปรแกรมประมวลผลข้อความรับเงื่อนไข "name is 'มงคล' " มาทำการเปรียบเทียบดังนี้

(ส่วนของเงื่อนไข, ความยาวส่วนของเงื่อนไข, ค่าแอดตริบิวต์ข้อความเดียว)

หรือ

('มงคล', 6, กนก)

ได้ผลลัพธ์ไม่เท่ากันแล้วส่งผลลัพธ์กลับคืนโปรแกรมประมวลผลหลัก

**3.2.4.2 การหาค่าเงื่อนไขความสัมพันธ์ข้อความที่ค่าแอดตริบิวต์มีหลายข้อความต่อกัน** แอดตริบิวต์ชนิดนี้ เช่น Province ดังกล่าวมาแล้ว ส่วนที่เป็นเงื่อนไข เช่น "province is 'กรุงเทพ' " โปรแกรมประมวลผลข้อความรับเงื่อนไข "province is 'กรุงเทพ' " มาทำการเปรียบเทียบดังนี้

(ส่วนของเงื่อนไข, ความยาวส่วนของเงื่อนไข, ค่าแอดตริบิวต์หลายข้อความต่อกัน)

หรือ

('กรุงเทพ', 9, ลพบุรี, ตรีง)

ได้ผลลัพธ์ไม่เท่ากันแล้วส่งผลลัพธ์กลับคืนโปรแกรมประมวลผลหลัก

**3.3 Header File** ทำหน้าที่เก็บคำศัพท์และความสัมพันธ์ต่าง ๆ

**3.3.1 เก็บคำศัพท์** เก็บคำศัพท์ต่าง ๆ คือ เก็บคำที่มีความหมายเหมือนกัน เช่น man มีความหมายเหมือนกับ name เก็บคำที่ต้องการตัดทิ้ง เช่น a, an, the เป็นต้น

**3.3.2 เก็บความสัมพันธ์** เก็บความสัมพันธ์ต่าง ๆ เช่น ความสัมพันธ์ PROXIMITY, YOUNG, OLD และ IALL เป็นต้น

## บทที่ 4

### การทดลองและผลการทดลอง

4. การทดลองและผลการทดลอง การทดลองแบ่งออกเป็น 2 ชุดใหญ่ ๆ คือ

ชุดแรกการทดลองหาค่าความสัมพันธ์เกี่ยวกับจำนวนเลข เช่น ทดลองหาค่าความสัมพันธ์ต่าง ๆ คือความสัมพันธ์ PROXIMITY, YOUNG, OLD และ TALL ในเงื่อนไขปกติ ทดลองหาค่าความสัมพันธ์ในเงื่อนไขขยายด้วย NOT และ VERY ทดลองหาค่าความสัมพันธ์ในเงื่อนไขขยายซ้ำด้วย NOT และ VERY รวมทั้งทดลองการจำกัดคำตอบด้วย WITH

ชุดที่สองการทดลองหาค่าความสัมพันธ์เกี่ยวกับข้อความ เช่น ทดลองหาค่าความสัมพันธ์ข้อความที่แอดตริบิวต์มีเพียงข้อความเดียวและทดลองหาค่าความสัมพันธ์ข้อความที่แอดตริบิวต์มีหลายข้อความต่อกัน

#### 4.1 การทดลองหาค่าความสัมพันธ์เกี่ยวกับจำนวนเลข

4.1.1 การทดลองหาค่าความสัมพันธ์ในเงื่อนไขปกติ ความสัมพันธ์ที่ใช้คือความสัมพันธ์ PROXIMITY, YOUNG, OLD และ TALL

4.1.1.1 คำสั่งแสดงการใช้  $m_{\text{PROXIMITY}}$  เช่น "list the man from person whose age is proximity 30" ผลการทดลองจะได้คำตอบทุกค่าตามตารางที่ 18

ตารางที่ 18

แสดงคำตอบจากคำสั่ง "list the man from person whose age is proximity 30"

Name	Age	$m_{\text{PROXIMITY}}^*$
กนก	25	0.328
ปัญญา	28	0.640
มงคล	30	1.000
ราตรี	45	0.035
อนันต์	50	0.012

\*คำนวณจากสมการที่ (1)

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ แสดงให้เห็นว่าทุกคนเป็นสมาชิกของความสัมพันธ์ PROXIMITY หรือด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“ใกล้เคียง” โดยแต่ละคนมีอายุใกล้เคียง 30 ปีมากน้อยไม่เท่ากัน เช่น คนที่มีอายุ 30 ปีจะมีระดับค่าใกล้เคียงเท่ากับ 1 แสดงว่าอายุ 30 ปีนั้นเท่ากับจำนวนเลขที่กำหนด สำหรับอายุอื่น ๆ ระดับค่าใกล้เคียงจะมากขึ้นอยู่กับอายุใกล้ 30 ปีมากน้อยเท่าใด จำนวนเลขจุดทศนิยมที่มีค่ามากจะใกล้เคียงมากกว่าจำนวนเลขจุดทศนิยมที่มีค่าน้อย

**4.1.1.2 คำสั่งแสดงการใช้ mTALL** เช่น "list the men from person whose height is tall" ผลการทดลองจะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 19

ตารางที่ 19

แสดงคำตอบจากคำสั่ง "list the men from person whose height is tall"

Name	Height	m <sub>TALL</sub> *
กนก	152	0.00
บัญญัติ	155	0.00
มงคล	160	0.20
ราตรี	167	0.50
อนันต์	178	0.84

\*คำนวณจากสมการที่ (4)

จากตารางที่ 19 แสดงให้เห็นค่า m<sub>TALL</sub> หรือ “สูง” โดยคนที่มีระดับค่า m<sub>TALL</sub> เท่ากับ 0.00 แสดงว่าไม่ได้เป็นสมาชิกของ m<sub>TALL</sub> จะไม่นำมาคิดในเทอมสูง สำหรับคนที่มีระดับค่า m<sub>TALL</sub> อื่น ๆ ที่เป็นจำนวนเลขจุดทศนิยมระหว่าง 0 ถึง 1 แสดงว่าเป็นสมาชิกที่ถูกต้องไม่สมบูรณ์ คือถ้าพูดในเทอม “สูง” แล้วจะอยู่ในลักษณะเกือบสูงหรือค่อนข้างสูงเท่านั้น

**4.1.1.3 คำสั่งแสดงการใช้ mYOUNG** เช่น "list the man from person whose age is young" ผลการทดลองจะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 20

ตารางที่ 20

แสดงคำตอบจากคำสั่ง "list the man from person whose age is young"

Name	Age	$m_{\text{YOUNG}}^*$
กนก	25	1.00
ปัญญา	28	0.74
มงคล	30	0.50
ราตรี	45	-0.06
อนันต์	50	0.04

\*คำนวณจากสมการที่ (2)

จากตารางที่ 20 แสดงให้ค่า  $m_{\text{YOUNG}}$  หรือ “หนุ่ม” โดยคนที่มีระดับค่า  $m_{\text{YOUNG}}$  เท่ากับ 1.00 แสดงว่าเป็นสมาชิกที่ถูกต้องสมบูรณ์คือถ้าพูดในทอม “หนุ่ม” แล้วถือว่าเป็นคนหนุ่ม สำหรับคนที่มีระดับค่า  $m_{\text{YOUNG}}$  อื่น ๆ ที่เป็นเลขจุดทศนิยมระหว่าง 0 ถึง 1 แสดงว่าเป็นสมาชิกที่ถูกต้องไม่สมบูรณ์ คือถ้าพูดในทอม “หนุ่ม” แล้วถือว่าเกินเลยความเป็นหนุ่มไปแล้ว

4.1.1.4 คำสั่งแสดงการใช้  $m_{\text{OLD}}$  เช่น "list the man from person whose age is old" ผลการทดลองจะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 21

ตารางที่ 21

แสดงคำตอบจากคำสั่ง "list the man from person whose age is old"

Name	Age	$m_{\text{OLD}}^*$
กนก	25	0.00
ปัญญา	28	0.00
มงคล	30	0.00
ราตรี	45	0.00
อนันต์	50	0.50

\*คำนวณจากสมการที่ (3)

จากตารางที่ 21 แสดงให้เห็นค่า  $m_{\text{OLD}}$  หรือ “แก่” โดยคนที่มีระดับ  $m_{\text{OLD}}$  เท่ากับ 0.00 แสดงว่าไม่ได้เป็นสมาชิกของ  $m_{\text{OLD}}$  สำหรับ  $m_{\text{OLD}}$  ค่าอื่น ๆ ที่เป็นเลขจุดทศนิยม เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่าง 0 ถึง 1 แสดงว่าเป็นสมาชิกที่ถูกต้องไม่สมบูรณ์ หรือถ้าพูดในเทอม “แก่” แล้วจะอยู่ในลักษณะเกือบแก่หรือค่อนข้างแก่

4.1.1.5 คำสั่งแสดงการไม่ใช่ Age เช่น "find the people from person who is young" ผลการทดลองจะได้คำตอบที่ไม่มี Age ตามตารางที่ 22

ตารางที่ 22

แสดงคำตอบจากคำสั่ง "find the people from person who is young"

Name	$m_{\text{YOUNG}}^*$
กนก	1.00
ปัญญา	0.74
มงคล	0.50
ราตรี	0.06
อนันต์	0.04

\*คำนวณจากสมการที่ (2)

จากตารางที่ 22 แสดงให้เห็นว่าสามารถถามคำถามด้วย YOUNG โดยคำตอบไม่มีค่า Age ก็ได้

4.1.2 การทดลองหาค่าความสัมพันธ์ในเงื่อนไขขยาย โดยจะทดลองขยายความสัมพันธ์ YOUNG ด้วย NOT และ VERY

4.1.2.1 คำสั่งแสดงการใช้ NOT ขยาย  $m_{\text{YOUNG}}$  เช่น "find the people from person that age is not young" ผลการทดลองจะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 23

## ตารางที่ 23

แสดงคำตอบจากคำสั่ง "find the people from person that age is not young"

Name	Age	$m_{\text{NOT\_YOUNG}}^*$
กนก	25	0.000
ปัญญา	28	0.265
มงคล	30	0.500
ราตรี	45	0.941
อนันต์	50	0.962

\*คำนวณจากสมการที่ (2) และตารางที่ (7)

จากตารางที่ 23 แสดงให้เห็นค่า NOT YOUNG จะเห็นว่าเมื่อขยายแล้วจะเกิดเทอม "ไม่หนุ่ม" โดยคนที่มีระดับค่าการเป็นสมาชิกของ NOT YOUNG เท่ากับ 0.000 แสดงว่าไม่เป็นสมาชิกของเทอม NOT YOUNG คือไม่จัดเข้าพวกคนไม่หนุ่ม สำหรับคนที่มีระดับค่าการเป็นสมาชิกของ NOT YOUNG ระหว่าง 0 ถึง 1 แสดงว่าเป็นสมาชิกของเทอม NOT YOUNG ที่ถูกต้องไม่สมบูรณ์ คือมีความไม่หนุ่มอยู่บางส่วนเท่านั้น

4.1.2.2 คำสั่งแสดงการใช้ VERY ขยาย  $m_{\text{YOUNG}}$  เช่น "find the people from person that age is very young" ผลการทดลองจะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 24

## ตารางที่ 24

แสดงคำตอบจากคำสั่ง "find the people from person that age is very young"

Name	Age	$m_{\text{VERY\_YOUNG}}^*$
กนก	25	1.000
ปัญญา	28	0.540
มงคล	30	0.250
ราตรี	45	0.003
อนันต์	50	0.001

\*คำนวณจากสมการที่ (2) และตารางที่ (7)

จากตารางที่ 24 แสดงให้ค่า VERY YOUNG จะเห็นว่าเมื่อขยายแล้วจะเกิดเทอม "หนุ่มมาก" โดยคนที่มีระดับค่าการเป็นสมาชิกของ VERY YOUNG เท่ากับ 1.000 แสดงว่าเป็นสมาชิกมากเกินกว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ VERY YOUNG ที่ถูกต้องสมบูรณ์ คือถ้าพูดถึงเทอมหนุ่มมากแล้วถือว่าเป็นจริง สำหรับคนที่มีระดับค่าการเป็นสมาชิกของ VERY YOUNG อยู่ระหว่าง 0 กับ 1 แสดงว่าเป็นสมาชิกของ VERY YOUNG ถูกต้องไม่สมบูรณ์ คือมีความเป็นหนุ่มมากอยู่บางส่วนเท่านั้น

**4.1.2.3 คำสั่งแสดงการใช้ NOT ขยาย mOLD** เช่น "find the people from person that age is not old" ผลการทดลองจะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 25

ตารางที่ 25

แสดงคำตอบจากคำสั่ง "find the people from person that age is not old"

Name	Age	m <sub>NOT_OLD</sub> *
กนก	25	1.000
ปัญญา	28	1.000
มงคล	30	1.000
ราตรี	45	1.000
อนันต์	50	0.500

\*คำนวณจากสมการที่ (3) และตารางที่ (7)

จากตารางที่ 25 แสดงให้เห็นค่า NOT OLD จะเห็นว่าเมื่อขยายแล้วจะเกิดเทอม "ไม่แก่" โดยคนที่มีระดับค่าการเป็นสมาชิกของ NOT OLD เท่ากับ 1.000 แสดงว่าเป็นสมาชิกของเทอม NOT OLD ถูกต้องสมบูรณ์ คือไม่แก่แน่นอน สำหรับคนที่มีระดับค่าการเป็นสมาชิกของ NOT OLD ระหว่าง 0 ถึง 1 แสดงว่าเป็นสมาชิกของเทอม NOT OLD ที่ถูกต้องไม่สมบูรณ์ คือมีความไม่แก่อยู่บางส่วนเท่านั้น

**4.1.2.4 คำสั่งแสดงการใช้ VERY ขยาย mOLD** เช่น "find the people from person that age is very old" ผลการทดลองจะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 26

ตารางที่ 26

แสดงคำตอบจากคำสั่ง "find the people from person that age is very old"

Name	Age	$m_{VERY\_OLD}^*$
กนก	25	0.000
ปัญญา	28	0.000
มงคล	30	0.000
ราตรี	45	0.000
อนันต์	50	0.250

\*คำนวณจากสมการที่ (3) และตารางที่ (7)

จากตารางที่ 26 แสดงให้เห็นค่า VERY OLD จะเห็นว่าเมื่อขยายแล้วจะเกิดเทอม "แก่ มาก" โดยคนที่มีระดับค่าการเป็นสมาชิกของ VERY OLD เท่ากับ 0.000 แสดงว่าไม่เป็นสมาชิกของเทอม VERY OLD จะไม่นำมาพิจารณา สำหรับคนที่มีระดับค่าการเป็นสมาชิกของ VERY OLD ระหว่าง 0 ถึง 1 แสดงว่าเป็นสมาชิกของเทอม VERY OLD ที่ถูกต้องไม่สมบูรณ์ คือมีความแก่มากอยู่บางส่วนเท่านั้น

4.1.2.4 คำสั่งแสดงการใช้ NOT ขยาย  $m_{TALL}$  เช่น "find the people from person that height is not tall" ผลการทดลองจะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 27

ตารางที่ 27

แสดงคำตอบจากคำสั่ง "find the people from person that height is not tall"

Name	Height	$m_{NOT\_TALL}^*$
กนก	152	1.000
ปัญญา	155	1.000
มงคล	160	0.800
ราตรี	167	0.500
อนันต์	178	0.159

\*คำนวณจากสมการที่ (4) และตารางที่ (7)

จากตารางที่ 27 แสดงให้ค่า NOT TALL จะเห็นว่าเมื่อขยายแล้วจะเกิดเทอม "ไม่สูง" โดยคนที่มีระดับค่าการเป็นสมาชิกของ NOT TALL เท่ากับ 1.000 แสดงว่าเป็นสมาชิกของเทอมด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NOT TALL ที่ถูกต้องสมบูรณ์คือไม่สูงจริง ๆ สำหรับคนที่มีระดับค่าการเป็นสมาชิกของ NOT TALL ระหว่าง 0 ถึง 1 แสดงว่าเป็นสมาชิกของเทอม NOT TALL ที่ถูกต้องไม่สมบูรณ์ คือมีความไม่สูงอยู่บางส่วนเท่านั้น

4.1.2.5 คำสั่งแสดงการใช้ VERY ขยาย mTALL เช่น "find the people from person that height is very tall" ผลการทดลองจะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 28

ตารางที่ 28

แสดงคำตอบจากคำสั่ง "find the people from person that height is very tall"

Name	Height	$m_{VERY\_TALL}^*$
กนก	152	0.000
บัญญัติ	155	0.000
มงคล	160	0.040
ราตรี	167	0.250
อนันต์	178	0.707

\*คำนวณจากสมการที่ (4) และตารางที่ (7)

จากตารางที่ 28 แสดงให้เห็นค่า VERY TALL จะเห็นว่าเมื่อขยายแล้วจะเกิดเทอม "สูงมาก" โดยคนที่มีระดับค่าการเป็นสมาชิกของ VERY TALL เท่ากับ 0.000 แสดงว่าไม่เป็นสมาชิกของเทอม VERY TALL ไม่ต้องนำมาพิจารณา สำหรับคนที่มีระดับค่าการเป็นสมาชิกของ VERY TALL ระหว่าง 0 ถึง 1 แสดงว่าเป็นสมาชิกของเทอม VERY TALL ที่ถูกต้องไม่สมบูรณ์ คือมีความสูงมากอยู่บางส่วนเท่านั้น

4.1.3 การทดลองการหาค่าความสัมพันธ์ในเงื่อนไขขยายซ้ำ โดยจะทดลองขยายความสัมพันธ์ VERY YOUNG, NOT YOUNG ด้วย NOT และ VERY

4.1.3.1 คำสั่งแสดงการใช้ VERY ขยาย VERY YOUNG เช่น "find the people from person that age is very very young" ผลการทดลองจะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตารางที่ 29

แสดงคำตอบจากคำสั่ง "find the people from person that age is very very young"

Name	Age	$m_{\text{VERY\_VERY\_YOUNG}}^*$
กนก	25	1.000
ปัญญา	28	0.029
มงคล	30	0.062
ราตรี	45	0.000
อนันต์	50	0.000

\*คำนวณจากสมการที่ (2) และตารางที่ (8)

จากตารางที่ 29 แสดงให้ค่า VERY VERY YOUNG จะเห็นว่าเมื่อขยายแล้วจะเกิดเทอม "หนุ่มมากมาก" หรือ VERY VERY YOUNG โดยคนที่มีระดับค่าการเป็นสมาชิกของ VERY VERY YOUNG เท่ากับ 0.000 แสดงว่าไม่เป็นสมาชิกของเทอม VERY VERY YOUNG ไม่ต้องนำมาพิจารณา คนที่มีระดับค่าการเป็นสมาชิกของ VERY VERY YOUNG ระหว่าง 0 ถึง 1 แสดงว่าเป็นสมาชิกของเทอม VERY VERY YOUNG ที่ถูกต้องไม่สมบูรณ์ คือมีความหนุ่มมากมากอยู่บางส่วนเท่านั้น สำหรับคนที่มีระดับค่าการเป็นสมาชิกของ VERY VERY YOUNG เท่ากับ 1.000 แสดงว่าเป็นสมาชิกของเทอม VERY VERY YOUNG ที่ถูกต้องสมบูรณ์ คือเป็นคนหนุ่มมากมากจริง

4.1.3.2 คำสั่งแสดงการใช้ VERY ขยาย NOT YOUNG เช่น "find the people from person that age is very not young" ผลการทดลองจะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 30

### ตารางที่ 30

แสดงคำตอบจากคำสั่ง "find the people from person that age is very not young"

Name	Age	$m_{\text{VERY\_NOT\_YOUNG}}^*$
กนก	25	0.000
ปัญญา	28	0.070
มงคล	30	0.250
ราตรี	45	0.885
อนันต์	50	0.925

\*คำนวณจากสมการที่ (2) และตารางที่ (8)

เอกสารนี้ จากตารางที่ 30 แสดงให้เห็นค่า VERY NOT YOUNG จะเห็นว่าเมื่อขยายแล้วจะเกิดขนด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทอม "หนุ่มไม่มาก" หรือ VERY NOT YOUNG โดยคนที่มีระดับค่าการเป็นสมาชิกของ VERY NOT YOUNG เท่ากับ 0.000 แสดงว่าไม่เป็นสมาชิกของเทอม VERY NOT YOUNG สำหรับคนที่มีระดับค่าการเป็นสมาชิกของ VERY NOT YOUNG ระหว่าง 0 ถึง 1 แสดงว่าเป็นสมาชิกของเทอม VERY NOT YOUNG ที่ถูกต้องไม่สมบูรณ์

**4.1.3.3 คำสั่งแสดงการใช้ NOT ขยาย VERY YOUNG** เช่น "find the people from person that age is not very young" ผลการทดลองจะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 31

ตารางที่ 31

แสดงคำตอบจากคำสั่ง "find the people from person that age is not very young"

Name	Age	$m_{\text{NOT\_VERY\_YOUNG}}^*$
กนก	25	0.000
ปัญญา	28	0.460
มงคล	30	0.750
ราตรี	45	0.996
อนันต์	50	0.998

\*คำนวณจากสมการที่ (2) และตารางที่ (8)

จากตารางที่ 31 แสดงให้ค่า NOT VERY YOUNG จะเห็นว่าเมื่อขยายแล้วจะเกิดเทอม "ไม่หนุ่มไม่มาก" หรือ NOT VERY YOUNG โดยคนที่มีระดับค่าการเป็นสมาชิกของ NOT VERY YOUNG เท่ากับ 0.000 แสดงว่าไม่เป็นสมาชิกของเทอม NOT VERY YOUNG สำหรับคนที่มีระดับค่าการเป็นสมาชิกของ NOT VERY YOUNG ระหว่าง 0 ถึง 1 แสดงว่าเป็นสมาชิกของเทอม NOT VERY YOUNG ที่ถูกต้องไม่สมบูรณ์

**4.1.3.4 คำสั่งแสดงการใช้ NOT ขยาย NOT YOUNG** เช่น "find the people from person that age is not not young" ผลการทดลองจะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 32

## ตารางที่ 32

แสดงคำตอบจากคำสั่ง "find the people from person that age is not not young"

Name	Age	$m_{\text{NOT\_NOT\_YOUNG}}^*$
กนก	25	1.000
ปัญญา	28	0.735
มงคล	30	0.500
ราตรี	45	0.059
อนันต์	50	0.038

\*คำนวณจากสมการที่ (2) และตารางที่ (8)

จากตารางที่ 32 แสดงให้ค่า NOT NOT YOUNG จะเห็นว่าเมื่อขยายแล้วจะเกิดเทอม "ไม่หนุ่ม" หรือ NOT NOT YOUNG โดยคนที่มีระดับค่าการเป็นสมาชิกของ NOT NOT YOUNG เท่ากับ 1.000 แสดงว่าเป็นสมาชิกของเทอม NOT NOT YOUNG ที่ถูกต้องสมบูรณ์ สำหรับคนที่มีระดับค่าการเป็นสมาชิกของ NOT NOT YOUNG ระหว่าง 0 ถึง 1 แสดงว่าเป็นสมาชิกของเทอม NOT NOT YOUNG ที่ถูกต้องไม่สมบูรณ์

4.1.4 การทดลองการจำกัดค่าคำตอบ การจำกัดค่าคำตอบเพื่อให้ได้คำตอบบางค่า และให้อยู่ในขอบเขตจำกัดสามารถกระทำได้โดยใช้ค่า "with" ตามด้วยค่า threshold (จำนวนเลขจุดทศนิยม) มีความหมายว่าระดับค่าการเป็นสมาชิกมากกว่าหรือเท่ากับค่า threshold ที่กำหนด threshold มีค่าระหว่าง 0 และ 1

4.1.4.1 คำสั่งแสดงการใช้ค่า WITH เช่น "list the man from person whose age is young with 0.5" ผลการทดลองจะได้คำตอบบางค่าตามตารางที่ 33

## ตารางที่ 33

แสดงคำตอบจากคำสั่ง "list the man from person whose age is young with 0.5"

Name	Age	$m_{\text{YOUNG}}^*$
กนก	25	1.00
ปัญญา	28	0.74
มงคล	30	0.50

\*คำนวณจากสมการที่ (2) และตารางที่ (10)

จากตารางที่ 33 แสดงให้เห็นคำตอบ โดยเลือกตอบรายการที่มีค่า YOUNG เท่ากับหรือมากกว่า 0.5 เท่านั้น ส่วนงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4.2 คำสั่งแสดงการใช้คำ VERY และ WITH เช่น "find the man from person whose age is very young with 0.5" ผลการทดลองจะได้คำตอบบางค่าตามตารางที่ 34

ตารางที่ 34

แสดงคำตอบจากคำสั่ง "find the man from person whose age is very young with 0.5"

Name	Age	$m_{VERY\_YOUNG}^*$
กนก	25	1.000
ปัญญา	28	0.547

\*คำนวณจากสมการที่ (2) ตารางที่ (7) และตารางที่ (10)

จากตารางที่ 34 แสดงให้เห็นคำตอบ โดยเลือกตอบรายการที่มีค่า VERY YOUNG เท่ากับหรือมากกว่า 0.5 เท่านั้น

4.1.4.3 คำสั่งแสดงการใช้คำ NOT และ WITH เช่น "find the man from person whose age is not young with 0.3" ผลการทดลองจะได้คำตอบบางค่าตามตารางที่ 35

ตารางที่ 35

แสดงคำตอบจากคำสั่ง "find the man from person whose age is not young with 0.3"

Name	Age	$m_{NOT\_YOUNG}^*$
มงคล	30	0.50
ราตรี	45	0.94
อนันต์	50	0.96

\*คำนวณจากสมการที่ (2) ตารางที่ (7) และตารางที่ (10)

จากตารางที่ 35 แสดงให้เห็นคำตอบโดยเลือกตอบรายการที่มีค่า NOT YOUNG เท่ากับหรือมากกว่า 0.3 เท่านั้น

4.1.4.4 คำสั่งแสดงการใช้ VERY VERY และ WITH เช่น "find the man from person whose age is very very young with 0.01" ผลการทดลองจะได้คำตอบบางค่าตามตารางที่ 36

ตารางที่ 36

แสดงคำตอบจากคำสั่ง "find the man from person whose age is very very young with 0.01"

Name	Age	$m_{VERY\_VERY\_YOUNG}^*$
กนก	25	1.000
ปัญญา	28	0.029
มงคล	30	0.062

\*คำนวณจากสมการที่ (2) ตารางที่ (8) และตารางที่ (10)

จากตารางที่ 36 แสดงให้เห็นคำตอบโดยเลือกตอบรายการที่มีค่า VERY VERY YOUNG เท่ากับหรือมากกว่า 0.01 เท่านั้น

4.1.4.5 คำสั่งแสดงการใช้ VERY NOT และ WITH เช่น "find the people from person that age is very not young with 0.6" ผลการทดลองจะได้คำตอบบางค่าตามตารางที่ 37

ตารางที่ 37

แสดงคำตอบจากคำสั่ง "find the people from person that age is very not young with 0.6"

Name	Age	$m_{VERY\_NOT\_YOUNG}^*$
ราตรี	45	0.885
อนันต์	50	0.925

\*คำนวณจากสมการที่ (2) ตารางที่ (8) และตารางที่ (10)

จากตารางที่ 37 แสดงให้เห็นคำตอบ โดยเลือกตอบรายการที่มีค่า VERY NOT YOUNG เท่ากับหรือมากกว่า 0.6 เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**4.1.4.6 คำสั่งแสดงการใช้คำ NOT VERY และ WITH** เช่น "find man from person that age is not very young with 0.5" ผลการทดลองจะได้คำตอบบางค่าตามตารางที่ 38

ตารางที่ 38

แสดงคำตอบจากคำสั่ง "find man from person that age is not very young with 0.5"

Name	Age	$m_{\text{NOT\_VERY\_YOUNG}}^*$
มงคล	30	0.750
ราตรี	45	0.996
อนันต์	50	0.998

\*คำนวณจากสมการที่ (2) ตารางที่ (8) และตารางที่ (10)

จากตารางที่ 38 แสดงให้เห็นคำตอบ โดยเลือกตอบรายการที่มีค่า NOT VERY YOUNG เท่ากับหรือมากกว่า 0.5 เท่านั้น

**4.1.4.7 คำสั่งแสดงการใช้ NOT NOT และ WITH** เช่น "find the people from person that age is not not young with 0.5" ผลการทดลองจะได้คำตอบบางค่าตามตารางที่ 39

ตารางที่ 39

แสดงคำตอบจากคำสั่ง "find the people from person that age is not not young with 0.5"

Name	Age	$m_{\text{NOT\_NOT\_YOUNG}}^*$
กนก	25	1.000
ปัญญา	28	0.735
มงคล	30	0.500

\*คำนวณจากสมการที่ (2) ตารางที่ (8) และตารางที่ (10)

จากตารางที่ 39 แสดงให้เห็นคำตอบ โดยเลือกตอบรายการที่มีค่า NOT NOT YOUNG เท่ากับหรือมากกว่า 0.5 เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองหาค่าความสัมพันธ์เกี่ยวกับข้อความ มี 2 แบบ คือ ทดลองหาค่าพีชชีข้อความที่แอดตริบิวต์มีเพียงข้อความเดียวและทดลองหาค่าพีชชีข้อความที่แอดตริบิวต์มีหลายข้อความต่อกัน

#### 4.2.1 การทดลองหาค่าความสัมพันธ์ข้อความที่แอดตริบิวต์มีข้อความเดียว

4.2.1.1 คำสั่ง เช่น "list the people from person whose name is 'มงคล'" ผลการทดลองจะได้คำตอบตามตารางที่ 40

ตารางที่ 40

แสดงคำตอบจากคำสั่ง 'list the people from person whose name is 'มงคล' '

Name
มงคล

จากตารางที่ 40 แสดงให้เห็นคำตอบ โดยเลือกตอบรายการที่ Name มีค่า 'มงคล' เท่านั้น

4.2.1.2 คำสั่ง เช่น 'list the people from person whose name is 'กนก' ' ผลการทดลองจะได้คำตอบตามตารางที่ 41

ตารางที่ 41

แสดงคำตอบจากคำสั่ง "list the people from person whose name is 'กนก' "

Name
กนก

จากตารางที่ 41 แสดงให้เห็นคำตอบ โดยเลือกตอบรายการที่ Name มีค่า 'กนก' เท่านั้น

## 4.2.2 การทดลองหาค่าความสัมพันธ์ข้อความที่แอดตริบิวต์มีหลายข้อความต่อกัน

4.2.2.1 คำสั่ง เช่น "select name from person who lives in 'กรุงเทพ' " ผลการทดลองจะได้คำตอบตามตารางที่ 42

ตารางที่ 42

แสดงคำตอบจากคำสั่ง 'select name from person who lives in 'กรุงเทพ' '

Name	Province
ชนันต์	กรุงเทพ,ชลบุรี,ตาก

จากตารางที่ 42 แสดงให้เห็นคำตอบ โดยเลือกตอบรายการที่ Province มีค่า 'กรุงเทพ' เท่านั้น

4.2.2.2 คำสั่ง เช่น 'select name from person who lives in 'อ่างทอง' ' ผลการทดลองจะได้คำตอบตามตารางที่ 43

ตารางที่ 43

แสดงคำตอบจากคำสั่ง 'select name from person who lives in 'อ่างทอง' '

Name	Province
ปัญญา	อ่างทอง

จากตารางที่ 43 แสดงให้เห็นคำตอบ โดยเลือกตอบรายการที่ Province มีค่า 'อ่างทอง' เท่านั้น

## บทที่ 5

### สรุปผลการทดลองและข้อเสนอแนะ

#### 5. สรุปผลการทดลองและข้อเสนอแนะ

จากการทดลองใช้คำสั่งที่เป็น "ภาษาธรรมชาติ" เพื่อทดสอบการทำงานของระบบ FDB ปรากฏว่าผลการทดลองเป็นไปตามจุดมุ่งหมายที่วางไว้

##### 5.1 บทสรุป

จากการที่ระบบคอมพิวเตอร์ธรรมดาไม่สามารถประมวลผลสารสนเทศที่มีความไม่สมบูรณ์หรือมีความคลุมเครือตามความต้องการของผู้ใช้ที่ต้องการสารสนเทศที่มีข้อเท็จจริงอย่างถูกต้องครบถ้วนเพื่อจำลองการหาเหตุผลของมนุษย์ เช่น ในระบบผู้เชี่ยวชาญ

"การขยายการใช้ระบบฐานข้อมูลแบบสัมพันธ์เพื่อสนับสนุนระบบผู้เชี่ยวชาญ" ได้จำลองระบบ FDB ขึ้นมา โดยใช้โครงสร้างของฐานข้อมูลแบบสัมพันธ์เพื่อสร้างฐานข้อมูลเสมือนผนวกกับทฤษฎีพีชคณิตนำมาสร้างความสัมพันธ์ต่าง ๆ นำการกระจายที่เป็นไปได้มาแจกจ่ายค่าเท่าที่เป็นไปได้และนำ threshold มาใช้จำกัดค่าคำตอบให้อยู่ในขอบเขตที่ต้องการ การที่ไม่ต้องเก็บสูตรทางคณิตศาสตร์ไว้ใน logic ของโปรแกรมทำให้ผู้ใช้สามารถแก้ไขสูตรต่าง ๆ ที่ใช้คำนวณหาค่าความสัมพันธ์ได้ตามความพอใจขณะใช้งาน รวมทั้งการนำภาษาธรรมชาติมาใช้สั่งงาน ทำให้สามารถกระทำต่อสารสนเทศที่มีความคลุมเครือซึ่งระบบคอมพิวเตอร์ธรรมดาไม่สามารถกระทำได้

ระบบ FDB อาศัยฐานข้อมูลหลักอันเป็นฐานข้อมูลเดิมซึ่งมีข้อมูลที่ใช้งานอยู่แล้วในภารกิจของหน่วยงานตามปกติ เพื่อนำมาสร้างฐานข้อมูลเสมือนขึ้นชั่วคราวประกอบการทำงานให้ได้คำตอบตามความต้องการของผู้ใช้

ระบบ FDB ไม่ได้เก็บชื่อฐานข้อมูลหลักและแอตทริบิวต์ฐานข้อมูลหลักไว้ในระบบจึงทำให้ระบบมีความยืดหยุ่นในการใช้งานสูง

ระบบ FDB ใช้คำสั่งที่เป็น "ภาษาธรรมชาติ-เหมือน-ภาษาอังกฤษ" จึงทำให้ผู้ใช้ใช้ได้สะดวกและเข้าใจง่าย

##### 5.2 ข้อจำกัดของระบบ

เนื่องจากการวิจัยนี้มุ่งเน้นความสามารถในการใช้ภาษาธรรมชาติและทฤษฎีพีชคณิตเพื่อ "การขยายการใช้ระบบฐานข้อมูลแบบสัมพันธ์เพื่อสนับสนุนระบบผู้เชี่ยวชาญ" ในขอบเขตที่แม้ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่กว้างขวางนัก ดังนั้นจะเห็นว่าระบบที่วิจัยจึงมีขีดจำกัดหลายประการ คือ ต้องใช้คำสั่งภาษาอังกฤษตามรูปแบบของ SQL คือ SELECT-FROM-WHERE การใช้ฐานข้อมูลหลักและฐานข้อมูลเสมือนได้อย่างละ 1 ฐานข้อมูล ชื่อแอตทริบิวต์จะใช้ article เช่น a, an, the ไม่ได้ เงื่อนไขจะใช้ตัวเชื่อม (connectives) and และ or เช่น young and tall ไม่ได้และผู้ใช้ไม่สามารถเพิ่มเติมความสัมพันธ์อื่น ๆ นอกเหนือจากที่กำหนดไว้ในระบบ

### 5.3 ข้อเสนอแนะ

จากข้อจำกัดของระบบ FDB ผู้วิจัยเห็นว่าสิ่งที่น่าสนใจเพิ่มเติมต่อไปคือพัฒนาให้สามารถใช้คำสั่งภาษาไทยในการทำงาน ขยายคำสั่งเพิ่มเติมนอกเหนือจากรูปแบบ SQL ให้ใช้ฐานข้อมูลหลักและฐานข้อมูลเสมือนได้มากกว่า 1 ฐานข้อมูล ให้สามารถใช้เงื่อนไขที่มีคำเชื่อม and และ or เช่น young and tall ได้และให้ผู้ใช้สามารถเพิ่มเติมความสัมพันธ์อื่น ๆ นอกเหนือจากที่กำหนดไว้ในระบบได้

ในการสนับสนุนระบบผู้เชี่ยวชาญ ระบบ FDB ซึ่งมีพีซีเทอมสามารถสนับสนุนระบบผู้เชี่ยวชาญได้หลายวิธีดังนี้

(1) ระบบผู้เชี่ยวชาญสามารถเชื่อมต่อเข้ามาในระบบ FDB ตามที่ระบบ FDB เปิดทางไว้ แล้วนำผลที่ได้กลับไปประกอบการทำงาน เช่น การอธิบาย การตัดสินใจ หรืองานอื่นใดก็ได้ ในระบบผู้เชี่ยวชาญ

(2) ระบบผู้เชี่ยวชาญสามารถเข้ามาใช้ VDB ที่ระบบ FDB สร้างขึ้น แล้วนำผลที่ระบบ FDB สร้างขึ้นกลับไปประกอบการทำงาน เช่น การอธิบาย การตัดสินใจ หรืองานอื่นใดที่ต้องการในระบบผู้เชี่ยวชาญ

## เอกสารอ้างอิง

- [1] Abraham Kandel "Fuzzy Mathematical Techniques with Applications", Florida State University, ADDISON-WESLEY PUBLISHING COMPANY, 1986.
- [2] Deyi Li and Dongbo Liu "A Fuzzy PROLOG Database System" Institute of Computer Engineering People's Republic of China, RESEARCH, STUDIES PRESS LTD, 1990.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*===== */
/* PROGRAM      : PLSMT4.PC          */
/* AUTHIOR      : COL. PIROM TIPSING */
/* PRECOMPILER  : PROC              */
/*              : ORACLE PRECOMPILER VERSION 1.3.17.0.4 */
/* COMPILER     : MICROSOFT (R) C OPTIMIZING COMPILER */
/*              : VERSION 6.00      */
/* LINKING      : MICROSOFT (R) SEGMENTED-EXECUTABLE */
/*              : LINKER VERSION 5.10 */
/* INVOKE       : MAIN PROGRAM      */
/* USING        : MAIN PROGRAM OF FUZZY DATA BASE SYSTEM*/
/*===== */

```

```

#include <stdio.h>
#include <string.h>
#include <setjmp.h>
#include <ctype.h>
#include <stdlib.h>
#include "natfunct.h"
#include "getfunct.h"
#include "gettext.h"

```

```
#define YES 1
```

```
#define NO 0
```

```
#define MAX_ITEMS 5
```

```
#define MAX_VNAME_LEN 40
```

```
#define MAX_INAME_LEN 10
```

```
#ifndef NULL
```

```
#define NULL 0
```

```
#endif
```

เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* โครงสร้าง dml_commands */
/* ACTIONS :ใช้เก็บคำสั่ง */
/*=====*/

char *dml_commands[] =
{
    "SELECT", "select", "INSERT", "insert",
    "UPDATE", "update", "DELETE", "delete"
};

/*=====*/
/* โครงสร้าง tab17,update */
/* ACTIONS : ใช้เก็บข้อความ,คำสั่ง */
/*=====*/

struct tab17 {char str[20];} tab7[]=
{
    "","","","","","","","","","","","","","","","","","","","","","",
    "","","","","","","","","","","","","","","","","","","","","",
};

struct update {char str[256];} sql_updat[]=
{
    "","","","","","","","","","","","","","","","","","","","","","",
};

extern int pass_prt;
extern int auto_updat;
extern int numb_updat;

int zz=0;
int z2=0;
int line_cnt=0;
int actn_cnt=0;
int page_cnt=1;

```

การที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int page_lim=5;
int max_stry=0;
int kmax_str=0;
int sw_link_ok=0;
int sw_transform=0;
int sw_treshold=0;
int sw_lenght =0;
char sql_heading[256];
char sql_body[256];
char sql_footing[256];

```

```
EXEC SQL BEGIN DECLARE SECTION;
```

```

char *sql_isrt1;
char *sql_isrt2;
char *sql_isrt3;
char *sql_isrt4;
char fmname[5][10];
char fmvalue[5][40];
char demon_sel_no[2];
char creat_drop_no[1];
char sql_tres_hold[10];

char sql_dropt[128];
char sql_null[128];
char sql_alter[128];
char sql_statement[256];

char sql_create[256];
char sql_update[256];
char sql_update0[256];
char sql_update1[256];
char sql_update2[256];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char sql_update3[256];
char sql_update4[256];
char sql_condition[256];
char *fp1 = (char *)sql_condition;
char linebuf[256];
extern char *temp_db;
char sql_t[10];
char sql_tt[10];
extern float anss;
char *blank = " ";
char *word;
char *uu;
varchar lst[2];
int cnt=0;
double tres_hold;
int sw_aposo=0;
int sw_digit=0;
int sql_t_leng=0;
int sw_prtx=0;
int sw_exclama=0;
int nl0=0;
int nl=0;
int l=0;
int k1=0;
int k2=0;
int k22=0;
int k3=0;
int k4=0;
int k4_1=0;
int k4_2=0;

```

เอ **int kmxy1=0;** สารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int kmxy2=0;
int g=0;
int sound;
varchar demo[0];
varchar newtab[0];
varchar userform[0];
EXEC SQL END DECLARE SECTION;

EXEC SQL INCLUDE sqlca;          /*SQL Communications Area */
EXEC SQL INCLUDE sqllda;        /*SQL Descriptor Area */

SQLDA *bind_dp;
SQLDA *select_dp;

extern SQLDA *sqlald();          /* allocate data buffers */

/*-----*/
/* find out if column allows nulls, */
/* and clear the datatype's null status bit */
/*-----*/

extern void sqlnul();
jmp_buf jmp_continue; /*buffer to hold longjmp state info*/
int parse_flag = 0; /*global flag for error routine*/
float tfloat;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* FUNCTION NAME: main() */
/* ACTIONS      : ฟังก์ชันหลักของระบบ FDB */
/*=====*/

main()
{
    int i=0;
    int oracle_connect();
    int alloc_descriptors(/*int, int, int*/);
    int get_sql_statement(/*void*/);
    char *strupr(/*char *s*/);
    welcome_to_fuzzy();
    sound_wait_20();

    /* Connect to ORACLE */
    if (oracle_connect() != 0)
        exit(1);

    /* alloc memory for select and bind descriptor */
    if (alloc_descriptors(MAX_ITEMS,
        MAX_VNAME_LEN, MAX_INAME_LEN) != 0)
        exit(1);

    /* process SQL statement */
    for (;;)
    {
        i = setjmp(jmp_continue);

        /*-----*/
        /* nat_func_statement(); */
        /* get the statement. Break on "Exit,X,Quit" */
        /*-----*/
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (nat_func_statement() != 0)
    exit(1);

print_func_msg();

/*-----*/
/* table management: create table / select table */
/*-----*/
if (pass_prt == 0)
{
    /*...*/
}
else
{
    select_table();
}

EXEC SQL WHENEVER SQLERROR GOTO sql_err;
parse_flag = 1; /*set a flage for sql_error().*/

/*-----*/
/* prepare the statement and declare a cursor */
/*-----*/
EXEC SQL PREPARE S FROM :sql_statement;
parse_flag = 0;          /* unset the flag */
EXEC SQL DECLARE C CURSOR FOR S;

/*-----*/
/* the bind variables for any placeholders */
/* in the SQL statement */
/*-----*/
set_bind_variables();

```

```

/*-----*/
/* open the cursor and execute the statement.    */
/* if the statement is not a query (SELECT), the */
/* statement processing is completed after OPEN */
/*-----*/
EXEC SQL OPEN C USING DESCRIPTOR bind_dp;

```

```

/*-----*/
/* Call the function that process the select-list */
/* If the statement is not a query, this function */
/* just returns, doing nothing.                  */
/*-----*/
process_select_list();

/*-----*/
/* Tell user how many rows were processed.      */
/*-----*/
for (i = 0; i <= MAX_ITEMS ; i++)
{
    if (strncmp(sql_statement,dml_commands[i],6)==0)
    {
        printf
        ("\n%d row%c processed.\n",sqlca.sqlerrd[2],
         sqlca.sqlerrd[2] == 1 ? '\0' : 's');
        *uu=getchar();
        line_cnt=0; actn_cnt=0;
        page_cnt=1; clear_mxy_tab7();
        break;
    } /*end if*/
} /* end for I<=MAX_ITEMS*/

```

```

} /* end of for(;;) statement-processing loop*/

```

```

/*-----*/
/* When done, free the memory allocated for pointers */
/* in the bind and select descriptors. */
/*-----*/

for (i = 0; i <= MAX_ITEMS; i++)
{
    if (bind_dp->V[i] != (char *) NULL)
        free(bind_dp->V[i]);
    free(bind_dp->I[i]); /*MAX_ITEMS were allocated*/

    if (select_dp->V[i] != (char *) NULL)
        free(select_dp->V[i]);
    free(select_dp->I[i]); /*MAX_ITEMS were allocated*/
}

/*-----*/
/* free space used by the descriptors themselves. */
/* dealloc the storage space for the descriptor */
/*-----*/

sqlclu(bind_dp); /*library function*/
sqlclu(select_dp); /*library function*/

EXEC SQL WHENEVER SQLERROR CONTINUE;

/*-----*/
/* Close the cursor */
/*-----*/

EXEC SQL CLOSE C;

```

```
EXEC SQL COMMIT WORK RELEASE;
```

เอกสารนี้เป็นทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
 เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
EXEC SQL WHENEVER SQLERROR GOTO sql_err;
```

```
printf ("\n\n*****[Normal End of Program]*****");
```

```
exit(0);
```

```
sql_err:
```

```
printf ("\n\nSQL_ERROR in main ");
```

```
printf ("\n\n      EXEC SQL DESCRIBE :temp_db; ");
```

```
printf ("                |");
```

```
/* see bottom */
```

```
printf ("\n\n%.70s\n",sqlca.sqlerrm.sqlerrmc);
```

```
if (parse_flag)
```

```
{
```

```
    printf ("Parse error at character offset ");
```

```
    printf ("%d in FSQL statement.\n",sqlca.sqlerrd[4]);
```

```
}
```

```
EXEC SQL WHENEVER SQLERROR CONTINUE;
```

```
EXEC SQL ROLLBACK WORK;
```

```
*uu=getchar();
```

```
longjmp(jmp_continue, 1);
```

```
return 0;
```

```
} /* end_main() */
```

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

```
/*      END MAIN PROGRAM      */
```

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

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* FUNCTION NAME: oracle_connect() */
/* ACTIONS : ติดต่อขอใช้ระบบฐานข้อมูลสัมพันธ์ ORACLE */
/* ถ้า ORACLE ยินยอมก็ทำงานต่อไปได้ */
/* ถ้า ORACLE ไม่ยินยอมจะจบโปรแกรม */
/*=====*/

oracle_connect()
{
EXEC SQL BEGIN DECLARE SECTION;
    VARCHAR username[20];
    VARCHAR password[20];
EXEC SQL END DECLARE SECTION;

    /*printf ("Enter user_name: ");*/
strcpy (username.arr,"PIROM"); /*gets(username.arr)*/
username.len = strlen(username.arr); /*printf ("Enter password : ");*/
strcpy (password.arr,"PIROM"); /*gets(password.arr)*/
password.len = strlen(password.arr);

EXEC SQL WHENEVER SQLERROR GOTO connect_error;
EXEC SQL CONNECT :username IDENTIFIED BY :password;
print_nlin10();
printf ("\n Connected to ORACLE as user");
printf (" %s.\n", username.arr); print_nlin10();
printf ("\n Press [ENTER] to continue...\n"); *uu=getchar();
return 0;
connect_error:
printf ("\n Cannot Connected to ORACLE ");
printf (" as user %s.\n", username.arr);
printf ("\n Press [ENTER] to end of program...\n");
*uu=getchar(); return -1;
} /* end oracle_connect */

```

```

/*=====*/
/* FUNCTION NAME: alloc_descriptors() */
/* ACTIONS :จองเนื้อที่เพื่อเก็บรายละเอียดของ BIND และ SELECT */
/* Allocate the BIND and SELECT descriptors using sqlald(). */
/* Also allocate the pointers to indicator variables */
/* in each descriptor. The pointers to the actual bind */
/* variable and the select-list items are realloc'ed in */
/* the set_bind_variables() or process_select_list() */
/* routines. This is possible since sqlald() sets a NULL */
/* pointer in each of MAX_ITEMS pointer spaces. */
/*=====*/
alloc_descriptors(size,max_vname_len,max_iname_len)
int size;
int max_vname_len;
int max_iname_len;
{
int i;
/*-----*/
/* The first sqlald parameter determines the maximum */
/* number of array elements in each variable in the */
/* descriptor. In other words, it determines the */
/* maximum number of bind variables or select-list */
/* items in the SQL statement. */
/* */
/* The second parameters determines the maximum length */
/* of string used to hold the names of select-list */
/* items or placeholders. The maximum length of column */
/* names in ORACLE is 30, but you can allocate more or */
/* less as need */
/* */
/* The third parameters determine the maximum length */

```

```

/* of strings used to hold the name of any indecator */
/* variables. To follow ORACLE standards, the maximum */
/* length of these should be 30. But, you can allocate */
/* more or less as needed. */
/*-----*/
if ((bind_dp = sqlald(size,max_vname_len,max_iname_len))
    == (SQLDA *)NULL)
{
    printf ("\nCannot allocate memory for ");
    printf (" bind descriptor.");
    *uu=getchar();
    return -1; /*Have to exit in this case.*/
}
if ((select_dp = sqlald(size, max_vname_len, 0))
    == (SQLDA *) NULL)
{
    printf ("\nCamot allocate memory for ");
    printf (" select descriptor.");
    *uu=getchar();
    return -1; /*Have to exit in this case.*/
}

select_dp->N = MAX_ITEMS;

/*-----*/
/* Allocate pointer to buffers */
/*-----bind-buffer indicator-----*/
for (i = 0; i < MAX_ITEMS; i++)
{
    bind_dp->I[i] = (short *) malloc(sizeof (short *));
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (i = 0; i < MAX_ITEMS; i++)
{
    bind_dp->V[i] = (char *) malloc(max_vname_len);
}

/*-----*/
/* Allocate select variable pointers          */
/*-----*/
for (i = 0; i < MAX_ITEMS; i++)
{
    select_dp->I[i] = (short *) malloc(sizeof (short *));
}
for (i = 0; i < MAX_ITEMS; i++)
{
    select_dp->V[i] = (char *) malloc(max_vname_len);
}

return 0;
} /* end alloc_descriptors */

```

```

/*=====*/
/* FUNCTION NAME: set_bind_variables() */
/* ACTIONS : รับตัวแปรจาก keyboard แล้วนำมาคำนวณ */
/*          หาค่าต่าง ๆ ที่เกี่ยวข้องใหม่ */
/*=====*/

set_bind_variables()
{
    int i, n;
    char bind_var[64];

    EXEC SQL WHENEVER SQLERROR goto sql_err;

    /* initial count of array elements */
    bind_dp->N = MAX_ITEMS;
    /*-----*/
    /* Describe any bind variable (input host variables) */
    /*-----*/
    EXEC SQL DESCRIBE BIND VARIABLES FOR S INTO bind_dp;

    /*-----*/
    /* If F is negative, there were more bind variables */
    /* than originally allocated by sqlald(). */
    /*-----*/

    if (bind_dp->F < 0)
    {
        printf
            ("\nToo many bind variables (%d), maximum is %d\n.",
             -bind_dp->F, MAX_ITEMS);
        *uu=getchar();
        return -1;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/* Set the maximum number of array element in the */
/* descriptor to the number found */
/*-----*/

```

```
bind_dp->N = bind_dp->F; /*N.K=0*/
```

```

/*-----*/
/*Get the value of each bind variables as a character */
/* strings. */
/* */
/* C[i] contains the length of the bind variable name */
/* used in FSQL statement. */
/* S[i] contain the actual name of the bind variable */
/* used in the FSQL statement */
/* L[i] will contain the lenght of data value entered */
/* V[i] will contain the address of the data value */
/* entered */
/* T[i] is always set to 1 because in this */
/* program... data values for all bind variables */
/* are enterd as character strings. ORACLE */
/* converts to the ....from CHAR. */
/* I[i] will point to the indicator valur, which is */
/* set to -1 when the bind variable null */
/* */
/*-----*/

```

```

for (i = 0; i < bind_dp->F; i++) /*Er bca all i if<add*/
{
    printf ("\nEnter value for bind variable,");
    printf (" No semi_colon. %.*s: ",
            (int)bind_dp->C[i], bind_dp->S[i]);

    /* รับค่า bind variable จาก keyboard*/
    fgets(bind_var, sizeof bind_var, stdin);

    /* Get length and remove newline character*/
    n = strlen(bind_var) - 1;

    /* Set it in the descriptor */
    bind_dp->L[i] = n;

    /*(re-)allocate the buffer for the value.
    sqlald() reverse a pointer location for
    V[i] but do not allocate the pointer
    *Error&NULL
    */

    bind_dp->V[i] = (char *) realloc(bind_dp->V[i],
            (int)bind_dp->L[i] + 1);

    /* And copy it in */
    strncpy(bind_dp->V[i], bind_var, n);

    /* Set the indicator variable's value.*/
    if ((strcmp(bind_dp->V[i], "NULL", 4) == 0) ||
        (strcmp(bind_dp->V[i], "null", 4) == 0))
        *bind_dp->I[i] = -1;

```

```

else
    *bind_dp->I[i] = 0;

    /* Set the bind datatype to 1 for CHAR.*/
    bind_dp->T[i] = 1;
} /*end for i < bind_dp->F*/

return 0;

sql_err:
/* ORACLE error header */
printf ("\n\nSQL_ERROR set_bind_variable ");
printf ("\n\n%.70s\n",sqlca.sqlerrm.sqlerrmc);

if (parse_flag)
{
    printf ("Parse error at character offset");
    printf (" %d in SQL statement\n",
        sqlca.sqlerrd[4]);
}

EXEC SQL WHENEVER SQLERROR CONTINUE;
EXEC SQL ROLLBACK WORK;

*uu=getchar();

longjmp(jmp_continue, 1); /*i*/
return 0;

} /*end_set_bind_variable*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* FUNCTION NAME: .process_select_list() */
/* ACTIONS : แยกแอดตริบิวต์ออกจากประโยคคำสั่ง */
/* และ บอกรายละเอียดของแต่ละตัวว่า */
/* ชื่ออะไร เป็นข้อมูลชนิดไหน มีความยาว */
/* เท่าใด สถานภาพ null หรือ not null */
/*=====*/

process_select_list()
{
    int i, null_ok, precision, scale;

    /*-----*/
    /* If the SQL statement is a SELECT, describe the */
    /* select-list items. DESCRIBE returns the name, */
    /* datatype, length (include precision and scale), */
    /* and null/not null status of each item */
    /*-----*/
    if ((strcmp(sql_statement, "SELECT", 6) != 0) &&
        (strcmp(sql_statement, "select", 6) != 0))
    {
        select_dp->F = 0;
        return 0;
    }

    select_dp->N = MAX_ITEMS;

    EXEC SQL DESCRIBE SELECT LIST FOR S INTO select_dp;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/* If F is negative, there were more select-list */
/* items than original allocated by sqlald(). */
/*-----*/
if (select_dp->F < 0)
{
    printf ("\nToo many select-list items ");
    printf (" (%d), maximum is %d\n",
            -(select_dp->F), MAX_ITEMS);
    *uu=getchar();
    return -1;
}

/*-----*/
/* Set the maximum number off arry element in the */
/* descriptor to the number found */
/*-----*/
select_dp->N = select_dp->F;

/*-----*/
/* Allocated storage for each select-list item. */
/*
/* sqlprc() is used to extract precision and scale */
/* from the length (select_dp->L[i]). */
/* sqlnull() is use to reset the high order bit of the */
/* datatype and to check wether the column */
/* is NOT NULL */
/* CHAR datatype have length, but zero precision and */
/* scale, Tthe length is defind at CREATE time */
/* NUMBER datatype have precision and scale only if */
/* defined at CREATE time. If the column */

```

```

/*      definition was just NUMBER, the precision      */
/*      and scale are zero, and you must allocate      */
/*      the required maximum length.                  */
/* DATE      datatype return a length of 7 if the default */
/*      format is used. etc.                          */
/* ROWID     datatype always returns a fixed length of 18 */
/*      if coerced to CHAR.                            */
/* LONG and                                         */
/* LONG RAW  datatypes return a length of 0 (zero), so */
/*      you need to set a maximum. in this ...        */
/*      example, it is 240 characters                  */
/*-----*/
print_head1();
print_head2();

/*-----*/
/*      HEADING PROCESS                               */
/*-----*/
printf ("\n");

for (i = 0; i < select_dp->F; i++)
{
/*-----*/
/* Turn off high-order bit of datatype in this      */
/* program, it does not matter if the column        */
/* is NOT NULL                                       */
/*-----*/
sqlmul (&(select_dp->T[i]), &(select_dp->T[i]),
        &null_ok);

```

```

switch (select_dp->T[i])
{
    /*-----*/
    /* CHAR datatype: no change in length */
    /* need, except possibly for TO_CHAR */
    /* conversions (not handle here) */
    /*-----*/

case 1 :
    break;

    /*-----*/
    /* NUMBER datatype: use sqlprc() to */
    /* extract precision and scale */
    /*-----*/

case 2 :
    sqlprc (& (select_dp->L[i]), &precision,
            &scale);
    /*Also allow for maximum size of NUMBER*/
    if (precision == 0)
        precision = 40;
    /* Also allow for decimal point and
       possible sign */
    select_dp->L[i] = precision + 2;
    break;

case 8 : /* LONG datatype*/
    select_dp->L[i] = 240;
    break;

case 11 : /* ROWID datatype*/
    select_dp->L[i] = 18;
    break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 12 : /* DATE datatype */
    select_dp->L[i] = 9;
    break;

case 23 : /* RAW datatype */
    break;

case 24 : /* LONG RAW datatype */
    select_dp->L[i] = 240;
    break;

} /*end case */

/*-----*/
/* Allocate space for select-list values. */
/*-----*/
select_dp->V[i] = (char *) realloc(select_dp->V[i],
    (int)select_dp->L[i]+1); /*Er&NULL*/

/*-----*/
/* print column heading, right-justifying number */
/* column headings */
/*-----*/

if (select_dp->T[i] == 2)
{
    printf ("<@%. *s>",(int)select_dp->L[i]+1,
        select_dp->S[i]); /*L+1*/
    strcat (sql_heading,"+");
    strncat (sql_heading,select_dp->S[i],
        (int)select_dp->L[i]+1);
}

```

เอกสาร else เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    printf ("<&%-.*s>",(int)select_dp->L[i],
           select_dp->S[i]); /*L-1.ER*/
    strcat (sql_heading,"-");
    strncat (sql_heading,select_dp->S[i],
            (int)select_dp->L[i]+1);
}

/*-----*/
/* Coerce(ใช้อำนาจบังคับ) all datatypes except */
/* LONG RAW to character */
/*-----*/
if (select_dp->T[i] != 24)
    select_dp->T[i] = 1;
} /* end for i < select_dp->F */

/*-----*/
/* END HEADING PROCESS */
/*-----*/
/*-----*/
/* PRINT HEADING */
/*-----*/

printf ("\n");
printf (" หน้าที<%d>",page_cnt);
strncat (sql_heading,"...",3);
strncat (sql_heading,sql_condition,
        strlen(sql_condition)+1);
printf("\n<%-.*s> ]\n",512,sql_heading);
print_dash();
/*-----*/
/* END PRINT HEADING */
/*-----*/

```

```

/*-----*/
/*      DETAIL      PROCESS      */
/*-----*/

sound_wait_20();

print_head3();

EXEC SQL WHENEVER NOT FOUND GOTO end_select_loop;

```

```
sw_exclama=0;
```

```

/*-----*/
/* FETCH each row select and print the column values. */
/*-----*/
for (;;)
  /*+ for (;;)+k1<>i++++*/
  {
    sw_link_ok=0;
    anss=0;
    tfloat=0;
    clear_sql_body(); /*k1<>i*/

    printf ("\n.");

```

```
EXEC SQL FETCH C USING DESCRIPTOR select_dp;
```

```

/*-----*/
/* Since each variable returned has been coerced */
/* to a character string, very little processing */
/* is required here. This routine just displays */
/* the values on the terminal. */
/*-----*/

```

```

/*-----*
* อ่านฐานข้อมูลมา 1 row แล้วตรวจสอบแต่ละคอลัมน์ *
* ถ้าค่าที่อ่านมาเป็น NULL ให้พิมพ์ under score แทน *
* ถ้าค่าที่อ่านมาเป็น "ข้อความ" หรือ "จำนวนเลข" ที่ *
* ไม่มีเงื่อนไขก็ส่งพิมพ์ได้เลย *
* *
* ถ้าค่าที่อ่านมาเป็น "ข้อความและมีเงื่อนไข" ต้องส่ง *
* ข้อความและเงื่อนไขให้ text processer *
* (gettext.c)แยกแยะค้นหาคำตอบ *
* เมื่อได้คำตอบตามเงื่อนไขที่ต้องการก็ส่งพิมพ์ได้เลย *
* หากยอมรับไม่ได้ตามเงื่อนไขก็ลบข้อความที่รอส่งพิมพ์ *
* แล้วกลับไปอ่าน row ต่อไป *
* *
* ถ้าค่าที่อ่านมาเป็น "จำนวนเลขและมีเงื่อนไข" ต้องส่ง *
* จำนวนเลขและเงื่อนไขให้ numerical proceser *
* (getfunct.c) ค้นหาผลลัพธ์ *
* เมื่อได้รับผลลัพธ์คืนมาแล้วก็นำผลลัพธ์ไปตรวจสอบว่า *
* ยอมรับได้หรือไม่ *
* ถ้ายอมรับได้ตามเงื่อนไขก็ส่งพิมพ์ได้เลย *
* หากยอมรับไม่ได้ตามเงื่อนไขก็ลบข้อความที่รอส่งพิมพ์ *
* แล้วกลับไปอ่าน row ต่อไป *
*-----*/

```

```
for (i = 0; i < select_dp->F; i++)
```

```
    /*!!! for (I<F) !!!*/
```

```
{
```

```
    sw_prtx=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (*select_dp->I[i] < 0)
    /*-( I[i] < 0)-----*/
    {
        /*L+1.ER*/
        printf("<%-*c>",(int)select_dp->L[i],'_');
        strcat(sql_body,"I1.");
        strncat(sql_body,"_",(int)select_dp->L[i]);
        strcat(sql_body,"]");
        sw_link_ok++;
    }
    /*-( I[i] < 0)-----*/
else
    /*-( I[i] != 0)-----*/
    {
        transform_lower_to_upper_case();
        compres_mxy_tab7();
        find_tres_hold();

        if (strcmp(tab7[0].str,"")==0)
            {
                sw_link_ok++;
            }
        else
            {
                /*...*/
            }
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/* GET_FUNCTION, GET_TEXT AND PRINT_DETAIL */
/*-----*/

for (k1=i; k1<i+1; k1++)
{
    /*---FOR K1<I+1---*/
    strncpy(sql_t,select_dp->S[k1],(int)select_dp->L[k1]);

for (k2=0;(k2<max_strylltab7[k2].str=='\0');k2++)
{
    /*---FOR K2<MxY---*/
    find_lenght();

if (strcmp(tab7[k2].str,sql_t,sql_t_leng)==0)
{
    /*-- T7 == SQLT --*/
    find_aposo();

if (sw_aposo !=0)
{
    /*--MAT APO TXT---*/

    get_text_func(tab7[k2+2].str,
        strlen(tab7[k2+2].str),
        (int)select_dp->L[k1],
        (int)select_dp->L[k1],
        select_dp->V[k1]);

    sql_text_process();

    sw_aposo = 0;

    break;
}
    /*--MAT APO TXT---*/

else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    /*-MAT UPO INT_TXT*/
    if ((strcmp(tab7[k2-1].str,"NOT")==0) ||
        (strcmp(tab7[k2-1].str,"VERY")==0) )
    {
        /*--K2-1==N,V-GT--*/
        get_funct_statement(tab7[k2-1].str,
            atof(select_dp->V[i]));
        next_get_funct_statement();
    }
    /*--K2-1==N,V-GT--*/
else
{
    /*--K2-1!=N,V-NG--*/
    if (sw_exclama==0)
    {
        sql_exclama();
        sw_exclama++;
    }
    anss = atof(select_dp->V[i]);
    if (anss > 1.0)
    {
        /*...*/
    }
    else
    {
        if ( (anss >= tres_hold) !=0 )
        {
            ecvtx_anss(); sw_link_ok++;
        }
        else
        {
            printf("\nGET_FUNCT -CLS-Clear Prt Rec"); clear_sql_body();
        }
    }
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพียงการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}          /*-MAT UPO INT_TXT*/
}          /*-- T7 == SQLT --*/
else
{          /*-- T7 != SQLT --*/
if ((strcmp(tab7[k2].str,"NOT")==0) ||
    (strcmp(tab7[k2].str,"VERY")==0) )
    /*.....== NOT,VERY.....*/
{
    /*...*/
}
    /*.....== NOT,VERY.....*/
else
    /*.....!= NOT,VERY.....*/
{
if (sw_aposo !=0)
    /*====HAVE=APOSO=====*/
{
    /*...*/
}
    /*====HAVE=APOSO=====*/
else
    /*====HAVEN'T=APOSO=====*/
{
    sw_link_ok++;
}
    /*====HAVEN'T=APOSO=====*/
}/*end strcmp(tab7[k2-1].str,not,vary*/
    /*.....!= NOT,VERY.....*/
}          /*-- T7 != SQLT --*/
}          /*EN-FOR K2<MxY---*/

/*-;END FOR K2 <MAX_STRY;-----*/

```

เอกสารนี้เป็นเอกสารที่สวชนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/*  COLLECTION DETAIL          */
/*-----*/

if (sw_link_ok !=0)
{
    strcat(sql_body,"[3.");
    strcat(sql_body,select_dp->V[i], /*k1<i*/
           (int)select_dp->L[i]);

    strcat(sql_body,"]",1);

    sw_prtx++;
    sw_link_ok=0;
}

/*-----*/
/*  END COLLECTION DETIAL      */
/*-----*/
} /*EN-FOR K1<I+1---*/
/*-----*/
/*  END_GET_FUNCTION, GET_TEXT AND PRINT_DETAIL  */
/*-----*/

}

/*---(I[i] != 0)---*/
} /*endifor I < select_dp->F*/

/*!!!endifor (i<F) !!!*/

print_detail_1();

} /* end for(;;) */

/*+++++++end for (;;;)++++++*/

/*-----*/
/*      END DETAIL      PROCESS  */
/*-----*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end_select_loop:
    print_dash();
    strcpy(sql_footing,"   END_OF_REPORT   ");
    printf("\n<%.*s>",512,sql_footing); *uu=getchar();
    strcpy(sql_heading,"");
    strcpy(sql_body,"");
    strcpy(sql_footing,"");
    /*-----*/
    sw_transform=0;
    /*-----*/
    return 0; /*go to for(;;)*/
sql_err:
    printf ("\n\nSQL_ERROR process_select_list");
    printf ("\n\n%.70s\n",sqlca.sqlerrm.sqlerrmc);
    if (parse_flag)
    {
        printf ("Parse error at character offset ");
        printf ("%d in SQL statement.\n",
            sqlca.sqlerrd[4]);
    }

EXEC SQL WHENEVER SQLERROR CONTINUE;
EXEC SQL ROLLBACK WORK;

*uu=getchar();

longjmp(jmp_continue, 1);
} /* end process_select_list */
/*-----*/
/*      END PROCESS SELECT LIST  */
/*-----*/

```

```

/*=====*/
/* FUNCTION NAME: sql_exclama() */
/* ACTIONS : ตรวจสอบตำแหน่งและ */
/*          เงื่อนไข */
/*=====*/

sql_exclama()
{
    if ((strcmp(tab7[k2+1].str,"!") ==0) ||
        (strcmp(tab7[k2+2].str,"") !=0))
    {
        for (k3=k2+2;;k3++)
        {
            if ((strcmp(tab7[k3].str,"NOT") !=0) ||
                (strcmp(tab7[k3].str,"VERY") !=0) )
            {
                k4=k3+2;
                break;
            }
        } /*for k3*/

        for (k3=k2+2;k3<k4+2;k3++)
        {
            strcpy(tab7[k4+5].str,tab7[k3].str);
            strcpy(tab7[k3].str,tab7[k4-k3+2].str);
            strcpy(tab7[k4-k3+2].str,tab7[k4+5].str);
        }

        compres_k4_tab7();
    } /*if !=*/

} /*end_sql_exclama()*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* FUNCTION NAME: sql_text_process() */
/* ACTIONS : body process รวบรวม text รอสั่งพิมพ์ */
/*=====*/

sql_text_process()
{
    if (sw_link_ok != 0)
    {
        strcat (sql_body,"[.2");
        strncat (sql_body,select_dp->V[k1],
            (int)select_dp->L[k1]); /*k1<i*/
        strncat(sql_body,"]",1);
        print_detail_2();
    }
    else
    {
        printf ("\n TEXT_PROC *** CLEAR ALL IN REC ***");
        clear_sql_body();
        sw_aposo=0;
    }
} /*end_sql_text_process()*/

/*=====*/
/* FUNCTION NAME: drop_table() */
/* ACTIONS : ลบ table ที่ */
/*=====*/

drop_table()
{
    printf ("\n DROPIng_TABLE"); /* ?drop? */
    created_table_msg(); *uu=getchar();
} /*end_drop_table()*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* FUNCTION NAME: print_detail_1() */
/* ACTIONS      : พิมพ์รายงานออกทางจอภาพ */
/*=====*/

print_detail_1()
{
    /*NeW*/
    if (sw_prtx !=0)
    {
        printf("\n<A%. *s (...see anS3..>",256,sql_body);
        line_cnt++;
        actn_cnt++;
        if (line_cnt >= page_lim)
        {
            print_dash();
            *uu=getchar();
            print_nlin10();
            print_nlin10();
            print_dash();
            page_cnt++;
            printf ("\n
                ");
            printf ("          หน้าที<%d>",page_cnt);
            printf("\n<%. *s> ]\n",512,sql_heading);
            print_head3();
            print_dash();
            line_cnt=0;
        }

        sw_link_ok=0;
        sw_prtx=0;
    }
}

```

```

}/*end_print_detail_1()*//*NeW*/

```

```

/*=====*/
/* FUNCTION NAME : print_detail_2() */
/* ACTIONS : พิมพ์รายงานออกทางจอภาพ */
/*=====*/

print_detail_2()
{
    printf("\nSAVE_ANS1<%.*s>",256,sql_body);
    sw_aposo=0; sw_link_ok=0;
} /*end_print_detail_2()*/

/*=====*/
/* FUNCTION NAME : compres_mxy_tab7() */
/* ACTIONS : จัดระเบียบข้อมูล */
/*=====*/

compres_mxy_tab7()
{
    for (kmxy1=0;(kmxy1<max_stry+1||
        tab7[kmxy1].str=='\0');kmxy1++)
    {
        for (kmxy2=0;(kmxy2<max_stry+1||
            tab7[kmxy2].str=='\0');kmxy2++)
        {
            if ((strcmp(tab7[kmxy2].str,"")==0) &&
                (strcmp(tab7[kmxy2+1].str,"")!=0) )
            {
                strcpy(tab7[kmxy2].str,tab7[kmxy2+1].str);
                strcpy(tab7[kmxy2+1].str,"");
            }
        }
    }
} /*end_compres_mxy_tab7()*/

```

```

/*=====*/
/* FUNCTION NAME : clear_mxy_tab7() */
/* ACTIONS : จัดระเบียบข้อมูล */
/*=====*/

clear_mxy_tab7()
{
    for (kmxy1=0;(kmxy1<max_stry+10 ||
        tab7[kmxy1].str=='\0');kmxy1++)
    {
        strcpy(tab7[kmxy1].str,"");
    }
} /*end_clear_mxy_tab7()*/

/*=====*/
/* FUNCTION NAME : compres_k4_tab7() */
/* ACTIONS : จัดระเบียบข้อมูล */
/*=====*/

compres_k4_tab7()
{
    for (k4_1=0;(k4_1<k4||tab7[k4_1].str=='\0');k4_1++)
    {
        for (k4_2=0;(k4_2<k4||tab7[k4_2].str=='\0');k4_2++)
        {
            if ((strcmp(tab7[k4_2].str,"")==0) &&
                (strcmp(tab7[k4_2+1].str,"")!=0) )
            {
                strcpy(tab7[k4_2].str,tab7[k4_2+1].str);
                strcpy(tab7[k4_2+1].str,"");
            }
        }
    }
} /*end_compres_k4_tab7()*/

```

```

/*=====*/
/* FUNCTION NAME: find_tres_hold() */
/* ACTIONS : การหาค่าจำกััดคำตอบ */
/*=====*/

find_tres_hold()
{
    for (k1=0;(k1<max_stry+1||tab7[k1].str=='\0');k1++)
    {
        if (strcmp(tab7[k1].str,"")==0)
            {break;}

        if ((strcmp(tab7[k1].str,"!")==0) ||
            (strcmp(tab7[k1].str,"NOT")==0) ||
            (strcmp(tab7[k1].str,"VERY")==0) )
            {
                /*...*/
            }
        else
            {
                get_tres_hold();
            }
    }
} /*end_find_tres_hold()*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* FUNCTION NAME: get_tres_hold() */
/* ACTIONS : การอ่านค่าจำกัดคำตอบ */
/*=====*/

get_tres_hold()
{
char *fp4 = (char *)tab7[k1].str;
char *fp4s; fp4s=fp4; *word =NO;
strcpy(sql_tres_hold,"");

while (*fp4!='\0')
{
if (*fp4 !=' ')
{
if ( isdigit(*fp4) || (*fp4=='.'))
{
*word=YES;
strncat(sql_tres_hold,fp4,1);
*word=NO;
}
}
else
{
/*...*/
}
fp4++;
}/*while*/

tres_hold=atof(sql_tres_hold);
fp4=fp4s;

```

เอ } /\*end\_get\_tres\_hold()\*/ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* FUNCTION NAME: transform_lower_to_upper_case() */
/* ACTIONS : ปรับสภาพข้อมูลให้เป็น upper case */
/*=====*/

transform_lower_to_upper_case() /*cndi*/
{
    char *fp1s;
    if (sw_transform ==0)
    {
        fp1s=fp1;
        *word =NO;
        max_stry=0;

        while (*fp1!='\0')
        {
            if (*fp1 !=' ')
            {
                if (strchr("?",*fp1) /*unds_*/
                {
                    *word=YES;
                    ++max_stry;
                    *word=NO;
                }
            }
            else
            {
                strncat(tab7[max_stry].str,strupr(fp1),1);
            }
        }
    }
    else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        strcat(tab7[max_stry].str, " ",1);
        if (*word)
        {
            *word=NO;
            ++max_stry;
        }
    }
    fp1++;
}/*while*/
sw_transform++;
fp1=fp1s;
}
} /*end_transform_lower_to_upper_case()*/

/*=====*/
/* FUNCTION NAME: find_aposo() */
/* ACTIONS : หาเครื่องหมายฟ้นทอง() */
/*=====*/

find_aposo()
{
    char *fp2s; char *fp2;
    fp2 = (char *)tab7[k2+2].str;

    while (*fp2!='\0')
    {
        if (strchr("'",*fp2))
            {sw_aposo++;}
        fp2++;
    }
}

}/*end_find_aposo()*/

```

```

/*=====*/
/* FUNCTION NAME: find_lenght() */
/* ACTIONS : หาคความยาวข้อมูล */
/*=====*/

find_lenght()
{
    char *fp3s; char *fp3;
    fp3 = (char *)sql_t;
    sql_t_leng=0;

    while (*fp3!='\0')
    {
        if (strchr(" ",*fp3))
            {break;}
        else
            {sql_t_leng++;}
        fp3++;
    }
}/*end_find_lenght()*/

/*=====*/
/* FUNCTION NAME: auto_update() */
/* ACTIONS : ปรับปรุงแก้ไขข้อมูล โดยอัตโนมัติ */
/*=====*/

auto_update()
{
    printf ("\n AUTOMATIC UPDATE"); *uu=getchar();
    cat_update();
    update_table_formula(); printf ("\n END AUTOMATIC UPDATE");
    *uu=getchar();
}/*end_auto_update()*/

```

```

/*=====*/
/* FUNCTION NAME: next_get_func_statement() */
/* ACTIONS : ถ้ามีเงื่อนไขต้องคำนวณต่อก็ส่งไปคำนวณ */
/* ถ้าไม่ต้องคำนวณก็ส่งไปตรวจว่ายอมรับได้หรือไม่ */
/*=====*/

next_get_func_statement()
{
    for (kmax_str=max_str-3; kmax_str > -1; kmax_str--)
    {
        if (kmax_str == 0 )
        {
            if ((anss >= tres_hold) !=0 )
            {
                ecvtx_anss();
                break;
            }
            else
            {
                printf("\n next_get_func ... Clear Prt Rec");
                clear_sqI_body();
                break;
            }
        }
        else
        {
            tfloat=anss;
            get_func_statement(tab7[kmax_str].str,tfloat);
        }
    }
} /*end for */

} /*end_next_get_func_statement()*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* FUNCTION NAME: clear_sql_body() */
/* ACTIONS : ลบข้อความที่รอส่งพิมพ์ */
/*=====*/

clear_sql_body()
{
    strncpy(sql_body,"",256);
    sw_link_ok=0;
} /*end_clear_sql_body()*/

/*=====*/
/* FUNCTION NAME: ecvtx_anss() */
/* ACTIONS : เปลี่ยนคำตอบที่เป็นจำนวนเลขให้เป็นตัวอักษร */
/*=====*/

ecvtx_anss()
{
    char *chr;
    char *buff;
    int *decpt;
    int ans_leng=5; /* fix*/
    int digits=7; /* fix*/
    anss=anss+10.00001;
    chr=gcvt(anss, digits,buff);
    strcat(sql_body,"?",2);
    strcat(sql_body,chr+1,ans_leng);
    strcat(sql_body,"?",2);
    return 0;
} /* end ecvtx_anss()*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* FUNCTION NAME: welcome_to_fuzzy() */
/* ACTIONS : แสดงสัญลักษณ์ (LOGO) ของระบบ FDB */
/*=====*/

welcome_to_fuzzy()
{
    print_nlin10();
    printf ("\n      W  W  EEEEE L   CCCCCC ");
    printf (" OOOOOO M  M  EEEEE ");
    printf ("\n      W  W  E   L   C   C");
    printf (" O  O MM MM E   ");
    printf ("\n      W WW W  EEEE L   C   ");
    printf (" O  O M MM M  EEEE ");
    printf ("\n      WW WW E   L   C   C ");
    printf (" O  O M  M E   ");
    printf ("\n      W  W  EEEEE LLLLLL CCCCCC ");
    printf ("OOOOOO M  M  EEEEE ");
    printf ("\n                                     ");
    printf ("\n                                     TO ");
    printf ("\n                                     ");
    printf ("\n          FFFFFFFF  DDDDD");
    printf ("DD  BBBBBBB ");
    printf ("\n          F          D  ");
    printf (" D  B  BB ");
    printf ("\n          FFFFFFFF  D  ");
    printf (" D  BBBBBBB ");
    printf ("\n          F          D  D");
    printf ("  B  BB ");
    printf ("\n          F          DDDDDDD ");
    printf ("  BBBBBBB ");
    print_nlin5();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
printf ("\n Press [ENTER] to continue...\n");
*uu=getchar();
}
```

```
/*=====*/
/* FUNCTION NAME:print_blank1() */
/* ACTIONS : พิมพ์รายงานทางจอภาพ */
/*=====*/
```

```
print_blank1()
```

```
{
printf("\n
");
printf("
");
}
```

```
/*-----*/
```

```
/* sound_wait_20 */
```

```
/*-----*/
```

```
sound_wait_20()
```

```
{
for (sound =0 ; sound < 20; sound++)
{ putchar('\a'); }
}
```

```
/*=====*/
/* FUNCTION NAME:print_head1() */
/* ACTIONS : พิมพ์รายงานทางจอภาพ */
/*=====*/
```

```
print_head1()
```

```
{
print_nlin10(); print_dash();
printf ("\n
F_S_Q_L
");
```

เอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf ("\n      P E R S O N A L      R E ");
printf ("P O R T          l");
printf ("\n          ");
printf ("          l");
printf ("\n          by          ");
printf ("          l");
printf ("\n          ");
printf ("          l");
printf ("\n          Lt.Col. PIROM TIPSIN");
printf ("G          l");
printf ("\n          ");
printf ("          l");
print_dash(); print_nlin10();
printf ("\n press  ENTER  to continue ... ");
*nu=getchar();
}

/*=====*/
/* FUNCTION NAME:print_head2() */
/* ACTIONS      : พิมพ์รายงานทางจอภาพ */
/*=====*/

print_head2()
{
    print_nlin10();
    printf ("\nNo.<%. *s>",2,demon_sel_no);
    printf ("[FSQL==> <%. *s> ]\n",75,linebuf);
    /*printf("\n=>[MT4].[SQL_CONDIT=><%. *s> ]\n", 60,sql_condition); */
    print_dash();
    printf ("\n      P E R S O N A L      ");
    printf (" R E P O R T      "); print_dash();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

print_head3()
{
}/*end_print_head3()*/

/*=====*/
/* FUNCTION NAME: select_table() */
/* ACTIONS : เมนูให้เลือกว่าจะจัดการอย่างไร กับ Virtual DataBase */
/*=====*/

select_table()
{
    if (auto_updat >= 1 ) {auto_update();}
    print_nlin5();
    print_dash();
    printf ("\n Memorandum for Fuzzy_DataB");
    printf ("ase ");
    print_blank1();
    print_blank1();
    print_blank1();
}/*
printf ("\n 0. Press [ENTER] for Existed Temporary");
printf (" Table ");
*/
printf ("\n 0. Press [ENTER] not upgrade Fuzzy_Dat");
printf ("aBase_Name ");
print_blank1();
/*
printf ("\n 1. Press [ANYKEY] for Create, Drop, or ");
printf ("Update Exited Temporary Table ");
*/
printf ("\n 1. Press [ANYKEY] for upgrade Fuzzy_Dat");
printf ("aBase ");

```

```

print_blank1();
print_blank1();
print_blank1();
printf ("\n          Note: For proximity is auto");
printf ("matic update          |");
print_dash();
printf("\n\n[Your Fuzzy_DataBase_Name is <%,*s> ]",
      32,temp_db);
print_nlin5();
printf ("\n  Would you like  to Create or Update Fu");
printf ("zzy_DataBase, Y(es) ?");
printf ("\n\n          ==> ");

gets(newtab.arr);
if (strlen(newtab.arr) == 0)
{
  use_existed_table_msg();
}
else
{
  print_nlin5();      print_dash();
  print_blank1();    print_blank1();
  printf ("\n          0. Press [ENTER] to Bypa");
  printf ("ss  1, 2 and 3          |");
  print_blank1();
  print_blank1();
  printf ("\n          1. Create New      Fuzz");
  printf ("y_DataBase          |");
  print_blank1();
  print_blank1();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf ("\n          2. Drop   Existed Fuzz");
printf ("y_DataBase          |");
print_blank1();
print_blank1();
printf ("\n          3. Update Existed Fuzz");
printf ("y_DataBase          |");
print_blank1();      print_dash();
print_nlin5();
strcpy(creat_drop_no,"");
printf ("\n Select No. :");
gets(creat_drop_no);
if (strcmp(creat_drop_no,"1")==0)
    {creating_table();}
if (strcmp(creat_drop_no,"2")==0)
    {drop_table();}
if (strcmp(creat_drop_no,"3")==0)
    {updating_table();}
else
    {/*...MVE TO CREATE & UPDATE_TABLE ...*/}
}
} /*end_select_table*/

```

```

/*=====*/
/* FUNCTION NAME: creating_table() */
/* ACTIONS : ให้เลือกว่าจะสร้างฐานข้อมูลเสมือนหรือไม่ */
/*=====*/

creating_table()
{
    printf ("\n CREATing_TABLE");
    print_blank1();
    print_blank1();

    printf ("\n| if create new table must run====> ");
    printf ("
|");

    printf ("\n|          1.upt@dbfx...NULL ");
    printf (" 2.alt@dbfx...float |");

    printf ("\n|          3.upt@dbf1...FORMULAR ");
    printf (" 4.alt@dbfx...FORMULAR |");

    /*
    create_vdbf_table();
    */

    created_table_msg();

    *uu=getchar();

}/*end_creating_table*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* FUNCTION NAME: updating_table() */
/* ACTIONS : ให้ผู้ใช้เลือกว่าเพิ่มความสัมพันธ์ของผู้ใช้เองทาง key board */
/* หรือจะใช้ความสัมพันธ์ที่มีอยู่แล้วในระบบ */
/*=====*/

updating_table()
{
    printf ("\n UPDATing_TABLE");
    printf ("\n Would you like to use your 'Formula' ");
    printf (" Y(es) ?");
    printf ("\n\n ==> ");
    gets(userform.arr);
    if (strlen(userform.arr) == 0)
    {
        /*...*/
    }
    else
    {
        printf ("\n System Formula");
        printf ("\n\n Numb_ Updat<%d>\n\n",numb_updat);
        printf("\n=>[System_Formula0=><%. *s> ]",
                256,sql_update0);

        printf("\n=>[System_Formula1=><%. *s> ]",
                256,sql_update1);

        printf("\n=>[System_Formula2=><%. *s> ]",
                256,sql_update2);

        printf("\n=>[System_Formula3=><%. *s> ]",
                256,sql_update3);

        printf("\n=>[System_Formula4=><%. *s> ]",
                256,sql_update4);
    }
}

```

```

strcpy(sql_update1,"");
strcpy(sql_update2,"");
strcpy(sql_update3,"");
strcpy(sql_update4,"");
for (g=0;g<numb_updat+1;g++)
{
    strcpy(sql_update,"");
    printf ("\nEnter Your own Formula, ");
    printf (" end with semi_colon ; \n");
    gets(sql_update);
    if (g==0)
    {
        strcpy(sql_update0," update ");
        strcat(sql_update0,temp_db);
        strcat(sql_update0," set ");
        strcat(sql_update0,sql_update);
        strncat(sql_update0," ",1);
    }
    if (g==1)
    {
        strcpy(sql_update1," update ");
        strcat(sql_update1,temp_db);
        strcat(sql_update1," set ");
        strcat(sql_update1,sql_update);
        strncat(sql_update1," ",1);
    }
    if (g==2)
    {
        strcpy(sql_update2," update ");
        strcat(sql_update2,temp_db);
        strcat(sql_update2," set ");

```

```

    strcat(sql_update2,sql_update);
    strncat(sql_update2," ",1);
}
if (g==3)
{
    strcpy(sql_update3," update ");
    strcat(sql_update3,temp_db);
    strcat(sql_update3," set ");
    strcat(sql_update3,sql_update);
    strncat(sql_update3," ",1);
}
if (g==4)
{
    strcpy(sql_update4," update ");
    strcat(sql_update4,temp_db);
    strcat(sql_update4," set ");
    strcat(sql_update4,sql_update);
    strncat(sql_update4," ",1);
}
}

printf("\n=>[Your_Formula0=><%.*s> ]",
        256,sql_update0);
printf("\n=>[Your_Formula1=><%.*s> ]",
        256,sql_update1);
printf("\n=>[Your_Formula2=><%.*s> ]",
        256,sql_update2);
printf("\n=>[Your_Formula3=><%.*s> ]",
        256,sql_update3);
printf("\n=>[Your_Formula4=><%.*s> ]\n",
        256,sql_update4);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

/*=====*/
/* FUNCTION NAME: cat_update() */
/* ACTIONS : การจ้กรวบรวมความสัมพันธ์ที่ผู้ใช้พิมพ์ผ่าน key board */
/*=====*/

cat_update()
{
    for (z2=0;z2<numb_updat;z2++)
    {
        if (z2==0)
        {
            strcat(sql_updat[z2].str,sql_update0);
        }
        if (z2==1)
        {
            strcat(sql_updat[z2].str,sql_update1);
        }
        if (z2==2)
        {
            strcat(sql_updat[z2].str,sql_update2);
        }
        if (z2==3)
        {
            strcat(sql_updat[z2].str,sql_update3);
        }
        if (z2==4)
        {
            strcat(sql_updat[z2].str,sql_update4);
        }
    }
}
}/*end_cat_update()*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* FUNCTION NAME: create_vdbf_table() */
/* ACTIONS : การสร้างฐานข้อมูลเสมือน */
/*=====*/

create_vdbf_table()
{
    EXEC SQL WHENEVER SQLERROR goto vdbf_notfound;
    EXEC SQL EXECUTE IMMEDIATE :sql_dropt;
    EXEC SQL WHENEVER SQLERROR goto sql_err;
    EXEC SQL EXECUTE IMMEDIATE :sql_create;
    return 0;

vdbf_notfound:
    if (sqlca.sqlcode != 0)
    {
        EXEC SQL EXECUTE IMMEDIATE :sql_create;
    }
    if (parse_flag)
    {
        printf ("\nParse error at character offset ");
        printf ("%d in SQL statement.\n",sqlca.sqlerrd[4]);
        *uu=getchar();
    }
    EXEC SQL ROLLBACK WORK;
    return 0;

sql_err:
    printf ("\n\nSQL_ERROR create_vdbf_table");
    sql_err_msg();
    return -1;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* FUNCTION NAME: sql_err_msg() */
/* ACTIONS : ข้อความเตือนเมื่อมีการทำงานผิดพลาด */
/*=====*/

sql_err_msg()
{
    sql_err:
        printf ("\n\n%.70s\n",sqlca.sqlerrm.sqlerrmc);
        *uu=getchar();
        if (parse_flag)
        {
            printf ("\nParse error at character offset ");
            printf ("%d in SQL statement.\n",sqlca.sqlerrd[4]);
            *uu=getchar();
        }
        EXEC SQL ROLLBACK WORK;
        *uu=getchar();
        return -1;
    }/*end_sql_err_msg()*/

/*=====*/
/* FUNCTION NAME: update_table_null() */
/* ACTIONS : การจัดระเบียบฐานข้อมูลเสมือน */
/*=====*/

update_table_null()
{
    EXEC SQL WHENEVER SQLERROR goto sql_err;
    strcpy(sql_null,sql_update0);
    EXEC SQL EXECUTE IMMEDIATE :sql_null;
    return 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sql_err:
    printf ("\n\nSQL_ERROR update_table_null ");
    sql_err_msg();
    return -1;
} /*end_update_table_null()*/

/*=====*/
/* FUNCTION NAME: alter_table_float() */
/* ACTIONS : การเปลี่ยนสภาพข้อมูลให้เป็นเลขทศนิยม */
/*=====*/

alter_table_float()
{
    EXEC SQL WHENEVER SQLERROR goto sql_err;
    EXEC SQL EXECUTE IMMEDIATE :sql_alter;

    return 0;
sql_err:
    printf ("\n\nSQL_ERROR alter_table_float ");
    sql_err_msg();
    return -1;
} /*end_alter_table_float()*/

/*=====*/
/* FUNCTION NAME: update_table_formula() */
/* ACTIONS : การปรับปรุงแก้ไขข้อมูล */
/*=====*/

update_table_formula()
{
    EXEC SQL WHENEVER SQLERROR goto sql_err;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (zz=1;zz<numb_updat+1;zz++)
{
    if (strcmp(sql_updat[zz].str,"")==0)
    {
        break;
    }
    strcpy(sql_update1,"");
    strcpy(sql_update1,sql_updat[zz].str);

    EXEC SQL EXECUTE IMMEDIATE :sql_update1;
    EXEC SQL COMMIT;
}
return 0;
sql_err:
printf ("\n\nSQL_ERROR update_table_formula");
printf ("\n\n%.70s\n",sqlca.sqlerrm.sqlerrmc);

if (parse_flag)
    printf ("\nParse error at character offset ");
    printf ("%d in SQL statement.\n",sqlca.sqlerrd[4]);

    EXEC SQL ROLLBACK WORK; *uu=getchar();
    return -1;
} /*end_update_table_formula*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

print_nlin5()
{
    printf ("\n\n\n\n\n");
}

print_dash()
{
    printf ("\n+-----");
    printf ("-----+");
}

/*=====*/
/* FUNCTION NAME: created_table_msg() */
/* ACTIONS : แสดงข้อความว่าสร้างฐานข้อมูลเสมือนเรียบร้อยแล้ว */
/*=====*/

created_table_msg()
{
    print_nlin10();
    print_dash();
    printf ("\n NEW Fuzzy_DataBase ");
    printf ("CREATED ");
    print_dash();
    print_nlin5();
}/*end_created_table_msg()*/

/*=====*/
/* FUNCTION NAME: updated_table_msg() */
/* ACTIONS : แสดงข้อความว่าปรับปรุงแก้ไขฐานข้อมูลเสมือนเรียบร้อยแล้ว */
/*=====*/

updated_table_msg()
{
    print_nlin10();

```

เอกสารนี้เป็นลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf ("\n          Fuzzy_DataBase      ");
printf ("UPDATED          ");
print_dash();
print_nlin5();
}/*end_created_table_msg()*/

/*=====*/
/* FUNCTION NAME: create_table() */
/* ACTIONS      : สร้าง table */
/*              / */
/*=====*/
create_table()
    /*??USE?? */
{
    EXEC SQL WHENEVER SQLERROR goto sql_err;
    EXEC SQL EXECUTE IMMEDIATE :sql_create;
    return 0;

sql_err:
    printf ("\n\nS_Q_L___ERROR_ create_table");
    printf ("\n\n%.70s\n",sqlca.sqlerrm.sqlerrmc);

    if (parse_flag)
    {
        printf ("Parse error at character offset ");
        printf (" %d in SQL statement.\n",sqlca.sqlerrd[4]);
    }
    EXEC SQL ROLLBACK WORK;
    *uu=getchar();
    return -1;
} /*end_create_table()*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* FUNCTION NAME: help() */
/* ACTIONS : ชื่อแนะนำในการใช้ระบบ */
/*=====*/

help()
{
    printf ("\n+=====");
    printf ("=====+");
    printf ("\n Help ");
    printf (" ");
    printf ("\n=====");
    printf ("=====");
    printf ("\n Enter a FQSL statement at the FSQL p");
    printf ("rompt. ");
    printf ("\n Use standard (not embedded) FSQL ");
    printf ("syntax. ");
    printf ("\n Statements can continue over several lines, but must be terminated with a semicolon.");
    printf ("olon(;). ");
    printf ("\n Typing \"exit\" or (no semicolon)");
    printf ("") exits the program.\ ");
    printf ("\n Typing \"?\" or \"help\" gets this");
    printf ("is messages. \ ");
    printf ("\n+-----");
    printf ("-----+");
    printf ("\n");
} /*end_help()*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* FUNCTION NAME: .void sql_error() */
/* ACTIONS : แสดงข้อความเตือนเมื่อมีข้อผิดพลาด */
/*=====*/

void sql_error()
{
    int i;
    printf ("\n\n%.70s\n",sqlca.sqlerrm.sqlerrmc);

    if (parse_flag)
    {
        printf ("Parse error at character offset %d ");
        printf ("in SQL statement.\n", sqlca.sqlerrd[4]);
    }

    EXEC SQL WHENEVER SQLERROR CONTINUE;
    EXEC SQL ROLLBACK WORK;
    longjmp(jmp_continue, 1);
} /*end_sql_error()*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* FUNCTION NAME: print_func_msg() */
/* ACTIONS : แสดง ข้อความ คำถาม */
/*=====*/

print_func_msg()
{
print_nlin10();
printf("\nTRANSLATION[MT4]\n");
printf("\nNo.<%. *s>",2,demon_sel_no);
printf("=>[MT4]..[FSQL==><%. *s> ]",75,linebuf);
printf("\n++++");
printf("++++");
printf("\n=>[SQL_CREATE=><%. *s> ]", 256,sql_create);
printf("\n=>[SQL_STATE(S)=><%. *s> ]\n",60,sql_statement);
printf("\n=>[SQL_CONDIT=><%. *s> ]\n",60,sql_condition);
printf("\n=>[SQL_UPDATE=><%. *s> ]",256,sql_update);
printf("\n=>[SQL_UPDATE0=><%. *s> ]",256,sql_update0);
printf("\n=>[SQL_UPDATE1=><%. *s> ]",256,sql_update1);
printf("\n=>[SQL_UPDATE2=><%. *s> ]",256,sql_update2);
printf("\n=>[SQL_UPDATE3=><%. *s> ]",256,sql_update3);
printf("\n=>[SQL_UPDATE4=><%. *s> ]\n",256,sql_update4);
printf("\n=>[SQL_DROPT=><%. *s> ]",60,sql_dropt);
printf("\n=>[SQL_ALTER=><%. *s> ]",60,sql_alter);
printf("\n=>[Fuzzy DataBase Name=><%. *s> ]",32,temp_db);
/*printf("\n=>[MT4].[SQL_NULL=><%. *s> ]",60,sql_null);*/
printf("\n++++");
printf("++++");
*uu=getchar();
}

/*===== ENX OF PROGRAM =====*/
/*===== END OF PROGRAM =====*/

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์หรือการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ก. 2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* PROGRAM_NAME : NAT.H */
/* AUTHOR : Lt.Col. Pirom Tipping */
/* CREATE DATE : */
/* LAST UPDATE : */
/* USING : include by "NATFUNCT.C" */
/*=====from === tre=====*/

/*-----*/
/*โครงสร้าง VDF เป็นการจองเนื้อที่สำหรับเก็บฟังก์ชันเพื่อคำนวณหาค่าพีชชี */
/* fname ใช้เก็บชื่อพีชชี */
/* fname1 ใช้เก็บชื่อแอดดริบิวต์ที่เกี่ยวข้อง */
/* fmvalue0 ถึง fmvalue4 ใช้เก็บสูตรการหาค่าของพีชชี */
/*-----*/

struct vdbf
{
    char fname[10];
    char fname1[15];
    char fmvalue0[105];
    char fmvalue1[105];
    char fmvalue2[105];
    char fmvalue3[105];
    char fmvalue4[105];
}

vdbf[] =
{
    /*00*/
    "young",
    "age",
    "= NULL",

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

"= POWER(1+POWER(((age-25)/5),2),-1) \
where age > 25 and age <= 130",
"= 1.0 where (age >= 0) and (age <= 25)",
"", "",
/*01*/
"old",
"age",
"= NULL",
"= 0.0 where age >= 0 and age <= 45",
"= POWER(1+POWER(((age-45)/5),-2),-1) \
where age > 45 and age <= 130",
"", "",
/*02*/
"proximity",
"DVAL1",
"...",
"...",
"",
"",
"",
/*03*/
"tall",
"height",
"= NULL",
"= 0.0 where height >= 0 and height <= 155",
"=POWER(1+POWER(((height-155)/10),-2),-1) \
where height > 155 and height <= 250",
"", "",
/*04*/  "", "", "", "", "", "", "",
/*05*/  "", "", "", "", "", "", ""

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/*โครงสร้าง VDBFX เป็นการจองเนื้อที่สำหรับเก็บฟังก์ชันเพื่อคำนวณหาค่าพีชชี่ */
/*      vmname ใช้เก็บชื่อพีชชี่ */
/*      vmname1 ใช้เก็บชื่อแอดดรีบิวต์ที่เกี่ยวข้อง */
/*      vmvalue0 ถึง vmvalue5 ใช้เก็บสูตรการหาค่าของพีชชี่ */
/*-----*/

struct vdbfx
{
    char vmname[10];
    char vmname1[15];
    char vmvalue0[64];
    char vmvalue1[64];
    char vmvalue2[64];
    char vmvalue3[64];
    char vmvalue4[64];
    char vmvalue5[64];
}

vdbfx[] =
{
    "proximity",
    "dvalue",
    "= NULL";
    /*"= EXP(-POWER(2.23,-1.0) * ABS( ", =>ORA-00919 invalid function */
    /*"= POWER(2.71828,(-0.223 * ABS( ", =>ORA-02017 integer value required*/
    "= POWER(0.800115,(1.0 * ABS( ",
    "dvalue",
    " - " ,
    "nvalue" ,
    " ) ) ) ",
    /*01*/  "" , "" , "" , "" , "" , "" , "" , "" , "" , "" ,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*----- */
/* โครงสร้าง SYNONYM เป็นการจองเนื้อที่สำหรับเก็บคำที่มีความหมาย */
/*      เหมือนกันสามารถนำมาใช้ทดแทนกันได้ */
/*      synonym1 ใช้เก็บคำที่ผู้ใช้ถาม */
/*      synonym2 ใช้เก็บคำที่มีความหมายเหมือนกับ synonym1 */
/*----- */

```

```
struct synonym
```

```
{
    char synonym1[15];
    char synonym2[15];
}
```

```
synonym[]=
```

```
{
    /*-----
    NOTE:
    การเพิ่มเติมคำศัพท์ใน synonym
    ต้องไม่มีคำศัพท์นี้ใน ignore
    -----*/
    "find" , "select" ,
    /*01*/ "select" , "select" ,
    /*02*/ "list" , "select" ,
    /*03*/ "people" , "name" ,
    /*04*/ "men" , "name" ,
    /*05*/ "who" , "where" ,
    /*06*/ "whose" , "where" ,
    /*07*/ "which" , "where" ,
    /*08*/ "see" , "select" ,
    /*09*/ "young" , "young" ,
    /*10*/ "man" , "name" ,
    /*11*/ "hair" , "hair_color",

```

```

/*12*/ "in" , "is" ,
/*13*/ "live" , "province" ,
/*14*/ "that" , "where" ,
/*15*/ "lives" , "province" ,
/*16*/ "person" , "name" ,
/*17*/ "peek" , "select" ,
/*18*/ ">>" , "very_very" ,
/*19*/ "are" , "is" ,
/*20*/ "look" , "select" ,
/*21*/ "tell" , "select" ,
/*22*/ "" , "" ,
};

```

```

/*-----*/
/*โครงสร้าง IGNORE เป็นการจองเนื้อที่สำหรับเก็บคำที่ระบบไม่ต้องการใช้ */
/* จำเป็นต้องตัดออกจากระบบ */
/* ignores ใช้เก็บคำที่ต้องตัดออกจากระบบ */
/*-----*/

```

```

struct ignore
{
    char ignores[10];
}

```

```

ignore[]=
{
    /*-----
    NOTE:
    การเพิ่มเติมคำศัพท์ใน ignore
    ต้องไม่มีคำศัพท์นั้นใน synonym
    -----*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

/\*02\*/ "me",  
 /\*03\*/ "what",  
 /\*04\*/ "give",  
 /\*05\*/ "her"  
 /\*06\*/ "please",  
 /\*07\*/ "to",  
 /\*08\*/ "how",  
 /\*09\*/ "many",  
 /\*10\*/ "living",  
 /\*11\*/ "there",  
 /\*12\*/ "do",  
 /\*13\*/ "does",  
 /\*14\*/ "number",  
 /\*15\*/ "a",  
 /\*16\*/ "an",  
 /\*17\*/ "any",  
 /\*18\*/ "some",  
 /\*19\*/ "his",  
 /\*20\*/ "",  
 /\*21\*/ "",

};



```

/*-----*/
/*โครงสร้าง DFAULT เป็นการจองเนื้อที่สำหรับเก็บคำที่ระบบต้องการใช้ */
/*      จำเป็นต้องเพิ่มเติมเข้าไปในระบบ      */
/*      dfault1 ถึง dfault4 ใช้เก็บคำที่ต้องเพิ่มเติมเข้าไปในระบบ      */
/*-----*/

struct dfault
{
    char dfault1[10];
    char dfault2[10];
    char dfault3[10];
    char dfault4[10];
}

dfault[] =
{
    "name", "name", "from", "person",
    /*01*/ "person", "name", "from", "person",
    /*02*/ "people", "name", "from", "person",
    /*03*/ "men", "name", "from", "person",
    /*04*/ "", "", "", "",
};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*----- */
/*โครงสร้าง ATTRB เป็นการจองเนื้อที่สำหรับเก็บชื่อแอตทริบิวต์ */
/*      attrbs ใช้เก็บชื่อแอตทริบิวต์ */
/*----- */

struct attrb
{
    char attrbs[15];
}

attrb[] =
{
    "12345"
    /*01*/ "",
    /*02*/ ""
};

/*=====ENx NAT.H=====*/
/*=====END NAT.H=====*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

๗



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# การขยายการใช้ระบบฐานข้อมูลแบบสัมพันธ์เพื่อตอบคำถามที่คลุมเครือ (Extensions of Relational Database Systems to Support Fuzzy Query)

ดร. เอื้อน ปิ่นเงิน \* พ.อ. ภิรมย์ ทิพย์สิงห์ \*\*

Ouen Pin-ngern Ph.D. Col. Pirom Tipsing

## บทคัดย่อ

ปัจจุบันการเรียกใช้ข้อมูลจากฐานข้อมูลจะใช้ข้อมูลที่ชัดเจน การกำหนดเงื่อนไขจะต้องชัดเจนและครบถ้วน แต่ปกตินมนุษย์จะคิดหาเหตุผลโดยใช้ข้อมูลข่าวสารที่ไม่ค่อยมีความชัดเจน ไม่สมบูรณ์และเป็นข้อมูลที่ ไม่เฉพาะเจาะจง งานวิจัยนี้ได้นำเสนอวิธีการเพิ่มความสามารถของฐานข้อมูลแบบสัมพันธ์โดยใช้แนวคิดฟัซซีเซตและความเป็นไปได้ซึ่งจะทำให้สามารถแทนและดำเนินการกับข้อมูลข่าวสารที่ไม่ชัดเจนได้

## Abstract

At the present, we retrieve precise information from databases. Satisfied conditions must be precise and complete. But human being usually reason with imprecise, incomplete, or vague information. This research deals with an extension of relational database by applying concepts of fuzzy sets and possibility to allow representation and manipulation of imprecise information.

## 1. บทนำ

การวิจัยในด้านวิทยาการคอมพิวเตอร์ เช่น การสร้างระบบผู้เชี่ยวชาญจำเป็นต้องใช้สารสนเทศที่มีข้อเท็จจริงอย่างถูกต้องครบถ้วนเพื่อจำลองการหาเหตุผลของมนุษย์ แต่เมื่อต้องนำสารสนเทศที่เกี่ยวข้องมากมายมาอ้างอิงในการประมวลผล ย่อมทำให้มีสารสนเทศบางส่วนมีความไม่สมบูรณ์หรือมีความคลุมเครือเกิดขึ้นทำให้ไม่สามารถประมวลผลได้ชัดเจนตามต้องการ การวิจัยเรื่อง "การขยายการใช้ระบบฐานข้อมูลแบบสัมพันธ์เพื่อตอบคำถามที่คลุม

เครือ" จึงได้จำลอง Fuzzy Data Base (FDB) ขึ้นมา เพื่อเป็นเครื่องมือในการแก้ปัญหาดังกล่าวนี้

FDB ออกแบบมาเพื่อให้สามารถแสดงและกระทำต่อสารสนเทศที่ไม่สมบูรณ์หรือมีความคลุมเครือ โดยสร้างจากพื้นฐานของระบบฐานข้อมูลแบบสัมพันธ์ ผสมกับทฤษฎีฟัซซีเซตและการกระจายความเป็นไปได้ (Possibilistic Distribution) จึงทำให้สามารถประมวลผลสารสนเทศที่มีความคลุมเครือ ไม่ชัดเจนตามความต้องการและสามารถประมวลผลคำถามที่เป็นภาษาธรรมชาติ (natural-language-like) เหมือนภาษาอังกฤษที่มนุษย์ใช้กันอยู่ทำให้การใช้งานง่ายขึ้น เช่น

\* อาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

\*\* นักศึกษาปริญญาโท สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

"find the people from person who is young"

"list the men from person whose height is tall"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามเหล่านี้ไม่สามารถประมวลผลได้โดยระบบฐานข้อมูลธรรมดา

นอกจากนี้ FDB ออกแบบมาเพื่อให้สามารถแสดงคุณลักษณะเฉพาะอย่างในการมองเห็นข้อมูล เช่น คำจำกัดความฟัซซีของคำว่า YOUNG ตามความเห็นของผู้ใช้คนหนึ่งอาจแตกต่างจากคนอื่น แม้ว่าอายุในฐานข้อมูลจะเป็นตัวเลขจำนวนเดียวกัน จะเห็นได้ว่าแบบจำลอง FDB ไม่เพียงแต่จะเหมาะสมกับการใช้งานในด้านระบบผู้เชี่ยวชาญเท่านั้นยังสามารถใช้ในการประมวลผลในสถานะการณ์ที่มีสารสนเทศไม่ครบถ้วนเพื่อจำลองการให้เหตุผลของมนุษย์ด้วย

2. ฟัซซีเซต

2.1 ความเป็นมาและความหมายของฟัซซีเซต ในปี ค.ศ. 1962 ศาสตราจารย์ ลอตฟี เอ. ซาเดห์ (Lotfi A. Zadeh) ได้นำเสนอบทความเกี่ยวกับฟัซซีเซตและต่อมาในปี ค.ศ. 1965 ศาสตราจารย์ซาเดห์ได้เสนอบทความเพิ่มเติมเป็นทฤษฎีฟัซซีเซต โดยฟัซซีเซตเป็นเซตของความถูกต้องไม่แน่นอนอน (imprecision) ชนิดหนึ่งที่ได้จากการรวบรวมจากบรรดาสมาชิกแล้วจัดเข้าไว้เป็นกลุ่ม (class) แต่ละกลุ่มมีขอบเขตไม่แน่นอน เราเรียกกลุ่มเหล่านี้ว่า "ฟัซซีเซต" [1]

2.2 การนำฟัซซีมาพัฒนา FDB ในชีวิตประจำวันของมนุษย์เรามักจะพบเห็นประโยคที่มีลักษณะคล้ายคลึงกับประโยคต่อไปนี้

"A เป็นเซตของคนหนุ่มในเมือง X"

จากประโยคดังกล่าวทำให้เกิดแนวทางในการนำฟัซซีมาสร้างความสัมพันธ์เพื่อสร้าง FDB ดังนี้

(1) กำหนดเกณฑ์ฟัซซี ใช้ความสัมพันธ์และฟังก์ชันทางคณิตศาสตร์เป็นตัวกำหนดเกณฑ์

(2) กำหนดขอบเขตของฟัซซี ฟัซซีเซตมีการกำหนดขอบเขตไม่แน่นอน เช่น จากข้อ 2.2

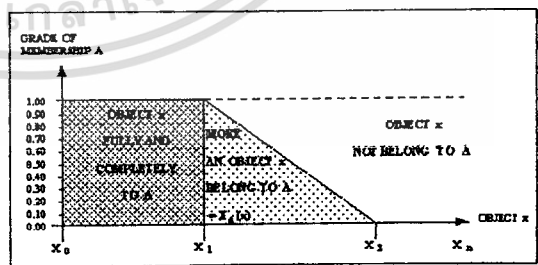
การกำหนดว่าอายุเท่าใดเป็นคนหนุ่ม มีความไม่แน่นอนและคลุมเครือ ในขณะที่ยูนิเวิร์สที่กล่าวถึงคือเมือง X มีการกำหนดขอบเขตที่แน่นอนว่าต้องเป็นเมือง X เท่านั้นจึงจะอยู่ในเกณฑ์

(3) กำหนดค่าฟัซซีเซต ฟัซซีเซตกำหนดค่าความจริงไว้ 3 ระดับคือ สิ่งที่ไม่เป็นสมาชิกของเซตให้ระดับค่าความจริงเป็น F (Fault) หรือ 0 สิ่งที่เป็นสมาชิกของเซตถูกต้องสมบูรณ์ให้ระดับค่าความจริงเป็นจริง T (True) หรือ 1 สิ่งที่เป็นสมาชิกของเซตถูกต้องไม่สมบูรณ์ให้ระดับค่าความจริงเป็นความคลุมเครือหรือฟัซซีหรือให้ระดับค่าความจริงอยู่ระหว่าง 0 กับ 1

(4) การกำหนดฟังก์ชันระดับการเป็นสมาชิก โดยมีเกณฑ์ทั่วไปว่า สมาชิกที่มีค่าถูกต้องสมบูรณ์ในเซต A เราให้ระดับค่าการเป็นสมาชิกเป็น 1 ในขณะที่สมาชิกที่ไม่อยู่ในเซต A เราให้ระดับค่าการเป็นสมาชิกเป็น 0 และให้ค่าสมาชิกอื่น ๆ ภายใต้เซต A ให้มีค่าอยู่ระหว่าง 0 กับ 1 การกำหนดระดับค่าการเป็นสมาชิกของสมาชิกทั้งหมดแสดงได้ดังรูปที่ 1

รูปที่ 1

แสดงกราฟระดับค่าการเป็นสมาชิกของฟัซซีเซต



รูปที่ 1 แสดงให้เห็นว่าสมาชิกของเซต A ที่ถูกต้องสมบูรณ์ (ตั้งแต่  $x_0$  ถึง  $x_1$ ) จะมีระดับค่าการเป็นสมาชิกเท่ากับ 1 สมาชิกของเซต A ที่ถูกต้องไม่สมบูรณ์ (มากกว่า  $x_1$  ถึง  $x_2$ ) จะมีระดับค่าการเป็นสมาชิกเท่ากับ  $X_A(x)$  ตามที่กำหนด สมาชิกอื่น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(มากกว่า  $x_2$  ถึง  $x_n$ ) ที่ไม่เป็นสมาชิกของ A จะมีระดับค่าการเป็นสมาชิกเท่ากับ 0

**2.2 ฟังก์ชันสมาชิก (Membership Functions)** เป็นตัวกำหนดระดับค่าการเป็นสมาชิกของข้อมูล เช่น  $m_{YOUNG}$ ,  $m_{OLD}$  (จะอธิบายรายละเอียดในหัวข้อต่อไป) ผลลัพธ์ที่ได้จากการคำนวณฟังก์ชันสมาชิกเรียกว่าระดับการเป็นสมาชิก (grade หรือ degree of membership)

**2.3 ความสัมพันธ์ (Relations)** ถ้า  $A_i$  เป็นแอตทริบิวต์ที่คลุมเครือภายใต้โดเมน  $D_i$  แล้ว ค่าแอตทริบิวต์จะเป็นค่าที่กระจายที่เป็นไปได้บน  $D_i$  เขียนแทนโดย  $\prod_{A_i}$

กำหนดให้  $D_1, \dots, D_n$  เป็นโดเมน ของสิ่งที่มีนัยเป็นจำนวนเลขได้หรือสิ่งที่มีนัยเป็นจำนวนเลขไม่ได้หรือ unit interval และ ให้ R เป็นความสัมพันธ์ที่เป็นไปได้ของ n เซต นั่นคือ R เป็นเซตของ n-tuple  $(\prod_{A_1}, \dots, \prod_{A_n})$  เมื่อ  $\prod_{A_i}$  ( $i=1, \dots, n$ ) เป็นค่ากระจายที่เป็นไปได้ของแอตทริบิวต์  $A_i$  ซึ่งกำหนดเหนือโดเมนเซต  $D_i$  ในการวิจัยนี้จะนำเสนอความสัมพันธ์ 4 ความสัมพันธ์ คือ

**(1) ความสัมพันธ์ใกล้เคียง (Proximity Relation)** ถ้าให้  $D_i$  เป็นโดเมนของสิ่งที่มีนัยเป็นจำนวนเลขได้ (numerical domain) และ  $x, y, z \in D_i$  แล้ว  $P(x, y) \in [0, 1]$  เป็นความสัมพันธ์ใกล้เคียงตามคุณสมบัติดังนี้

- **คุณสมบัติการสะท้อนกลับ (reflexive)** เป็นความสัมพันธ์ระหว่างตัวเอง เช่น ความสัมพันธ์ระหว่าง  $x$  กับ  $x$  จะมีค่าความจริงเป็น 1 นั่นคือ

$$P(x, x) = 1$$

- **คุณสมบัติสมมาตร (symmetric)** เป็นความสัมพันธ์ระหว่าง  $x$  กับ  $y$  จะมีค่าความจริงเท่ากับความสัมพันธ์ระหว่าง  $y$  กับ  $x$  นั่นคือ

$$P(x, y) = P(y, x)$$

- **คุณสมบัติการถ่ายทอด (transitive)** เป็นความสัมพันธ์ระหว่าง  $x$  กับ  $z$  จะมีค่าความจริงเท่ากับหรือมากกว่าความสัมพันธ์ระหว่าง  $x$  กับ  $y$  คูณกับความสัมพันธ์ระหว่าง  $y$  กับ  $z$  นั่นคือ

$$P(x, z) \geq \max_{y \in D_i} P(x, y) * P(y, z)$$

ความสัมพันธ์ใกล้เคียงได้มีผู้เชี่ยวชาญเสนอไว้หลายรูปแบบด้วยกัน ในบทวิจัยนี้ได้เลือกใช้ความสัมพันธ์ใกล้เคียงที่นำมาอ้างอิงคือ

$$m_{PROXIMITY}(x, y) = e^{-c|x-y|} \text{ เมื่อ } c > 0$$

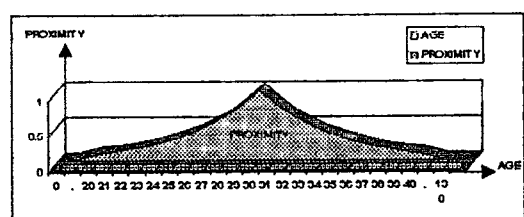
... สมการที่ (1)

ค่า  $c$  จะกำหนดคนทำก็ได้แต่ต้องมากกว่า 0 โดยคำนึงถึงความเหมาะสมกับค่าของข้อมูลและความต้องการของผู้ใช้ เช่น ข้อมูลอายุของมนุษย์กับข้อมูลของเงินเดือนค่าของข้อมูลอยู่ในช่วงต่างกันค่า  $c$  ย่อมไม่เท่ากัน ในที่นี้กำหนดค่า  $c = 2.23 \times 10^{-1}$  และค่า  $e$  คือค่า exponential มีค่า = 2.71828 [2]

เมื่อ  $x$  เป็นเลขจำนวนเต็มที่กำหนด  $y$  เป็นค่าแอตทริบิวต์ที่มีนัยเป็นจำนวนเลขได้ คุณสมบัติของ  $m_{PROXIMITY}$  คือ ถ้า  $P(x, y) = 0$  แล้ว  $x$  กับ  $y$  จะไม่มีความใกล้เคียงกันเลย และถ้า  $P(x, y) = 1$  แล้ว  $x$  กับ  $y$  จะเป็นจำนวนเลขเดียวกัน ตัวอย่างของความสัมพันธ์ PROXIMITY แสดงได้ดังรูปที่ 2

รูปที่ 2

แสดงกราฟของความสัมพันธ PROXIMITY



รูปที่ 2 แสดงให้เห็นค่าของความสัมพันธ์ PROXIMITY จากข้อความ “AGE IS PROXIMITY 30” หรือ “อายุใกล้เคียง 30” เมื่อ x แทนจำนวนเลขที่กำหนดคือ 30 และ y แทนค่าของแอดตริบิวต์ AGE จากกราฟจะเห็นว่าค่า AGE ที่ถูกต้องสมบูรณ์คืออายุเท่ากับ 30 ปี ระดับการเป็นสมาชิกจะมีค่าเท่ากับ 1 และค่า AGE ที่ถูกต้องไม่สมบูรณ์คืออายุไม่เท่ากับ 30 ปี ระดับการเป็นสมาชิกจะมีค่าระหว่าง 1 ถึง 0 โดยระดับการเป็นสมาชิกจะเพิ่มขึ้นเข้าใกล้ 1 ตามค่าอายุที่ใกล้ 30

(2) ความสัมพันธ์ YOUNG ให้เอกภพสัมพัทธ์ (relative universe) เป็นเลขจำนวนเต็มอยู่ในช่วงอายุ [0,130] ปี และให้ x แทนค่าอายุที่มีหน่วยเป็นปีแล้ว YOUNG เป็นชื่อฟังก์ชันที่ระนิยามระดับการเป็นสมาชิก x ต่อ YOUNG ดังนี้

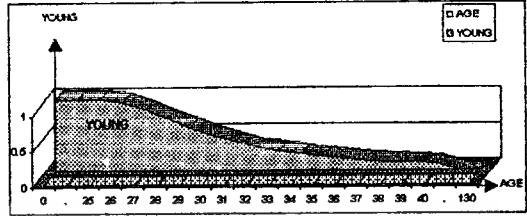
$$m_{YOUNG}(x) = \begin{cases} 1 & \text{เมื่อ } 0 \leq x \leq 25 \\ \left[ 1 + ((x-25)/5) \right]^{-2} & \text{เมื่อ } 25 < x \leq 130 \end{cases} \quad [2]$$

... สมการที่ (2)

จากความสัมพันธ์ YOUNG คนที่มีอายุ 0 ถึง 25 ปี  $m_{YOUNG}$  จะมีค่าเท่ากับ 1 หมายความว่าอายุยังอ่อนวัยหรือเยาว์วัยแน่นอน แต่คนที่มีอายุมากกว่า 25 ปีจนถึงอายุ 130 ปี  $m_{YOUNG}$  จะมีค่าค่อย ๆ ลดลง หมายความว่าความเยาว์วัยจะลดลงตามลำดับอายุ คุณสมบัติของ  $m_{YOUNG}$  คือ ถ้าอายุมาก  $m_{YOUNG}$  จะมีค่าน้อย ถ้าอายุน้อย  $m_{YOUNG}$  จะมีค่ามาก อย่างไรก็ตาม  $m_{YOUNG}$  สามารถกำหนดเป็นรูปแบบอื่นได้ตามความต้องการของผู้ใช้ กราฟของความสัมพันธ์ YOUNG แสดงได้ดังรูปที่ 3

รูปที่ 3

แสดงกราฟของความสัมพันธ์ YOUNG



รูปที่ 3 แสดงให้เห็นคุณสมบัติของ  $m_{YOUNG}$  จากกราฟจะเห็นว่าค่า AGE ที่ถูกต้องสมบูรณ์คืออายุระหว่าง 0 - 25 ปีระดับการเป็นสมาชิกมีค่าเท่ากับ 1 และค่า AGE ที่ถูกต้องไม่สมบูรณ์คืออายุระหว่าง 26 - 130 ปีระดับการเป็นสมาชิกมีค่าระหว่าง 1 ถึง 0 โดยระดับการเป็นสมาชิกจะเพิ่มขึ้นเข้าใกล้ 1 ตามค่าอายุที่ลดน้อยลง

(3) ความสัมพันธ์ OLD ให้เอกภพสัมพัทธ์เป็นเลขจำนวนเต็มอยู่ในช่วงอายุ [0,130] ปี และให้ x แทนค่าอายุที่มีหน่วยเป็นปีแล้ว OLD เป็นชื่อฟังก์ชันที่ระนิยามระดับการเป็นสมาชิก x ต่อ OLD ดังนี้

$$m_{OLD}(x) = \begin{cases} 0 & \text{เมื่อ } 0 \leq x \leq 45 \\ \left[ 1 + ((x-45)/5) \right]^{-2} & \text{เมื่อ } 45 < x \leq 130 \end{cases} \quad [2]$$

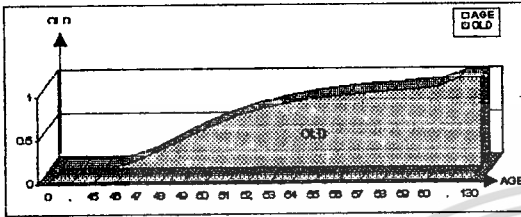
... สมการที่ (3)

จากความสัมพันธ์ OLD คนที่มีอายุ 0 ถึง 45 ปี  $m_{OLD}$  จะมีค่าเท่ากับ 0 หมายความว่าอายุยังไม่แก่ แต่คนที่มีอายุมากกว่า 45 ปีจนถึงอายุ 130 ปี  $m_{OLD}$  จะมีค่าค่อย ๆ เพิ่มขึ้น มีความหมายว่าความแก่เพิ่มขึ้นตามลำดับอายุ คุณสมบัติของ  $m_{OLD}$  คือ ถ้าอายุน้อย  $m_{OLD}$  จะมีค่าน้อย ถ้าอายุมากแล้ว  $m_{OLD}$  จะมีค่ามากตามไปด้วย อย่างไรก็ตาม  $m_{OLD}$  สามารถ

กำหนดเป็นรูปแบบอื่นได้ตามความต้องการของผู้ใช้ กราฟของความสัมพันธ์ OLD แสดงได้ดังรูปที่ 4

รูปที่ 4

แสดงกราฟของความสัมพันธ์ OLD



รูปที่ 4 แสดงให้เห็นคุณสมบัติของ  $m_{OLD}$  จากกราฟจะเห็นว่าค่า AGE คืออายุระหว่าง 0 - 45 ปี ไม่เป็นสมาชิกของ OLD ดังนั้นระดับการเป็นสมาชิกจะมีค่าเท่ากับ 0 และค่า AGE ที่ถูกต้องไม่สมบูรณ์คืออายุระหว่าง 46 - 130 ปีระดับการเป็นสมาชิกมีค่าระหว่าง 0 ถึง 1 โดยระดับการเป็นสมาชิกจะเพิ่มขึ้นเข้าใกล้ 1 ตามอายุที่มากขึ้น

(4) ความสัมพันธ์ TALL ให้เอกภพสัมพัทธ์เป็นเลขจำนวนเต็มอยู่ในช่วงความสูง [0,250] เซนติเมตร และให้  $x$  แทนค่าความสูงที่มีหน่วยเป็นเซนติเมตรแล้ว TALL เป็นชื่อฟังก์ชันที่ใช้นิยามระดับการเป็นสมาชิก  $x$  ต่อ TALL ดังนี้

$$m_{TALL}(x) = \begin{cases} 0 & \text{เมื่อ } 0 \leq x \leq 155 \\ \left[ 1 + \left( \frac{x-155}{10} \right)^{-2} \right]^{-1} & \text{เมื่อ } 155 < x \leq 250 \end{cases}$$

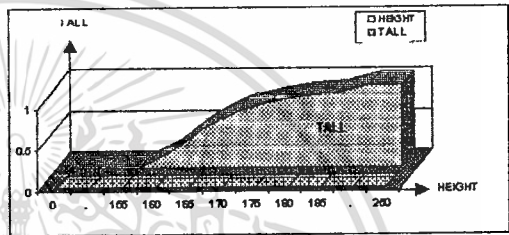
... สมการที่ (4)

จากความสัมพันธ์ TALL คนที่มีความสูง 0 ถึง 155 เซนติเมตร  $m_{TALL}$  จะมีค่าเท่ากับ 0 หมายความว่าจัดอยู่ในกลุ่มคนไม่สูง แต่คนที่มีความสูงมากกว่า 155 เซนติเมตรถึง 250 เซนติเมตร  $m_{TALL}$  จะมีค่าค่อย ๆ เพิ่มขึ้น หมายความว่ากลุ่มของความสูงจะ

เพิ่มขึ้นตามลำดับความสูง คุณสมบัติของ  $m_{TALL}$  คือ ถ้าความสูงน้อย  $m_{TALL}$  จะมีค่าน้อย ถ้าความสูงมากแล้ว  $m_{TALL}$  จะมีค่ามากตามไปด้วย อย่างไรก็ตาม  $m_{TALL}$  สามารถกำหนดเป็นรูปแบบอื่นได้ตามความต้องการของผู้ใช้ กราฟของความสัมพันธ์ TALL แสดงได้ดังรูปที่ 5

รูปที่ 5

แสดงกราฟของความสัมพันธ์ TALL



รูปที่ 5 แสดงให้เห็นคุณสมบัติของ  $m_{TALL}$  จากกราฟแสดงให้เห็นว่าค่าความสูงที่อยู่ระหว่าง 0 - 155 เซนติเมตรระดับการเป็นสมาชิกมีค่าเท่ากับ 0 และค่าความสูงที่ถูกต้องไม่สมบูรณ์คือความสูงที่อยู่ระหว่าง 156 - 250 เซนติเมตร ระดับการเป็นสมาชิกมีค่าระหว่าง 0 ถึง 1 โดยระดับการเป็นสมาชิกจะเพิ่มขึ้นเข้าใกล้ 1 ตามความสูงที่มากขึ้น

2.4 การประยุกต์ใช้ความสัมพันธ์ ความสัมพันธ์ที่เป็นจำนวนเลข

นอกจากใช้งานตามปกติแล้วยังสามารถขยายความสัมพันธ์ตามต้องการได้ดังต่อไปนี้

- (1) การขยายค่าความสัมพันธ์ การขยายค่าความสัมพันธ์ทำได้โดยใช้ขยายค่าความสัมพันธ์ที่เป็นจำนวนเลข คำขยายที่ใช้ เช่น VERY และ NOT การขยายโดย VERY ทำได้หลายวิธี เช่น การถอดราก การยกกำลัง ในบทวิจัยนี้ใช้วิธียกกำลังสอง การขยายโดย VERY จะเพิ่มความเข้มข้นให้กับความสัมพันธ์ เช่น จาก YOUNG (หนุ่ม) เป็น VERY YOUNG (หนุ่มมาก)

การขยายโดย NOT ในบทวิชัยนี้ใช้วิธีหา complement การขยายโดย NOT จะเพิ่มความหมายเชิงปฏิเสธให้กับความสัมพัทธ์ เช่น จาก YOUNG (หนุ่ม) เป็น NOT YOUNG (ไม่หนุ่ม) เป็นต้น รูปแบบการขยายค่าความสัมพัทธ์มีรูปแบบดังตารางที่ 1

คำขยาย	ฟัซซีเทอม	สมาชิกฟังก์ชัน
not	very young	not(very young) = 1 - (very young)
	very old	not(very old) = 1 - (very old)
	very tall	not(very tall) = 1 - (very tall)
not	not young	not(not young) = 1 - (not young)
	not old	not(not old) = 1 - (not old)
	not tall	not(not tall) = 1 - (not tall)

ตารางที่ 1

แสดงรูปแบบการขยายความสัมพัทธ์ที่เป็น "จำนวนเลข"

คำขยาย (modifier)	ฟัซซีเทอม (fuzzy term)	สมาชิกฟังก์ชัน (membership function)
very	young	very(young) = (young) <sup>2</sup>
	old	very(old) = (old) <sup>2</sup>
	tall	very(tall) = (tall) <sup>2</sup>
not	young	not(young) = 1 - young
	old	not(old) = 1 - old
	tall	not(tall) = 1 - tall

ผลลัพธ์การขยายค่าความสัมพัทธ์ เช่น ความสัมพันธ์ YOUNG แสดงได้ดังตารางที่ 3 และรูปที่ 6

(2) การขยายค่าความสัมพัทธ์ การขยายค่าความสัมพัทธ์ทำได้โดยการใช้ขยายค่าความสัมพัทธ์ที่เป็นจำนวนเลข คำขยายที่ใช้ เช่น VERY และ NOT เหมือนการขยายค่าความสัมพัทธ์ในข้อ 2.4(1) การขยายค่าความสัมพัทธ์มีรูปแบบดังตารางที่ 2

ตารางที่ 2

แสดงรูปแบบการขยายค่าความสัมพัทธ์ที่เป็น "จำนวนเลข"

คำขยาย	ฟัซซีเทอม	สมาชิกฟังก์ชัน
very	very young	very(very young) = (very young) <sup>2</sup>
	very old	very(very old) = (very old) <sup>2</sup>
	very tall	very(very tall) = (very tall) <sup>2</sup>
very	not young	very(not young) = (not young) <sup>2</sup>
	not old	very(not old) = (not old) <sup>2</sup>
	not tall	very(not tall) = (not tall) <sup>2</sup>

ผลลัพธ์การขยายค่าความสัมพัทธ์ เช่น ความสัมพันธ์ YOUNG แสดงได้ดังตารางที่ 3 และรูปที่ 6

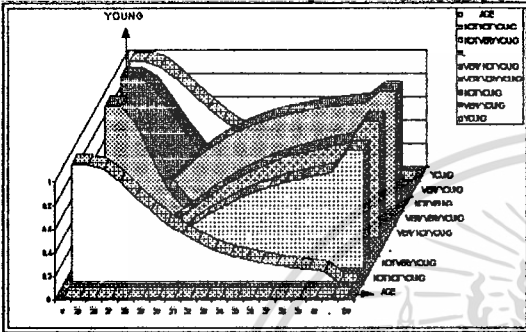
ตารางที่ 3

แสดงระดับการเป็นสมาชิกที่ได้จากการหาค่าความสัมพันธ์การขยายค่าความสัมพัทธ์และการขยายค่าความสัมพัทธ์ YOUNG

Age	YOUNG	VERY	NOT	VERY	VERY	NOT	NOT
		YOUNG	YOUNG	VERY	NOT	VERY	NOT
00	1.000	1.000	0.000	1.000	0.000	0.000	1.000
25	1.000	1.000	0.000	1.000	0.000	0.000	1.000
26	0.962	0.925	0.038	0.586	0.001	0.074	0.962
27	0.862	0.743	0.138	0.552	0.019	0.256	0.862
28	0.735	0.540	0.265	0.291	0.070	0.459	0.735
29	0.610	0.372	0.390	0.138	0.152	0.627	0.610
30	0.500	0.250	0.500	0.062	0.250	0.750	0.500
31	0.410	0.168	0.590	0.028	0.348	0.831	0.410
32	0.338	0.114	0.662	0.013	0.438	0.885	0.338
33	0.281	0.078	0.719	0.006	0.516	0.921	0.281
34	0.236	0.055	0.764	0.003	0.583	0.944	0.236
35	0.200	0.040	0.800	0.001	0.640	0.960	0.200
36	0.171	0.029	0.829	0.000	0.687	0.970	0.171
37	0.148	0.021	0.852	0.000	0.725	0.978	0.148
38	0.129	0.016	0.871	0.000	0.758	0.983	0.129
39	0.113	0.012	0.887	0.000	0.786	0.987	0.113
40	0.100	0.010	0.900	0.000	0.810	0.990	0.100
130	0.000	0.000	1.000	0.000	1.000	1.000	0.000

รูปที่ 6

กราฟแสดงระดับการเป็นสมาชิกที่ได้จากการหาค่าความสัมพันธ์การขยายค่าความสัมพันธ์และการขยายซ้ำค่าความสัมพันธ์ YOUNG



จากรูปที่ 6 แสดงกราฟความสัมพันธ์ ดูจากหลังสุดทางด้านหน้าคือ YOUNG VERY\_YOUNG NOT\_YOUNG VERY\_VERY\_YOUNG VERY\_NOT\_YOUNG NOT\_VERY\_YOUNG NOT\_NOT\_YOUNG ตามลำดับ

(3) การจำกัดคำตอบ การจำกัดคำตอบเพื่อให้ได้จำนวนคำตอบลดลงทำได้โดยใช้ with ตามด้วย threshold (เลขจุดทศนิยมที่ต้องการ) เช่น "age is young with 0.5" จะได้คำตอบเฉพาะที่มีระดับการเป็นสมาชิกมากกว่าหรือเท่ากับ 0.5 เท่านั้น รูปแบบในการคำนวณคือ

$$\text{คำตอบ} = \text{ผลลัพธ์} \text{ เมื่อผลลัพธ์} \geq \text{threshold}$$

2.5 คำถาม (Query) เป็นสิ่งที่ผู้ใช้ใช้เพื่อค้นหาคำตอบจากฐานข้อมูล คำถามที่ใช้สามารถใช้ภาษาธรรมชาติได้ เช่น "find a man from person who is young" ในทวิลักษณ์นี้จะทำการสร้างคำถามภาษา SQL จากภาษาธรรมชาติโดยมีหลักการและขั้นตอนดังนี้

2.5.1 การสร้างประโยคคำถาม ภาษา SQL จากภาษาธรรมชาติ เพื่อประกอบการ

อธิบายจึงขอยกตัวอย่างประโยคคำถามและชื่อฐานข้อมูลดังนี้ คือ

ตัวอย่างประโยคคำถามภาษาธรรมชาติ

"find the people from a person whose age is young"

ตัวอย่างชื่อฐานข้อมูลเสมือน

"tempfdbx"

(1) แฉงประโยคคำถามออกเป็นคำ จากประโยคคำถามภาษาธรรมชาติสามารถแฉงประโยคออกเป็นคำต่าง ๆ ได้ดังนี้

find the people from a person whose age is young

(2) ตัดคำที่ไม่ต้องการออก เลือกคำที่ตัดออกแล้วความหมายไม่เปลี่ยนไป เช่น พวก article อันมี a, an, the เป็นต้น หลังการตัดแล้วจะเหลือกลุ่มคำต่าง ๆ ดังนี้

find people from person whose age is young

(3) หากคำที่มีความหมายเหมือนกันมาแทน FDB ได้เก็บคำศัพท์ที่มีความหมายเหมือนกันเอาไว้ อาจเป็นคำที่มีความหมายเหมือนกันจริงๆ หรือความหมายเหมือนกันโดยอนุโลม เช่น people, person, citizen, man, men และ name เป็นต้น เมื่อนำคำที่มีความหมายเหมือนกันมาแทนจะได้กลุ่มคำต่าง ๆ ดังนี้

select name from person whose age is young

(4) ตัดส่วนเงื่อนไขของประโยคออก ส่วนที่เป็นเงื่อนไขคือส่วนของประโยคตั้งแต่คำว่า who, that, whose เป็นต้นไปจนจบประโยค ดังนั้นจึงทำการตัดส่วนของประโยคอันได้แก่กลุ่มคำต่อไปนี้ คือ

whose age is young

เมื่อตัดส่วนที่เป็นเงื่อนไขของประโยคออกแล้วจะเหลือกลุ่มคำต่าง ๆ ดังนี้

select name from person

(5) หากคำที่ขามาเพิ่มเติมแล้วจัดรูปแบบประโยคใหม่ เพื่อให้ประโยคคำถามสมบูรณ์จึงต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หาสิ่งที่ขาดไปมาเพิ่มเติมให้เป็นคำสั่งที่สมบูรณ์ เช่น แทนชื่อฐานข้อมูลหลัก "person" ด้วยชื่อฐานข้อมูลเสมือน "tempfdbx" เพิ่มเครื่องหมายจุลภาค ( , ) แล้วจัดวางคำบางคำ เช่น age และ young ให้ถูกตำแหน่ง หลังจากนั้นก็จัดเรียงประโยคเสียใหม่ให้สมบูรณ์จะได้ประโยคคำถามภาษา SQL ดังนี้

select	name	,	age	,	young	from	tempfdbx
--------	------	---	-----	---	-------	------	----------

**2.5.2 การแปลภาษาธรรมชาติเป็นภาษา SQL** การแปลภาษาธรรมชาติเป็นภาษา SQL ใช้หลักการในการทำงานเดียวกันกับการสร้างประโยคคำถามดังกล่าวมาแล้ว ระบบนี้ได้แปลภาษาธรรมชาติจากแต่ละคำถามเป็นภาษา SQL รูปแบบต่าง ๆ รวมทั้งส่วนเงื่อนไข เช่น จากตัวอย่างคำถามและชื่อฐานข้อมูลเสมือนที่กล่าวมาแล้ว คือ "find the people from a person whose age is young" และ "tempfdbx" สามารถแปลเป็นชุดคำสั่ง SQL หลายรูปแบบดังต่อไปนี้

(1) SELECT-FORM คือ select name , age , young from tempfdbx

(2) CREATE-FORM คือ create table tempfdbx (name , age , young) as select name , age , age from person

(3) UPDATE-FORM คือ update tempfdbx set young = 1.0 where ((age >= 0) and (age <= 25))

และ

update tempfdbx set young = power(1+power(((age-25)/5),2)-1) where ((age > 25) and (age <= 130))

(4) ALTER-FORM คือ alter table tempfdbx modify young number (4, 3)

(5) DROP-FORM คือ drop table tempfdbx

(6) CONDITION-FORM คือ age is young

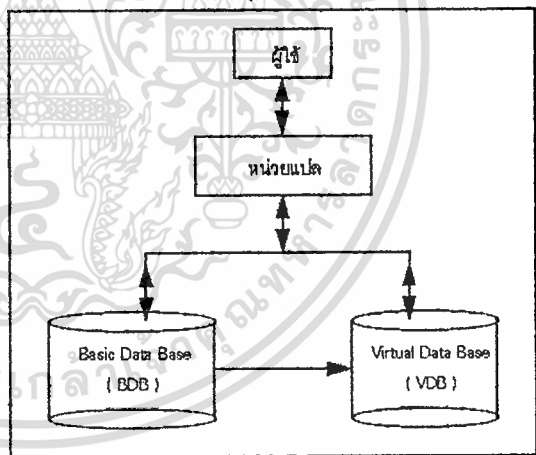
การแปลจะกระทำทุกครั้งที่มีคำถามสำหรับการใช้รูปแบบต่าง ๆ ขึ้นอยู่กับคำตอบคำถามของผู้ใช้ระหว่างการใช้งาน

**3. การสร้างและการออกแบบระบบ FDB**

**3.1 โครงสร้างของระบบ** โครงสร้างของระบบประกอบด้วยส่วนสำคัญ 3 ส่วนคือ ส่วนฐานข้อมูลหลัก ส่วนฐานข้อมูลเสมือนและส่วนแปลคำถาม (ดังรูปที่ 7) การออกแบบฐานข้อมูลทั้งสองจะเป็นไปตามกฎเกณฑ์ของการออกแบบฐานข้อมูลแบบสัมพันธ์

รูปที่ 7

โครงสร้างพื้นฐานของระบบ FDB เพื่อตอบคำถามที่คลุมเครือ



**(1) ฐานข้อมูลหลัก (Basic Data Base หรือ BDB)**

คือฐานข้อมูลแบบสัมพันธ์เดิมที่ใช้อยู่ในภารกิจของหน่วยงานตามปกติ ข้อมูลเหล่านี้จะไม่มีการเปลี่ยนแปลงตลอดเวลาการใช้งานของผู้ใช้ ดังนั้นการกระทำใด ๆ เพื่อหาผลลัพธ์สุดท้ายในระบบ FDB สามารถทำได้โดยไม่เปลี่ยนแปลงข้อเท็จจริงในฐานข้อมูลหลัก ตัวอย่างของฐานข้อมูลหลักแสดงได้ดังตารางที่ 4

## ตารางที่ 4

แสดงความสัมพันธ์ของ PERSON ใน BDB

Name	Age	Height	Province	Hair_Color	Smart
กนก	25	152	ทพบุรีศรี	0.8/brown + 0.6/black	0.6
ปัญญา	28	155	อำเภอ	0.5/brown + 0.4/red	0.7
มงคล	30	160	-	0.3/blue + 0.7/red	0.4
ราตรี	45	167	นนทบุรี,สงขลา	1/brown	0.9
อนันต์	50	178	กรุงเทพฯ,ทพบุรี,คท	0.4/brown + 0.9/red	0.2

จากตารางที่ 4 ค่าแอตทริบิวต์ภายใต้โดเมน Name เป็นเซตของข้อความเดียว ค่าแอตทริบิวต์ภายใต้โดเมน Age และ Height เป็นเซตของตัวเลขที่นับจำนวนได้ ค่าแอตทริบิวต์ภายใต้โดเมน Province เป็นเซตกลุ่มข้อความ ค่าแอตทริบิวต์ภายใต้โดเมน Hair\_color เป็นเซตของการกระจายที่เป็นไปได้ ค่าแอตทริบิวต์ภายใต้โดเมน Smart เป็นเซตของจำนวนจริงในช่วง [0,1]

(2) ฐานข้อมูลเสมือน (Virtual Data Base หรือ VDB) ใช้เก็บรวบรวมความสัมพันธ์หรือฟังก์ชันต่าง ๆ (เช่น ความใกล้เคียง) ซึ่งสิ่งเหล่านี้ "อธิบาย" ถึงวิธีการคำนวณระดับของการใช้ค่าข้อมูลเพื่อตอบคำถามของผู้ใช้ นิยามของฐานข้อมูลส่วนนี้สามารถใช้เพื่อสะท้อนให้เห็นความต้องการคำตอบของผู้ใช้แต่ละคน เราสามารถเปลี่ยนแปลงตามความต้องการของผู้ใช้โดยแอตทริบิวต์ในฐานข้อมูลจะเปลี่ยนค่าของความสัมพันธ์ที่กำหนดไว้ในข้อ 2.3

ตัวอย่างของฐานข้อมูลเสมือนแสดงได้ดังตารางที่ 5

## ตารางที่ 5

แสดงความสัมพันธ์ YOUNG ใน VDB

Name	Age	m <sub>YOUNG</sub>
กนก	25	1.00
ปัญญา	28	0.74
มงคล	30	0.50
ราตรี	45	0.06
อนันต์	50	0.04

\* คำนวณจากสมการ (2)

จากตารางที่ 5 Name Age เป็นข้อมูลที่นำมาจากฐานข้อมูลหลัก สำหรับ m<sub>YOUNG</sub> เป็นข้อมูลที่ไม่ได้อยู่ในฐานข้อมูลหลักแต่ได้จากการคำนวณความสัมพันธ์ที่กำหนดไว้ จึงถือว่าเป็นข้อมูลที่มีความคลุมเครือหรือไม่สมบูรณ์ m<sub>YOUNG</sub> จะเปลี่ยนแปลงตลอดเวลาการใช้งานขึ้นอยู่กับความต้องการของผู้ใช้ ถ้าผู้ใช้มีความเห็นไม่สอดคล้องกับข้อมูลที่มีอยู่ก็สามารถเปลี่ยนได้ตามความเห็นของผู้ใช้เอง

3.2 การสร้างระบบ ในการทดลองผู้วิจัยได้วิเคราะห์และออกแบบฐานข้อมูลทั้งสองส่วนโดยใช้ ORACLE RDBMS version 6.0 ตลอดจนสร้างโปรแกรมส่วนประมวลผลโดยใช้โปรแกรม Pro\*C ของบริษัท Oracle Corporation และ Microsoft C Optimize Compiler ของบริษัท Microsoft Corporation มีรายละเอียดดังนี้

3.2.1 ฐานข้อมูล ทั้งฐานข้อมูลหลักและฐานข้อมูลเสมือนใช้ระบบฐานข้อมูลแบบสัมพันธ์มีรายละเอียดดังนี้

(1) ฐานข้อมูลหลัก PERSON เป็นฐานข้อมูลที่ใช้สำหรับงานใด ๆ ก็ได้ แต่ต้องมีแอตทริบิวต์ที่เกี่ยวข้องกับความสัมพันธ์ที่ต้องการ เช่น ความสัมพันธ์ YOUNG, OLD จะต้องมีแอตทริบิวต์ชื่อ Age และความสัมพันธ์ TALL จะต้องมีแอตทริบิวต์ชื่อ Height เป็นต้น ตัวอย่างฐานข้อมูลหลัก PERSON ได้แสดงไว้ตามตารางที่ 4

(2) ฐานข้อมูลเสมือน TEMPFDBX เป็นฐานข้อมูลเสมือนที่ผู้ใช้กำหนดขึ้นมาระหว่างการใช้งานเพื่อให้ระบบ FDB สร้างขึ้นชั่วคราวสำหรับเก็บข้อมูลความสัมพันธ์ที่ผู้ใช้ต้องการ ฐานข้อมูล TEMPFDBX จะมีแอตทริบิวต์ไม่แน่นอน เพราะระบบจะสร้างขึ้นตามคำสั่งของผู้ใช้และลบทิ้งเมื่อผู้ใช้ แต่สามารถเก็บไว้ใช้งานต่อไปได้ถ้าต้องการ

ตัวอย่างฐานข้อมูลเสมือน TEMPFDBX ได้แสดงไว้ตามตารางที่ 5

3.2.2 โปรแกรม ในการทดลอง ผู้วิจัย ได้เขียนโปรแกรมขึ้นมาประกอบด้วยชุดโปรแกรมต่าง ๆ ดังนี้

(1) **ตัวประมวลผลภาษาธรรมชาติ (Natural Language Processor)** ทำหน้าที่เป็นส่วนรับคำสั่งจากผู้ใช้เป็นภาษาธรรมชาติ โดยขอบเขตของคำสั่งคือ ต้องการเลือกอะไร (SELECT) มาจากที่ไหน (FROM) ด้วยเงื่อนไข (WHERE) อย่างไร และทำหน้าที่แปลคำสั่งภาษาธรรมชาติให้เป็นภาษา SQL. เพื่อใช้สำหรับกระทำกับข้อมูลและฐานข้อมูล เช่น SELECT, UPDATE ข้อมูล CREATE TABLE และ DROP TABLE เป็นต้น

(2) **ตัวประมวลผลหลัก (Main Processor)** ทำหน้าที่ติดต่อกับระบบฐานข้อมูลแบบสัมพันธ์ เช่น การขอเข้าไปทำงานในระบบฐานข้อมูลแบบสัมพันธ์ การจองหน่วยความจำเพื่อทำงานการค้นหาข้อมูล การสร้างปรับปรุงแก้ไขลบทั้งฐานข้อมูล รวมทั้งการออกรายงานทางจอภาพ

(3) **ตัวประมวลผลตัวเลข (Numerical processor)** ทำหน้าที่คำนวณหาคำตอบที่เป็นจำนวนเลขตามเงื่อนไข

(4) **ตัวประมวลผลตัวอักษร (Text Processor)** ทำหน้าที่จัดการหาคำตอบที่เป็นข้อความตามเงื่อนไข

4. การทำงานของระบบ

การใช้ระบบ FDB ทำได้ง่ายเพียงแต่เรียกโปรแกรมแล้วพิมพ์คำสั่งว่าต้องการอะไร จากฐานข้อมูลหลักฐานใด (เช่น PERSON) ด้วยเงื่อนไขอย่างไร บอกชื่อความสัมพันธ์ (เช่น YOUNG) และบอกชื่อฐานข้อมูลเสมือน (เช่น TEMPFDBX) ก็จะได้คำตอบที่ต้องการ ตัวอย่างขั้นตอนการทำงานมีดังนี้

(1) **ป้อนคำสั่ง** ตัวอย่างคำสั่งภาษาธรรมชาติที่ใช้ เช่น "find the people from person whose age is young "

(2) **แปลคำสั่ง** จากคำสั่งภาษาธรรมชาติระบบ FDB จะนำมาวิเคราะห์และแปลตามข้อ 2.5.2 เช่น คำสั่ง "find the people from person whose age is young" จะแปลออกมาเป็นภาษา SQL ได้ดังนี้ "select name , age , young from tempfdbx"

(3) **สร้างเสริมและหาคำตอบ** เมื่อแปลคำสั่งเรียบร้อยแล้วระบบ FDB จะสร้างฐานข้อมูลเสมือนขึ้นชั่วคราวแล้วปรับสภาพข้อมูลให้ถูกต้องเป็นปัจจุบัน ต่อจากนั้นระบบ FDB จะอ่านข้อมูลแล้วนำไปคำนวณหาคำตอบตามเงื่อนไขที่ผู้ใช้งานต้องการ

5. การทดลอง ผู้วิจัยได้สร้างฐานข้อมูลและโปรแกรมตามข้อ 3 แล้วได้ทำการทดลองกับคำสั่งและความสัมพันธ์ดังนี้

(1) คำสั่งแสดงการใช้ m<sub>PROXIMITY</sub> เช่น "list the man from person whose age is proximity 30" จะได้คำตอบทุกค่าตามตารางที่ 6

ตารางที่ 6 แสดงคำตอบจากคำสั่ง "list the man from person whose age is proximity 30"

Name	Age	m <sub>PROXIMITY</sub>
กนก	25	0.328
ปัญญา	28	0.640
มงคล	30	1.000
ราตรี	45	0.035
อนันต์	50	0.012

\* คำนวณจากสมการที่ (1)

(2) คำสั่งแสดงการใช้  $m_{TALL}$  เช่น "list the men from person whose height is tall" จะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 7

ตารางที่ 7

แสดงคำตอบจากคำสั่ง "list the men from person whose height is tall"

Name	Height	$m_{TALL}$
กนก	152	0.00
ปัญญา	155	0.00
มงคล	160	0.20
ราตรี	167	0.50
อนันต์	178	0.84

\* จำนวนจากสมการที่ (4)

(3) คำสั่งแสดงการใช้  $m_{YOUNG}$  เช่น "list the man from person whose age is young" จะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 8

ตารางที่ 8

แสดงคำตอบจากคำสั่ง "list the man from person whose age is young"

Name	Age	$m_{YOUNG}$
กนก	25	1.00
ปัญญา	28	0.74
มงคล	30	0.50
ราตรี	45	0.06
อนันต์	50	0.04

\* จำนวนจากสมการที่ (2)

(4) คำสั่งแสดงการใช้  $m_{OLD}$  เช่น "list the man from person whose age is old" จะได้คำตอบทุกค่าเท่าที่เป็นไปได้ตามตารางที่ 9

ตารางที่ 9

แสดงคำตอบจากคำสั่ง "list the man from person whose age is old"

Name	Age	$m_{OLD}$
กนก	25	0.00
ปัญญา	28	0.00
มงคล	30	0.00
ราตรี	45	0.00
อนันต์	50	0.50

\* จำนวนจากสมการที่ (3)

(5) คำสั่งแสดงการไม่ใช้ Age เช่น "find the people from person who is young" จะได้คำตอบที่ไม่มี Age ตามตารางที่ 10

ตารางที่ 10

แสดงคำตอบจากคำสั่ง "find the people from person who is young"

Name	$m_{YOUNG}$
กนก	1.00
ปัญญา	0.74
มงคล	0.50
ราตรี	0.06
อนันต์	0.04

\* จำนวนจากสมการที่ (2)

(6) คำสั่งแสดงการใช้ with เพื่อจำกัดคำตอบ เช่น "list the man from person whose age is young with 0.5" จะได้คำตอบบางค่าตามตารางที่ 11

## ตารางที่ 11

แสดงคำตอบจากคำสั่ง "list the man from person whose age is young with 0.5"

Name	Age	$m_{\text{YOUNG}}$
กนก	25	1.00
ปัญญา	28	0.74
มงคล	30	0.50

\* จำนวนจากสมการที่ (2) และข้อ 2.4 (3)

(7) คำสั่งแสดงการใช้ very เช่น "find the man from person whose age is very young with 0.5" จะได้คำตอบตามตารางที่ 12

## ตารางที่ 12

แสดงคำตอบจากคำสั่ง "find the man from person whose age is very young with 0.5"

Name	Age	$m_{\text{VERY_YOUNG}}$
กนก	25	1.000
ปัญญา	28	0.547

\* จำนวนจากสมการที่ (2) ตารางที่ (1) และข้อ 2.4 (3)

(8) คำสั่งแสดงการใช้ not เช่น "find the man from person whose age is not young with 0.3" จะได้คำตอบตามตารางที่ 13

## ตารางที่ 13

แสดงคำตอบจากคำสั่ง "find the man from person whose age is not young with 0.3"

Name	Age	$m_{\text{NOT_YOUNG}}$
มงคล	30	0.50
ราตรี	45	0.94
อนันต์	50	0.96

\* จำนวนจากสมการที่ (2) ตารางที่ (1) และข้อ 2.4 (3)

(9) คำสั่งแสดงการใช้ very very เช่น

"find the man from person whose age is very very young with 0.01" จะได้คำตอบตามตารางที่ 14

## ตารางที่ 14

แสดงคำตอบจากคำสั่ง "find the man from person whose age is very very young with 0.01"

Name	Age	$m_{\text{VERY_VERY_YOUNG}}$
กนก	25	1.000
ปัญญา	28	0.029
มงคล	30	0.062

\* จำนวนจากสมการที่ (2) ตารางที่ (2) และข้อ 2.4 (3)

(10) คำสั่งแสดงการใช้ not very เช่น "find the man from person that age is not very young with 0.5" จะได้คำตอบตาม ตารางที่ 15

## ตารางที่ 15

แสดงคำตอบจากคำสั่ง "find the man from person that age is not very young with 0.5"

Name	Age	$m_{\text{NOT_VERY_YOUNG}}$
มงคล	30	0.750
ราตรี	45	0.997
อนันต์	50	0.991

\* จำนวนจากสมการที่ (2) ตารางที่ (2) และข้อ 2.4 (3)

(11) คำสั่งแสดงการหาข้อมูลประเภทข้อความ(string)เดี่ยว เช่น "list the people from person whose name is 'มงคล'" จะได้คำตอบตาม ตารางที่ 16

## ตารางที่ 16

แสดงคำตอบจากคำสั่ง "list the people from person whose name is 'มงคล'"

Name
มงคล

(12) คำสั่งแสดงการหาข้อมูลประเภทที่มีหลายข้อความต่อกัน เช่น "select name from person who lives in 'กรุงเทพ'" จะได้คำตอบตามตารางที่ 17

ตารางที่ 17

แสดงคำตอบจากคำสั่ง "select name from person who lives in 'กรุงเทพ'"

Name	Province
อนันต์	กรุงเทพ.ชลบุรี.ตาก

## 6. ผลการทดลองและข้อจำกัด

ผลการทดลองเป็นไปตามจุดมุ่งหมายที่วางไว้ แต่เนื่องจากการวิจัยนี้มุ่งเน้นความสามารถในการใช้ภาษาธรรมชาติและทฤษฎีฟัซซีเซตเพื่อ "การขยายการใช้ระบบฐานข้อมูลแบบสัมพันธ์เพื่อตอบคำถามที่คลุมเครือ" ในขอบเขตที่ไม่กว้างขวางนัก ดังนั้นจะเห็นว่าระบบที่วิจัยมีขีดจำกัดหลายประการคือ ต้องใช้คำสั่งภาษาอังกฤษตามรูปแบบของ SQL คือ SELECT-FROM-WHERE การใช้ฐานข้อมูลหลักและฐานข้อมูลเสมือนได้อย่างละ 1 ฐานข้อมูล ชื่อแอตทริบิวต์จะใช้ article เช่น a, an, the ไม่ได้ เงื่อนไขจะใช้ตัวเชื่อม (connectives) and และ or เช่น "young and tall" หรือ "young or tall" ไม่ได้และผู้ใช้ไม่สามารถเพิ่มความสัมพันธ์อื่น ๆ นอกเหนือจากที่กำหนดไว้ในระบบ

## 7. สรุปและข้อเสนอแนะ

การวิจัยได้จำลองระบบ FDB ขึ้นมา โดยใช้โครงสร้างของฐานข้อมูลแบบสัมพันธ์ผนวกกับทฤษฎีฟัซซีเซตและการกระจายที่เป็นไปได้ เพื่อให้สามารถทำงานกับระบบสารสนเทศที่มีความคลุมเครือได้

ระบบ FDB อาศัยฐานข้อมูลหลักอันเป็นฐานข้อมูลเดิมซึ่งมีข้อมูลที่ใช้งานอยู่แล้วในการกิจของหน่วยงานตามปกติ เพื่อนำมาสร้างฐานข้อมูลเสมือนขึ้นชั่วคราวประกอบการทำงานให้ได้คำตอบตามความต้องการของผู้ใช้ โดยผู้ใช้จะใช้คำสั่งที่เป็น "ภาษาธรรมชาติ-เหมือน-ภาษาอังกฤษ" จึงทำให้ผู้ใช้ใช้ได้สะดวกและเข้าใจง่าย

จากข้อจำกัดของระบบ FDB ผู้วิจัยมีความคิดว่าสิ่งที่น่าจะวิจัยเพิ่มเติมต่อไปคือ พัฒนาให้สามารถใช้คำสั่งภาษาไทยในการทำงาน ขยายคำสั่งเพิ่มเติมนอกเหนือจากรูปแบบ SQL ให้ใช้ฐานข้อมูลหลักและฐานข้อมูลเสมือนได้มากกว่า 1 ฐานข้อมูล ให้สามารถใช้เงื่อนไขที่มีตัวเชื่อม เช่น and และ or ได้และให้ผู้ใช้สามารถเพิ่มความสัมพันธ์อื่น ๆ นอกเหนือจากที่กำหนดไว้ในระบบได้ด้วย

เอกสารอ้างอิง

1. Abraham Kandel "Fuzzy Mathematical Techniques with Applications", Florida State University, ADDISON-WESLEY PUBLISHING COMPANY, 1986.
2. Deyi Li and Dongbo Liu "A Fuzzy PROLOG Database System" Institute of Computer Engineering People's Republic of China, RESEARCH STUDIES PRESS LTD, 1990.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน

ชื่อผู้เขียน	พันเอก ภิรมย์ ทิพย์สิงห์
วันเดือนปีเกิด	วันที่ ๕ มกราคม ๒๔๘๕
สถานที่เกิด	นครศรีธรรมราช
วุฒิการศึกษาระดับปริญญาตรี	ปริญญาการศึกษาบัณฑิต
สถานที่สำเร็จการศึกษา	มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีที่สำเร็จการศึกษา	๒๕๑๖
ผลงานทางวิชาการที่ได้รับการตีพิมพ์	การขยายการใช้ระบบฐานข้อมูลแบบสัมพันธ์ เพื่อตอบคำถามที่คลุมเครือ
ประสบการณ์การทำงาน	รับราชการที่ ศูนย์กรรมวิธีข้อมูล สำนักปลัดบัญชา ทหาร กองบัญชาการทหารสูงสุด ตำแหน่ง ประจำศูนย์กรรมวิธีข้อมูล สบช.ทหาร สน.สบ.ทหารสูงสุด ตั้งแต่ปี ๒๕๒๓
อาชีพปัจจุบัน	รับราชการทหาร ที่ กรมการสนเทศทหาร กองบัญชาการทหารสูงสุด ตำแหน่ง ผู้ช่วยผู้อำนวยการกองปฏิบัติการ กรมการสนเทศทหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้