สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การศึกษาและพัฒนาระบบดาต้าแวร์เฮาส์

DATA WAREHOUSE

โดย

นางสาวจิตติญาณ์ แก้วแพรก

นางสาวเฉลิมลักษณ์ พันธ์ช่วย

เลขหมู่.....
เลขทะเบียน 37079
วัน, เดือน, ปี 30 ธ.ค. 2543

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การศึกษาและพัฒนาระบบดาต้าแวร์เฮาส์
DATA WAREHOUSE

โดย

นางสาวจิตติญาณ์ แก้วแพรก

นางสาวเฉลิมลักษณ์ พันธ์ช่วย

อาจารย์ที่ปรึกษา

ดร.วรวัฒน์ ลิ้มโภคา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2542

ภาควิชา วิศวกรรมคอมพิวเตอร์

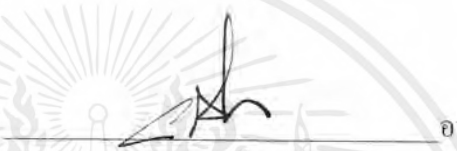
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การศึกษาและพัฒนาระบบดาต้าแวร์เฮาส์

DATA WAREHOUSE

ผู้จัดทำ

1. นางสาวจิตติญาณ์ แก้วแพรก รหัสประจำตัว 39014077
2. นางสาวเฉลิมลักษณ์ พันธ์ช่วย รหัสประจำตัว 39014110



อาจารย์ที่ปรึกษา

(ดร.วรวัฒน์ ลิ้มโภคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การศึกษาและพัฒนาระบบดาต้าแวร์เฮาส์

นางสาวจิตติญาณ์ แก้วแพรก 39014077
นางสาวเฉลิมลักษณ์ พันธ์ช่วย 39014110
ดร. วรวัฒน์ ลิ้มโภคา อาจารย์ที่ปรึกษา
ปีการศึกษา 2542

บทคัดย่อ

โครงงานนี้เป็นการศึกษาและออกแบบระบบฐานข้อมูลดาต้าแวร์เฮาส์ (Data Warehouse) ซึ่งเป็นระบบฐานข้อมูลแบบใหม่ โดยเปรียบเทียบกับระบบฐานข้อมูลทั่วไป (Relational Database) โดยระบบฐานข้อมูลดาต้าแวร์เฮาส์ที่ศึกษาจะใช้หลักการจัดเก็บฐานข้อมูลแบบหลายมิติ (Multidimensional) ซึ่งได้ทำการศึกษาจากซอฟท์แวร์ที่มีความสามารถในการทำงานกับระบบฐานข้อมูลหลายมิติ (Multidimensionality) ที่มีชื่อว่า ไพลอทดิซิชัน ซัพพอร์ท ซุท รุ่น 5.0 และทำการพัฒนาโปรแกรมประยุกต์ (Application) โดยใช้โปรแกรม Delphi รุ่น 4.0 ทำการติดต่อกับผู้ใช้เพื่อให้สามารถเรียกใช้ข้อมูลที่เก็บในระบบฐานข้อมูลดาต้าแวร์เฮาส์ที่สร้างขึ้นบนระบบจัดการฐานข้อมูลออราเคิลได้ โดยโปรแกรมประยุกต์ที่พัฒนาขึ้นจะสามารถนำตารางต่าง ๆ ที่มีโครงสร้างฐานข้อมูลเป็นแบบหลายมิติ มาสร้างเป็นแบบจำลอง (Modal) และทำการวิเคราะห์ได้ตามความต้องการ ทำให้เกิดความยืดหยุ่นในการใช้งานมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Warehouse

Jittiya Kaewprag
Chaloemlak Panchuoi
Dr.Worrawat Limpoka Advisor

Abstract

This project deals with the studying and designing of Data Warehouse, a new kind of Database, and compares with Relational Database. The studying about the multidimensional database theory and concept from a software based on multidimensionality, namely "Pilot Decision Support Suite version 5.0" and the application has been developed with Delphi version 4.0 for searching data from the Oracle database. This application can build models from any tables that a structure is multidimensional database and users can analyze the data from data warehouse with flexibility.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ

อาจารย์ ดร.วรวัฒน์ ลิ้มโภคา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา

อาจารย์ทุก ๆ ท่านที่ประสิทธ์ประสาทวิชาความรู้ต่าง ๆ ตลอดสี่ปีที่ผ่านมา

เพื่อน ๆ 4D ทุกคน สำหรับคำแนะนำ คอยให้ความช่วยเหลือทุกครั้งที่เกิดปัญหา และคอยให้กำลังใจเสมอมา

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูข้าพเจ้ามาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุกๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

สุดท้ายขอมอบคุณความดีจากปริญญานิพนธ์ ฉบับนี้ให้แด่ คุณ มารดา บิดา และครูอาจารย์ผู้มีพระคุณทุกท่าน

นางสาวจิตติญาณ์ แก้วแพรก
นางสาวเฉลิมลักษณ์ พันธ์ช่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญภาพ	VII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของการทำโครงงาน	2
1.3 ขอบเขตของโครงงาน	2
1.4 ขั้นตอนการดำเนินงาน	3
บทที่ 2 ทฤษฎีเบื้องต้น	4
2.1 นิยามของดาต้าแวร์เฮาส์	4
2.2 จุดมุ่งหมายของดาต้าแวร์เฮาส์	4
2.3 จุดอ่อนของระบบฐานข้อมูลแบบทรานเซคชั่น (OLTP)	5
2.4 ความแตกต่างของดาต้าแวร์เฮาส์กับฐานข้อมูลประจำวัน	6
2.5 ความแตกต่างของการจัดการฐานข้อมูล multidimensional database และ relational database	10
2.6 ทำไมฐานข้อมูลประจำวันสำหรับ OLTP จึงทำการนอร์มอลไลซ์	10
2.7 ทำไมในดาต้าแวร์เฮาส์จึงไม่ทำการนอร์มอลไลซ์เหมือนกับฐานข้อมูลประจำวัน	11
บทที่3 โครงสร้างและการออกแบบของดาต้าแวร์เฮาส์	12
3.1 Dimension Modeling	12
3.2 Star join schema	13
3.3 The fact table	15
3.4 The dimension table	16
3.5 Drilling up และ Drilling down	17
3.6 Aggregation	18
3.7 Snow flake	19
3.8 หลักในการทำ query บนตารางที่ออกแบบบนดาต้าแวร์เฮาส์	22
3.9 ขั้นตอนการออกแบบดาต้าแวร์เฮาส์	23
บทที่ 4 Pilot Decision Support Suite Version 5.0	24
4.1 Pilot Desktop	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
4.2 รายละเอียดของแต่ละโมดูล	25
4.3 ประโยชน์ทั่วไปของ software pilot	26
4.4 วิธีการใช้งานซอฟท์แวร์ไฟลอทในส่วนของไพลอทเดสท์ทอป	26
บทที่ 5 การออกแบบและพัฒนาโปรแกรมประยุกต์	35
5.1 การเลือกใช้ทูลส์ที่จะนำมาพัฒนาโปรแกรมประยุกต์	35
5.2 ระบบจัดการฐานข้อมูลออราเคิล 8 (Oracle 8 Database Management System)	38
5.3 การออกแบบฐานข้อมูล (Database design)	38
5.4 การทำ SQL แบบไดนามิก (Dynamic SQL)	40
5.5 การออกแบบส่วนที่ติดต่อกับผู้ใช้ (User interface)	41
บทที่ 6 การใช้งานของโปรแกรมประยุกต์	44
6.1 ทูลส์ที่นำมาใช้ในการพัฒนาโปรแกรมประยุกต์	44
6.2 การทำงานของโปรแกรมประยุกต์	44
6.3 คอมโพเนนท์ใน Delphi4.0 ที่นำมาใช้ในการสร้างโปรแกรมประยุกต์ทางด้านฐานข้อมูล	50
6.4 คอมโพเนนท์พิเศษที่ใช้สำหรับ Multidimensional Database	51
6.5 Cross-tabulations หรือ Crosstabs	56
6.6 อธิบายส่วนต่างๆ ของโปรแกรมประยุกต์	57
บทที่ 7 การทดสอบสมรรถนะของโปรแกรมประยุกต์	68
7.1 การทำ Aggregation fact table	68
7.2 การทดลองสร้างอินเด็กซ์บนฐานข้อมูลดาต้าแวร์เฮาส์ที่มีขนาดเล็ก	70
7.3 ผลทดลองสร้างอินเด็กซ์บนฐานข้อมูลดาต้าแวร์เฮาส์ที่มีขนาดเล็ก	70
7.4 การทดลองสร้างอินเด็กซ์บนฐานข้อมูลดาต้าแวร์เฮาส์ที่มีขนาดใหญ่	71
7.5 ผลทดลองสร้างอินเด็กซ์บนฐานข้อมูลดาต้าแวร์เฮาส์ที่มีขนาดใหญ่	72
7.6 สรุปผลการทดลอง	73
ภาคผนวกที่ ก การสร้างเซอร์วิสเพื่อทำการติดต่อกับเซิฟเวอร์ของออราเคิล	74
ภาคผนวกที่ ข วิธีการสร้าง BDE ในการเชื่อมต่อกับระบบจัดการฐานข้อมูลออราเคิล	80
ภาคผนวก ค คู่มือการใช้โปรแกรมประยุกต์	84
ภาคผนวก ง คู่มือสำหรับโปรแกรมเมอร์	107
เอกสารอ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 1.1 สรุปความแตกต่างของดาต้าแวร์เฮาส์กับระบบฐานข้อมูลประจำวัน	9
ตารางที่ 3.1 แสดงตัวอย่างการนำแอตทริบิวท์ในตารางไดเมนชั่นมาเป็น Row header	17
ตารางที่ 3.2 แสดงรายงานที่มี 3 row header	17
ตารางที่ 3.3 แสดงการ Drill down ฟิลด์ ProductName ให้เป็น Flavor	18
ตารางที่ 7.1 แสดงเวลาที่ใช้ในการ query ของแต่ละ fact table	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

	หน้า
รูปที่ 3.1 แสดงภาพ Dimensional model ของธุรกิจอย่างหนึ่ง	12
รูปที่ 3.2 แสดงภาพการ dice เพื่อเลือกเฉพาะส่วนที่ต้องการ	13
รูปที่ 3.3 แสดงภาพของ Star join schema	14
รูปที่ 3.4 แสดงตัวอย่างของ Fact Table	15
รูปที่ 3.5 แสดงตัวอย่างของ Dimension Table	16
รูปที่ 3.6 แสดงโครงสร้างฐานข้อมูลดาต้าแวร์เฮาส์	19
รูปที่ 3.7 แสดงตารางไดเมนชั่นโปรดักแบบสโนว์เฟลก(Snowflake)	20
รูปที่ 3.8 แสดงโครงสร้างฐานข้อมูลดาต้าแวร์เฮาส์ที่ออกแบบเป็นสโนว์เฟลก(Snowflake)	21
รูปที่ 3.9 แสดง star schema เพื่อทำ star query	22
รูปที่ 4.1 แสดงหน้าจอ Welcome Screen ของ Pilot Desktop	24
รูปที่ 4.2 แสดงหน้าจอสำหรับฐานข้อมูลที่จะให้เชื่อมต่อกับไพลอท	27
รูปที่ 4.3 แสดงหน้าจอการเลือกฐานข้อมูลที่ได้สร้างเอาไว้	27
รูปที่ 4.4 แสดงหน้าจอในการกำหนดค่าการเชื่อมต่อกับฐานข้อมูลที่ได้สร้างเอาไว้	28
รูปที่ 4.5 แสดงหน้าจอที่ใช้ในการสร้างโมเดล	29
รูปที่ 4.6 แสดงไดอะลอกเพื่อใส่ชื่อโมเดลที่ต้องการสร้าง	29
รูปที่ 4.7 แสดงหน้าจอสำหรับเลือก Source table	30
รูปที่ 4.8 แสดงหน้าจอให้ผู้ใช้เลือกตารางไดเมนชัน	30
รูปที่ 4.9 แสดงหน้าจอเมื่อสร้างตารางไดเมนชั่น	31
รูปที่ 4.10 หน้าจอของการกำหนดคุณสมบัติของแต่ละไดเมนชั่น	32
รูปที่ 4.11 หน้าจอแสดงคุณสมบัติของ Level Category	32
รูปที่ 4.12 หน้าจอแสดงคุณสมบัติของ Level Brand	33
รูปที่ 4.13 หน้าจอแสดงคุณสมบัติของ Level Product	33
รูปที่ 4.14 แสดงหน้าจอเมื่อได้ทำการสร้างโมเดลเสร็จเรียบร้อยแล้ว	34
รูปที่ 5.1 แสดงรูปหน้าจอของ BDE Administrator	36
รูปที่ 5.2 แสดงหน้าจอของ Datbase Desktop	37
รูปที่ 5.3 แสดงหน้าจอของ SQL Explorer	38
รูปที่ 5.4 แสดงการออกแบบฐานข้อมูลแบบสโนว์เฟลก	39
รูปที่ 5.5 แสดงฟอร์มที่ใช้การสร้างโมเดล	42
รูปที่ 5.6 แสดงฟอร์มที่ใช้การวิเคราะห์โมเดลที่ได้สร้างเอาไว้	43
รูปที่ 6.1 หน้าจอหลักของโปรแกรมประยุกต์	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

หน้า

รูปที่ 6.2 ฟอร์มในการสร้างโมเดล 46
รูปที่ 6.3 ฟอร์มในการเพิ่มตารางไดเมนชัน 47
รูปที่ 6.4 ฟอร์มที่ใช้ในการเพิ่มตัววัดทางธุรกิจ 47
รูปที่ 6.5 ฟอร์มในการวิเคราะห์ข้อมูล 48
รูปที่ 6.6 ฟอร์มในการเลือกชื่อฟิลด์ที่ต้องการวิเคราะห์ 49
รูปที่ 6.7 ฟอร์มแสดงเงื่อนไขที่กำหนดให้กับแต่ละฟิลด์ของตารางไดเมนชัน 50
รูปที่ 6.8 แสดง component ต่างๆ ที่ใช้ในการจัดการด้านฐานข้อมูล 51
รูปที่ 6.9 แสดง component พิเศษที่ใช้สำหรับ Multidimensional Database 56
รูปที่ 6.10 เปรียบเทียบการแสดงข้อมูลแบบตารางปกติ และแบบ CrossTabs 57
รูปที่ 6.11 ฟอร์มแสดงการเข้าสู่โปรแกรม 58
รูปที่ 6.12 แสดงฟอร์มหลักในการทำงานของโปรแกรม 59
รูปที่ 6.13 แสดงฟอร์มในการวิเคราะห์ข้อมูล 59
รูปที่ 6.14 แสดงฟอร์มในการสร้างโมเดล 61
รูปที่ 6.15 แสดงฟอร์มในการเพิ่มตารางไดเมนชัน 63
รูปที่ 6.16 แสดงฟอร์มในการเพิ่มฟิลด์ในตารางแฟ็ก 63
รูปที่ 6.17 แสดงฟอร์มในการสร้างตารางสโนว์เฟลกสำหรับแต่ละไดเมนชัน 64
รูปที่ 6.18 แสดงฟอร์มในการเลือกตารางไดเมนชันเพื่อการวิเคราะห์ข้อมูล 65
รูปที่ 6.19 แสดงฟอร์มในการใส่เงื่อนไขให้กับแต่ละฟิลว์ในตารางไดเมนชัน 66
รูปที่ 6.20 แสดงฟอร์มสำหรับแสดงข้อมูลในรูปของกราฟ 67
รูปที่ 7.1 โครงสร้างของฐานข้อมูลดาต้าแวร์เฮาส์ที่ได้ทำการจำลองขึ้นมา 68
รูปที่ 7.2 แสดงโครงสร้างสำหรับตารางที่ทำการสร้างอินเด็กซ์ที่มีจำนวนเรคอร์ดน้อย 69
รูปที่ 7.3 แสดงโครงสร้างสำหรับตารางที่ทำการสร้างอินเด็กซ์ที่มีจำนวนเรคอร์ดมาก 69
รูปที่ 7.4 แสดงผลใน log file เมื่อมีการดึงข้อมูลขึ้นมาจากตารางขนาดเล็กที่ยังไม่มีการสร้างอินเด็กซ์ 70
รูปที่ 7.5 แสดงผลใน log file เมื่อมีการดึงข้อมูลขึ้นมาจากตารางขนาดเล็กที่มีการสร้างอินเด็กซ์แล้ว 71
รูปที่ 7.6 แสดงผลใน log file เมื่อมีการดึงข้อมูลขึ้นมาจากตารางขนาดใหญ่
ที่ยังไม่มีการสร้างอินเด็กซ์ 72
รูปที่ 7.7 แสดงผลใน log file เมื่อมีการดึงข้อมูลขึ้นมาจากตารางขนาดใหญ่ที่มีการสร้างอินเด็กซ์แล้ว 73
รูปที่ ก-1 แสดงการเปิดโปรแกรม Oracle Net8 Easy Config 74
รูปที่ ก-2 แสดงหน้าจอให้ใส่ชื่อเซอร์วิสที่ต้องการ 75
รูปที่ ก-3 แสดงหน้าจอให้เลือกโปรโตคอลในการเชื่อมต่อกับฐานข้อมูล 75
รูปที่ ก-4 แสดงหน้าจอให้ใส่ชื่อ Host Name ที่ต้องการติดต่อด้วย 76
รูปที่ ก-5 แสดงหน้าจอให้ใส่ชื่อ Database SID 77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

หน้า

รูปที่ ก-6 แสดงหน้าจอให้ทำการทดสอบเซอร์วิสที่ได้สร้างไว้ 77
รูปที่ ก-7 แสดงหน้าจอให้ใส่ Username และ password เพื่อทดสอบการเชื่อมต่อกับเซิฟเวอร์ 78
รูปที่ ก-8 แสดงหน้าจอสุดท้ายของการสร้างเซอร์วิส 79
รูปที่ ข-1 แสดงหน้าจอ Control Panel เพื่อเปิดโปรแกรม BDE Administrator 80
รูปที่ ข-2 แสดงหน้าจอ Configuration ของออราเคิล 81
รูปที่ ข-3 แสดงหน้าจอการเพิ่ม Alias ที่เป็นออราเคิล 81
รูปที่ ข-4 แสดงหน้าจอการเลือก Service name ที่ได้สร้างขึ้นมา 82
รูปที่ ข-5 แสดงหน้าจอที่ให้ผู้ใช้ใส่ User Name และ Password 82
รูปที่ ข-6 แสดงหน้าจอเมื่อทำการเชื่อมต่อได้เรียบร้อยแล้ว 83
รูปที่ ค-1 แสดงรูปหน้าจอแรกที่ปรากฏเมื่อเข้าสู่โปรแกรม 84
รูปที่ ค-2 แสดงรูปหน้าจอการทำงานหลักของโปรแกรม 85
รูปที่ ค-3 แสดงรูปหน้าจอสำหรับสร้างโมเดล 86
รูปที่ ค-4 แสดงรูปหน้าจอสำหรับการเพิ่มไดเมนชัน 87
รูปที่ ค-5 แสดงรูปหน้าจอสำหรับการเพิ่มฟิลด์ในตารางแฟ็ก 87
รูปที่ ค-6 แสดงรูปหน้าจอเมื่อมีการเชื่อมต่อกับฐานข้อมูลออกราเคิล 88
รูปที่ ค-7 แสดงรูปหน้าจอเมื่อทำการเชื่อมต่อกับฐานข้อมูล 88
รูปที่ ค-8 แสดงรูปหน้าจอเมื่อมีการเลือกตารางไดเมนชันเรียบร้อยแล้ว 89
รูปที่ ค-9 แสดงรูปหน้าจอเมื่อทำการเชื่อมต่อกับฐานข้อมูล ในการเพิ่มฟิลด์ของตารางแฟ็ก 90
รูปที่ ค-10 แสดงรูปหน้าจอเมื่อมีการเลือกตารางไดเมนชัน และตัววัดเรียบร้อยแล้ว 90
รูปที่ ค-11 แสดงรูปหน้าจอเมื่อมีการดับเบิลคลิกไดเมนชัน Customer เพื่อสร้างสโนว์เฟลก 91
รูปที่ ค-12 แสดงรูปหน้าจอเมื่อมีการเลือกไดเมนชัน และตัววัดได้ตรงตามความต้องการแล้ว 92
รูปที่ ค-13 แสดงรูปหน้าจอเมื่อคลิกที่ปุ่ม Save หรือ Save As... 93
รูปที่ ค-14 แสดงรูปหน้าจอเมื่อมีการเลือกไฟล์ที่จะนำมาใช้ในการวิเคราะห์ 93
รูปที่ ค-15 แสดงรูปหน้าจอเมื่อมีการคลิกที่ปุ่ม View 94
รูปที่ ค-16 แสดงรูปหน้าจอเมื่อมีการเลือกฟิลด์ของตารางไดเมนชัน 94
รูปที่ ค-17 แสดงรูปหน้าจอเมื่อมีการเลือกตัววัดทางธุรกิจ 95
รูปที่ ค-18 แสดงรูปหน้าจอเมื่อมีการ Query ข้อมูลที่ได้ทำการเลือกไว้สรุปตามรายปี 96
รูปที่ ค-19 แสดงรูปหน้าจอเมื่อมีการเลือกที่ปุ่ม Show SQL เป็นข้อมูลที่สรุปตามรายเดือน 96
รูปที่ ค-20 แสดงหน้าจอการกำหนดค่า Time frame ให้สรุปข้อมูลเป็นรายเดือน 97
รูปที่ ค-21 แสดงหน้าจอสำหรับการกำหนดเงื่อนไขในฟิลด์ Parent ของไดเมนชัน Customer 98
รูปที่ ค-22 แสดงหน้าจอสำหรับการ Query ตามเงื่อนไขที่ได้กำหนดไว้ 99
รูปที่ ค-23 แสดงกราฟของราคาขายสำหรับ Flavor ของสินค้าต่างๆ ในแต่ละเดือน 100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

หน้า

รูปที่ ค-24 การ Drill down จาก DistrictName ลงมาเป็น CustomerName 101

รูปที่ ค-25 แสดงหน้าจอการวิเคราะห์ข้อมูลที่ได้มาจากการ Drill down จาก DistrictName ไปเป็น CustomerName 101

รูปที่ ค-26 การ Drill up จาก DistrictName ขึ้นไปเป็น RegionName 102

รูปที่ ค-27 แสดงหน้าจอการวิเคราะห์ข้อมูลที่ได้มาจากการ Drill up จาก BrandName ไปเป็น CategoryName 102

รูปที่ ค-28 แสดงหน้าจอในการเลือก option aggregation 103

รูปที่ ค-29 แสดงหน้าจอสำหรับเลือก aggregation fact table สำหรับรายเดือน 104

รูปที่ ค-30 แสดงหน้าจอสำหรับเลือก aggregation fact table สำหรับรายปี 104

รูปที่ ค-31 แสดงหน้าจอสำหรับการวิเคราะห์ข้อมูลตามรายเดือน 105

รูปที่ ค-32 แสดงหน้าจอสำหรับการวิเคราะห์ข้อมูลตามรายปี 106

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันนี้มีการใช้ฐานข้อมูลอย่างกว้างขวางในระบบงานทั่วๆ ไป จึงมีการวิจัยและพัฒนาวิธีการเก็บข้อมูลจำนวนมาก รวมถึงการค้นหาและนำข้อมูลที่ต้องการออกมาจากระบบฐานข้อมูลด้วย แต่เนื่องจากระบบฐานข้อมูลทั่วไป (Operational Database) ที่นิยมใช้กันอยู่ในปัจจุบันนี้ มีหลักในการเก็บข้อมูลที่เน้นในเรื่องการลดความซ้ำซ้อน (Redundancy) รักษาความถูกต้อง (Integrity) ลดการสูญหายของข้อมูล (Information Lost) และลดความผิดพลาดที่เกิดขึ้นจากการแก้ไขข้อมูล (Update Anomalies)

เนื่องจากฐานข้อมูลทั่วไป มีลักษณะดังได้กล่าวมาแล้ว จึงมีความสามารถเพียงแค่การเรียกใช้ข้อมูลที่มีอยู่ แต่ไม่สามารถจะนำมาช่วยในการสนับสนุนการตัดสินใจได้ เพราะเมื่อมีการเรียกใช้ข้อมูลจะต้องเรียกใช้ข้อมูลจากฐานข้อมูลขนาดใหญ่ ซึ่งมีข้อมูลจำนวนมหาศาล และมีการแตกตารางที่นอร์มอลไลซ์ (Normalized tables) แล้วออกเป็นหลายตารางจึงไม่รองรับคำถามที่ต้องการจะนำมาใช้ช่วยสนับสนุนการตัดสินใจ (decision support queries) เพราะคำถาม (queries) ทั้งหมดของระบบที่ช่วยสนับสนุนการตัดสินใจ (Decision Support System) จะมีการรวม (join) กันของตารางต่างๆ ที่ซับซ้อน และนอกจากนี้การเก็บข้อมูลในระบบฐานข้อมูลทั่วไป ยังไม่มีการเก็บข้อมูลย้อนหลัง (historical data) เพื่อใช้ช่วยในการคาดคะเนแนวโน้มที่คาดว่าจะเป็นไปในอนาคต

ดังนั้นจึงทำการศึกษาถึงระบบฐานข้อมูลที่สามารถให้ข้อมูลที่นำมาช่วยสนับสนุนการตัดสินใจได้เพื่อจะได้สามารถใช้ข้อมูลที่มีอยู่ได้อย่างมีประสิทธิภาพสูงสุด ซึ่งฐานข้อมูลที่ได้ทำการศึกษานั้นก็คือ ระบบฐานข้อมูลดาต้าแวร์เฮาส์ (Data Warehouse) เนื่องจากระบบฐานข้อมูลดาต้าแวร์เฮาส์ ถูกคิดขึ้นมาเพื่อช่วยให้ผู้ใช้เรียกใช้ข้อมูลด้วยวิธีที่สร้างสรรค์ เพราะธรรมชาติที่แตกต่างกันระหว่างระบบฐานข้อมูลดาต้าแวร์เฮาส์และระบบฐานข้อมูลทั่วไป

โดยระบบฐานข้อมูลดาต้าแวร์เฮาส์ จะแยกกลุ่มข้อมูลสารสนเทศที่ใช้ในการวิเคราะห์ทางธุรกิจออกจากฐานข้อมูลที่ใช้ประจำวัน มาเก็บอยู่ในระบบจัดการฐานข้อมูล (Relational Database Management Systems) ประสิทธิภาพสูง และทำให้การเรียกใช้ข้อมูลชุดนี้ทำได้อย่างยืดหยุ่น จากเครื่องมือ (Tool) ที่อยู่บนเครื่องเดสก์ทอปทั่วไป โดยลดออฟโหลดดิ้ง (off-loading) เพิ่มกลไกการช่วยตัดสินใจ ปรับปรุงเวลาที่ตอบสนอง (response time) ให้รวดเร็วขึ้นอย่างมากและผู้บริหารสามารถเรียกข้อมูลรายละเอียดที่จำเป็น ที่ถูกเก็บมาก่อนหน้านี้ (historical data) มาใช้ช่วยให้การตัดสินใจทางธุรกิจให้เกิดความแม่นยำมากยิ่งขึ้น โดยระบบนี้พิจารณาถึงการใช้ประโยชน์จากข้อมูลเป็นหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของการทำโครงงาน

1.2.1 ศึกษาว่าระบบฐานข้อมูลดาต้าแวร์เฮาส์ คืออะไร มีประโยชน์อย่างไรในเชิงธุรกิจ

1.2.2 ศึกษาว่าระบบฐานข้อมูลดาต้าแวร์เฮาส์นี้ มีโครงสร้างข้อมูล (Data Structure) เป็นอย่างไร และมีหลักการในการออกแบบระบบฐานข้อมูลดาต้าแวร์เฮาส์ อย่างไร

1.2.3 ศึกษาระบบจัดการฐานข้อมูลออราเคิล (Oracle) ว่ามีฟังก์ชันการทำงานเป็นอย่างไร และสามารถสร้างระบบฐานข้อมูลดาต้าแวร์เฮาส์บนออราเคิลได้อย่างไร

1.2.4 ศึกษาการออกแบบและการสร้างระบบฐานข้อมูลดาต้าแวร์เฮาส์บนระบบจัดการฐานข้อมูลออราเคิล ว่าต้องมีการออกแบบส่วนใดบ้าง และจะมีวิธีการใดในการทดสอบว่าระบบฐานข้อมูลดาต้าแวร์เฮาส์ ที่ทำการออกแบบไว้นั้นสามารถดึงข้อมูลที่ต้องการออกมาได้อย่างถูกต้อง

1.2.5 ศึกษา Pilot Decision Support Suite Version 5.0 ซึ่งเป็นทูลส์ (Tools) ที่ใช้ในการเรียกดูข้อมูลที่เก็บในฐานข้อมูลดาต้าแวร์เฮาส์ได้โดยไม่จำกัด ทำให้มีความยืดหยุ่นในการใช้งานมากยิ่งขึ้น

1.2.6 ศึกษาวิธีการและทดลองทำการเขียนทูลส์ ทำการติดต่อกับผู้ใช้ (User Interface) เพื่อให้ผู้ใช้สามารถเรียกใช้ข้อมูลที่เก็บในระบบฐานข้อมูลดาต้าแวร์เฮาส์ที่สร้างบบระบบจัดการฐานข้อมูลออราเคิลได้โดยง่าย

1.3 ขอบเขตของโครงงาน

โครงงานนี้เป็นการศึกษาระบบฐานข้อมูลดาต้าแวร์เฮาส์ ที่สามารถให้ข้อมูลที่นำมาช่วยสนับสนุนการตัดสินใจเพื่อจะได้รู้ถึงประโยชน์ โครงสร้าง การออกแบบ และวิธีการใช้งานระบบฐานข้อมูลดาต้าแวร์เฮาส์ เพื่อให้ได้ประสิทธิภาพสูงสุด โดยศึกษาการทำงานของทูลส์ที่มีชื่อว่า Pilot Decision Support Suite Version 5.0 ซึ่งเป็นทูลส์ที่มีประสิทธิภาพสูงสามารถนำฐานข้อมูลดาต้าแวร์เฮาส์ขึ้นมาใช้งานได้โดยไม่จำกัด ทำให้เกิดความยืดหยุ่นแก่ผู้ใช้งานเป็นอย่างมาก

หลังจากที่ได้ทำการศึกษาแล้ว จึงทำการออกแบบระบบฐานข้อมูลดาต้าแวร์เฮาส์โดยใช้ข้อมูลจากโปรแกรมไพลอทเป็นกรณีศึกษา และสร้างโปรแกรมประยุกต์ (Application) โดยใช้โปรแกรมเดลไฟเวอร์ชั่น 4.0 (Delphi 4.0) เพื่อให้ผู้ใช้สามารถทำการเรียกใช้และติดต่อกับระบบจัดการฐานข้อมูลออราเคิลได้ โดยโปรแกรมประยุกต์ที่พัฒนาขึ้นนี้จะมีความอ่อนตัวเป็นอย่างมาก เนื่องจากทูลส์ที่สร้างขึ้นนี้สามารถใช้ในการขับเคลื่อนตารางใดๆ ก็ได้ที่มีโครงสร้างฐานข้อมูลเป็นแบบมัลติไดเมนชันนอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ทำการศึกษาเกี่ยวกับระบบฐานข้อมูลดาต้าแวร์เฮาส์ และประโยชน์ของระบบฐานข้อมูลดาต้าแวร์เฮาส์ในเชิงธุรกิจ โดยการศึกษาจากหนังสืออ้างอิง

1.4.2 ทำการศึกษาโครงสร้างของระบบฐานข้อมูลดาต้าแวร์เฮาส์ และศึกษาหลักการในการออกแบบระบบฐานข้อมูลดาต้าแวร์เฮาส์

1.4.3 ศึกษาโปรแกรมไพลอท เพื่อดูฟังก์ชันการทำงาน และแนวคิดในการวิเคราะห์ข้อมูลของโปรแกรม

1.4.4 ทดลองออกแบบระบบฐานข้อมูลดาต้าแวร์เฮาส์จากข้อมูลที่ได้ศึกษาจากโปรแกรมไพลอท และทดสอบว่าสามารถใช้คำสั่ง SQL ดึงข้อมูลที่ต้องการออกมาได้อย่างถูกต้อง

1.4.5 ศึกษารายละเอียดและประสิทธิภาพของทูลส์ที่จะนำมาใช้ในการสร้างโปรแกรมประยุกต์ (Application) ซึ่งทูลส์ที่จะนำมาใช้นั้นก็คือโปรแกรมเดลไฟเวอร์ชัน 4.0

1.4.6 ศึกษาระบบจัดการฐานข้อมูลออราเคิล เพื่อให้ติดต่อกับโปรแกรมเดลไฟได้ และทำการสร้างระบบฐานข้อมูลดาต้าแวร์เฮาส์บนออราเคิลได้

1.4.7 ทำการสร้างโปรแกรมประยุกต์ด้วยโปรแกรมเดลไฟ ซึ่งตอนแรกทดลองให้ติดต่อกับฐานข้อมูลพาราดอกซ์ก่อน เมื่อสามารถใช้งานได้ จึงทำให้ติดต่อกับฐานข้อมูลออราเคิล

1.4.8 ทำการทดสอบประสิทธิภาพโปรแกรมประยุกต์ที่สร้างขึ้นมา ถ้ายังพบข้อผิดพลาดหรือไม่พอใจในประสิทธิภาพการทำงานก็แก้ไขจนกว่าจะสามารถใช้งานได้เป็นอย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีเบื้องต้น

2.1 นิยามของดาต้าแวร์เฮาส์

ดาต้าแวร์เฮาส์ คือการแยกฐานข้อมูลที่จะใช้งานกับระบบช่วยเหลือการตัดสินใจออกจากระบบฐานข้อมูลประจำวัน (Operational Database) โดยฐานข้อมูลดาต้าแวร์เฮาส์ อาจจะนำข้อมูลมาจากฐานข้อมูลประจำวัน หรือมีการประมวลผลข้อมูลจากฐานข้อมูลประจำวัน ให้กลายเป็นสารสนเทศ (Information) หรือกระทั่งนำข้อมูลอื่นเพิ่มเติมเข้ามาก็ได้

นิยามของดาต้าแวร์เฮาส์ คือฐานข้อมูลที่ช่วยในการจัดหาสำหรับกระบวนการตัดสินใจโดยฐานข้อมูลดังกล่าวจะต้องมีคุณสมบัติดังนี้

2.1.1 Subject Oriented : ข้อมูลจะต้องถูกสร้างขึ้นมาจากหัวข้อ (subject) ธุรกิจที่สนใจ เช่น ถ้าบริษัทประกันภัยต้องการใช้ดาต้าแวร์เฮาส์ ฐานข้อมูลที่ได้จะต้องสร้างขึ้นจากประวัติลูกค้า เบี้ยประกัน และการเรียกร้อง แทนที่จะแยกตามชนิดของผลิตภัณฑ์ หรือบริการประกันภัย/ประกันชีวิต ข้อมูลที่สร้างขึ้นจะประกอบด้วยหัวข้อที่เก็บเฉพาะข่าวสารที่จำเป็นสำหรับกระบวนการตัดสินใจเท่านั้น

2.1.2 Integrated : ข้อมูลถูกรวบรวมจากแหล่งต่างๆ จากระบบปฏิบัติการ รูปแบบของข้อมูล เพลตฟอร์ม (plat form) ที่หลากหลาย สร้างขึ้นเป็นฐานข้อมูลที่สอดคล้องเป็นหนึ่งเดียว เช่น ค่าของตัวแปรตัวเดียวกันในแต่ละฐานข้อมูลอาจต่างกัน ฐานข้อมูลหนึ่งอาจใช้ 0 และ 1 อีกฐานข้อมูลหนึ่งอาจใช้ T และ F ดังนั้นฐานข้อมูลที่สร้างใหม่จะต้องได้รับการกำหนดค่าตัวแปรให้เหมือนกันเป็นหนึ่งเดียว

2.1.3 Time-variant : ข้อมูลซึ่งใช้ตัดสินใจที่เก็บไว้ จะต้องมีอายุประมาณ 5 ถึง 10 ปี เพื่อใช้เปรียบเทียบ หาแนวโน้ม และทำนายผลลัพธ์ในอนาคตได้

2.1.4 Non-volatile : ข้อมูลจะไม่อัพเดตหรือถูกทำให้เปลี่ยนแปลงง่ายๆ ผู้ใช้สามารถใช้ฐานข้อมูลดาต้าแวร์เฮาส์ ได้เพียงแค่โหลด (load) และเข้าถึง (access) เท่านั้น

2.2 จุดมุ่งหมายของดาต้าแวร์เฮาส์

เป้าหมายของการออกแบบดาต้าแวร์เฮาส์ คือ การแยกกลุ่มข้อมูลสารสนเทศที่ใช้ในการวิเคราะห์ทางธุรกิจออกจากฐานข้อมูลประจำวันมาเก็บอยู่ใน RDBMS (Relational Database Management System) ประสิทธิภาพสูง และทำให้การเรียกใช้ข้อมูลชุดนี้ทำได้อย่างยืดหยุ่น จากทูลส์ที่อยู่บนเครื่องเดสก์ทอปทั่วไป โดยลดออฟโหลดดิ้ง (off-loading) เพิ่มกลไกการช่วยตัดสินใจ ปรับปรุงเวลาที่ตอบสนอง (response time) รวดเร็วขึ้นอย่างมาก และผู้บริหารสามารถเรียกข้อมูลรายละเอียดที่จำเป็น ที่ถูกเก็บมาก่อนหน้านี้ (historical data) มาใช้ช่วยให้การตัดสินใจทางธุรกิจแม่นยำขึ้น

เป้าหมายในการจัดทำดาต้าแวร์เฮาส์มีดังต่อไปนี้

2.2.1 ดาต้าแวร์เฮาส์ทำให้สามารถเข้าถึงข้อมูลขององค์กรได้ ผู้จัดการและนักวิเคราะห์ขององค์กรสามารถเชื่อมต่อเข้าไปยังดาต้าแวร์เฮาส์จากเครื่องคอมพิวเตอร์ของตนได้ ซึ่งการเชื่อมต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถทำได้ในทันที ตามความต้องการ และมีประสิทธิภาพสูง ทูลส์ที่มีให้กับผู้จัดการ และนักวิเคราะห์สามารถใช้งานได้ง่าย สามารถออกรายงาน (report) ได้ด้วยการคลิกปุ่มๆ เดียว

2.2.2 ข้อมูลในดาต้าแวร์เฮาส์จะต้องถูกต้องตรงกันหมด คำถามเดียวกันต้องได้รับคำตอบที่เหมือนกันเสมอ ไม่ว่าผู้ถามจะเป็นใคร ถามเวลาใด

2.2.3 ข้อมูลในดาต้าแวร์เฮาส์สามารถถูกตัดและหมุนดูได้ทุกแกน หมายถึง ข้อมูลสามารถถูกวิเคราะห์จากหัวข้อในธุรกิจประเภทนั้น โดยแบ่งข้อมูลหรือรวมข้อมูลมาวิเคราะห์ตามความต้องการได้

2.2.4 60% ของดาต้าแวร์เฮาส์ คือฮาร์ดแวร์ ของ central data warehouse, relational database software และข้อมูล (data) ส่วนที่เหลืออีก 40% นั้นคือ กลุ่มของฟร้อนท์เอ็นด์ทูลส์ (set of front end tools) ที่ทำหน้าที่ในการป้อนคำถาม วิเคราะห์ และแสดงข้อมูลออกมา

2.2.5 ดาต้าแวร์เฮาส์เป็นส่วนที่ผลิตข้อมูลจาก OLTP ข้อมูลไม่เพียงแต่ถูกรวบรวมมาไว้ที่ศูนย์กลางอย่างเดียว แต่จะถูกรวบรวมอย่างระมัดระวังจากแหล่งข้อมูลหลายๆ แห่งภายนอกองค์กรด้วย แล้วมาปรับปรุงให้เหมาะสมกับการใช้งานเท่านั้น ถ้าข้อมูลเชื่อถือไม่ได้ หรือไม่สมบูรณ์ จะไม่ถูกอนุญาตให้นำไปใช้ได้

2.2.6 คุณภาพของข้อมูลในดาต้าแวร์เฮาส์เป็นตัวผลักดันให้สามารถทำการรีเอ็นจิเนียนริ่งธุรกิจได้

2.3 จุดอ่อนของระบบฐานข้อมูลแบบทรานแซคชั่น (OLTP)

ตามธรรมชาติของระบบฐานข้อมูลประจำวันจะเป็นระบบทรานแซคชั่น (Transaction Systems) แต่เนื่องจากทรานเซคชั่นมักจะต้องเรียกใช้ข้อมูลจากฐานข้อมูลขนาดใหญ่ เมื่อต้องการเรียกใช้ข้อมูลเพื่อช่วยในการตัดสินใจ ทำให้มีข้อมูลจำนวนมหาศาล ทรานแซคชั่นจึงมีปัญหาในการเรียกข้อมูลทรานแซคชัน (transaction data) ดังต่อไปนี้

2.3.1 ข้อมูลทรานแซคชันเรียกใช้ได้ยาก

2.3.2 มีการแตกตารางที่นอร์มอลไลซ์ (normalized tables) แล้วออกเป็นหลายตาราง

2.3.3 ไม่รองรับคำถามที่สนับสนุนการตัดสินใจ (decision support queries) เพราะคำถาม (Query) ทั้งหมดของระบบสนับสนุนการตัดสินใจ (Decision support system) มีการรวม (join) กันของตารางต่างๆ ที่ซับซ้อน ซึ่งจะทำให้ประสิทธิภาพของ transaction operation database ตกลง และทำงานช้าลง

2.3.4 มีข้อมูลย้อนหลัง (historical data) น้อย

ดังนั้นดาต้าแวร์เฮาส์จึงถูกคิดขึ้นมาเพื่อช่วยให้ผู้ใช้เรียกใช้ข้อมูลด้วยวิธีที่สร้างสรรค์เพราะธรรมชาติที่แตกต่างกันระหว่างดาต้าแวร์เฮาส์และฐานข้อมูลประจำวันโดยดาต้าแวร์เฮาส์จะแยกฐานข้อมูลออกมา เก็บข้อมูลสารสนเทศ (Information) ที่สรุปมาเฉพาะหัวข้อที่สนใจ เพื่อวิเคราะห์ใช้ในการบริหาร และควบคุมธุรกิจ โดยระบบนี้พิจารณาถึงการใช้ประโยชน์จากข้อมูลเป็นหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ความแตกต่างของ ดาต้าแวร์เฮาส์กับฐานข้อมูลประจำวัน

Online transaction processing (OLTP) และ dimensional data warehousing (DDW) นั้น มีความแตกต่างกันอย่างมากมายดังต่อไปนี้

2.4.1 ความถูกต้อง (Consistency) : ทั้ง OLTP และดาต้าแวร์เฮาส์ต่างก็ให้ความสำคัญในเรื่องของ ความถูกต้องของข้อมูล (data consistency)

สำหรับ OLTP ซึ่งมีการทำทรานแซคชั่นจำนวนมากๆ นั้น สิ่งที่ต้องการ คือการทำทรานแซคชั่นให้ครบ(ไม่มีการสูญหาย) ดังนั้นจึงมีความจำเป็นผู้ส่งและผู้รับจะต้องรับรู้และตรวจสอบอยู่ตลอดเวลาว่า ขณะนี้มีการทำทรานแซคชันเกิดขึ้นหรือไม่

สำหรับดาต้าแวร์เฮาส์จะไม่สนใจการทำทรานแซคชั่นแต่ละครั้ง แต่จะสนใจว่า การโหลดข้อมูลใหม่เข้ามานั้นทำเสร็จแล้วหรือยัง (โหลดข้อมูลเข้ามาทั้งหมดและข้อมูลถูกต้อง)

2.4.2 ทรานแซคชั่น (Transaction)

สำหรับระบบ OLTP นั้นในแต่ละวันอาจมีการทำทรานแซคชั่นเป็นหมื่นเป็นแสนครั้ง ซึ่งการทำทรานแซคชั่นแต่ละครั้งจะใช้ข้อมูลเพียงเล็กน้อยเท่านั้น

สำหรับดาต้าแวร์เฮาส์แต่ละวันจะทำเพียงแค่ 1 ทรานแซคชั่น ซึ่งทรานแซคชั่นนี้อาจต้องใช้ข้อมูลเป็นล้าน เรคอร์ด (record) เลยก็ได้ ดังนั้นเราจึงเรียกกระบวนการนี้ว่า production data load แทน สิ่งที่เราสนใจในกระบวนการนี้มีเพียงแค่ production data load เท่านั้น ถ้าการทำ production data load ถูกทำให้หยุดกลางคัน ระบบจะทำการเอาข้อมูลที่เคยมีมาก่อนที่ production data load จะทำมาเขียนทับลงทันที

2.4.3 ผู้ใช้และผู้จัดการ (User and managers)

OLTP นั้น ผู้ใช้คือผู้ที่ทำงานอยู่ในองค์กรนั้น (อาจจะเป็นเจ้าหน้าที่ที่ทำหน้าที่รับออร์เดอร์ (order) เจ้าหน้าที่ที่ทำหน้าที่ตรวจสอบและดูแลเงิน เจ้าหน้าที่ที่ทำหน้าที่คอยบริการลูกค้าใหม่ เจ้าหน้าที่ที่ทำหน้าที่คีย์ข้อมูลเข้าไป) ซึ่งส่วนใหญ่ผู้ใช้เหล่านี้ ในช่วงเวลาหนึ่งจะทำงานกับ 1 แอคเคาท์ (account) เท่านั้น ผู้ใช้ OLTP มักจะทำงานที่มีลักษณะเป็นงานซ้ำๆ เดิม รายงานส่วนใหญ่ที่ได้จากการทำงานบนระบบ OLTP นี้มักจะมีลักษณะเป็นลิสต์ (list) ของทั้งตาราง (table) เลย

ดาต้าแวร์เฮาส์นั้น ผู้ใช้คือผู้ที่ทำหน้าที่คอยดูแลการทำงานของพนักงานในองค์กรเท่านั้น (คอยนับจำนวนออร์เดอร์ใหม่ๆ, หาเหตุผลว่าทำไมลูกค้าถึงบ่น, คอยตรวจดูว่ามีข้อมูลอะไรใหม่ๆ เข้ามาบ้าง, คอยตรวจสอบและแก้ไขข้อผิดพลาดของข้อมูล) ผู้ใช้ของดาต้าแวร์เฮาส์จะไม่ทำงานโดยใช้ทีละ 1 แอคเคาท์ แต่จะพิจารณาจากแอคเคาท์ทั้งหมด (อาจจะมีจำนวนเป็นร้อยเป็นพันก็ได้) แล้วหาคำตอบที่ต้องการออกมา (Answer set ขนาดเล็กๆ) และคำถามที่ให้กับดาต้าแวร์เฮาส์ก็สามารถเปลี่ยนแปลงได้เรื่อยๆ (ไม่ใช่คำถามที่เคยถามก็ได้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 The Time Dimension

OLTP จะทำงานอย่างรวดเร็ว และทำทรานแซคชั่นอย่างสม่ำเสมอ (การวัดเวลาใช้หน่วยเป็นนาทีและวินาที) สถานะของข้อมูลต่างๆ มีการเปลี่ยนแปลงอยู่ตลอดเวลา และความสัมพันธ์ระหว่าง entity ต่างๆ ก็เปลี่ยนแปลงไปด้วย

OLTP database จะขาดการสนับสนุน (การอ้างอิง) จากข้อมูลในอดีต เพราะในดาต้าแวร์เฮาส์มักจะมีคำถามถึงการวิเคราะห์ข้อมูลในอดีต แม้ว่าเราสามารถเก็บข้อมูลอดีตไว้ใน OLTP ได้ แต่ก็เป็นภาระหนักของระบบในการทำให้มองเห็นภาพในอดีต และจะทำให้ยากในการทำทรานแซคชั่น (ข้อมูลมีมากขึ้น ก็ต้องใช้เวลาในการทำมากขึ้นด้วย)

"Data Warehouse is a time series"

ดาต้าแวร์เฮาส์จะต้องไม่ถูกเปลี่ยนแปลงตลอดวัน ดังนั้น Twinkling ก็จะไม่มีทางเกิดขึ้น และมีการระมัดระวังในการเก็บข้อมูลลงไปใน ดาต้าแวร์เฮาส์แต่ละครั้งด้วย

2.4.5 Data Model ความแตกต่างที่สำคัญที่สุดของ OLTP และดาต้าแวร์เฮาส์คือการสร้าง data หรือ data model นั่นเอง

OLTP จะใช้ entity relation modeling คือการกำจัดความซ้ำซ้อนให้หมดไป เพื่อการทำทรานแซคชั่นจะได้สามารถทำได้ง่ายและรวดเร็วยิ่งขึ้น (สามารถทำทรานแซคชั่นได้โดยเข้าถึง ข้อมูลเพียงตัวเดียวเท่านั้น) การทำ entity relation modeling นั้น จะใช้วิธีแยกข้อมูลออกเป็นตารางเล็กๆ และแต่ละตารางสามารถเชื่อมต่อไปยังตารางอื่นได้

ข้อสังเกตเกี่ยวกับ OLTP ซึ่งใช้ entity relation diagram

- ไดอะแกรมนี้ symmetric หมายความว่า ทุกๆ ตารางเหมือนกันหมด ไดอะแกรม (diagram) นี้ไม่สามารถบอกได้ว่าตารางไหนสำคัญกว่าหรือใหญ่กว่า และไม่สามารถบอกได้ว่าตารางใดบรรจุ ตัววัดที่เป็นตัวเลข (numerical measurement) ทางธุรกิจ ซึ่ง ไดอะแกรมรูปแบบนี้ ผู้ใช้จะทำความเข้าใจและจดจำได้ยาก
- ถ้ามี 2 ตารางในไดอะแกรมต้องการรวม (join) กัน ก็มีหลายวิธีที่จะทำได้ และไม่ว่าจะวิธีไหนก็จะได้คำตอบเดียวกัน ซึ่งเป็นเรื่องยากในการเลือกว่าจะใช้เส้นทาง (path) ใดในการเชื่อม (link) กันระหว่างตารางที่ต่างกัน เพราะถึงแม้ท้ายสุดจะได้คำตอบเดียวกัน แต่ในระหว่างการทำ inner join นั้นจะมีการใช้ data element ไม่เหมือนกัน

สำหรับดาต้าแวร์เฮาส์จะใช้ไดเมนชันนอลโมเดล (Dimensional model) หรือ star join schema ซึ่งเป็นชื่อที่นักออกแบบฐานข้อมูล (Database) ใช้กันมานานแล้ว เนื่องจากไดอะแกรมมีรูปร่างคล้ายดาว ซึ่งมี ตารางใหญ่ 1 ตาราง อยู่ตรงกลางและมีตารางเล็กๆ ที่มีความสัมพันธ์กับตารางหลักนั้นอยู่รอบๆ ซึ่งตารางหลักนี้เป็นตารางเดียวที่ใช้ multiple join เพื่อเชื่อมต่อกับตารางอื่นๆ แต่ตารางอื่นๆ ที่อยู่รอบๆ นั้นจะมีเพียงแค่ single join เพื่อเชื่อมเข้ากับตารางหลักเท่านั้น

ตารางหลักเรียกว่า Fact table

ตารางอื่นๆ เรียกว่า Dimension table

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.6 The standard template query

คาด้าแวร์เฮาส์จะใช้ SQL เป็น template มาตรฐานสำหรับการทำ query ทั้งหมดในคาด้าแวร์เฮาส์ ซึ่งเกี่ยวกับ fact table

- Select list : จะเลือกคอลัมน์ (column) ที่ต้องการให้ปรากฏใน answer set ของผู้ใช้ไว้ ซึ่งปกติแล้วใน select list เราจะจัด row header ไว้เป็น item แรกใน select list
- From clause : from clause ใน SQL จะได้มาจากชื่อของตารางที่จะต้องใช้ใน query
- Where clause : การทำ เงื่อนไขของการรวม (join constrain) ซึ่งในคำสั่งนั้นจะเขียนถึงความสัมพันธ์ระหว่าง fact table และ dimension table โดย primary key ใน dimension table จะมีสถานะเป็น foreign key ใน fact table และหลังจากที่ทำการ join จะพบว่ามีการทำ application constrain สำหรับ dimension การสร้าง application constrain ทำได้โดยพิจารณาแต่ละ dimension ดูว่ามีค่าอะไรบ้างที่อยู่ในขอบเขตที่ต้องการ
- Group by clause : บอก SQL ให้สรุปข้อมูลตาม row header ซึ่ง row header นี้มีลิสต์อยู่ใน select list นั่นเอง
- Order by clause : ทำการตัดสินใจว่าจะทำการเรียงลำดับค่าใน answer set ให้ผู้ใช้เห็นอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operational Database Vs Data warehouse		
	Operational Database	Data warehouse
การใช้งาน	ข้อมูล	วิเคราะห์
ลักษณะการจัดการข้อมูล	แอพพลิเคชั่นกับแอพพลิเคชั่น (Application Oriented)	ตามหัวข้อเรื่องที่ต้องการ (Subject Oriented)
โครงสร้างข้อมูล	ซับซ้อน แล้วแต่เครื่องและการคำนวณ แต่เป็นรูปแบบชัดเจน ประมวลผลซ้ำเรื่องเดิม	มีโครงสร้างไม่แน่นอน ประมวลผลแบบวิเคราะห์ แต่ง่าย เหมาะกับองค์กร
เนื้อหาและช่วงเวลา	ปัจจุบัน	อดีตและปัจจุบัน
การอัพเดท	เป็นเรื่องๆ ไป มีจำนวนน้อยและทำเป็นประจำ	แล้วแต่สถานการณ์และความต้องการ ไม่มีการอัพเดทโดยตรง
การเคลื่อนไหวของข้อมูล	ตลอดเวลา	คงที่จนกว่าจะปรับปรุงใหม่
เวลาในการทำงาน	เสี้ยววินาทีถึง 2-3 วินาที	ไม่แน่นอน ตั้งแต่หลายวินาทีจนถึงนาที
ความแน่นอนในการใช้ข้อมูล	แน่นอน	ไม่แน่นอน
แหล่งข้อมูล	ภายในองค์กร	ทั้งภายในและภายนอกองค์กร
ขนาดของข้อมูล	กิกะไบต์	กิกะไบต์ถึงเทราไบต์

Subject Oriented : ข้อมูลสารสนเทศสำหรับการทำธุรกิจจะถูกจัดวางตามหัวข้อที่องค์กรนั้นสนใจ ยกตัวอย่าง หัวข้อของอุตสาหกรรมโรงงานอาจจะเป็นรายชื่อ ลูกค้า, สินค้า, บัญชีการเงิน และตัวแทนจำหน่าย ส่วนมหาวิทยาลัยอาจจะเป็นรายชื่อนักเรียน, วิชาที่สอน และผู้สอน สำหรับโรงพยาบาลก็มีอย่างรายชื่อคนไข้, เจ้าหน้าที่จัดยา, การจัดยา ฯลฯ ที่สำคัญข้อมูลจากดาต้าแวร์เฮาส์คือข้อมูลที่มาจากแหล่งที่ยืนยันว่าถูกต้องแล้ว

Operation Oriented : อาทิ แอพพลิเคชั่นด้านการสั่งของอาจจะมีข้อมูลเกี่ยวกับลูกค้า สินค้า และบัญชี ฐานข้อมูลก็จะรวมเอาข้อมูลมาใส่ในช่องว่างที่แบบฟอร์มเว้นว่างเอาไว้

ข้อมูลในอดีต : เป็นสิ่งที่ระบบฐานข้อมูลทั่วไปใช้น้อยมากเพราะมีจุดประสงค์เพื่อจับความเป็นไปในปัจจุบันมากกว่า แต่ทางธุรกิจจะใช้ข้อมูลเหล่านี้เพื่อหาความเป็นไปและแนวโน้มในอนาคต

ข้อมูลไม่เคลื่อนไหว : เนื่องจากการวิเคราะห์และคำนวณข้อมูลทางธุรกิจจะทำได้ก็ต่อเมื่อข้อมูลมีปริมาณเคงที่เท่านั้น แต่ข้อมูลจากฐานข้อมูลจะมีการเคลื่อนไหวตลอดเวลา (Dynamic) ซึ่งเป็นเหตุผลหนึ่งที่ทำให้ข้อมูลในดาต้าแวร์เฮาส์ไม่มีการอัพเดทโดยตรง แต่จะใช้ข้อมูลที่ประมวลผลใหม่ซ้อนทับขึ้นไปเรื่อยๆ

ตารางที่ 1.1 สรุปความแตกต่างของ Data Warehouse กับ Operational Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความแตกต่างของการจัดการฐานข้อมูล Multidimensional Database และ Relational Database

1. การตอบคำถามของการจัดการฐานข้อมูลแบบ Multidimensional Database จะสามารถทำได้ง่ายกว่า การจัดการฐานข้อมูลแบบ Relational Database เช่น ผลิตภัณฑ์ใดที่ขายดี, ที่สาขาใดที่มียอดขายต่ำลง เป็นต้น

2. การจัดการฐานข้อมูลแบบ Multidimensional Database ใช้ดูแนวโน้มของแอททริบิวท์ (attribute) ว่าค่าของแอททริบิวท์นั้นมีทิศทางการเปลี่ยนแปลงในช่วงเวลาที่เราสนใจเป็นอย่างไร เพราะ Dimensional Database สามารถสรุปออกมาให้อยู่ในรูปแบบที่เข้าใจได้ง่ายและรวดเร็วกว่า เช่น ในการค้นหามีข้อมูลมากมายเกินกว่าการแสดงผล 1 หน้าจอ ผู้ใช้ที่ใช้ Relational Database จะรู้สึกไม่สะดวกในการดูแนวโน้มของข้อมูลเท่ากับการใช้ Multidimensional Database (เพราะต้องทำการเลื่อนหน้าจอ)

3. การจัดการฐานข้อมูลแบบ Relational Database แสดงข้อมูลแบบเล็กที่สุด ไม่สามารถแยกย่อยได้อีก (atomize data) และเป็นข้อมูล ณ เวลาใดเวลาหนึ่ง (record at a time) แต่การจัดการฐานข้อมูล Multidimensional จะแสดงข้อมูลโดยรวมซึ่งเน้นผลสรุปของข้อมูลและมีประโยชน์ในการวิเคราะห์ข้อมูลนั้นๆ

ทำไมฐานข้อมูลประจำวันสำหรับ OLTP จึงทำการนอร์มอลไลซ์

การนอร์มอลไลซ์ (Normalization) เป็นขั้นตอนการออกแบบฐานข้อมูล โดยใช้เทคนิค Decomposition ซึ่งจะแตกแอททริบิวท์ของ unsatisfactory relation schema ไปเป็น relation schema ที่เล็กกว่าซึ่งมีคุณสมบัติตามที่ต้องการ ตาม normal form ขั้นต่างๆ

กระบวนการของการนอร์มอลไลซ์ซึ่งผ่านการ decomposition จะต้องมีคุณสมบัติ 2 ประการดังนี้ คือ

1. Lossless join or nonadditive join property เป็นคุณสมบัติที่รับประกันว่าจะไม่เกิด spurious tuple problem ขึ้น

2. Dependency preservation property เมื่อมีการ update ฐานข้อมูล ระบบควรจะสามารถตรวจสอบได้ว่าการ update ไม่ได้สร้าง illegal relation ขึ้น

Normal form ตั้งแต่ขั้นที่หนึ่งถึงสาม และ Boyce-codd อ้างอิงอยู่บน data dependency แบบ functional dependencies ระหว่าง attribute ของ relation ดังนี้

1NF : A relation R is in first normal form (1NF) if and only if all underlying domains contain atomic values only.

Relation R จะอยู่ใน normal form ที่หนึ่งได้ก็ต่อเมื่อแอททริบิวท์ (attribute) ทุกตัวใน domain มีค่าเพียงค่าเดียว (atomic value) เท่านั้น

2NF : A relation R is in second normal form (2NF) if and only if it is in 1 NF and every nonkey attributes is fully dependent on the primary key.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Relation R จะอยู่ใน normal form ที่ 2 ได้ก็ต่อเมื่อ relation R อยู่ใน normal form ที่หนึ่งแล้ว และทุกๆ nonkey attribute fully depend อยู่บน primary key นั่นคือ nonkey attribute จะขึ้นกับ (depend on) primary key โดยไม่ขึ้นกับสับเซ็ตของ primary key

3NF : A relation R is in 3NF if and only if it is in 2NF and every nonkey attribute is nontransitively dependent on the primary key

Relation R จะอยู่ใน normal form ที่สามได้ก็ต่อเมื่อ relation R อยู่ใน normal form ที่สองแล้ว และทุกๆ nonkey attribute ไม่ขึ้น (depend) กันเอง

Boyce-Codd Normal Form (BCNF) : A relation R is in Boyce-Codd Normal Form (BCNF) if and only if every determinant is a candidate key

Relation R จะอยู่ใน Boyce-Codd Normal Form (BCNF) ก็ต่อเมื่อ determinant ทุกๆ ตัว เป็น candidate key

จุดประสงค์ของการทำนอร์มอลไลซ์คือ เพื่อสร้างกลุ่มของ Relation schema ซึ่งเก็บข้อมูลโดยปราศจากความซ้ำซ้อน (redundancy) ที่ไม่จำเป็น เพราะฐานข้อมูลที่ดีควรจะมีความซ้ำซ้อนน้อยที่สุด (ซึ่งจะทำให้เรา retrieve ข้อมูลได้ง่ายด้วย) เนื่องจากความซ้ำซ้อนมีผลเสียคือเปลืองพื้นที่, เปลืองค่าใช้จ่ายในการป้อนข้อมูล (data entry cost), ทำให้การอัพเดท (update) ซับซ้อน เกิดปัญหา update anomalies

Update Anomalies แบ่งเป็น 3 ประเภท คือ

- Insert Anomalies
  - insert fact ซึ่งขัดแย้งกับ fact เดิม
  - insert fact บาง fact ไม่ได้
- Deletion Anomalies
  - การลบ fact หนึ่งออก อาจทำให้ fact อื่นถูกลบไปด้วย
- Modification Anomalies
  - เนื่องจาก fact ซ้ำซ้อน เมื่อเปลี่ยนแปลงค่าในแอททริบิวท์ใดแอททริบิวท์หนึ่ง เราจะต้อง multiple update ให้ครบ ซึ่งอาจทำให้ข้อมูล inconsistent กันได้

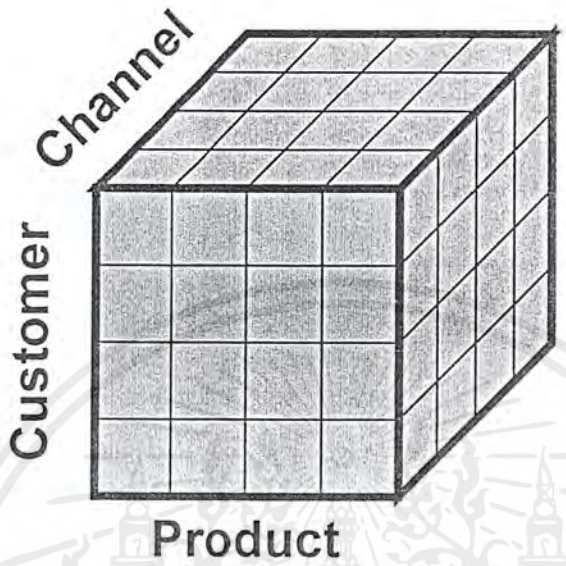
ทำไมในดาต้าแวร์เฮาส์จึงไม่ทำการนอร์มอลไลซ์เหมือนกับฐานข้อมูลประจำวัน

ในฐานข้อมูลประจำวัน ข้อมูลจำนวนมหาศาลมีการอัพเดทอยู่ตลอดเวลา ดังนั้นประเด็นสำคัญจึงอยู่ที่การอัพเดท การออกแบบฐานข้อมูลประจำวันจึงต้องทำให้มีความซ้ำซ้อนน้อยที่สุด วิธีการที่จะทำให้ได้ตามวัตถุประสงค์นี้ก็คือการนอร์มอลไลซ์

แต่สำหรับในดาต้าแวร์เฮาส์ ข้อมูลเป็นข้อมูลที่มีการกลั่นกรองมาแล้ว ใช้ในการวิเคราะห์ ตอบคำถามของผู้บริหาร ประเด็นสำคัญจึงไม่อยู่ที่การอัพเดท จึงทำให้ข้อมูลในดาต้าแวร์เฮาส์สามารถมีความซ้ำซ้อนได้ เพราะความซ้ำซ้อน มีข้อดีคือการตอบคำถามและการสร้างรายงาน สามารถทำได้เร็วเนื่องจากไม่ต้องมีการรวม (join) กันของหลายตาราง ดังนั้นในดาต้าแวร์เฮาส์จึงไม่มีความจำเป็นต้องทำการนอร์มอลไลซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
โครงสร้างและการออกแบบของดาต้าแวร์เฮาส์

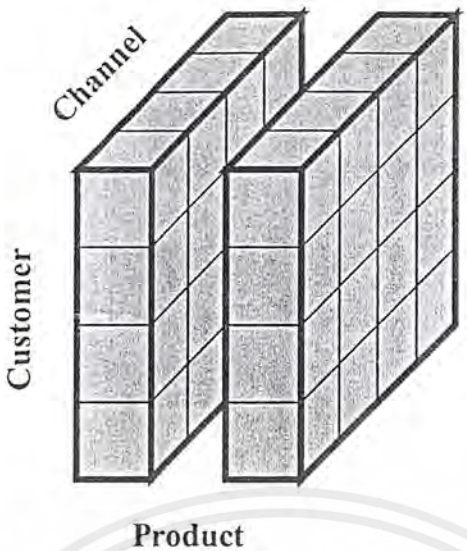


รูปที่ 3.1 แสดงภาพ Dimensional model ของธุรกิจอย่างหนึ่ง

3.1 Dimensional Modeling

Dimensional Modeling เป็นชื่อเรียกของเทคนิคในการทำให้ฐานข้อมูลง่ายต่อการทำความเข้าใจ โดยมองภาพของฐานข้อมูลเป็นลูกบาศก์ที่มี 3,4,5 มิติ หรือมากกว่านั้น ทำให้สามารถจินตนาการ การ slice และ dice ลูกบาศก์ที่มีลักษณะเหมือนลูกเต๋านี้ได้ กล่าวคือ สามารถตัดข้อมูลมาวิเคราะห์ดูในช่วงใดก็ได้ (slice) และหมุนข้อมูลดูได้จากทุกๆ ด้านของลูกเต๋า (dice)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงภาพการ dice เพื่อเลือกเฉพาะส่วนที่ต้องการ

Dimensional Model ทำให้เราสามารถมองภาพของข้อมูลได้โดยง่าย ดังตัวอย่างที่เห็นในรูปที่ 3.1

ตัวอย่างเช่น เรามี สินค้า (Product) ขายให้กับลูกค้า (customer) หลายๆ เจ้าในช่องทางการจัดจำหน่าย (Channel) ที่แตกต่างกัน เราสามารถสร้าง Dimensional modeling ได้ดังรูปที่ 3.1

เมื่อเรามองข้อมูลในลักษณะของลูกเต๋าแบบนี้ จะพบว่ามีชื่อ (Label) บนแต่ละด้านของลูกบาศก์ แต่ละจุดภายในลูกบาศก์เกิดจากการตัดกันของ coordinate ซึ่งมีชื่ออยู่ที่ขอบของลูกบาศก์ เช่น

ชื่อที่ขอบของลูกบาศก์ Product, Customer, Channel ทำให้เราสามารถเข้าใจได้เลยว่า จุดต่างๆ ภายในลูกบาศก์นั้น คือ ผลลัพธ์ทางด้านธุรกิจที่พิจารณาจากทั้ง 3 เรื่อง (Product, Customer, Channel) พร้อมๆ กัน

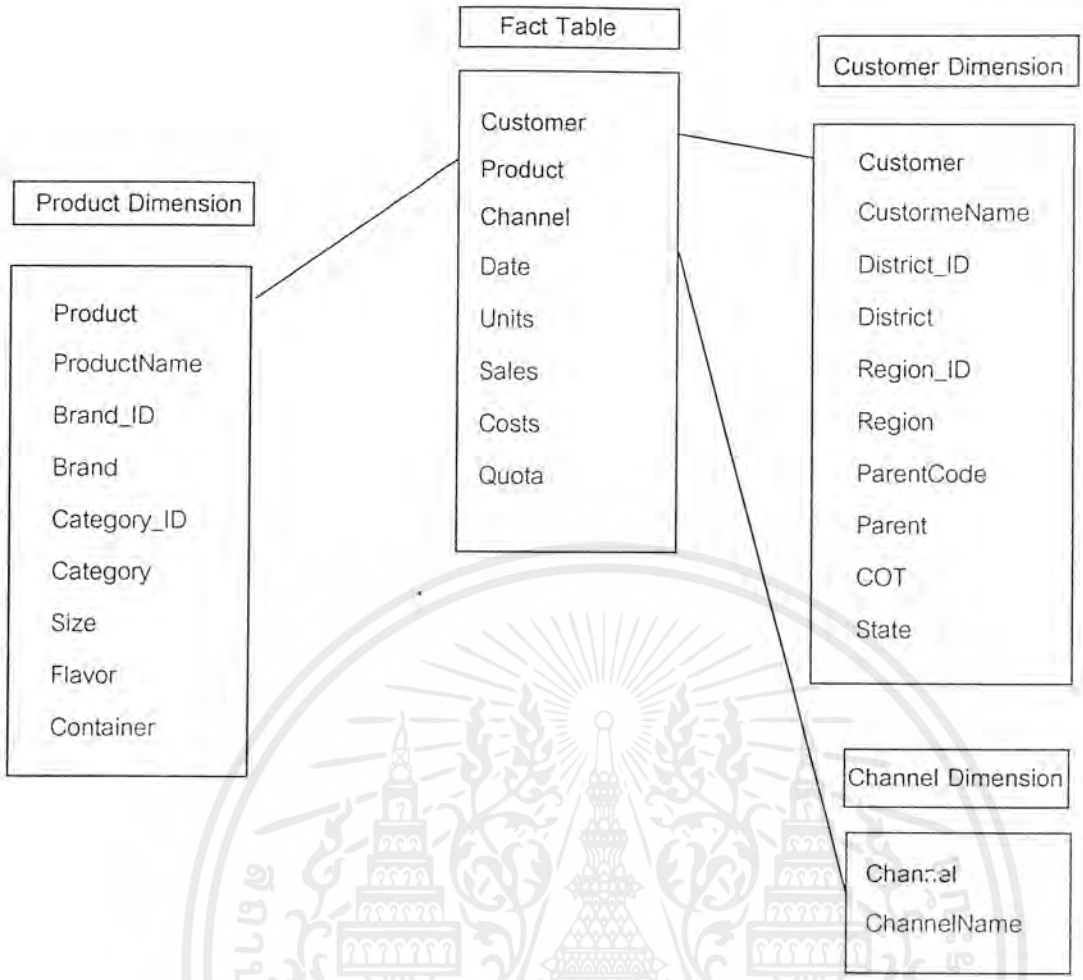
3.2 Star Join Schema

Star join schema เป็นอีกชื่อหนึ่งของ dimensional model ซึ่งเป็นชื่อที่ใช้กันมานานแล้ว เนื่องจาก diagram มีรูปร่างคล้ายดาว ซึ่งมีตาราง (table) ใหญ่ 1 ตาราง อยู่ตรงกลาง และมีตารางเล็กๆ ที่มีความสัมพันธ์กับ ตารางหลักอยู่รอบๆ ซึ่งตารางหลักนี้เป็นตารางเดียวที่ใช้ multiple join เพื่อเชื่อมต่อกับตารางอื่นๆ แต่ตารางอื่นๆ ที่อยู่รอบๆ นั้นจะมีเพียงแค่ single join เพื่อเชื่อมเข้ากับ ตารางหลัก เท่านั้น

ตารางหลักเรียกว่า Fact Table

ตารางอื่นๆ จะเรียกว่า Dimensional Table

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงภาพของ Star join schema

รูปที่ 3.3 เป็นโมเดล (Model) ของธุรกิจง่าย ที่ขายสินค้าให้แก่ลูกค้าต่างๆ ด้วยช่องทางการจัดจำหน่าย (channel) ที่แตกต่างกัน ซึ่งตัวเลขที่ใช้ในการวัดก็จะปรากฏอยู่ใน Fact table โดยที่เรคอร์ด (record) ที่อยู่ใน fact table เหล่านี้เรียกว่าเกรน (grain) ของ fact table จะได้มาจากการรวม (combination) ของไดเมนชั่น (dimension) ต่างๆ ก็คือ product, customer, channel dimension นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 Fact table

รูปที่ 3.4 แสดงตัวอย่างของ Fact Table

Fact table จะเป็นที่เก็บตัวเลขที่วัดได้ของธุรกิจหนึ่งๆ ตัวเลขที่วัดได้แต่ละตัวนี้มาจากการ intersection กันของทุกๆ dimension จากรูป ตัวเลขที่วัดได้คือ Units, Sales, Costs, Quota

ข้อมูลใน Fact table ควรจะมีลักษณะ คือ numeric, continuously valued, additive หมายความว่า เป็นตัวเลข, มีค่าต่อเนื่อง และมาจากทุกๆ dimension table

- Numeric : เหตุผลที่เป็น numeric ก็คือ ในความเป็นจริงคำถาม (query) ทุกคำถามที่ทำกับ fact table นั้นระบบจัดการฐานข้อมูล (DBMS) จะใช้เรคคอร์ด (record) เป็นพันๆ หมื่น หรือกระทั่งเป็นล้านๆ เรคอร์ดเพื่อสร้าง answer set ขึ้นมา จำนวนเรคอร์ดที่มากมายนี้จะถูก compressed เป็นจำนวนแถว (row) เพียงไม่กี่แถว ใน answer set ของผู้ใช้นั่นก็คือ การทำคำสั่ง select list หรือ built-in function : sum ของ SQL นั่นเอง วิธีเดียวที่จะ compress record เหล่านี้ให้กลายเป็น answer set ได้ก็คือ การบวกค่าของเรคอร์ดต่างๆ เข้าด้วยกัน ดังนั้นเรคอร์ดที่มีลักษณะที่เป็นตัวเลข และ additive จะทำให้สร้าง answer set ได้ง่าย

- Continuously valued : ค่าใน fact table ควรจะเป็นค่าที่มีความต่อเนื่องเพื่อที่จะสามารถไกด์ผู้ออกแบบฐานข้อมูล ให้สามารถแยกได้ว่าอะไรคือ fact อะไรคือแอททริบิวท์ (attribute) ของ dimension table สำหรับในกรณีตัวอย่างข้างต้น fact ของ จำนวนยอดขาย (sales) เป็นค่าที่ต่อเนื่อง เพราะในสถานการณ์จริง ค่าใดๆ ของจำนวนยอดขายนี้จะมีขอบเขตที่กว้างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Additive : จากตัวอย่างดังกล่าวข้างต้น ค่าของ fact sales, costs, units, quota เป็นค่า additive หมายความว่า เราสามารถเพิ่มค่าของยอดขายและตัววัดอื่นๆ ได้ในทุก combination ของ customer, product, channel ไม่ว่าจะ slice ฐานข้อมูลอย่างไร ค่าที่รวมจำนวนยอดขายทั้งหมด (sum total) จะสื่อความหมาย ค่าที่รวมจำนวนทั้งหมดจะสามารถนำไปใช้ได้

แต่อย่างไรก็ตาม จะมีค่า fact ที่ไม่เป็น additive ด้วย เป็น fact ที่เป็น semiadditive และ fact ที่เป็น nonadditive ซึ่ง semiadditive คือ grain ที่ไม่ได้มาจากทุก dimension table ซึ่งจะสามารถหาผลรวมตาม dimension บาง dimension เท่านั้น ส่วน nonadditive คือ grain ที่ไม่สามารถหาผลรวมตาม dimension ใดๆ ได้

3.4 Dimension Table

ใน Dimension table จะเก็บคำอธิบายของแต่ละ dimension ของธุรกิจเอาไว้ ซึ่งคำอธิบายเหล่านี้จะช่วยในการอธิบายถึงสมาชิกในทุกๆ ไดเมนชั่น (dimension) และใน dimension table จะประกอบด้วยหลายๆ แอททริบิวท์ (attribute or field) ซึ่ง แอททริบิวท์ ที่ดีจะต้องเป็นตัวอักษร และแต่ละ แอททริบิวท์ ต้องแยกออกจากกัน ซึ่ง แอททริบิวท์ เหล่านี้ถูกใช้เป็นแหล่งที่มาของข้อบังคับ (constrains) และ row headers ใน answer set ของผู้ใช้ ดังรูปที่ 3.5

รูปที่ 3.5 แสดงตัวอย่างของ Dimension Table

ในกรณีที่มี Numeric data fields อยู่ เราต้องทำการสำรวจว่า field นั้นเป็น fact หรือเป็น attribute

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

- Fact : ถ้าพิจารณาแล้วพบว่า numeric data field นี้มาจากการวัด (Measurement) การคำนวณในช่วงเวลาที่ต่อกันช่วงหนึ่ง ซึ่งอาจมีการเปลี่ยนแปลงค่าได้ถ้าเวลาเปลี่ยนไป ก็จะกล่าวได้ว่า numeric data field นั้นเป็น fact
- Attribute : ถ้าพิจารณาแล้วพบว่า numeric data field ที่มีนั้นมีค่าที่ไม่ต่อเนื่องกัน และมีลักษณะเป็นการบรรยายบางสิ่งบางอย่างซึ่งมีค่าคงที่ ก็จะกล่าวได้ว่า numeric data field นั้นเป็น dimension attribute

หน้าที่ของ Key ใน dimension table คือเป็นแหล่งที่มาของกฎข้อบังคับ (constrain) ในการทำ query หรือที่หน้าที่เป็น row header ใน answer set ของผู้ใช้

Brand	Sales	Quota
Courtyard	25468	42518
Savory summer	25414	12454
Citrus Plantation	12457	6541
Zapper	11524	14521

ตารางที่ 3.1 แสดงตัวอย่างการนำ Attribute ใน dimension table มาเป็น row header

3.5 Drilling up และ Drilling down

การ Drill down เป็นการเข้าไปในชั้นที่ต่ำลงของตารางไดเมนชัน หรืออาจกล่าวได้ว่า การ drill down หมายถึง การแสดงรายละเอียดในลำดับชั้นให้มากขึ้นเพื่อช่วยในการอธิบายในสิ่งที่สนใจอยู่ ซึ่งก็คือการเพิ่ม row header เข้าไปในรายงาน (report) นั่นเอง การเพิ่ม row header เข้าไปในรายงานที่มีอยู่แล้วจะทำให้เห็นส่วนประกอบภายใน row header นั้นๆ ซึ่งก็คือ row header อันใหม่ ตัวอย่างเช่นตารางที่ 3.2 เป็นรายงานที่ยังไม่ได้ทำการ drill down และตารางที่ 3.3 เป็นการ drill down ฟิลด์ ProductName ให้เป็น Flovor

CustomerName	ProductName	Units
7Eleven-Nashu NH	Courtyard	32
7Eleven-Nashu NH	Citrus Plantation	50
7Eleven-Nashu NH	Savory Summer	23
7Eleven-Nashu NH	Regatta	12

ตารางที่ 3.2 แสดงรายงานที่มี 3 row header

37079

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CustomerName	ProductName	Flavor	Units
7Eleven-Nashu NH	Courtyard	Orange	12
7Eleven-Nashu NH	Courtyard	Raspberry	20
7Eleven-Nashu NH	Citrus Plantation	Lemon	25
7Eleven-Nashu NH	Citrus Plantation	Grape	25
7Eleven-Nashu NH	Savory Summer	Orange	12
7Eleven-Nashu NH	Savory Summer	Lemon	11
7Eleven-Nashu NH	Regatta	Raspberry	6
7Eleven-Nashu NH	Regatta	Raspberry/Grape	6

ตารางที่ 3.3 แสดงการ Drill down ฟิลด์ ProductName ให้เป็น Flavor

การ Drill up จะตรงกันข้ามกับการ Drill down นั่นคือเป็นการเอา row header ออกจากรายงาน เช่น การ drill up ตารางที่ 3.3 สำหรับฟิลด์ flavor ก็จะได้ผลดังตารางที่ 3.2 นั่นเอง

3.6 Aggregation

เนื่องจากข้อมูลพื้นฐานของ ไดเมนชั่นนอลดาต้าแวร์เฮาส์ (Dimensional Data Warehouse) จะประกอบด้วยเรคอร์ด จำนวนมาก ถ้านักวิเคราะห์ต้องการทำ query โดยไม่มีการกำหนดข้อบังคับ ให้กับบางไดเมนชั่น เนื่องจากในดาต้าแวร์เฮาส์จะมีแต่เฉพาะข้อมูลระดับพื้นฐานเท่านั้น (Based-level data) จึงทำให้เวลา query จะต้องทำการ sum ข้อมูลภายในเรคอร์ดจำนวนมากมาย ซึ่งถ้าเป็นเช่นนี้จะทำให้การทำ query เกิดความสิ้นเปลืองสูงมาก ดังนั้นจึงได้มีการทำการรวบรวม (Aggregate) ข้อมูลเอาไว้ล่วงหน้าเพื่อเร่งให้การทำ query สามารถทำได้เร็วขึ้น มีสมรรถนะ (Performance) ที่ดียิ่งขึ้น

การรวบรวม (Aggregate) คือ fact table record ที่แสดงถึงข้อมูลสรุปที่ได้จาก based-level fact table

การรวบรวมมีหลายชนิด และแต่ละชนิดจะมี fact table เป็นของมันเอง ซึ่ง fact table เหล่านั้นจะถูกเรียกว่าเป็น derivative fact table เพราะข้อมูลเหล่านั้นได้รับ (derive) มาจาก based-level fact table

เราสามารถสร้าง Aggregate ได้มากมายตามที่เราต้องการ ซึ่งในความเป็นจริงเราจะสร้าง Aggregate ไว้เฉพาะส่วนที่เราต้องการเท่านั้น ไม่จำเป็นต้องสร้าง Aggregate ทุกๆ ฟิลด์ (fields) ในแต่ละไดเมนชั่น และไม่จำเป็นต้องทำการ Aggregate ทุกๆ ไดเมนชั่นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 Snowflake

ตารางไดเมนชั่นที่มีข้อมูลแบบเป็นลำดับขั้น (Hierarchy) อาจจะมีการแตกข้อมูลออกมาเป็นตารางย่อยๆ เพื่อให้ผู้ออกแบบสามารถเข้าใจได้ง่ายยิ่งขึ้น ซึ่งเรียกโครงสร้างนี้ว่าสโนว์เฟลก (Snowflake) ความสัมพันธ์ที่เป็นแบบ many - to – one แต่ละอันจะถูกแยกออกเป็นตารางย่อยๆ โดยพิจารณาจากข้อมูลของเรคอร์ด 1 เรคอร์ด ต่อไปนี้ โดยโครงสร้างของฐานข้อมูลดาต้าแวร์เฮาส์จะมีลักษณะดังรูปที่ 3.6

ProductName : Savory Summer 12oz Raspberry

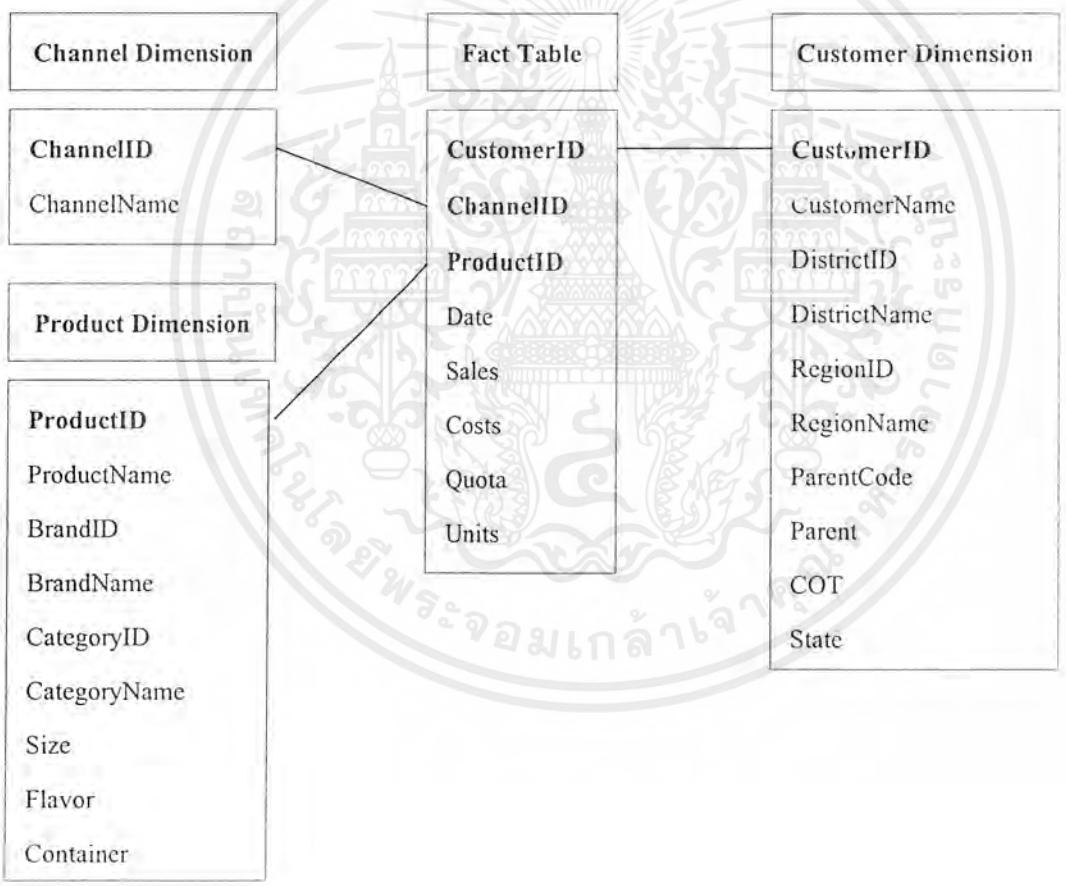
Size : 12oz

Flavor : Raspberry

Container : Can

BrandName : Savory Summer

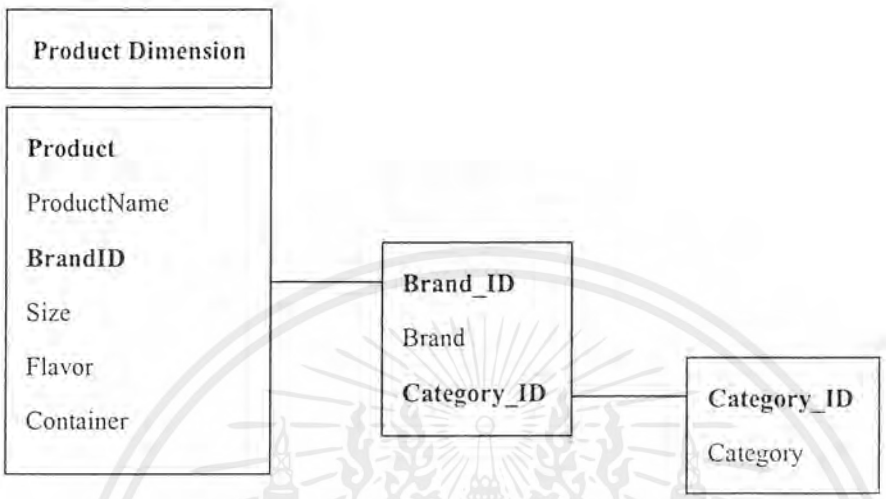
CategoryName : Fruit Juice



รูปที่ 3.6 แสดงโครงสร้างฐานข้อมูลดาต้าแวร์เฮาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

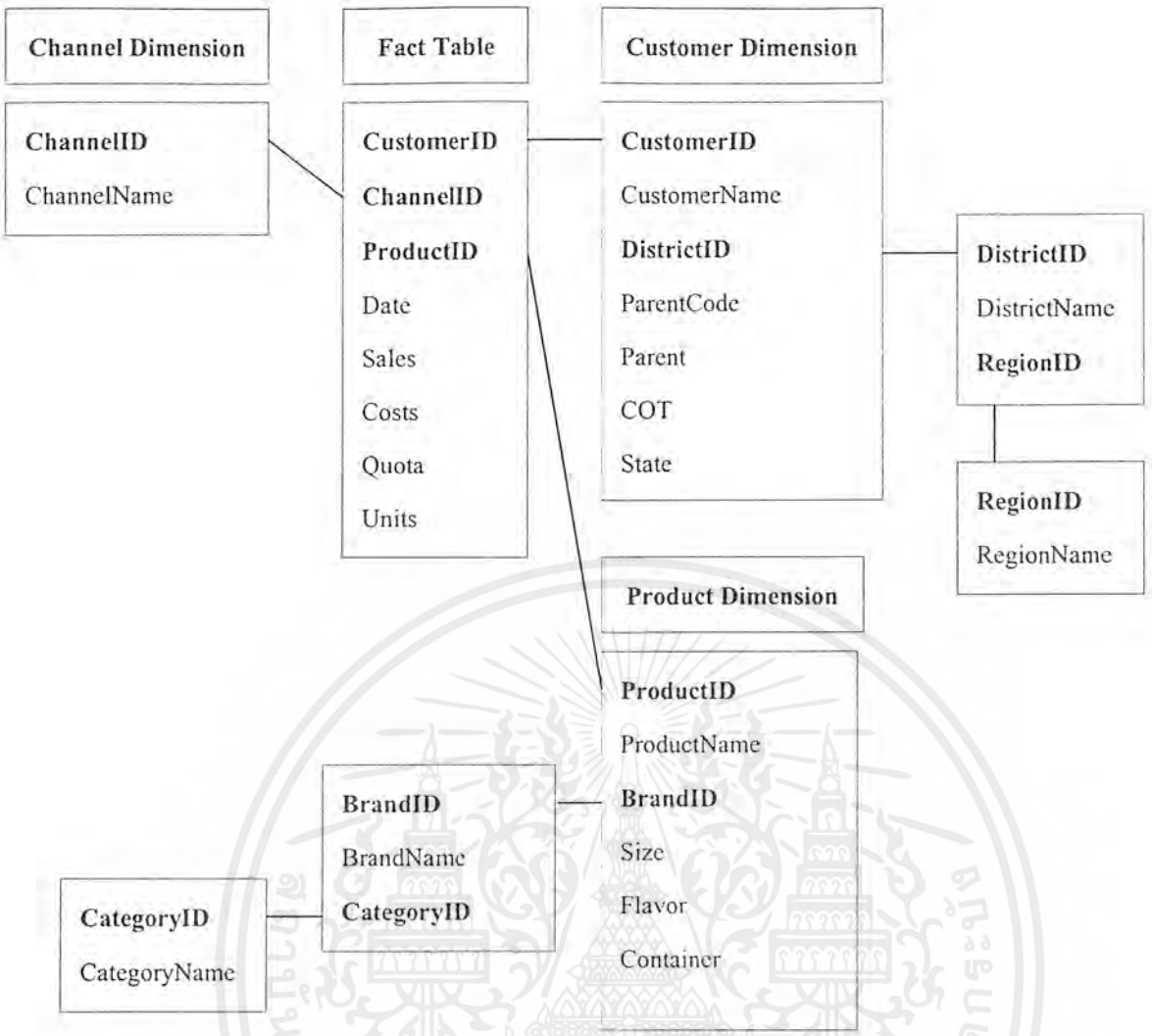
จากข้อมูลภายในเรคอร์ด 1 เรคอร์ด จะเห็นว่ามีฟิลด์ที่เป็น ProductName, BrandName และ CategoryName และจากข้อมูลภายในสามารถสรุปได้ว่า category 1 category จะมี brand ได้หลายๆ brand และ brand 1 brand ก็จะมีได้หลายๆ product จึงสามารถแตกตาราง Product dimension ออกเป็นโครงสร้างสโนว์เฟลก ดังรูปที่ 3.7 รูปนี้ไดเมนชั่น Product จะถูกแตกออกเป็นตารางแบรนด์ (Brand) และตาราง Category



รูปที่ 3.7 แสดงตารางไดเมนชั่นโปรดักแบบสโนว์เฟลก(Snowflake)

โดยจะนำ primary key ของตารางย่อย มาเป็น foreign key ของตารางที่ตัวเองไปเกาะอยู่ เช่น Brand_ID, Category_ID และสามารถโครงสร้างฐานข้อมูลแบบสโนว์เฟลกจะมีลักษณะดังรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงโครงสร้างฐานข้อมูลดาต้าแวร์เฮาส์ที่ออกแบบเป็นสโนว์เฟลก(Snowflake)

การทำโครงสร้างแบบสโนว์เฟลกอาจจะทำให้ประสิทธิภาพในการขับเคลื่อนลดต่ำลง เนื่องจากต้องมีการรวมกันระหว่างตารางย่อยกับตารางไดเมนชั่นอีกทีหนึ่ง รวมทั้งต้องสร้าง SQL ที่มากขึ้น แต่อย่างไรก็ตาม สโนว์เฟลกเหมาะสำหรับฐานข้อมูลที่มีขนาดใหญ่ มีจำนวนเรคอร์ดหลายแสนเรคคอร์ด เพราะมีส่วนช่วยลดความซ้ำซ้อนลงได้ รวมทั้งโครงสร้างแบบสโนว์เฟลกมีข้อดี เช่น เมื่อต้องการเรียกดูเฉพาะ ProductName, Flavor ก็ทำแค่เพียงดึงข้อมูลจากตารางไดเมนชั่น Product เพียงตารางเดียว และทำการ join กับตารางแฟ็กเท่านั้น ไม่ต้องมีการ join กันระหว่างตารางไดเมนชั่น Product กับตารางย่อยคือ ตาราง Brand และ Category และจะไม่ทำให้เกิดปัญหาการอัพเดท โดยไม่ต้องทำ multiple update

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

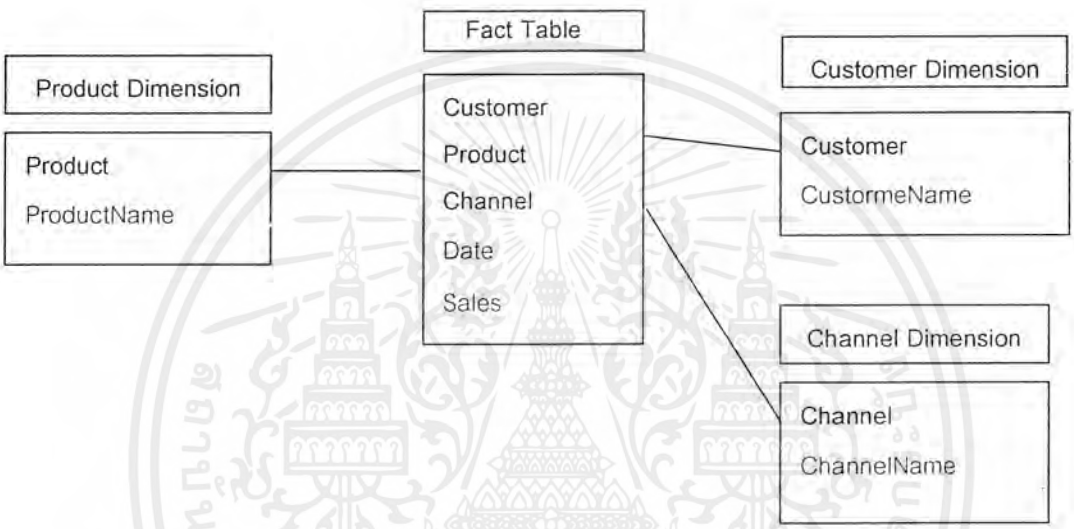
3.8 หลักในการทำ query บนตารางที่ออกแบบบนดาต้าแวร์เฮาส์

3.7.1 การเขียนคำสั่ง SQL

เมื่อมัลติไดเมนชันนอลดาต้าเบส (Multidimensional database) ถูก Query จะต้องมีการใช้ ตารางไดเมนชั่น (Dimension table) ในการ query เสมอ เพราะในตารางไดเมนชั่นจะเก็บชื่อและคำอธิบายที่ query ต้องการใช้เอาไว้ ซึ่งคำถามของ star schema จะประกอบด้วย Where clause 2 ส่วน คือ

- การรวม (join) กันระหว่าง fact table และ ตารางไดเมนชั่น
- เงื่อนไขที่จะมีสำหรับคอลัมน์ในตารางไดเมนชั่น

ในตัวอย่างจะเป็นการแสดง star query ที่ใช้ สำหรับตารางไดเมนชั่น product, customer, channel



รูปที่ 3.9 แสดง star schema เพื่อทำ star query

จาก star schema ด้านบน ถ้าต้องการให้ผลลัพธ์แสดงคอลัมน์เป็น ProductName, CustomerName, ChannelName และแสดงค่าของ Sales เป็นของ product ที่ชื่อว่า 'courtyard' ในช่วงระยะเวลา 3 เดือนที่แล้ว ที่ลูกค้าที่ชื่อ 7Eleven เป็นคนซื้อไป ในช่องทางตรง (Direct channel) จะสามารถเขียน SQL ได้ดังนี้

```
select   Product.ProductName,
         Customer.CustomerName,
         Channel.ChannelName,
         Fact.Sales
From     Product,  Customer, Channel, Fact
Where    Product.ProductName = 'Courtyard'
And      Customer.CustomerName = '7Eleven'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

And Channel.ChannelName = 'Direct'

And Fact.Date = 'Last 3 months'

And Fact.Product = Product.Product

And Fact.Customer = Customer.Customer

And Fact.Channel = Channel.Channel

Where clause ส่วนที่เป็นเงื่อนไขสำหรับคอลัมน์ในตารางไดเมนชั่น คือ

Where Product.ProductName = 'Courtyard'

And Customer.CustomerName = '7Eleven'

And Channel.ChannelName = 'Direct'

Where clause ส่วนที่เป็นการรวมกันระหว่างตารางแฟ็ก (Fact table) กับตารางไดเมนชั่น คือ

And Fact.Product = Product.Product

And Fact.Customer = Customer.Customer

And Fact.Channel = Channel.Channel

3.9 ขั้นตอนการออกแบบดาต้าแวร์เฮาส์

การออกแบบดาต้าแวร์เฮาส์มีอยู่ 4 ขั้นตอนคือ

3.9.1 เลือกกระบวนการทางธุรกิจ (Business process) ที่ต้องการสร้าง

กระบวนการทางธุรกิจ (Business process) : เป็นกระบวนการหลักที่ต้องการทำในองค์กร ซึ่งกระบวนการนั้นมีระบบเดิมสนับสนุนอยู่ โดยข้อมูลในระบบนั้นสามารถนำมารวบรวมเพื่อทำเป็นดาต้าแวร์เฮาส์ได้ เช่น ใบสั่งของ (Order), ใบส่งของ (Invoices), สินค้าที่ส่งไป (Shipments), รายการสินค้า (Inventory) ฯลฯ

3.9.2 เลือกเกรน (Grain) ของกระบวนการทางธุรกิจ

เกรน (Grain) : เป็นข้อมูลพื้นฐาน ข้อมูลที่เป็นอะตอมมิก (Atomic) ซึ่งถูกแสดงอยู่ใน fact table สำหรับกระบวนการนี้ เกรนที่มีอยู่ทั่วไป เช่น ข้อมูลการทำทรานแซคชั่นในแต่ละครั้ง (Individual transaction), ข้อมูลของการทำงานในแต่ละวันหรือสรุปในแต่ละวัน (Individual daily snapshots), ข้อมูลสรุปการทำงานในแต่ละเดือน (Individual monthly snapshots)

3.9.3 เลือกไดเมนชั่นที่จะถูกนำมาใช้กับแต่ละเรกอร์ดของ fact table

ไดเมนชั่นที่มีอยู่โดยทั่วไป เช่น time, product, customer,channel ซึ่งแต่ละไดเมนชั่นจะถูกอธิบายแยกกันในลักษณะของแอททริบิวท์ และจะถูกอธิบายเป็นตัวหนังสือ

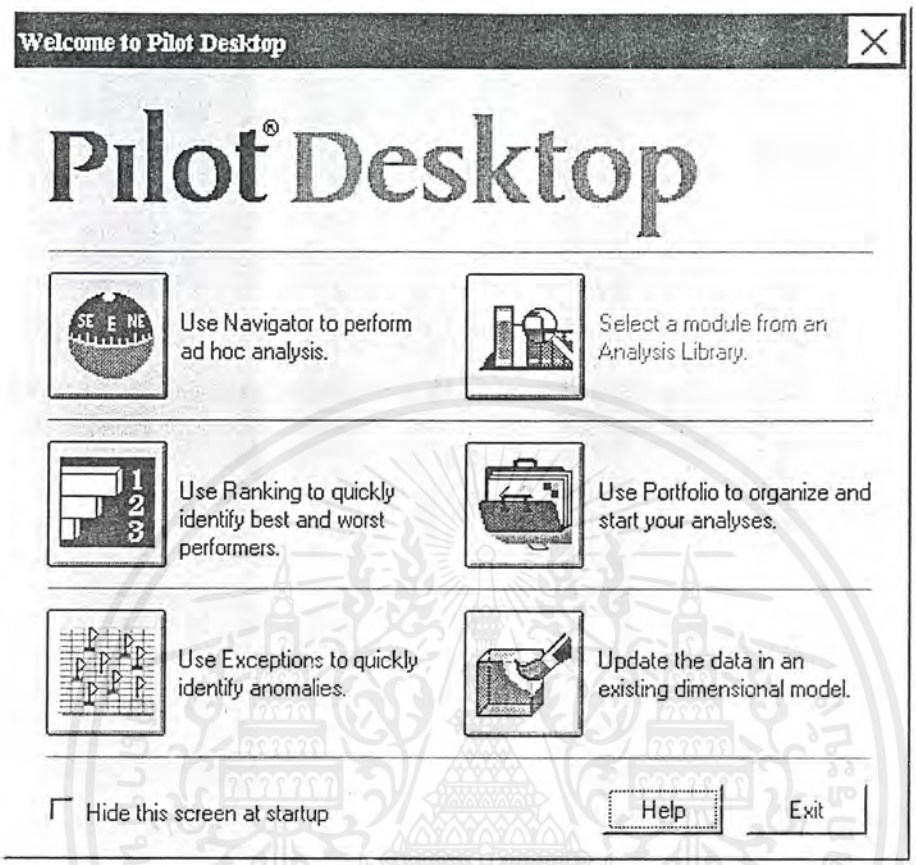
3.9.4 เลือก fact ที่มีการวัด การประมวลผล หรือการคำนวณไว้แล้ว (measured fact) ที่จะเก็บอยู่ในแต่ละเรกอร์ดของ fact table

ตัววัดที่มีอยู่โดยทั่วไป เช่น ปริมาณต่างๆ ซึ่งจะมีลักษณะเป็นตัวเลข ได้แก่ ยอดที่ขายได้, โควต้าของยอดขาย, จำนวนชิ้นที่ขายได้, ค่าใช้จ่ายทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

Pilot Decision Support Suite version 5.0



รูปที่ 4.1 แสดงหน้าจอ Welcome Screen ของ Pilot Desktop

4.1 Pilot Desktop

การใช้ Pilot Desktop จะทำให้สามารถ

- วิเคราะห์ข้อมูลและการเปลี่ยนแปลงของตลาด
- เข้าใจแนวโน้มของยอดขาย รูปแบบการซื้อขาย และ market performance
- มีแนวทางที่ยืดหยุ่นที่จะสร้าง report เฉพาะ

Desktop จะมีโมดูล (module) สำหรับการวิเคราะห์อยู่หลาย module ซึ่งจะช่วยให้สามารถพัฒนา decision support ได้ดียิ่งขึ้น โดยจะมีดังนี้

- Navigator Module
- Ranking Module
- Exception Classic Module
- Exception Plus Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ละ Module ก็จะหาคำตอบของคำถามที่แตกต่างกัน เพื่อให้สามารถวินิจฉัยสถานะของธุรกิจจากมุมมองที่แตกต่างกันออกไป ซึ่งในการวิเคราห์ก็จะสามารถหาแนวโน้ม และความสัมพันธ์ของข้อมูล จุดแข็ง จุดอ่อนทางของธุรกิจ และ trace performance โดยการใช้ exception จึงทำให้มีการตัดสินใจทางธุรกิจที่ฉลาด และได้รับโอกาสในทางการตลาดมากขึ้น

4.2 รายละเอียดของแต่ละโมดูล

4.2.1 Ranking Analysis

- Ranking Analysis จะตรวจสอบจุดแข็งและจุดด้อยของประสิทธิภาพทางธุรกิจในองค์กร ซึ่งจะดูได้โดยการจัดลำดับ เช่น เรียงจากมากไปน้อย หรือน้อยไปมาก ของตัววัด (measure) ที่ต้องการ
- Ranking Analysis จะทำให้สามารถจัดลำดับ (level) ใดๆ ของไดเมนชั่น (dimension) ก็ได้ หรือจะเปลี่ยนมุมมอง เปลี่ยนไทม์เฟรม (time frame) และสามารถระบุจำนวนของสิ่ง (item) ที่จะจัดลำดับได้ รวมทั้งยังสามารถแสดงข้อมูลให้เป็นลักษณะของแผนภูมิ (chart) เพื่อที่จะหาการเปลี่ยนแปลงของแต่ละข้อมูล ซึ่งวิธีการดังกล่าวจะทำให้สามารถมองลึกลงไปถึงสาเหตุของปัญหาทางธุรกิจได้

4.2.2 Exception Analysis

- Exception Analysis จะแสดงประสิทธิภาพ (Performance) ของธุรกิจที่แตกต่างไปจากที่คาดหวังไว้ โดยจะใช้ trigger เป็นตัวระบุว่าธุรกิจอันไหนที่อยู่สูงหรือต่ำกว่าเป้าหมาย (goal) ที่ต้องการ
- Trigger จะเป็นตัวที่ได้จากการคำนวณมาจากตัววัด 2 ตัว หรือ 2 ไดเมนชั่นที่แตกต่างกัน
- Exception Classic กับ Exception Plus Module จะมีลักษณะที่เหมือนกัน แตกต่างกันก็เพียงแต่ Exception Classic จะใช้ไดเมนชั่น 2 ไดเมนชั่น เป็น trigger ส่วน Exception Plus จะใช้ measure 2 ตัว เป็น trigger

4.2.3 Ad Hoc Analysis หรือ Navigator module

- Ad hoc analysis จะแสดงมุมมองของข้อมูลที่เป็นลักษณะยืดหยุ่น ซึ่งสามารถวิเคราะห์แนวทางใดก็ได้ที่ต้องการ โดยการสร้าง scenario, switch มุมมอง และแสดงแนวโน้มภายใต้ time frame ต่างๆ
- Ad hoc analysis จะให้ตัวเลือกที่เกี่ยวกับการตรวจสอบและการมองข้อมูลทางธุรกิจ ซึ่งสามารถตัดสินใจได้ว่าจะใช้ time frame หรือ measure อันไหนที่จะวิเคราะห์ข้อมูล โดยจะสามารถ drill down และ drill up ในระดับใดๆ ของ dimension ดูได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 Portfolio

- เราสามารถสร้าง Portfolio ใน analysis ที่จำเป็นต้องใช้บ่อยๆ โดยการ save analysis ที่สร้างไว้ที่ module ต่างๆ ได้

4.3 ประโยชน์ทั่วไปของซอฟท์แวร์ไพลอท

4.3.1 เป็นเครื่องมือค้นหาแบบมัลติไดเมนชันนอล (Multidimensional Query Tool) ที่อนุญาตให้มีการค้นหา หรือกำหนดมุมมองของข้อมูล (Query Data View) หลายๆ อันได้ สามารถเก็บหลายๆ ไดเมนชันได้อย่างมีประสิทธิภาพ

หลักการ : คือการจัดเตรียมและเก็บข้อมูลที่เกี่ยวข้องกัน ซึ่งสามารถมองและวิเคราะห์ข้อมูลเหล่านั้นได้หลายๆ ทาง โดยจัดเป็นมุมมองข้ามตาราง (เรียกว่า Cross-tabular view)

4.3.2 สะดวกในการวิเคราะห์แบบเปรียบเทียบ

4.3.3 ใช้ทำนายแนวโน้มของธุรกิจ

4.3.4 ไพลอทจะเก็บข้อมูลที่รวบรวมแล้ว (Consolidate) ไว้ในหน่วยความจำของเครื่องคอมพิวเตอร์ทำให้สามารถนำออกมาดูได้ทันที ไม่ต้องทำการคำนวณใหม่ทุกๆ ครั้ง

4.4 วิธีการใช้งานซอฟท์แวร์ไพลอทในส่วนของไพลอทเดสก์ทอป (Pilot Desktop)

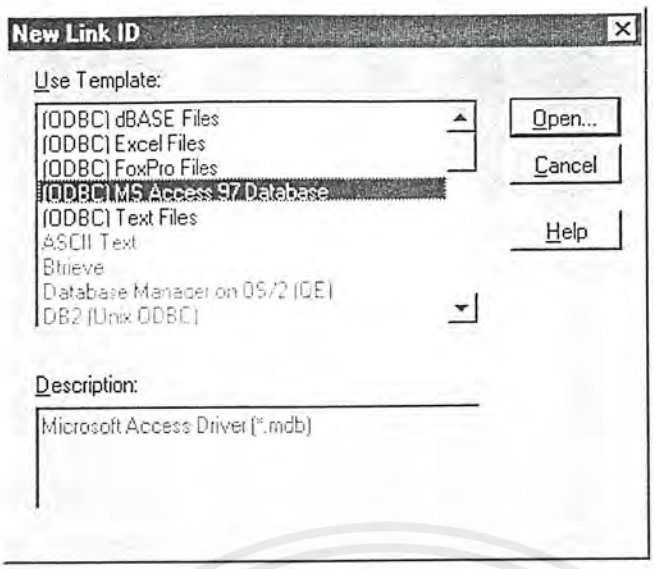
4.4.1 การสร้าง Model ใน Pilot

Model ที่กล่าวถึงนี้จะสร้างมาจากฐานข้อมูลที่มีอยู่แล้ว ซึ่งอาจจะได้จากการสร้างฐานข้อมูลใน Microsoft Access, Foxpro หรือ application อื่นๆ ก็ได้ โดยมีขั้นตอนดังต่อไปนี้

4.4.1.1 สร้าง Link เพื่อให้เชื่อมต่อกับ Pilot

1. เปิด Program Pilot Link Configurator ที่ปุ่ม Start
2. Click ที่ File>New จะเป็น dialog box ปรากฏขึ้นมาดังรูปที่ 4.2

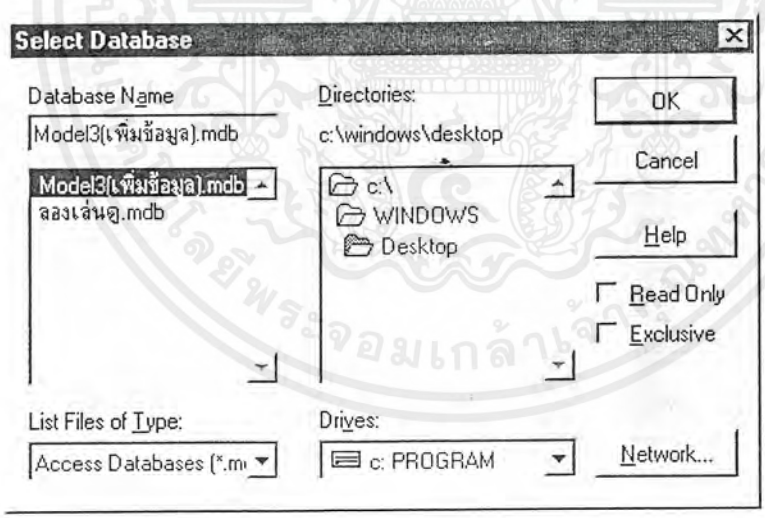
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงหน้าจอสำหรับฐานข้อมูลที่จะให้เชื่อมต่อกับไพลอท

3. ในที่นี้เราถ้าเรามีฐานข้อมูลที่สร้างโดย Microsoft Access ก็จะเลือก Template ดังรูปที่ 4.2

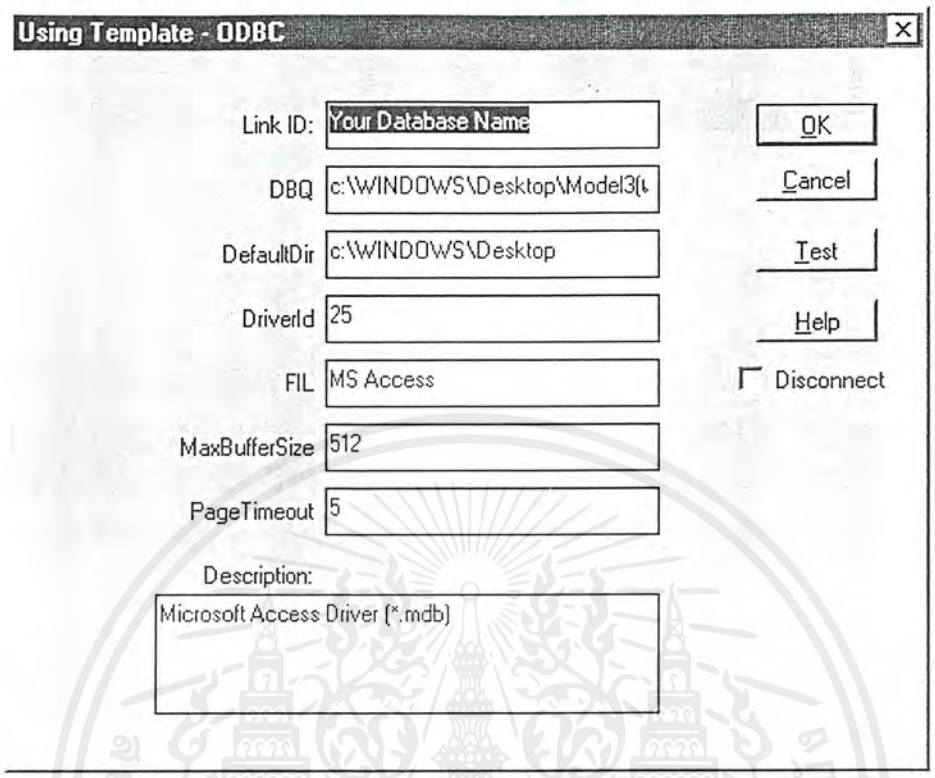
4. Click ที่ปุ่ม Open... ก็จะเข้าสู่ dialog box ดังรูปที่ 4.3



รูปที่ 4.3แสดงหน้าจอการเลือกฐานข้อมูลที่ได้สร้างเอาไว้

5. เลือกไฟล์ฐานข้อมูลที่ต้องการจะสร้างเป็น Model แล้ว click ที่ปุ่ม OK จะปรากฏ dialog box ดังรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



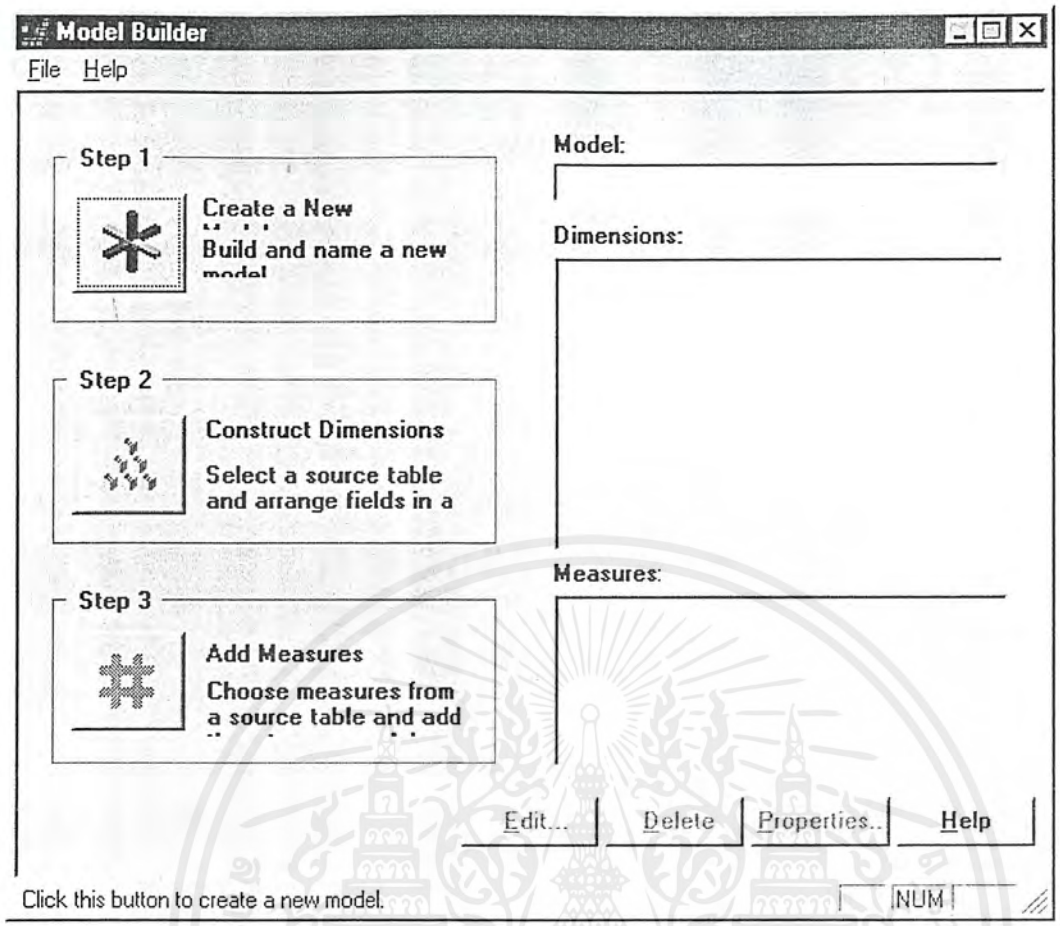
รูปที่ 4.4 แสดงหน้าจอในการกำหนดค่าการเชื่อมต่อกับฐานข้อมูลที่ได้สร้างเอาไว้

5. พิมพ์ชื่อ Link ID ที่ต้องการ แล้ว click ที่ปุ่ม OK
6. จากนี้ไฟล์ฐานข้อมูลที่เราได้สร้างขึ้นก็จะเชื่อมต่อกับ Pilot ได้แล้ว จากนั้นเราจะนำฐานข้อมูลนี้ไปสร้างเป็นโมเดลที่ต้องการต่อไป

4.4.1.2 การสร้าง Model โดยใช้ Program Pilot Model Builder

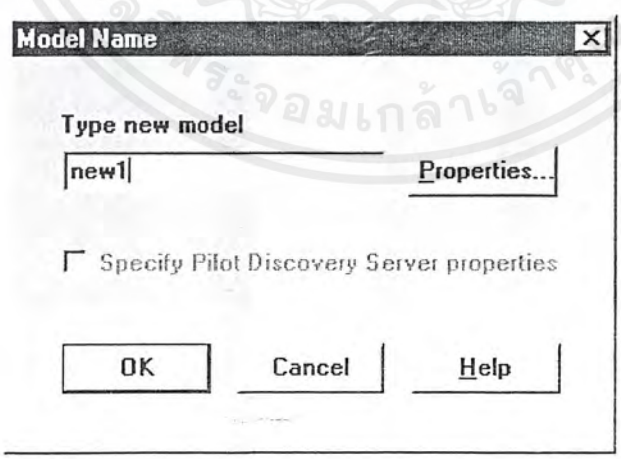
1. เปิด program Pilot Model Builder จากปุ่ม Start จะปรากฏหน้าจอดังรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงหน้าจอที่ใช้ในการสร้างโมเดล

2. คลิกที่ปุ่ม step 1 จะปรากฏ dialog box ดังรูปที่ 4.6

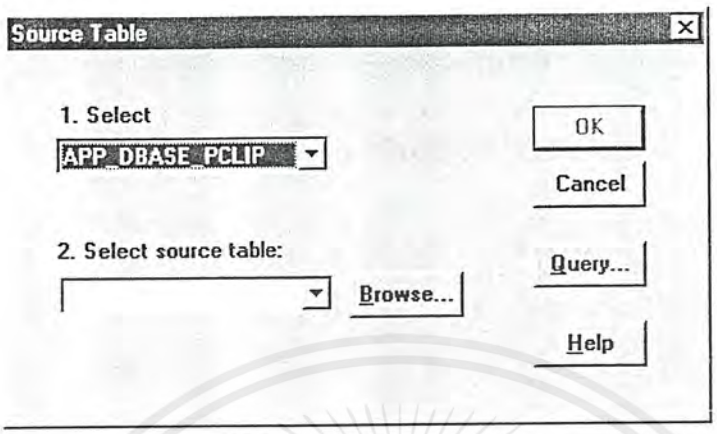


รูปที่ 4.6 แสดงไดอะลอกเพื่อใส่ชื่อโมเดลที่ต้องการสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จะมีช่อง Type new model ให้ใส่ชื่อโมเดลที่ต้องการลงไป แล้วคลิกปุ่ม OK

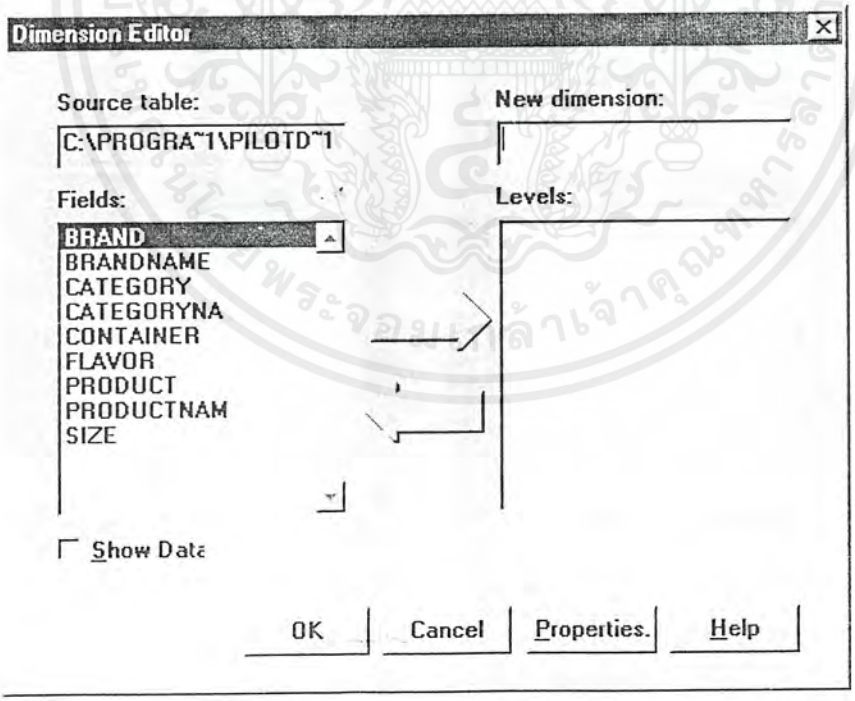
4. จากนั้นจะกลับมาสู่หน้าจอหลัก แล้วทำขั้นตอน 2 ต่อ คือจะให้สร้าง dimension จาก source table ที่เราต้องการ จะปรากฏ dialog box ดังรูปที่ 4.7



รูปที่ 4.7 แสดงหน้าจอสำหรับเลือก Source table

5. ที่ช่อง Select ให้เลือกชื่อ Link ID ของ file ฐานข้อมูลที่เราจะทำการสร้างโมเดล

6. ที่ช่อง Select source table ก็ให้เลือกตารางที่เราต้องการสร้างเป็น dimension แล้วคลิกปุ่ม OK จะปรากฏ dialog box ดังรูปที่ 4.8

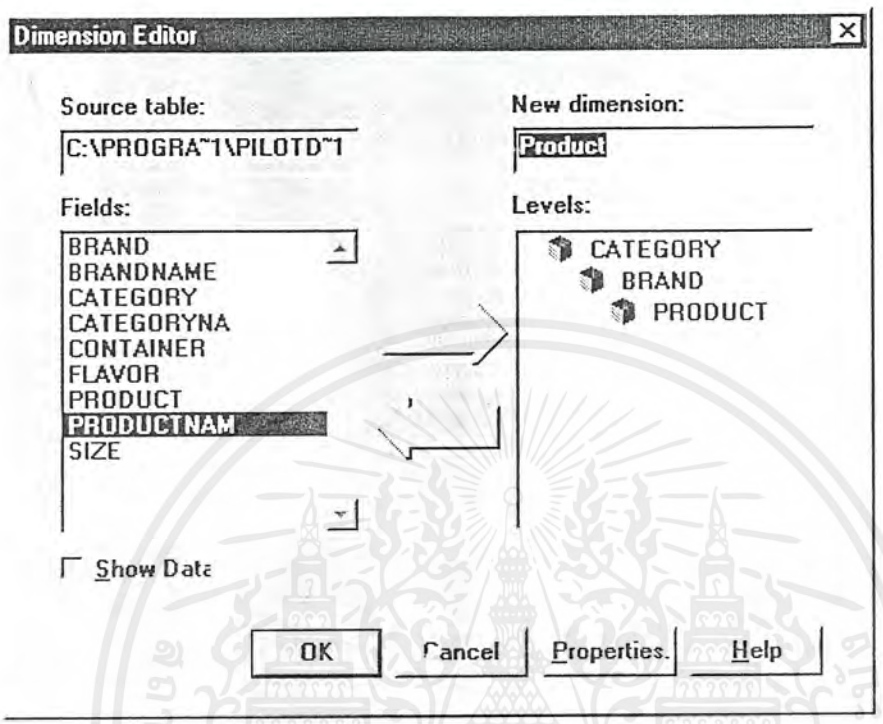


รูปที่ 4.8 แสดงหน้าจอให้ผู้ใช้เลือกตารางไดเมนชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ในที่นี้ dimension แรกที่ต้องการสร้างคือ Product dimension โดยที่ช่อง New dimension ให้ใส่ชื่อ dimension ที่ต้องการ

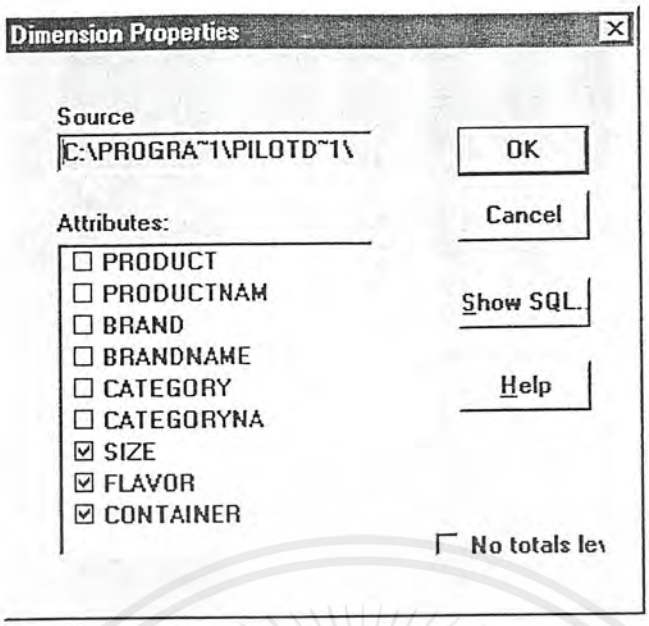
8. ช่อง level ก็เลือก field ที่ต้องการทำเป็น level ใน dimension นั้นออกมา ดังรูปที่ 4.9



รูปที่ 4.9 แสดงหน้าจอเมื่อสร้างตารางไดเมนชั่น

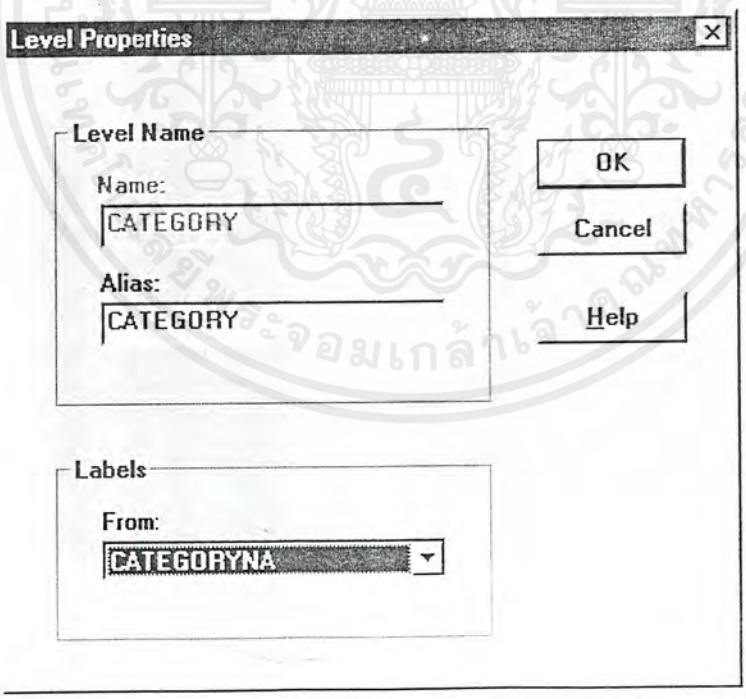
9. ถ้าต้องการให้ dimension นั้นมี attribute ก็คลิกปุ่ม properties จะปรากฏ dialog box property ของdimension นั้นให้ทำการ check ในช่องของ field ที่ต้องการให้เป็น attribute ของ dimension นั้น ดังรูปที่ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 หน้าจอของการกำหนดคุณสมบัติของแต่ละไดเมนชั่น

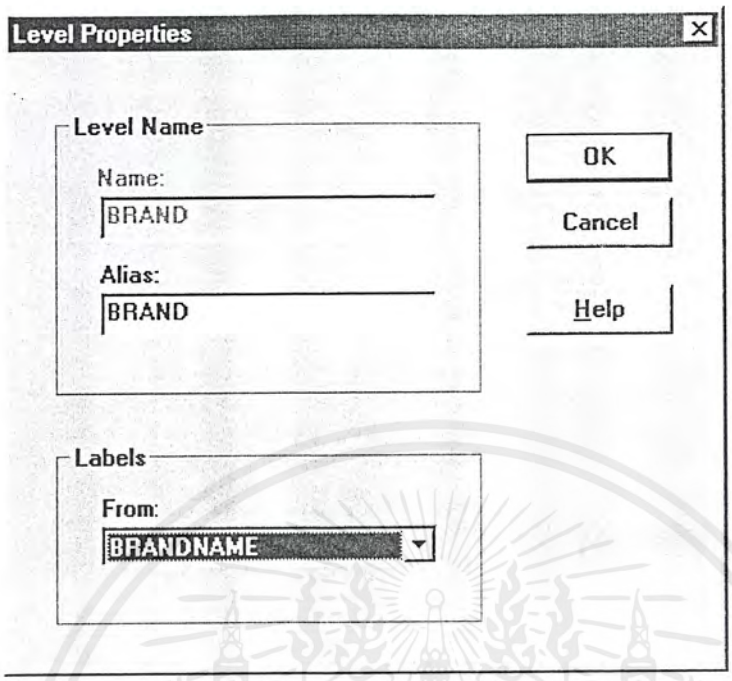
10. จากนั้นคลิกที่คำว่า Category แล้วดูค่า properties ของ level นั้น และเปลี่ยนตรง Labels From: จาก None เป็น Categoryna ซึ่งเป็นชื่อ field ใน table เพื่อที่เวลาแสดงผลจะได้แสดงออกมาเป็นข้อมูลที่เป็นชื่อ ไม่ใช่เป็นรหัส (ID) ดังรูปที่ 4.11



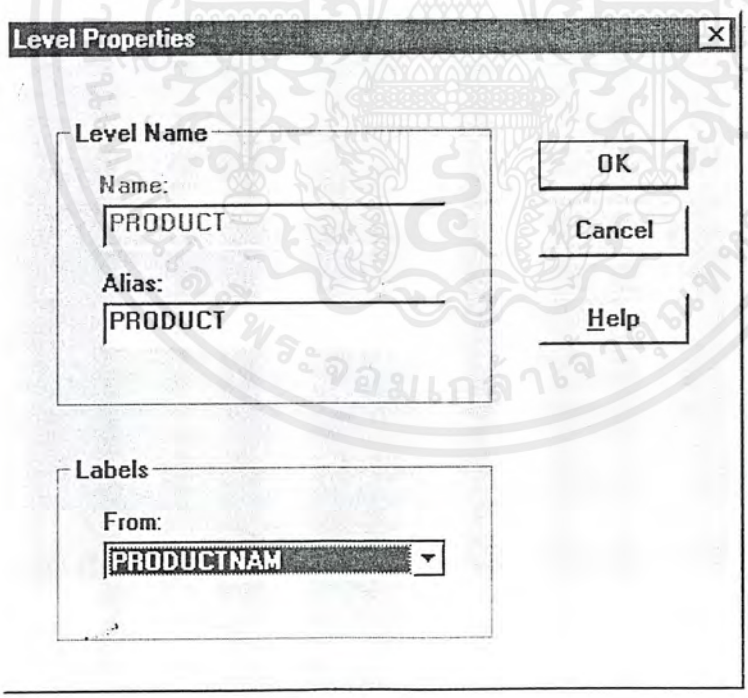
รูปที่ 4.11 หน้าจอแสดงคุณสมบัติของ Level Category

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. Level Brand และ Product ก็ทำเช่นเดียวกัน เนื่องจากทั้ง 2 field นี้ เป็น field ของ ID จะได้ดังรูปที่ 4.12 และรูปที่ 4.13 ตามลำดับ



รูปที่ 4.12 หน้าจอแสดงคุณสมบัติของ Level Brand



รูปที่ 4.13 หน้าจอแสดงคุณสมบัติของ Level Product

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. สร้าง dimension เพิ่มตามที่ต้องการ ในที่นี้จะสร้างอีก 2 dimension คือ channel dimension และ customer dimension โดยวิธีการสร้างเหมือนตอนต้นตามที่อธิบายไปแล้ว โดยที่ customer dimension จะมี attribute 2 field คือ

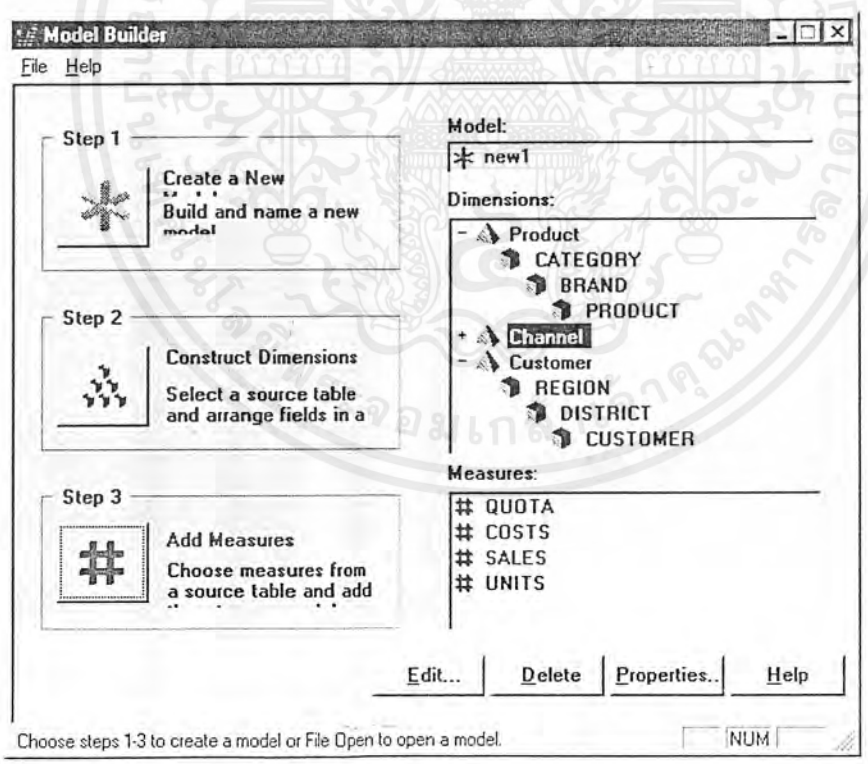
- COT
- Parent

13. Step 3 จะเป็นการ add measure ซึ่งจะเป็นตัววัดทางธุรกิจที่เราได้สร้างไว้ในฐานข้อมูล ซึ่ง measure ในธุรกิจนี้จะมีด้วยกัน 4 ตัวคือ

- Quota
- Sales
- Costs
- Units

14. สำหรับ Calculate measure ก็สามารถทำการสร้างได้ โดยการกำหนด Formula เช่น ต้องการ Margin = Sales-Costs เป็นต้น

15. หน้าจอที่สำเร็จจะเป็นดังรูปที่ 4.14



รูปที่ 4.14 แสดงหน้าจอเมื่อได้ทำการสร้างโมเดลเสร็จเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การออกแบบและพัฒนาโปรแกรมประยุกต์

5.1 การเลือกใช้ทูลส์ที่จะนำมาพัฒนาโปรแกรมประยุกต์

สำหรับทูลส์ที่นำมาใช้ในการพัฒนาระบบนั้น ผู้จัดทำได้ใช้ Delphi เป็นเครื่องมือที่ช่วยในการพัฒนางานด้านฐานข้อมูล เนื่องจาก Delphi มีเครื่องมือที่ช่วยในการพัฒนางานด้านฐานข้อมูลอย่างครบถ้วน จากการออกแบบเป็นอย่างดีของ Delphi ทำให้การพัฒนาแอปพลิเคชั่นทางด้านฐานข้อมูลกลายเป็นเรื่องง่าย และมีประสิทธิภาพสูง โดย Delphi ได้แบ่งส่วนต่างๆ ของระบบงานทางด้านฐานข้อมูลอย่างเป็นระเบียบ ตั้งแต่การเชื่อมต่อกับฐานข้อมูล การเข้าถึงฐานข้อมูล การติดต่อกับผู้ใช้ และการอำนวยความสะดวกต่างๆ

ฐานข้อมูลมาตรฐานที่ Delphi ใช้เป็นตัวเก็บข้อมูล คือ Paradox และ dBase แต่ Delphi สามารถทำงานกับฐานข้อมูลภายนอกอื่นๆ ได้ ไม่ว่าจะเป็นไฟล์ฐานข้อมูลทั่วไป เช่น Microsoft Access, Microsoft FoxPro (ซึ่งเรียกฐานข้อมูลเหล่านี้ว่า ฐานข้อมูลภายใน) และชุด Client/Server ยังสามารถใช้กับระบบฐานข้อมูลที่อยู่บนเครื่อง Server เช่น Oracle, MS SQL, SyBase, Interbase (ซึ่งจะเรียกฐานข้อมูลเหล่านี้ว่า ฐานข้อมูล Server) หรืออาจกล่าวได้ว่า

- ฐานข้อมูลภายใน คือฐานข้อมูลที่อยู่ในรูปแบบของไฟล์สำหรับเก็บข้อมูลปกติจะอยู่ภายในเครื่องเดียวกันกับแอปพลิเคชั่นที่ใช้งาน เช่น Paradox, dBase, MS Access, MS FoxPro เป็นต้น
- ฐานข้อมูลภายนอก คือฐานข้อมูลที่มีระบบจัดการฐานข้อมูลคอยควบคุมการใช้ข้อมูล ปกติจะอยู่บนเครื่องที่ทำหน้าที่เป็น Database Server เช่น ฐานข้อมูล Oracle, MS SQL Server, SyBase, Interbase เป็นต้น

Delphi มีการเชื่อมต่อกับฐานข้อมูลผ่านระบบที่ชื่อว่า Borland Database Engine (BDE) ซึ่งเป็นตัวจัดการระบบทางด้านฐานข้อมูลทั้งหมด

5.1.1 เครื่องมือที่ช่วยพัฒนาแอปพลิเคชั่นทางด้านฐานข้อมูลของ Delphi

Delphi มีโปรแกรมที่ช่วยอำนวยความสะดวกในการพัฒนางานด้านฐานข้อมูลให้อย่างเพรียบพร้อม โปรแกรมต่างๆ เหล่านี้ จะช่วยให้การพัฒนางานสะดวก และง่ายขึ้นเป็นอย่างมาก ตั้งแต่การสร้าง และปรับปรุงโครงสร้างฐานข้อมูล การโอนย้ายข้อมูล การแก้ไขข้อมูล และการแสดงข้อมูล

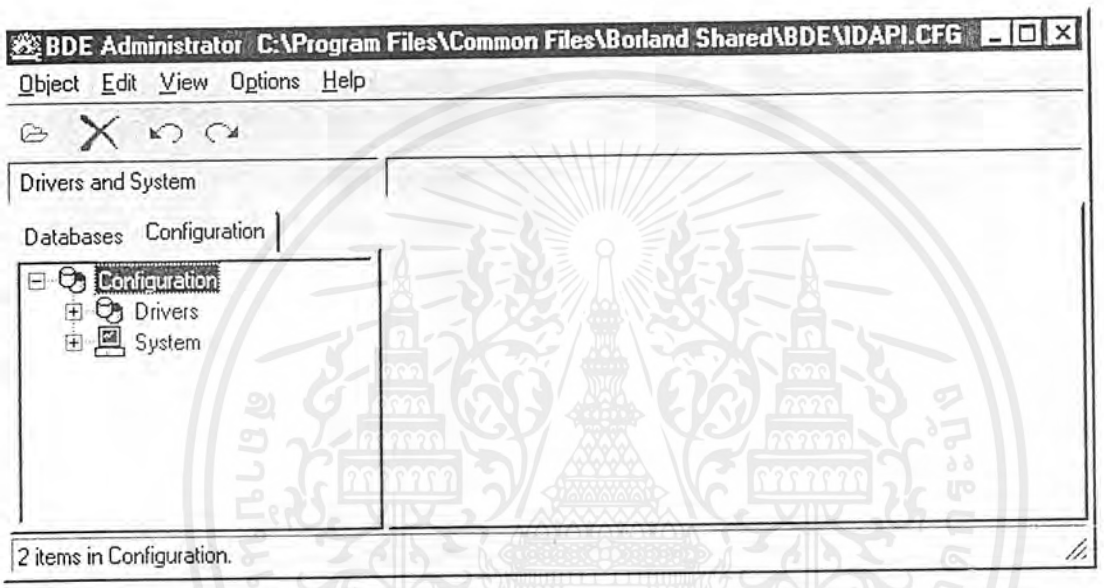
โปรแกรมสำหรับช่วยงานทางด้านฐานข้อมูลใน Delphi มีดังนี้

- BDE Administrator
- Database Desktop
- Database Explorer
- Data Migration Expert
- SQL Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับโครงงานที่จัดทำขึ้นนี้ ได้ใช้งานเฉพาะโปรแกรม BDE Administrator, Database Desktop และ Database Explorer ซึ่งจะอธิบายรายละเอียดต่อไป ในตอนแรกที่ทำโครงงานได้ทำการสร้างตารางโดยใช้ฐานข้อมูล Paradox เพื่อทำการทดสอบโปรแกรมก่อน เมื่อโปรแกรมสามารถใช้งานได้จึงทำการติดต่อกับฐานข้อมูลออราเคิล

5.1.1.1 BDE Administrator เป็นโปรแกรมสำหรับสร้าง และกำหนดคุณสมบัติต่างๆ ที่จำเป็นต้องใช้ในการเชื่อมต่อกับฐานข้อมูล เช่น การกำหนดคุณสมบัติต่างๆ ของ BDE, สร้างและกำหนดคุณสมบัติของ Driver และ Alias ซึ่งการทำงานต่างๆ เหล่านี้ ล้วนเป็นส่วนสำคัญมากในการเชื่อมต่อกับฐานข้อมูลดังรูปที่ 5.1



รูปที่ 5.1 แสดงรูปหน้าจอของ BDE Administrator

หน้าที่การทำงานของ BDE Administrator มีดังนี้

- กำหนดคุณสมบัติต่างๆ ของ BDE (Borland Database Engine)
- กำหนดคุณสมบัติของ Driver ในการเชื่อมต่อกับฐานข้อมูล STANDARD (Paradox, dBase/FoxPro), SQL, Access และ ODBC driver
- สร้างและลบ ODBC driver
- สร้างและแก้ไข Database Aliases

หน้าต่างของ DBE Administrator มีลักษณะคล้ายกับหน้าต่างของโปรแกรม Windows Explorer คือประกอบด้วย 2 ด้านคือ

- ด้านซ้ายเป็น Tree View ซึ่งประกอบด้วยแทป 2 แทป คือ
  - แทป Database ซึ่งจะแสดง Alias ที่ได้มีการสร้างเอาไว้
  - แทป configuration ใช้ในการกำหนดค่างต่างๆ ของ Driver และ BDE

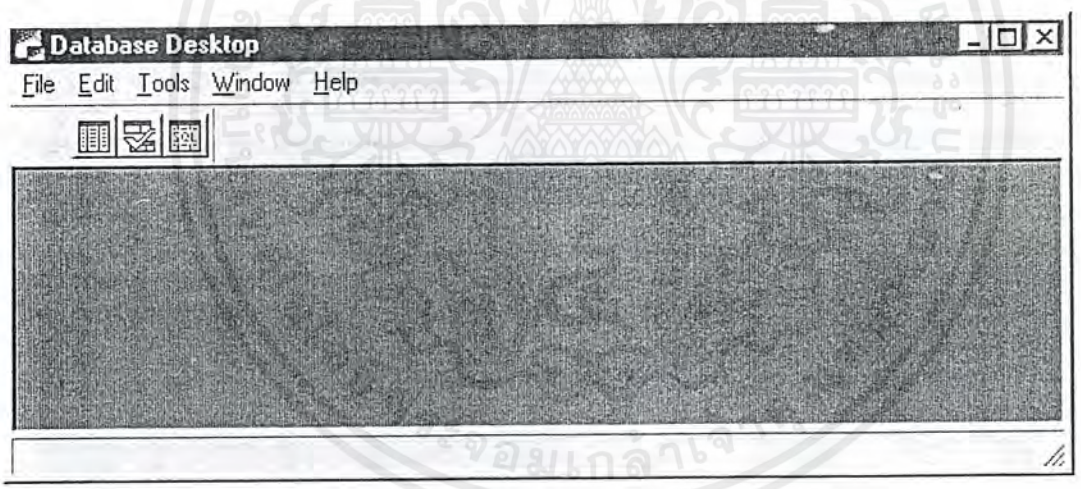
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ด้านขวาเป็น List View ซึ่งจะแสดงคุณสมบัติและรายละเอียดต่างๆ ของออบเจกต์ที่ถูกเลือกในด้าน Tree View โดยลักษณะของ List View จะเปลี่ยนไปตามออปเจกต์ที่ถูกเลือกในด้าน Tree View

เราอาจจะต้องกำหนดคุณสมบัติต่างๆ ของ Driver และ Alias ให้ถูกต้อง เพื่อใช้ในการเชื่อมต่อกับฐานข้อมูล รายละเอียดในการกำหนดคุณสมบัติของ Driver และ Alias ในแต่ละตัวจะแตกต่างกันไปตามความต้องการของระบบนั้นๆ เราสามารถดูรายละเอียดในการกำหนดคุณสมบัติของ Alias และ Driver ในแต่ละตัวได้จาก Help ของมัน สำหรับการกำหนดคุณสมบัติต่างๆ ของ Driver และ Alias ของแอปพลิเคชั่นที่ได้ออกแบบขึ้นมา มีรายละเอียดใน ภาคผนวก ข.

5.1.1.2 Database Desktop เป็นโปรแกรมสำหรับจัดการเกี่ยวกับงานทางด้านฐานข้อมูลต่างๆ ไม่ว่าจะเป็นการสร้าง การแสดง และการแก้ไขตาราง และอินเด็กซ์ของฐานข้อมูล รวมไปถึงการแก้ไขข้อมูลในตารางและการสร้างไฟล์คำสั่ง SQL

เรามักจะใช้ Database Desktop ในการสร้างตารางและอินเด็กซ์ ตลอดจนการกรอกข้อมูลเริ่มต้น นอกจากนี้ Database Desktop ยังเป็นเครื่องมือที่มีความสามารถพร้อมในการปรับปรุงเปลี่ยนแปลงค่าต่างๆ ในตาราง ไม่ว่าจะเป็นการเพิ่ม/ลดฟิลด์ในตาราง การ Pack ข้อมูล เป็นต้น ดังรูปที่ 5.2



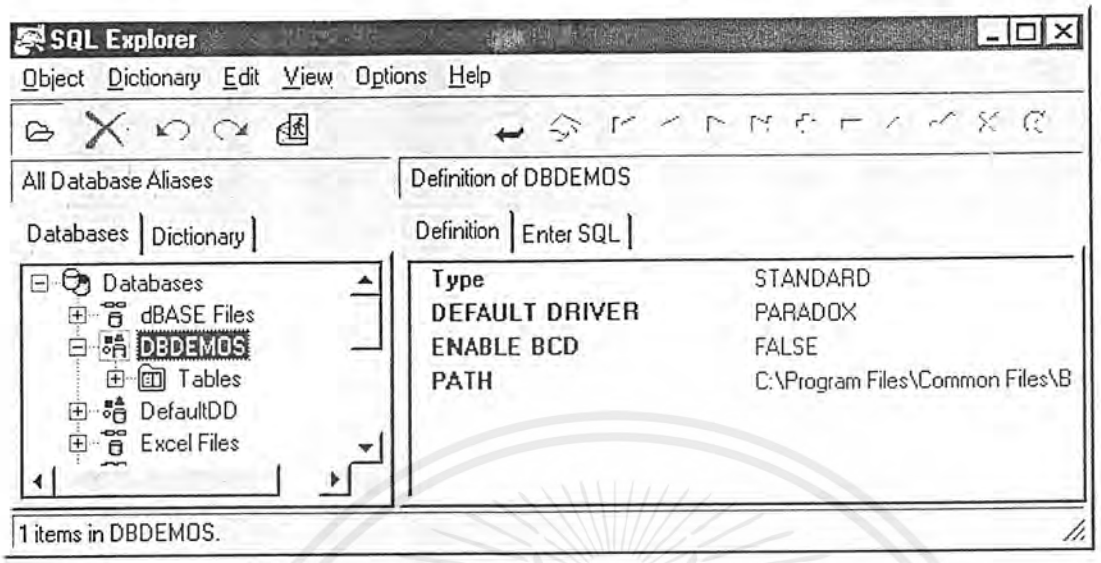
รูปที่ 5.2 แสดงหน้าจอของ Datbase Desktop

5.1.1.3 SQL Explorer หรือ Database Explorer เป็นโปรแกรมที่ใช้ในการแสดงรายละเอียดต่างๆ เกี่ยวกับฐานข้อมูล ตั้งแต่รายละเอียดของ Alias ตาราง อินเด็กซ์ ฟิลด์ ตลอดจนข้อมูลภายในตาราง นอกจากนี้ใน Database Explorer เรายังสมารถเขียนคำสั่งภาษา SQL เพื่อแสดงและจัดการกับข้อมูล รวมทั้งโครงสร้างของฐานข้อมูลได้

Database Explorer เป็นโปรแกรมที่มีประโยชน์มากในการแสดงและแก้ไขรายละเอียดต่างๆ ของตาราง และข้อมูลในตาราง ซึ่ง Database Explorer เป็นโปรแกรมที่ใช้งานง่าย และมีความสามารถที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลากหลาย โดยเฉพาะความสามารถในการรับคำสั่ง SQL ซึ่งจะมีส่วนช่วยเป็นอย่างมากในการแสดงและแก้ไขข้อมูล ตลอดจนการทดสอบคำสั่งภาษา SQL ที่จะเอาไปใช้ในแอปพลิเคชั่น ดังรูปที่ 5.3



รูปที่ 5.3 แสดงหน้าจอของ SQL Explorer

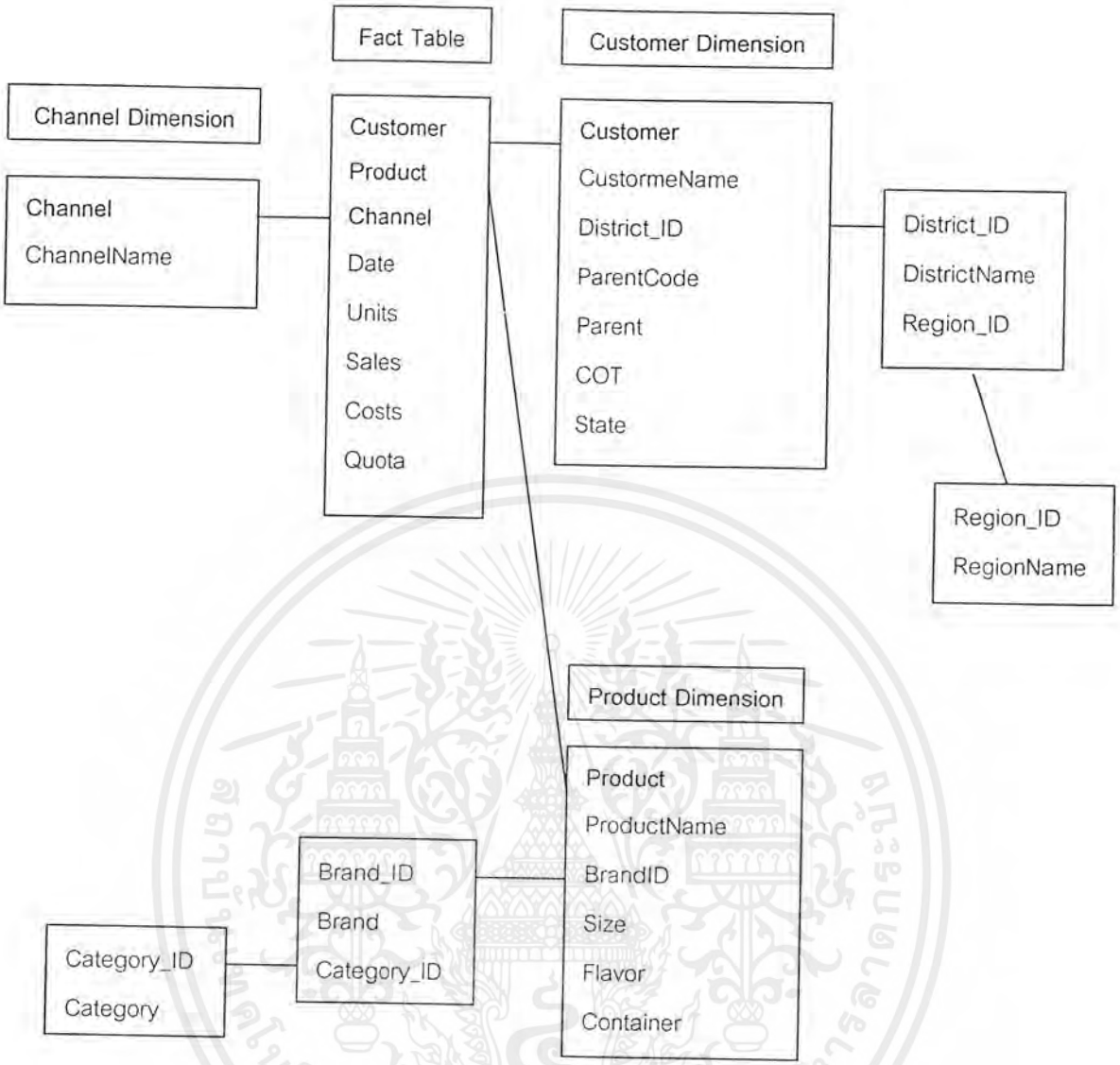
5.2 ระบบจัดการฐานข้อมูลออราเคิล 8 (Oracle 8 Database Management System)

ออราเคิล 8 เป็นระบบฐานข้อมูลแบบเชิงวัตถุสัมพันธ์ (Object Relation Database Management System : ORDBMS) ซึ่งเป็นระบบฐานข้อมูลที่สามารถรองรับข้อมูลที่มีความซับซ้อนมากๆ เช่นข้อมูลมัลติมีเดีย (Multimedia), รูปภาพ, เสียง เป็นต้น สามารถจัดการกับข้อมูลเหล่านี้ได้อย่างมีประสิทธิภาพ โดยออราเคิล 8 ได้จัดเตรียมชนิดข้อมูลชนิดใหม่ขึ้นมาเพื่อรองรับกับข้อมูลที่มีความซับซ้อนมากๆ เหล่านี้ โดยเฉพาะ ยอมให้มีข้อมูลชนิดที่ผู้ใช้กำหนดขึ้นมาเอง (User-Defined Datatypes) ข้อมูลชนิดนี้จะมีคุณสมบัติทาง Object Oriented เช่นเดียวกับ Object ในโปรแกรมภาษาที่สามารถเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ต่างๆ แต่คุณสมบัติบางอย่างอาจยังไม่เทียบเท่าโปรแกรมภาษาเหล่านั้น เช่น คุณสมบัติการสืบทอด (Inheritance)

5.3 การออกแบบฐานข้อมูล (Database Design)

ฐานข้อมูลที่ได้ทำการออกแบบจะเป็นแบบมัลติไดเมนชั่นนอลดาต้าเบส (Multidimensional database) ซึ่งหากตารางไดเมนชันมีข้อมูลที่มีโครงสร้างเป็นลำดับชั้นภายในแล้ว จะต้องทำให้เป็นโครงสร้างแบบสโนว์เฟลก (Snowflake) ดังรูปที่ 5.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 แสดงการออกแบบฐานข้อมูลแบบสโนว์เฟลก

ข้อดีในการออกแบบฐานข้อมูลให้มีโครงสร้างสโนว์เฟลกนี้ก็เพื่อรองรับการใช้งานกับฐานข้อมูลขนาดใหญ่ที่มีจำนวนข้อมูลหลายแสนเรคอร์ด ทำให้ง่ายต่อการเข้าใจ รวมทั้งลดความซ้ำซ้อนซึ่งจะทำให้ลดปัญหาเรื่องการอัพเดทฐานข้อมูลลงได้ และอาจจะเพิ่มความเร็วในการ select ข้อมูลขึ้นมา เช่น ถ้าต้องการ select เฉพาะ ProductName โดยไม่ทำการ select BrandName ก็ไม่จำเป็นต้องทำการ join ระหว่าง ตารางโปรดักส์กับตารางแบรนด์ ก็จะลดเวลาในการ scan ตารางลงได้

ตารางที่สร้างต้องเก็บในระบบจัดการฐานข้อมูลออราเคิล และฐานข้อมูลจะสามารถนำมาใช้ได้ต้องไปสร้าง Alias ใน BDE (Borland Database Engine) ของ Delphi เพื่อให้สามารถเชื่อมต่อกับโปรแกรมที่ได้ทำการออกแบบไว้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับฐานข้อมูลนั้นต้องมีการกำหนดให้ Primary key ของตารางแฟ็ก มาจากการรวมกันของทุกๆ คีย์จากตารางไดเมนชั่น หรือก็คือ foreign key ของตารางแฟ็กจะต้องได้มาจาก primary key ของตารางไดเมนชั่นทั้งหมด ส่วนตารางย่อย หรือสโนว์เฟลกก็จะต้องมีชื่อฟิลด์หรือ primary key ที่เหมือนกับ foreign key ของตารางไดเมนชั่นที่แตกออกมา

สำหรับในตารางแฟ็กนั้น จะต้องมีชื่อฟิลด์ที่มีชนิดของข้อมูลเป็นเวลา (time/date) ด้วย รวมทั้งจะต้องมีฟิลด์ที่เป็นตัวเลขที่สามารถวัดได้ เช่นฟิลด์ยอดขาย ฟิลด์ค่าใช้จ่าย ฯลฯ เพื่อนำฟิลด์ดังกล่าวมาสรุปเป็นข้อมูลโดยรวมเพื่อนำมาสนับสนุนการตัดสินใจได้

5.4 การทำ SQL แบบไดนามิก (Dynamic SQL)

5.4.1 Dynamic SQL

Dynamic SQL เป็นการสร้าง SQL statement ในขณะที่โปรแกรมกำลังวิ่ง (Run-time) แล้วส่ง statement เหล่านั้นไปให้กับระบบจัดการฐานข้อมูล (DBMS) ทำการประมวลผล (execute) หรือกล่าวอย่างง่ายๆ ว่าการทำ dynamic SQL ระบบจัดการฐานข้อมูลจะไม่มีทางทราบ SQL statement จนกระทั่งโปรแกรมวิ่ง

5.4.2 การเขียนคำสั่ง SQL

การเขียนคำสั่ง SQL ได้อธิบายไว้แล้วในบทที่ 3 หัวข้อที่ 3.7 ซึ่งเมื่อ Star schema ถูก Query จะต้องมีการใช้ ตารางไดเมนชั่น (Dimension table) ในการ query เสมอ เพราะในตารางไดเมนชั่นจะเก็บชื่อและคำอธิบายที่ query ต้องการใช้เอาไว้ ซึ่งคำถามของ star schema จะประกอบด้วย Where clause 2 ส่วน คือ

- การรวม (join) กันระหว่าง fact table และ ตารางไดเมนชั่น
- เงื่อนไขที่จะมีสำหรับคอลัมน์ในตารางไดเมนชั่น

เช่น

```
select   Product.ProductName,
         Customer.CustomerName,
         Channel.ChannelName,
         Fact.Sales
From     Product, Customer, Channel, Fact

Where    Product.ProductName = 'Courtyard'
And      Customer.CustomerName = '7Eleven'
And      Channel.ChannelName = 'Direct'
And      Fact.Date = 'Last 3 months'
```

เงื่อนไขที่จะมีสำหรับคอลัมน์ในตารางไดเมนชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
And     Fact.Product = Product.Product
And     Fact.Customer = Customer.Customer
And     Fact.Channel = Channel.Channel
```

เงื่อนไขการ join กันระหว่างตาราง fact และ ตารางไดเมนชั่น

5.4.3 การสร้าง Dynamic SQL

การสร้าง Dynamic SQL ทำได้โดยการนำชื่อฟิลด์ต่างๆ ที่ผู้ใช้ต้องการเรียกดู เก็บไว้ในตัวแปรที่กำหนดให้มีชนิดของข้อมูลเป็น string โดยเมื่อผู้ใช้เลือกชื่อฟิลด์ต่างๆ จาก component เช่น ComboBox, ListBox ก็จะทำการเก็บข้อมูลเหล่านั้นให้อยู่ในรูปของตัวแปรที่กำหนดขึ้นมา เมื่อผู้ใช้ต้องการนำข้อมูลเหล่านั้นมาวิเคราะห์ ก็นำตัวแปรเหล่านั้นมารวมกันเพื่อสร้างให้เป็น SQL Script ดังนั้นโปรแกรมที่จัดทำขึ้นจึงต้องทราบให้ได้ว่า แต่ละฟิลด์มาจากตารางไหน และต้องทราบว่าฟิลด์ในตารางแต่ละอันต้อง join กับตารางไหนบ้าง รวมทั้งต้องทราบเงื่อนไขการ join กันระหว่างตาราง fact กับตารางไดเมนชั่น และเงื่อนไขสำหรับแต่ละคอลัมน์ที่ผู้ใช้ต้องการด้วย ตัวอย่างการเขียน dynamic SQL

```
runscript := 'Select  ' + ListField +','+ TimeKey + ',Sum(' + FactFieldCombobox.Text + ')';
runscript := runscript+ #13#10;
runscript := runscript+'From ' + ListTable;
runscript := runscript+ #13#10;
runscript := runscript+'Where ' + ListJoin;
runscript := runscript+ #13#10;
if setcondition = true then
        runscript := runscript+ 'and '+ StrCondition + #13#10;
runscript := runscript+'Group By ' + ListField + ',' +TimeKey;
Listbox1.Items.Text := runscript;
```

5.5 การออกแบบส่วนที่ติดต่อกับผู้ใช้ (User Interface)

User interface ที่ได้ทำการออกแบบแบ่งการใช้งานออกเป็น 2 ส่วนหลักๆ คือ

5.5.1 การสร้างโมเดล

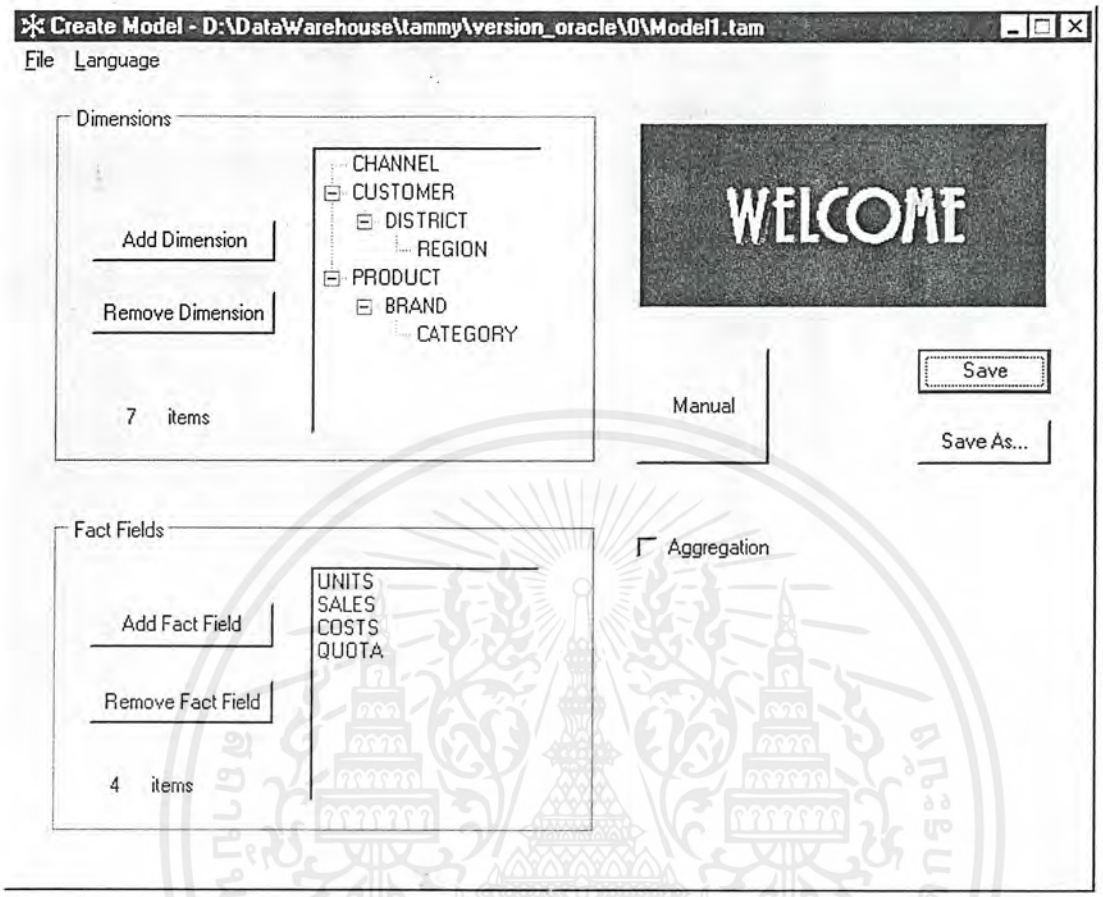
ส่วนนี้จะเป็นส่วนที่ใช้ในการสร้างโมเดล โดยจะต้องสร้างจากฐานข้อมูลดาต้าแวร์เฮาส์ที่อยู่บนระบบฐานข้อมูลออราเคิล และเป็นฐานข้อมูลที่มีอยู่แล้ว ซึ่งในส่วนนี้ผู้ใช้สามารถที่จะบันทึกโครงสร้างที่ได้สร้างไว้ และนำมาเรียกดูในภายหลังได้

หากต้องการสร้างโมเดลใหม่ก็สามารถเลือกที่ New หรือหากต้องการดูโมเดลที่สร้างไว้แล้วก็สามารถเรียกดูได้จาก Open

เมื่อต้องการสร้างโมเดลใหม่ ขั้นตอนแรกก็ต้องทราบ Alias ของฐานข้อมูลที่มีอยู่ว่าชื่ออะไร จากนั้นก็ติดต่อกับ Alias นั้นๆ แล้วเลือกตารางไดเมชั่น และตัววัดที่ต้องการจากฐานข้อมูลที่ได้ทำการเอาเลือกไว้ เมื่อได้ทำการสร้างโมเดลดังกล่าวแล้ว ก็จะต้องนำโมเดลที่สร้างไว้มาทำการวิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.5 เป็นตัวอย่างของ User interface ของการสร้างโมเดล ซึ่งต้องมีการเพิ่มตารางไดเมนชั่น และเพิ่มตัววัด



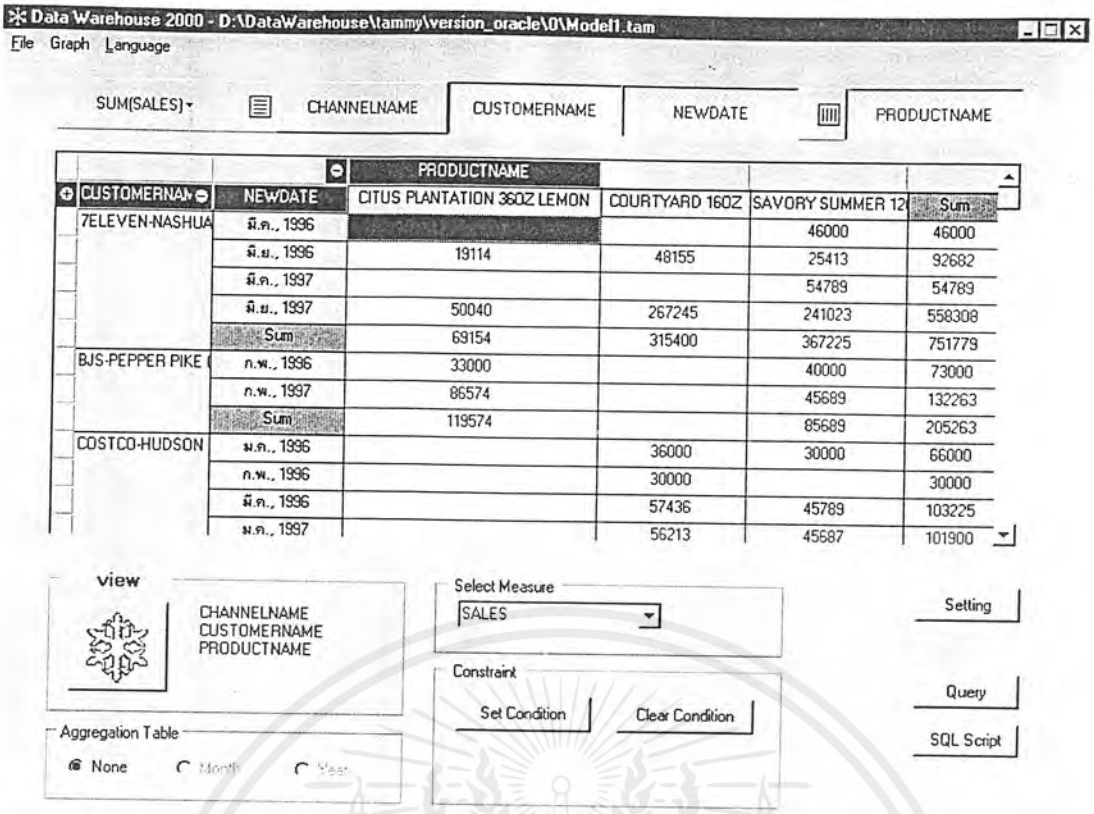
รูปที่ 5.5 แสดงฟอร์มที่ใช้การสร้างโมเดล

5.5.2 การวิเคราะห์

หลังจากสร้างโมเดลแล้วก็จะต้องนำโมเดลนั้นมาวิเคราะห์เพื่อช่วยสนับสนุนการตัดสินใจของผู้บริหาร ดังนั้นเมื่อผู้ใช้เปิดทูลส์ตัวนี้ขึ้นมาจะต้องเลือกเมนู Load Model จากนั้นจึงเลือกโมเดลที่ได้ทำการบันทึกเอาไว้ แล้วเลือกไดเมนชั่น และตัวที่ต้องการจะนำมาวิเคราะห์ คลิกที่ปุ่ม Query เพื่อให้ข้อมูลแสดงออกมาในรูปแบบตาราง

ผู้ใช้สามารถ Drill up, drill down และใส่เงื่อนไขที่ต้องการได้ เพื่อดูถึงแนวโน้มที่เกิดขึ้น รวมทั้งยังสามารถเปลี่ยนแกนของเวลา เพื่อดูข้อมูลรวมตามที่ต้องการได้ เช่น ดูข้อมูลรวมเป็นรายปี รายเดือน รายสัปดาห์ ฯลฯ รูปที่ 5.6 เป็นตัวอย่างของ User interface ของการวิเคราะห์โมเดลที่ได้ทำการสร้างเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 แสดงฟอร์มที่ใช้การวิเคราะห์โมเดลที่ได้สร้างเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การใช้งานของโปรแกรมประยุกต์

6.1 ทูลส์ที่นำมาใช้ในการพัฒนาโปรแกรมประยุกต์

ในโปรแกรมประยุกต์ที่ได้สร้างขึ้นมา ได้ใช้ Delphi เป็นทูลส์ในการพัฒนาโปรแกรม เนื่องจากทูลส์ตัวนี้มีความเหมาะสมที่จะใช้ในการพัฒนาโปรแกรมทางด้านฐานข้อมูล (Database application) กล่าวคือ

ในการพัฒนาโปรแกรมทางด้านฐานข้อมูล ทูลส์ที่ใช้นั้นจะต้องมีความสามารถในการทำงานได้หลายแพล็ทฟอร์ม (scaleability) ตัวอย่างเช่น ทูลส์นั้นจะต้องสามารถทำงานได้ดีทั้ง ตารางของ dBase, ตารางของ Sybase เป็นต้น ความสามารถต่างๆ ของ Delphi ที่เหมาะสมกับการนำมาใช้เป็นทูลส์ในการพัฒนาโปรแกรมทางด้านฐานข้อมูลมีดังต่อไปนี้

- Delphi สนับสนุนทั้ง ตารางแบบโลคอล (local database) และตารางที่อยู่บนเครื่องเซิฟเวอร์ฐานข้อมูลจากที่อื่น (remote database server)
- Delphi สนับสนุนการถามคำถาม (Query) ได้หลายประเภท และสามารถเข้าถึงแพล็ทฟอร์มของระบบจัดการฐานข้อมูลได้หลายแพล็ทฟอร์มภายในโปรแกรมประยุกต์ (application) เพียงอันเดียว
- Delphi สามารถย้ายโปรแกรมประยุกต์จากการใช้งานระบบจัดการฐานข้อมูลหนึ่งไปยังอีกระบบจัดการฐานข้อมูลหนึ่งได้ง่าย โดยเป็นการเข้าถึงฐานข้อมูลผ่านทาง BDE (Borland Database Engine) ทำให้มีลักษณะเป็นแบบไม่ขึ้นกับแพล็ทฟอร์ม (platform-independent)
- Delphi มี native drivers ที่มีความรวดเร็วสำหรับแพล็ทฟอร์มของ client/server ได้หลายๆ แพล็ทฟอร์ม
- Delphi สนับสนุน ODBC อย่างสมบูรณ์แบบ

6.2 การทำงานของโปรแกรมประยุกต์

เนื่องจากหลักการของดาต้าแวร์เฮาส์เป็นหลักการที่มุ่งเน้นในเรื่องการตั้งคำถาม (Query) ฐานข้อมูลที่มีพื้นฐานของดาต้าแวร์เฮาส์ (Data warehouse based) เพื่อให้ได้มาซึ่งคำตอบที่ต้องการจะนำไปใช้ตัดสินใจได้อย่างรวดเร็ว ดังนั้นโปรแกรมประยุกต์ที่เกี่ยวกับดาต้าแวร์เฮาส์นี้จะไม่มีการอัพเดทฐานข้อมูลเหมือนในโปรแกรมประยุกต์สำหรับฐานข้อมูลทั่วไป การทำงานของโปรแกรมประยุกต์ที่ได้ออกแบบขึ้นมาจะเป็นการสร้าง SQL Script ขึ้นมาตามคำสั่งที่ผู้ใช้ต้องการใช้ในการตั้งคำถามแต่ละครั้ง หรือเรียกว่าเป็น Dynamic SQL นั่นเอง

รูปแบบของคำสั่ง SQL จะประกอบด้วย select list, from clause, where clause ซึ่ง where clause นั้นจะมี 2 ส่วนคือส่วนที่ใช้ join ตารางไดเมนชั่นกับตารางแฟ็ก (join constrain) และส่วนที่ใช้เลือกเฉพาะแถวที่ต้องการ (application constrain) หรืออาจเรียกว่า search condition ก็ได้, group by clause

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโปรแกรมที่ได้ทำการออกแบบขึ้นมาฟอร์มที่ใช้รับคำสั่งจากผู้ใช้จะมีอยู่ 6 ฟอร์มด้วยกัน ซึ่งจะทำการรับคำสั่งตามที่ผู้ใช้ต้องการ โดยแบ่งเป็นส่วนๆ ตามลักษณะการใช้งาน ดังนี้

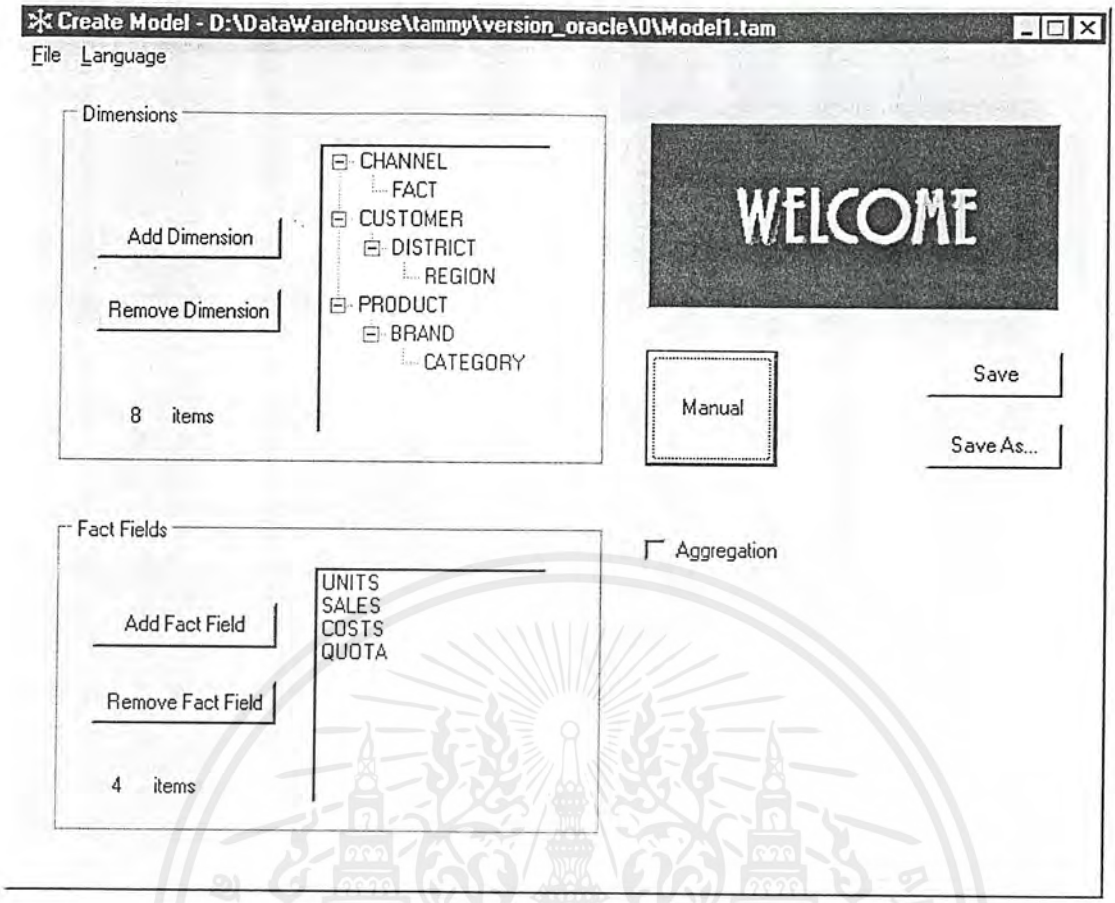
- ฟอร์มที่ 1 เป็นฟอร์มหลักในการทำงาน ซึ่งจะแบ่งการทำงานออกเป็น 2 ส่วนหลักๆ คือ
  - ส่วนที่ใช้สร้างโมเดล
  - ส่วนที่ใช้ในการวิเคราะห์ข้อมูลที่ได้สร้างเอาไว้แล้ว

ดังรูปที่ 6.1

รูปที่ 6.1 หน้าจอหลักของโปรแกรมประยุกต์

- ฟอร์มที่ 2 เป็นฟอร์มที่ใช้ในการสร้างโมเดล โดยผู้ใช้จะต้องเลือกตารางไดเมนชัน และตารางแฟ็กที่ต้องการ และบันทึกข้อมูลนั้นๆ เก็บเอาไว้เพื่อนำโมเดลนี้ไปวิเคราะห์ต่อไป ดังรูปที่ 6.2

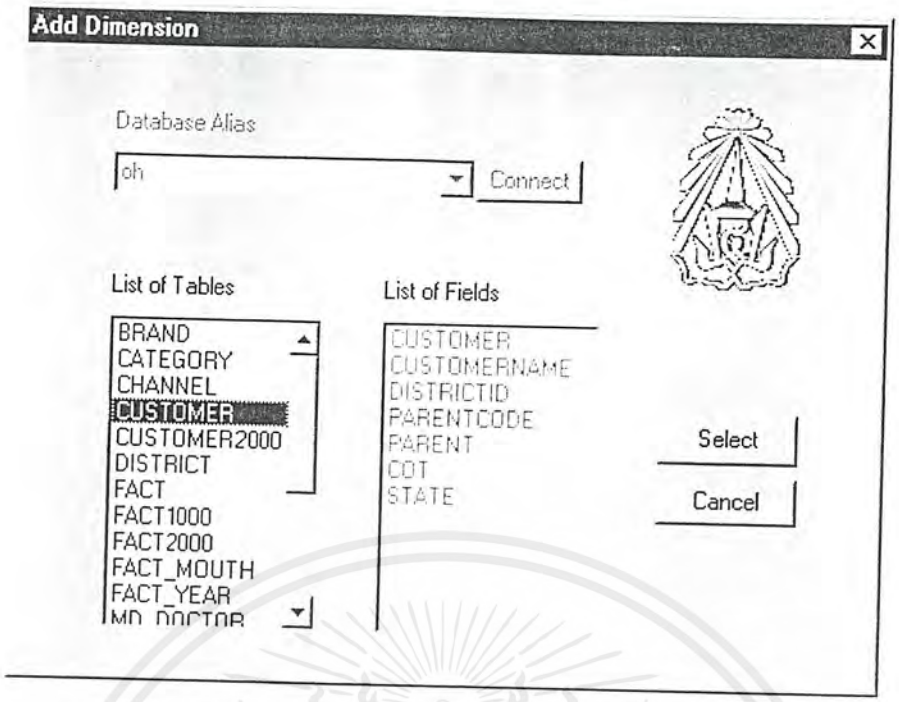
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



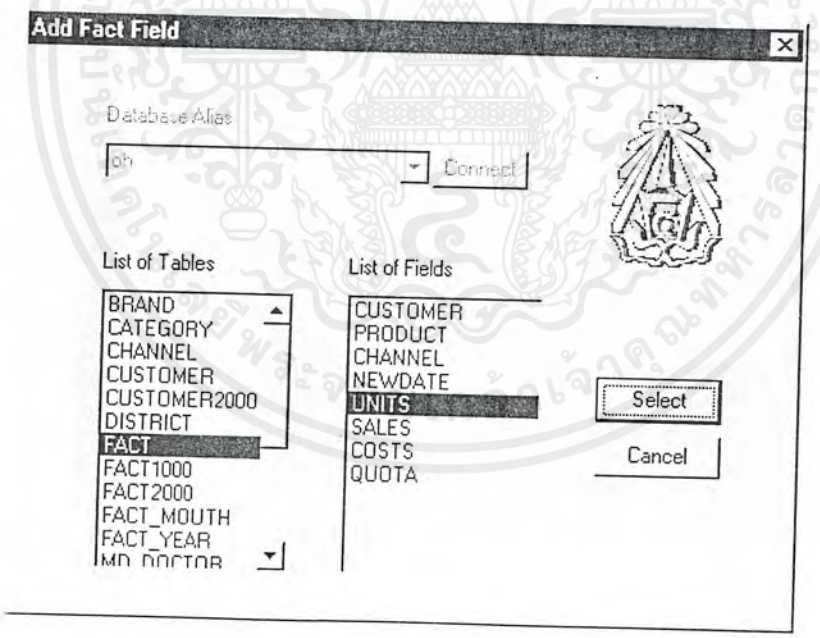
รูปที่ 6.2 ฟอร์มในการสร้างโมเดล

- ฟอร์มที่ 3 เป็นส่วนที่ใช้ในการเพิ่มตารางไดเมนชั่น และเพิ่มค่าตัววัด โดยจะทำการติดต่อกับฟอร์ที่ 2 คือเมื่อคลิกที่ปุ่ม Add Dimension หรือ Add Fact Field ก็จะปรากฏฟอร์มที่ 3 นี้ขึ้นมา ฟอร์มนี้จะเป็นฟอร์มที่ทำการติดต่อกับระบบจัดการฐานข้อมูล เพื่อดึงเอาชื่อของตารางทั้งหมดที่มีในระบบจัดการฐานข้อมูลนั้นออกมาให้ผู้ใช้ได้เลือก ดังรูปที่ 6.3 เป็นการเลือกตารางไดเมนชั่น และรูปที่ 6.4 เป็นการเลือกตัววัดที่มาจากตารางแฟ็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



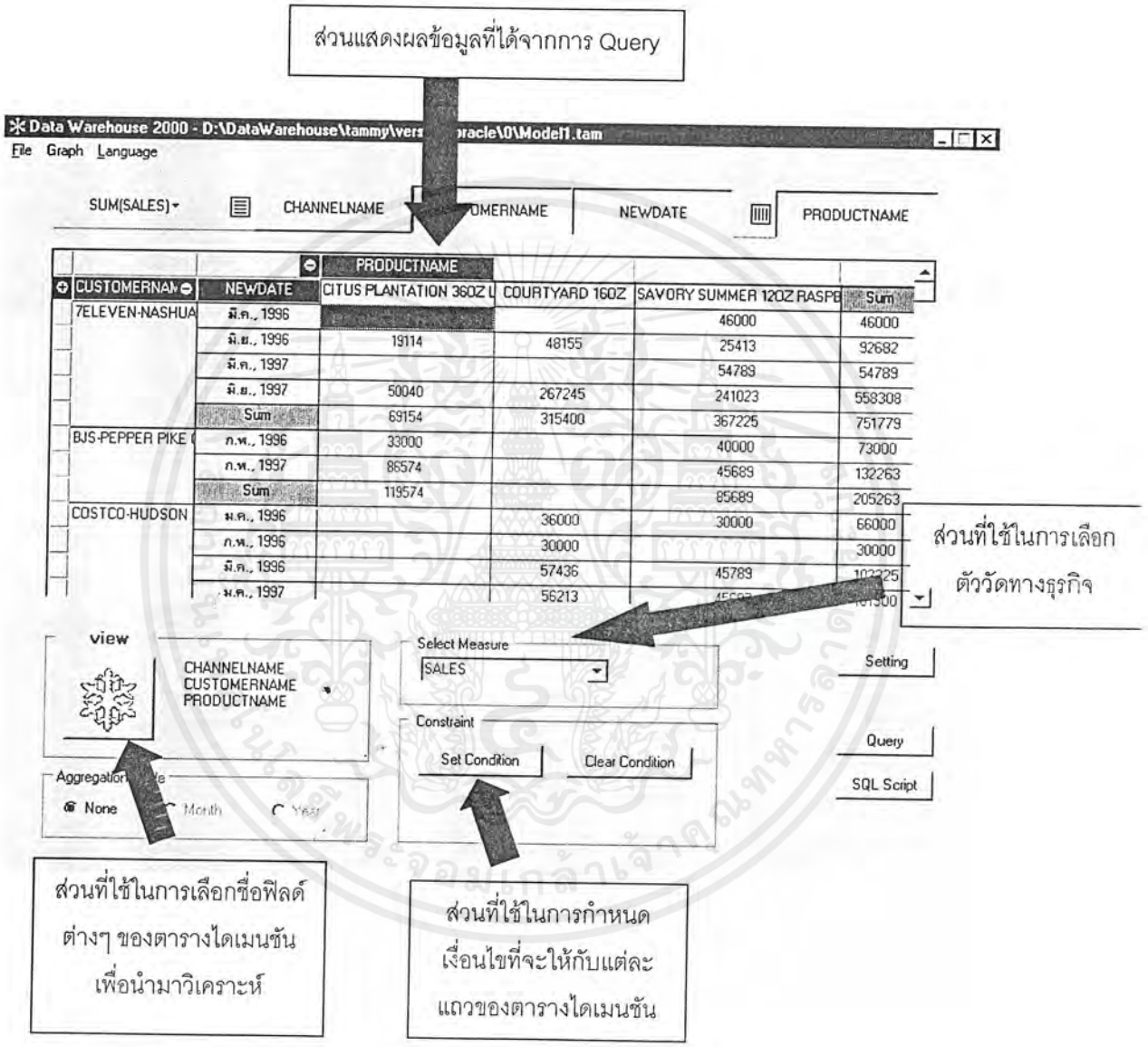
รูปที่ 6.3 ฟอร์มในการเพิ่มตารางไดเมนชัน



รูปที่ 6.4 ฟอร์มที่ใช้ในการเพิ่มตัววัดทางธุรกิจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

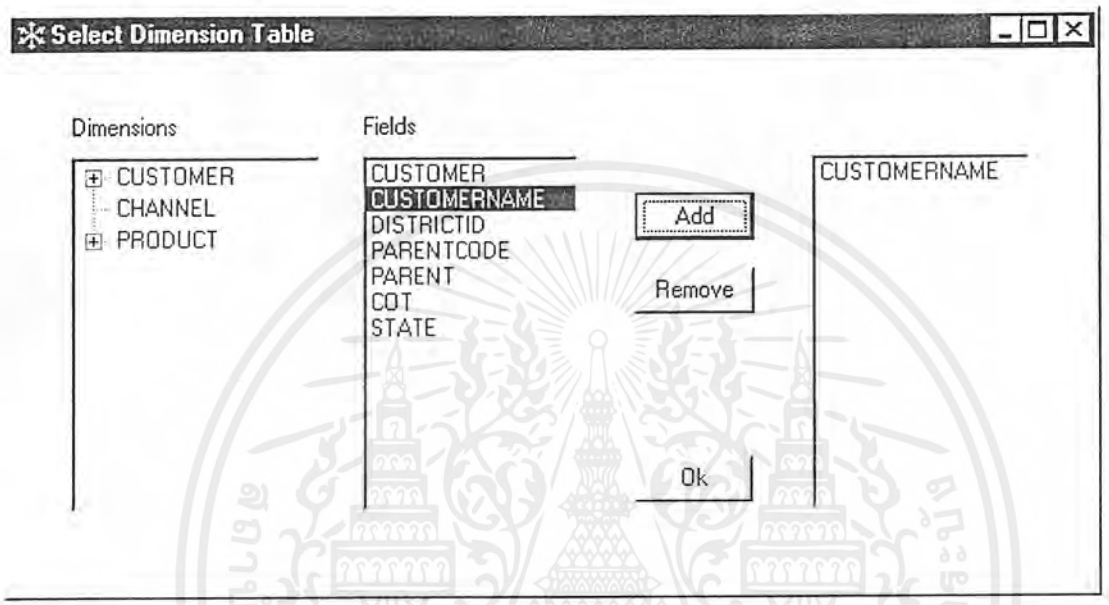
- ฟอร์มที่ 4 เป็นฟอร์มที่ใช้ในการวิเคราะห์ข้อมูล โดยจะทำการโหลดจากโมเดลที่ได้สร้างไว้แล้ว จากนั้นก็มาเลือกเงื่อนไข เลือกฟิลด์ ที่ต้องการจะวิเคราะห์ จากตัวอย่างรูปที่ 6.5 เป็นการเลือกฟิลด์ที่ใช้ในการวิเคราะห์คือ ฟิลด์ CustomerName จากตารางไดเมนชั่น Customer, ProductName จากตาราง Product, ChannelName จากตาราง Channel และเลือกตัววัดทางธุรกิจคือยอดขาย (sales) เมื่อทำการ Query ข้อมูลที่ได้เลือกมา ก็จะแสดงผลในรูปของตารางซึ่งเป็นเป็น Cross-tabulation



รูปที่ 6.5 ฟอร์มในการวิเคราะห์ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฟอร์มที่ 5 เป็นฟอร์มที่ได้มาจากการคลิกที่ปุ่ม View เพื่อเลือกที่จะดูเฉพาะฟิลด์ของตารางไดเมนชันที่ต้องการ ดังรูปที่ 6.6 เป็นการเลือกดูเฉพาะฟิลด์ CustomerName ของตาราง Customer ผู้ใช้สามารถเพิ่มหรือลดฟิลด์ที่ต้องการวิเคราะห์ได้จากฟอร์มนี้ ซึ่งวิธีการเช่นนี้จะเรียกว่าการ Drill up/Drill down ข้อมูลนั่นเอง

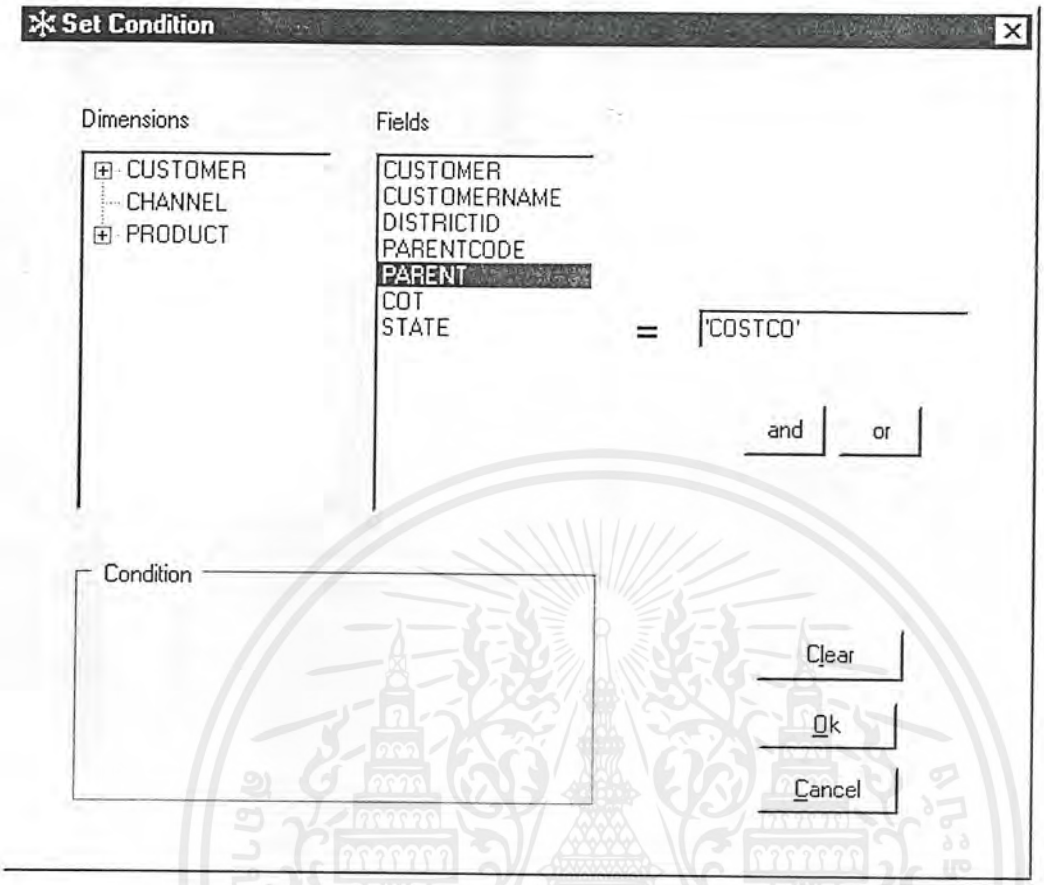


รูปที่ 6.6 ฟอร์มในการเลือกชื่อฟิลด์ที่ต้องการวิเคราะห์

- ฟอร์มที่ 6 เป็นฟอร์มที่ได้มาจากการคลิกที่ปุ่ม Condition จากรูปที่ 6.5 ซึ่งเป็นการเพิ่มเงื่อนไขในการวิเคราะห์ข้อมูลให้กับแต่ละแถวของตารางไดเมนชัน โดยเงื่อนไขที่ทำการสร้างขึ้นนี้จะต้องนำไปสร้างเป็น SQL Script ในส่วนของ Where clause รวมกับข้อมูลที่ได้ทำการเลือกไว้ข้างต้น จากนั้นก็ทำการ Query ข้อมูลอีกครั้งหนึ่ง เพื่อแสดงผลออกทางตาราง จากรูปที่ 6.7 เป็นการใส่เงื่อนไขให้กับฟิลด์ Parent ในตารางไดเมนชัน Customer ให้มีค่าเท่ากับ Costco ดังนั้น SQL ที่ผลิตได้จะต้องเพื่อ Where clause เข้าไป ดังนี้

```
.....and Customer.parent = 'COSTCO'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.7 ฟอร์มแสดงเงื่อนไขที่กำหนดให้กับแต่ละฟิลด์ของตารางไดเมนชัน

6.3 คอมโพเนนท์ใน Delphi4.0 ที่นำมาใช้ในการสร้างโปรแกรมประยุกต์ทางด้านฐานข้อมูล

Delphi4.0 จะมีคอมโพเนนท์ต่างๆ ที่สามารถนำมาใช้ในการพัฒนาโปรแกรมประยุกต์ทางด้านฐานข้อมูล ซึ่งมีคอมโพเนนท์ที่สำคัญดังนี้

6.3.1 DBGrid : เป็นเหมือนตารางสำหรับแสดงและแก้ไขข้อมูลในฐานข้อมูล

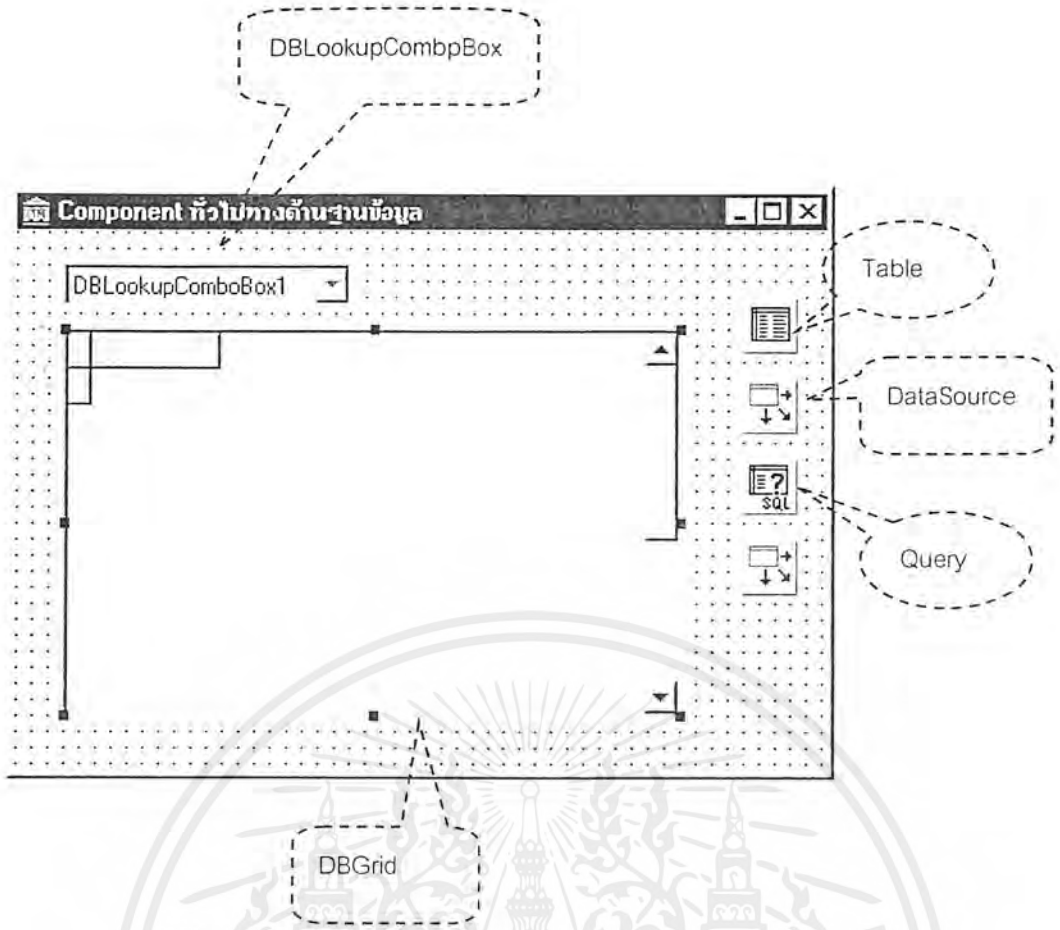
6.3.2 DBLookupComboBox : เป็นคอมโพเนนท์ที่สามารถแสดงรายการข้อมูล ซึ่งรายการข้อมูลที่แสดงในลิสต์จะมาจากข้อมูลในฐานข้อมูล

6.3.3 Table : เป็นคอมโพเนนท์ที่ใช้ในการเข้าถึงตารางฐานข้อมูลโดยตรง โดยสามารถแก้ไข ค้นหา คัดลอก และลบข้อมูลในตารางได้

6.3.4 Query : เป็นการเข้าถึงข้อมูลที่สำคัญอีกแบบหนึ่งโดยจะทำผ่านคำสั่งภาษา SQL ทั้งคำสั่งเพื่อดึงข้อมูล (Select) และคำสั่งเพื่อแก้ไขข้อมูล (Insert, Update, Delete)

6.3.5 DataSource : เป็นตัวกลางในการเชื่อมต่อระหว่าง data set (Table หรือ Query) กับคอมโพเนนท์ประเภท Data Control (เช่น DBGrid, DBEdit) ที่ใช้สำหรับการแสดงหรือแก้ไขข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.8 แสดง component ต่างๆ ที่ใช้ในการจัดการด้านฐานข้อมูล

6.4 คอมโพเนนท์พิเศษที่ใช้สำหรับ Multidimensional Database จะประกอบด้วย

6.4.1 DecisionCube : เป็นคอมโพเนนท์ที่จัดการข้อมูลให้อยู่ในรูปของ cross-tabulate เมื่อทำการวางออบเจ็ค decisioncube ลงบน form หรือ data module เพื่อให้โปรแกรมประยุกต์สามารถนำเสนอข้อมูลที่เป็น crosstabulate ได้ Descisioncube จะวิเคราะห์ข้อมูลจาก dataset ในโครงสร้างแบบ multidimensional ซึ่งแต่ละ dimension ก็คือ field ใน dataset ซึ่งเป็นข้อมูลที่ถูกสรุปมา dataset ใดๆ สามารถที่จะใช้เป็น source ของออบเจ็ค decisioncube ก็ได้ แต่ decision query จะดีที่สุด

โดยปกติค่า summary จะถูกคำนวณโดย dataset ซึ่งเป็นแหล่งข้อมูลของ DecisionCube อย่างไรก็ตามสามารถคำนวณค่าเฉลี่ยภายในโดย DecisionCube ได้ ซึ่งมีข้อดีคือสามารถวิเคราะห์ให้เป็น pivot, subtotal และ drill-in ได้โดยมี decision grid เป็นคอมโพเนนท์ที่จะแสดงข้อมูลจาก Decision Cube

TDecisionCube มี property และ method ที่อนุญาตให้ข้อมูลที่เก็บอยู่ในรูปของ crosstabulated เป็นตัวจัดหาข้อมูลให้กับออบเจ็ค DecisionSource, DeciosionGrid และ DecisionGraph ซึ่งคอมโพเนนท์เหล่านี้จะไม่ดึงข้อมูลจาก DecisionCube โดยตรงแต่มันจะใช้ออบเจ็ค DecisionSource ซึ่งจะเป็นตัวที่นำเสนอ configuration ของ dimension จากออบเจ็ค DecisionCube

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Property ของ DecisionCube มีดังนี้

- TDecisionCube.Active : เมื่อ set ให้ Active เป็น true Decision Cube ก็จะ fetch และ analyze ข้อมูลจาก dataset ที่ถูกระบุใน dataset property และเมื่อ Active เป็น false Decision Cube จะถูกปิด และ DecisionSource จะไม่สามารถอ่านข้อมูลจาก DecisionCube ได้ รวมทั้ง active property จะไม่สามารถ set ให้เป็น true ได้ ถ้า dataset ที่ระบุใน dataset property ยังไม่ Active

- TDecisionCube.DesignState : ระบุว่ามีข้อมูลปริมาณเท่าไรที่จะให้ Decision Cube แสดงในตอนออกแบบ เราสามารถ set design state เพื่อจำกัดปริมาณข้อมูลเมื่อ Decision Cube กำลัง active ในขณะที่กำลังออกแบบ การสร้าง cache ของ Decision Cube จะทำให้สิ้นเปลืองเวลา และ memory การจำกัดข้อมูลที่ DecisionCube จะใช้ในขณะออกแบบจะทำให้ preference ดีขึ้นเมื่อ Decision Cube มีหลาย dimension หรือเมื่อ แต่ละ dimension มี value จำนวนมากๆและ value ของ design state สามารถที่จะเปลี่ยนแปลงได้ในขณะออกแบบโดยการใช้ Decision Cube editor

- TCustomDataStore.BinData : แสดง dimension ทั้งหมดใน multidimensional array รวมทั้ง Cell หนึ่งๆ สำหรับทุกๆ ค่าตาม dimension การอ่าน Bindata ก็เพื่อดูว่า active dimension ใดถูก group เป็นช่วงหรือถูก drill อย่างถาวร เมื่อ Bindata เป็นจริง จะมีอย่างน้อย 1 dimension ที่ถูกยุบให้เป็นช่วงของ value เมื่อ BinData เป็นเท็จทุก dimension รวมทั้ง entry สำหรับค่าของ field ของ dimension จะปรากฏใน dataset

- TCustomDataStore.Capacity : เป็นการจำกัดหน่วยความจำที่อาจจะถูกใช้โดย cached array ของค่า summary ทำการกำหนดค่า capacity เพื่อจำกัดจำนวนไบท์ที่อาจจะถูกใช้โดย crosstabulated data store สำหรับเก็บ multidimension array ของค่า summary ถ้าจำนวนของ active dimension และ summary ทำให้ cache เกินค่านี้ ก็จะทำให้เกิด ElowCapacityError

- TCustomDataStore.CurrentSummary : ระบุอินเด็กซ์ของ summary ปัจจุบันจาก set ของ active summary ที่ถูกเก็บเอาไว้โดย crosstabulated data store

- TCustomDataStore.Dataset : กำหนด dataset เพื่อให้ dataset ซึ่งบรรจุ crosstabulated data สำหรับเก็บข้อมูล โดยปกติ dataset จะเป็นของออบเจ็ค TdecisionQuery ออบเจ็ค TdecisionQuery เป็น SQL Query ที่ถูก group อย่างเหมาะสมเพื่อคำนวณ summary สำหรับ crosstabulated data store คุณสมบัติ dataset จะสามารถ กำหนดให้เป็นออบเจ็ค TQuery ซึ่งมีรูปแบบ SQL เหมือนกับของ TdecisionQuery หรืออาจเป็น dataset อื่นๆ ที่บรรจุค่า summary ในโครงสร้างเดียวกันกับผลลัพธ์ที่ได้จาก Query ของ TdecisionQuery ถ้าคุณสมบัติ dataset เป็นออบเจ็ค TQuery หรือ TdecisionQuery crosstabulated data store จะตัดสินใจโดยอัตโนมัติว่า field ไหนเป็น dimension และ field ไหนเป็น summary ถ้าคุณสมบัติ dataset ไม่ใช่ Query field ใน dimension Map จะต้องถูกกำหนดให้ชัดเจนว่าเป็น dimension หรือ summary และถ้ามันนำเสนอ summary ก็จะต้องกำหนดชนิดของ summary ด้วย สำหรับค่า summary สำหรับ crosstabulated data store จะต้องถูกคำนวณโดยออบเจ็ค TDataSet data store จะไม่สรุปข้อมูลดิบ แต่จะวิเคราะห์ข้อมูลที่สรุปแล้วใน dimension

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- TcustomDataStore.DimensionCount : ระบุจำนวนของ dimension ที่ถูก cache โดย crosstabulated data store
- TCustomDataStore.DimensionMap : อธิบาย dimension และ summary ที่เก็บไว้โดย crosstabulated data store
- TCustomDataStore.DimensionMapCount : ระบุจำนวนของ field ใน dataset ซึ่งกำหนดโดยคุณสมบัติ Dataset ร่วมกับจำนวนของ field ที่ derive มาจาก field เหล่านั้น เช่น field สำหรับค่าเฉลี่ย ซึ่ง derive มาจาก field count และ summary ค่าของ DimensionMapCount จะไม่เหมือนกับ DimensionCount + SummaryCount DimensionCount และ SummaryCount จะนับจำนวนของ dimension และ summary ซึ่งจะถูกเก็บไว้โดย crosstabulated data cube และจะมีข้อมูลให้กับ DecisionSource DimensMapcount จะนับ dimension หรือ summary ที่ไม่ active ซึ่งจะไม่รวมอยู่ใน cache
- TCustomDataStore.MaxCell : จำกัดจำนวนของ Cell ซึ่งสามารถจะถูกรวมใน cache ที่เวลาใดเวลาหนึ่งได้ ใช้ MaxCell เพื่อจำกัดการใช้หน่วยความจำของ crosstabulated data store
- TCustomDataStore.MaxDimension : จำกัดจำนวนของ dimension ที่สามารถ cache โดย crosstabulated data store ที่เวลาใดเวลาหนึ่งได้
- TCustomDataStore.MaxSummaries : จำกัดจำนวนของ summary ที่สามารถถูก cache โดย crosstabulated data store ที่เวลาหนึ่ง
- TCustomDataStore.ShowProgressDialog : ใช้พิจารณาว่าจะให้ผู้ใช้เห็น progress bar ขณะที่ cache กำลังถูกสร้างหรือเปล่า
- TCustomDataStore.SummaryCount : กำหนดจำนวนของฟิลด์ summary ที่ถูก cache โดย crosstabulated data store
- TComponent.Name : กำหนดชื่อของ component ซึ่งถูกอ้างอิงเป็น code ใช้ Name เพื่อเปลี่ยนชื่อของ Component เพื่อสื่อถึงวัตถุประสงค์ใน application ปัจจุบัน โดย default เดลไฟล์จะกำหนดลำดับชื่อตามชนิดของ component เช่น 'Button1', 'Button2' เป็นต้น

Method ของ DecisionCube มีดังนี้

- TComponent.Tags : Tag ไม่ได้ถูกกำหนดความหมายไว้ล่วงหน้า คุณสมบัติ Tag มีไว้สำหรับอำนวยความสะดวกในการเก็บค่า integer เพิ่มเติมหรือข้อมูล pointer สำหรับความต้องการพิเศษใน application
- TDecisionCube.Create : สร้าง instance ของ TdecisionCube
- TDecisionCube.Destroy : ทำลาย instance ของ DecisionCube และ ห้าม call method destroy โดยตรง ต้อง Call free ก่อน โดย free จะรับรองว่า memory สำหรับ decision cube ยังไม่ว่าง แล้วค่อย call destroy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- TDecisionCube.GetDetailSQL : จะ return SQL statement ที่ถูกใช้ในการ สร้างข้อมูลของ decision cube
- TDesionCube.GetSQL : เหมือนกับ TDecisionCube.GetDetailSQL
- TDesionCube.ShowCubeDialog : แสดง Decision Cube Editor โดยจะทำการเรียก ShowDecisionCube Dialog เพื่ออนุญาตให้ผู้ใช้เปลี่ยนแปลง active dimension สำหรับ data cube โดยใช้ DecisionCube Editor รวมทั้งจะอนุญาตให้ผู้ใช้สามารถทำให้ dimension active หรือ inactive และเปลี่ยนแปลงชื่อของ dimension รวมทั้งมันยังสามารถควบคุมว่าจะให้ information ถูกแสดงใน decision control มากแค่ไหน ได้ในเวลาออกแบบ

Event ของ DecisionCube มีดังนี้

- TCustomDataStore.AfterClose : จะเกิดขึ้นทันทีหลังจาก crosstabulated data store ถูก deactive เรียก after close event handler เพื่อระบุ action เมื่อ crosstabulated data store ถูกปิด
- TCustomDataStore.AfterOpen : จะเกิดขึ้นทันทีหลังจาก crosstabulated data store ถูก activeเรียก after open event handler เพื่อระบุ active เมื่อ crosstabulated data store ถูกทำให้ active ซึ่งจะเป็นโอกาสแรกที่จะตรวจสอบว่า memory ถูกใช้ไปเท่าไร โดย multidimensional cache
- TCustomDataStore.BeforeClose : จะเกิดขึ้นก่อนที่ crosstabulated data store ถูก deactive เมื่อเขียน before close event handler เพื่อระบุ active เมื่อ crosstabulated data store จะปิด
- TCustomDataStore.BeforeOpen : เกิดขึ้นก่อนที่ crosstabulated data store จะถูก active เขียน before open handler เพื่อระบุ action ก่อน crosstabulated data store จะถูก active เป็นเวลาที่เหมาะสมกับการปรับค่า capacity หรือการกำหนด showProgressDialog ซึ่งจะขึ้นอยู่กับจำนวน active dimension ใน dimensionMap และเป็นโอกาสสุดท้ายที่จะปรับเปลี่ยน entry ใน dimensionMap ก่อนที่จะสร้าง cache ถ้า data store ไม่ถูก active ก่อน event นี้ DimensionMap property ก็จะเป็น nil BeforeOpen event handler สามารถสร้าง dimension map ถ้ามันเป็น nil ถ้า dimension map ไม่เป็น nil property ของ dimension item จะถูกเปลี่ยนแปลง แต่ order และจำนวนของ dimension item จะไม่ถูกเปลี่ยนแปลง
- TCustomDataStore.OnLowCapacity : จะเกิดเมื่อ multidimensional cache ต้องการ memory มากเกินกว่าที่มันได้รับจาก คุณสมบัติ capacity โดยจะเขียน OnLowCapacity event handler เพื่อจะตอบสนองกับ ElowCapacityError expression OnLowCapacity event handler สามารถเปลี่ยนค่าของ Capacity หรือ Deactive บาง dimension ใน dimension map เพื่อให้ cache พอดีกับ memory ที่มีอยู่ ถ้าปัญหา memory ถูกจัดการโดย OnLowCapacity event handler ก็จะมีการเปลี่ยน Action parameter ไปเป็น eaContinue เพื่ออนุญาตให้ data store สามารถสร้าง cache ต่อไปได้ มิฉะนั้น ElowCapacityError exception ก็จะเกิดขึ้น
- TCustomDataStore.OnRefresh : เกิดขึ้นทันทีก่อนที่ dimension map จะถูกเปลี่ยนแปลง ใช้ OnRefresh event handler เพื่อเปลี่ยนแปลง property ของ dimension item ก่อนที่ dimension Map property

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะถูก set และ cache จะถูกสร้างใหม่ ซึ่งจะเป็นโอกาสที่จะ reset Maximum Dimension หรือ Max Summary property

6.4.2 DecisionSource : เป็นตัวระบุสถานะปัจจุบันของ decision grid หรือ decision graphs โดยมี property และ event ที่สำคัญที่จะเป็นตัวควบคุมลักษณะ และพฤติกรรมของ decision source ดังต่อไปนี้

Property ของ DecisionSource มีดังนี้

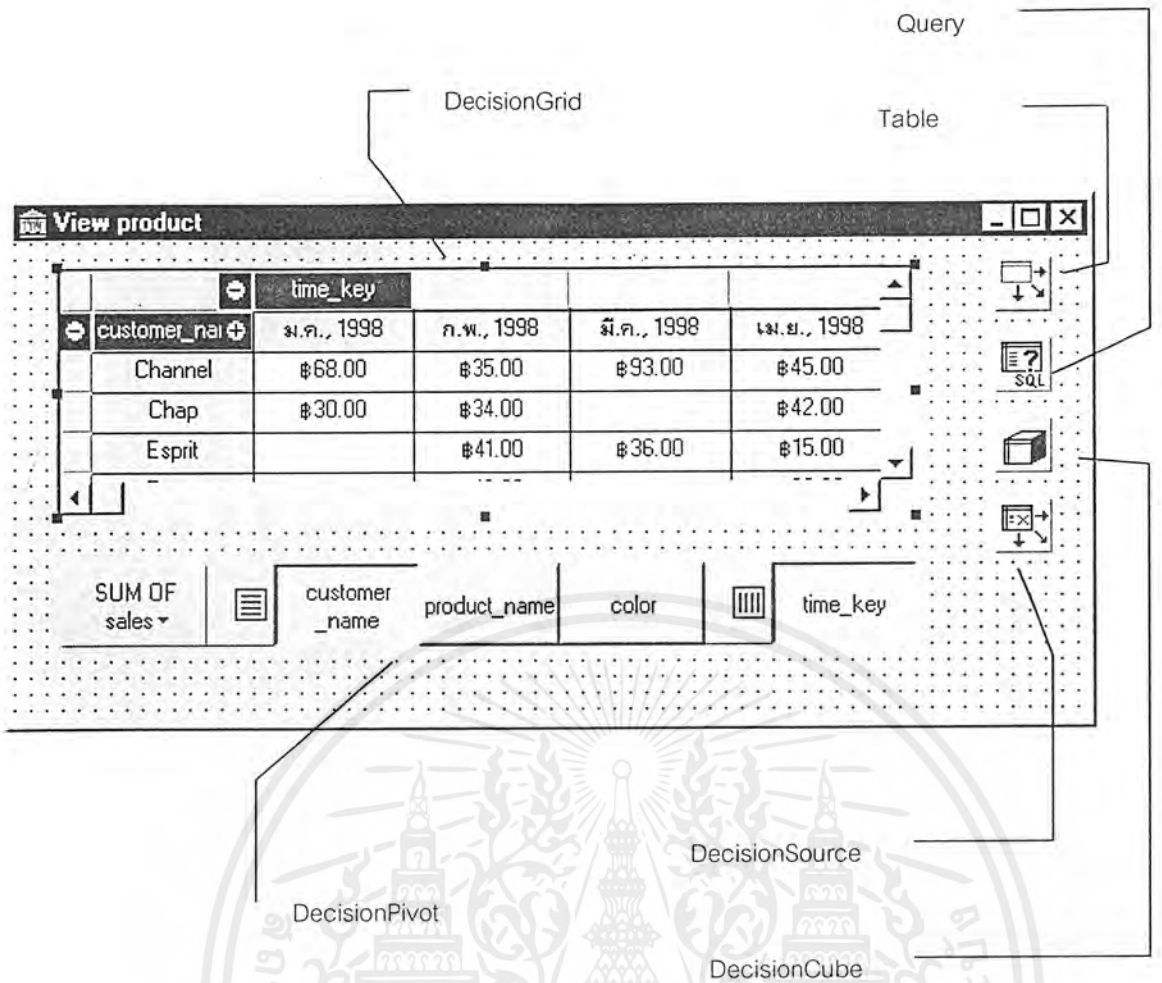
- ControlType property ของ TDecisionSource จะเป็นตัวระบุว่าปุ่มของ decision pivot ควรจะมีลักษณะเป็น check boxes (เลือกได้ครั้งละหลายๆ อัน) หรือว่า radio buttons (เลือกได้อย่างใดอย่างหนึ่ง)
- SparseCols and SparseRows properties ของ TDecisionSource ระบุว่าถ้าคอลัมน์และแถวที่แสดงไม่มีข้อมูลเลยก็จะให้แสดงออกมาเป็นคอลัมภ์และแถวที่ว่างเปล่า

Event ของ DecisionSource มีดังนี้

- OnLayoutChange จะเกิดขึ้นเมื่อผู้ใช้เรียกใช้ pivot หรือทำการ drill down ข้อมูล
- OnNewDimensions จะเกิดขึ้นเมื่อข้อมูลถูกเปลี่ยนแปลง เช่น เมื่อมีการ summary หรือฟิลด์ของไดเมนชั่นถูกแก้ไข
- OnSummaryChange จะเกิดขึ้นเมื่อ summary ปัจจุบันถูกเปลี่ยนแปลง
- OnStateChange เกิดขึ้นเมื่อ Decision Cube activates หรือ deactivates
- OnBeforePivot เกิดขึ้นเมื่อการเปลี่ยนแปลงถูก commit แต่ยังไม่สะท้อนไปยังผู้ใช้ โดยผู้พัฒนาสามารถที่จะเปลี่ยนแปลง เช่น เปลี่ยนแปลงความจุ หรือสถานะของ pivot ก่อนที่ผู้ใช้จะเห็น
- OnAfterPivot จะเกิดขึ้นหลังจากเปลี่ยนสถานะของ pivot โดยผู้พัฒนาสามารถเก็บข้อมูลเอาไว้

6.4.3 DecisionGrid : เป็นคอมโพเนนท์ที่แสดงข้อมูล cross-tabulate จาก DecisionSource ในรูปแบบของตาราง (Grid) ซึ่งตารางนี้จะถูกเรียกว่า Crosstabs Decision grids จะแสดงค่าโดยสรุปของแต่ละไดเมนชั่น ฟิลด์แต่ละฟิลด์สามารถถูกสรุปได้ในหลายรูปแบบ โดยอาจทำการเปลี่ยนที่ dimension map การเปลี่ยนไดเมนชั่น และค่า summery จะทำให้เกิดการประมวลผลที่ decision grid โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

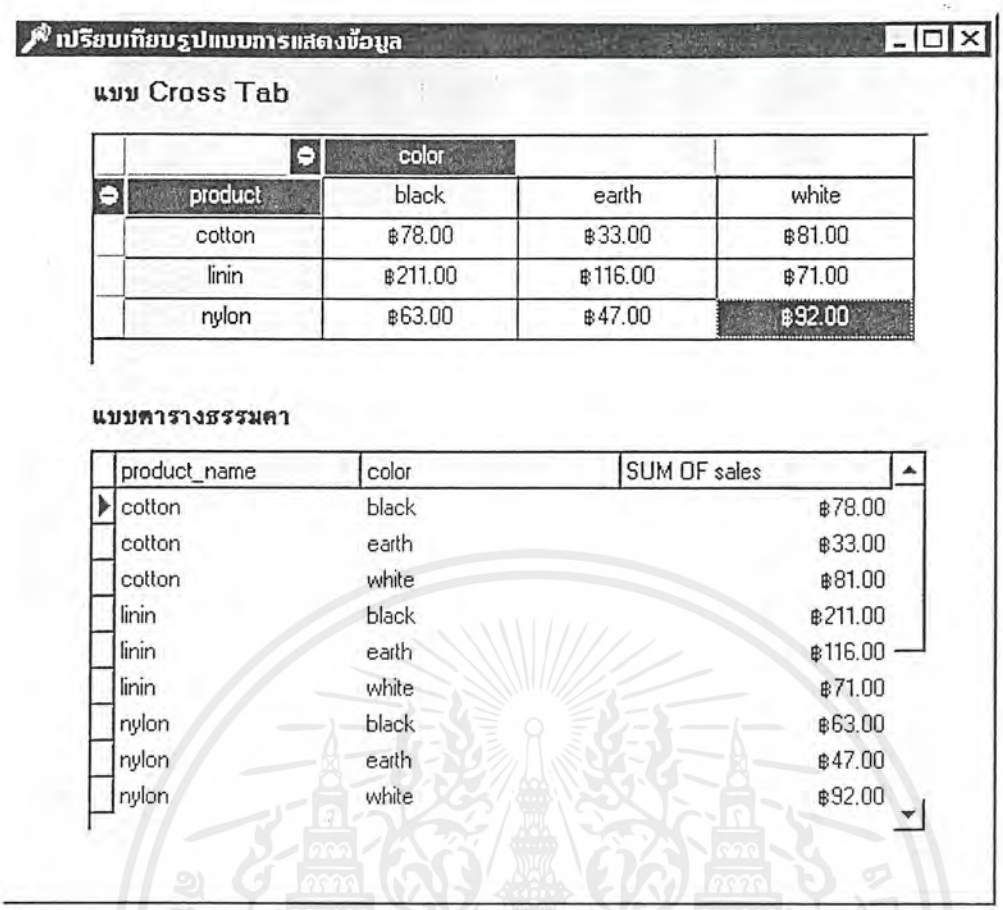


รูปที่ 6.9 แสดง component พิเศษที่ใช้สำหรับ Multidimensional Database

6.5 Cross-tabulations หรือ Crosstabs

CrossTabs เป็นวิธีในการแสดงกลุ่มของข้อมูลเพื่อให้มีความสัมพันธ์กันและสามารถมองเห็นแนวโน้มได้ง่าย ฟิลด์ของตารางไดเมนชันจะอยู่ในรูปของ Crosstabs และค่าในแต่ละฟิลด์จะถูกแบ่งกลุ่มและถูกสรุปภายในไดเมนชัน ทำให้สามารถมองได้ง่ายกว่าการแสดงแบบตารางปกติ ดังรูป 6.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.10 เปรียบเทียบการแสดงข้อมูลแบบตารางปกติ และแบบ CrossTabs

6.6 อธิบายส่วนต่างๆ ของโปรแกรมประยุกต์

โปรแกรมที่ได้ทำการออกแบบขึ้นมานั้นประกอบด้วยฟอร์มทั้งหมด 8 ฟอร์ม ดังนี้

- ฟอร์มรูปภาพที่ปรากฏครั้งแรกที่เปิดโปรแกรม
- ฟอร์มของ Main Menu
- ฟอร์มในการสร้างโมเดล
- ฟอร์มสร้างตารางไดเมนชัน และตารางแฟ็ก
- ฟอร์มสร้างตารางสโนว์เฟลกสำหรับแต่ละไดเมนชัน
- ฟอร์มที่ใช้ในการเลือกตารางไดเมนชันเพื่อวิเคราะห์ข้อมูล
- ฟอร์มสำหรับใส่เงื่อนไข
- ฟอร์มสำหรับแสดงข้อมูลในรูปของกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมโพเนนท์ต่างๆ ในแต่ละฟอร์มอธิบายได้ดังนี้

1. ฟอร์มก่อนเข้าสู่โปรแกรม เป็นฟอร์มที่แสดงรูปภาพ และชื่อผู้จัดทำโปรแกรม ผู้ใช้สามารถเลื่อนเมาส์ไปที่รูปภาพเพื่อเมาส์เป็นรูปมือแล้วคลิกเพื่อใช้เข้าสู่โปรแกรมได้ และหากต้องการออกจากโปรแกรมก็สามารถเลื่อนเมาส์ไปที่ Exit แล้วคลิกเพื่อออกจากโปรแกรม

รูปที่ 6.11 ฟอร์มแสดงการเข้าสู่โปรแกรม

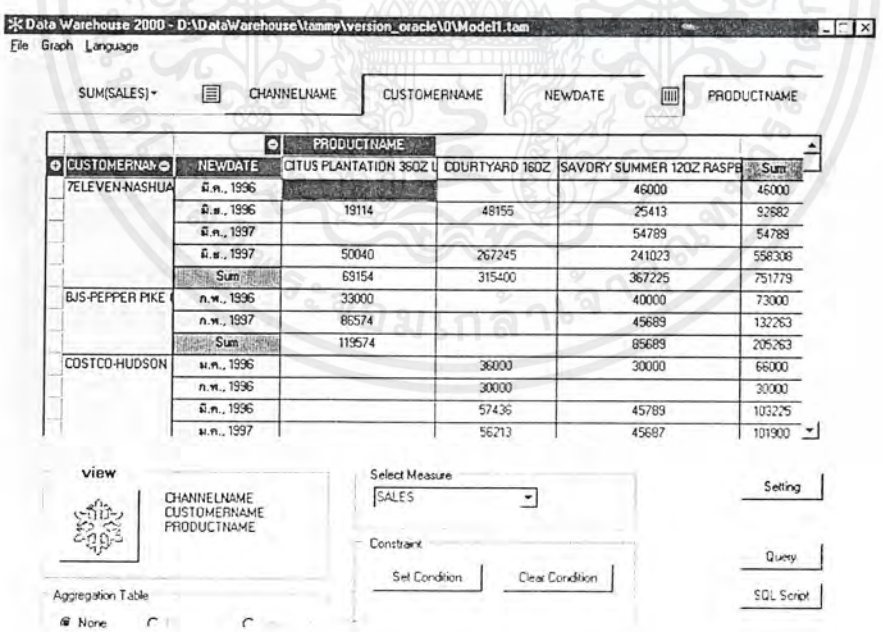
คอมโพเนนท์ที่ใช้ในฟอร์มนี้ได้แก่

- TImage : เป็นคอมโพเนนท์ที่ใช้ในการแสดงรูปภาพ
- GIFImage : เป็นคอมโพเนนท์ที่ใช้ในการแสดงรูปภาพสำหรับไฟล์ที่มีนามสกุล Gif ทำให้สามารถแสดงภาพเคลื่อนไหวได้
- TTimer : เป็นคอมโพเนนท์ที่ใช้ในการกำหนดระยะเวลาที่จะให้ตัวอักษรเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ฟอร์ม Main menu เป็นฟอร์มเริ่มต้นการใช้งานของโปรแกรม โดยจะมีเมนูอยู่ 3 เมนูคือ File, Graph และ Language ซึ่งฟอร์มนี้เป็นหน้าจอหลักในการทำงาน โดยเมนู File จะมีเมนูย่อยอีก 3 เมนูคือ Create Model, Load Model และ Exit ส่วน Language จะมีเมนูย่อย 2 เมนู คือ English และ Thai เมื่อเข้าสู่โปรแกรมครั้งแรกฟอร์มนี้จะมีลักษณะดังรูปที่ 6.12 แต่เมื่อต้องการจะวิเคราะห์โมเดล หรือทำการเลือกเมนู Load model ฟอร์มนี้จะปรากฏดังรูปที่ 6.13

รูปที่ 6.12 แสดงฟอร์มหลักในการทำงานของโปรแกรม



รูปที่ 6.13 แสดงฟอร์มในการวิเคราะห์ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมโพเนนท์ที่สำคัญของฟอร์มมีดังต่อไปนี้

- TPanel : เป็นคอมโพเนนท์ที่นำมาปิดคอมโพเนนท์ทั้งหมดที่มีอยู่ในฟอร์มนี้ และจะให้ค่า visible เป็น false เมื่อมีการเลือกเมนูเป็น Load Model นั่นคือทำให้ผู้ใช้เห็นคอมโพเนนท์ทั้งหมดเมื่อต้องการจะวิเคราะห์ข้อมูล
- TOpenDialog : เป็นไดอะลอกที่ใช้สำหรับเลือกไฟล์ซึ่งจะแสดงรายชื่อไฟล์และโฟล์เดอร์ปรากฏขึ้นมา โดยสามารถเรียกใช้ OpenDialog ได้จากเมดถอด Execute
- TMainMenu : ใช้สำหรับแสดงคำสั่งการทำงานที่มีอยู่ในโปรแกรม
- TTable : เป็นคอมโพเนนท์ที่ใช้ในการเข้าถึงตารางฐานข้อมูลโดยตรง โดยสามารถแก้ไข ค้นหา คัดลอก และลบข้อมูลในตาราง สามารถกำหนดให้มีการกรองเอาเฉพาะข้อมูลที่ต้องการ และสามารถกำหนดความสัมพันธ์แบบ Master/Detail กับตารางอื่นได้
- TQuery : เป็นการเข้าถึงข้อมูล โดยการเข้าถึงข้อมูลด้วยวิธีนี้ จะผ่านคำสั่งภาษา SQL ทั้งคำสั่งเพื่อดึงข้อมูล (Select) และคำสั่งเพื่อแก้ไขข้อมูล (Insert, Update, Delete) การใช้ Query สามารถเข้าถึงข้อมูลจากตารางฐานข้อมูลได้มากกว่า 1 ตาราง หรือสามารถเข้าถึงเพียงบางส่วนของข้อมูลในตารางได้
- TDecisionCube : เป็นคอมโพเนนท์ที่ใช้ในการจัดการข้อมูลให้อยู่ในรูปของ Cross-tabulation
- TDecisionSource : เป็นตัวกลางในการเชื่อมต่อระหว่าง data set (Table หรือ Query) กับ DecisionCube
- TDecisionPivot : เป็นคอมโพเนนท์ที่แสดงชื่อฟิลด์ทั้งหมดที่ได้ทำการเลือกมา ผู้ใช้สามารถปิด-เปิดชื่อฟิลด์ที่ต้องการดูได้โดยใช้ DecisionPivot ในการขับเคลื่อน
- TDecisionGrid : เป็นคอมโพเนนท์ที่แสดงข้อมูล cross-tabulate จาก DecisionSource ในรูปแบบของตาราง (Grid)
- TButton : เป็นคอมโพเนนท์ที่ใช้ในการติดต่อกับผู้ใช้งาน ซึ่งจะเป็นปุ่มให้คลิก และเมื่อถูกคลิกก็จะเกิดอีเวนต์ OnClick ขึ้น ซึ่งในโปรแกรมนี้จะมีคอมโพเนนท์ Button อยู่หลายอันด้วยกัน เช่น Button Setting, Query, View, Show SQL ฯลฯ
- TLabel : เป็นคอมโพเนนท์ที่ใช้สำหรับแสดงข้อความบนโปรแกรมประยุกต์ คอมโพเนนท์ชนิดนี้จะไม่สามารถรับ Input ได้โดยตรง แต่ใช้สำหรับทำการ setfocus ไปยังคอมโพเนนท์อื่นๆ ได้โดยคุณสมบัติ FocusControl สำหรับในโปรแกรมนี้คอมโพเนนท์ Label ได้แก่ Measure, CustomerName ฯลฯ
- TComboBox : เป็นคอมโพเนนท์ที่สามารถเรียก List ออกมาเลือกได้ เมื่อเลือกเสร็จแล้วข้อความที่อยู่ใน List จะถูกนำมาใส่ที่ Edit Box โดยที่จะเป็นการรวมข้อดีระหว่างคอมโพเนนท์ List Box กับ Edit คือสามารถแสดงข้อความได้ทีละหลายๆ ข้อความและในขณะเดียวกันก็ใช้เนื้อที่น้อยกว่า List Box ซึ่งในโปรแกรมที่ได้ทำการออกแบบไว้นี้ ใช้คอมโพเนนท์ ListBox ในการแสดง Fact Field โดยให้ผู้ใช้เลือกว่าจะเอาฟิลด์ของตารางแฟ็กตัวไหนที่จะนำมาใช้ในการวิเคราะห์ เช่นยอดขาย รายจ่าย ฯลฯ

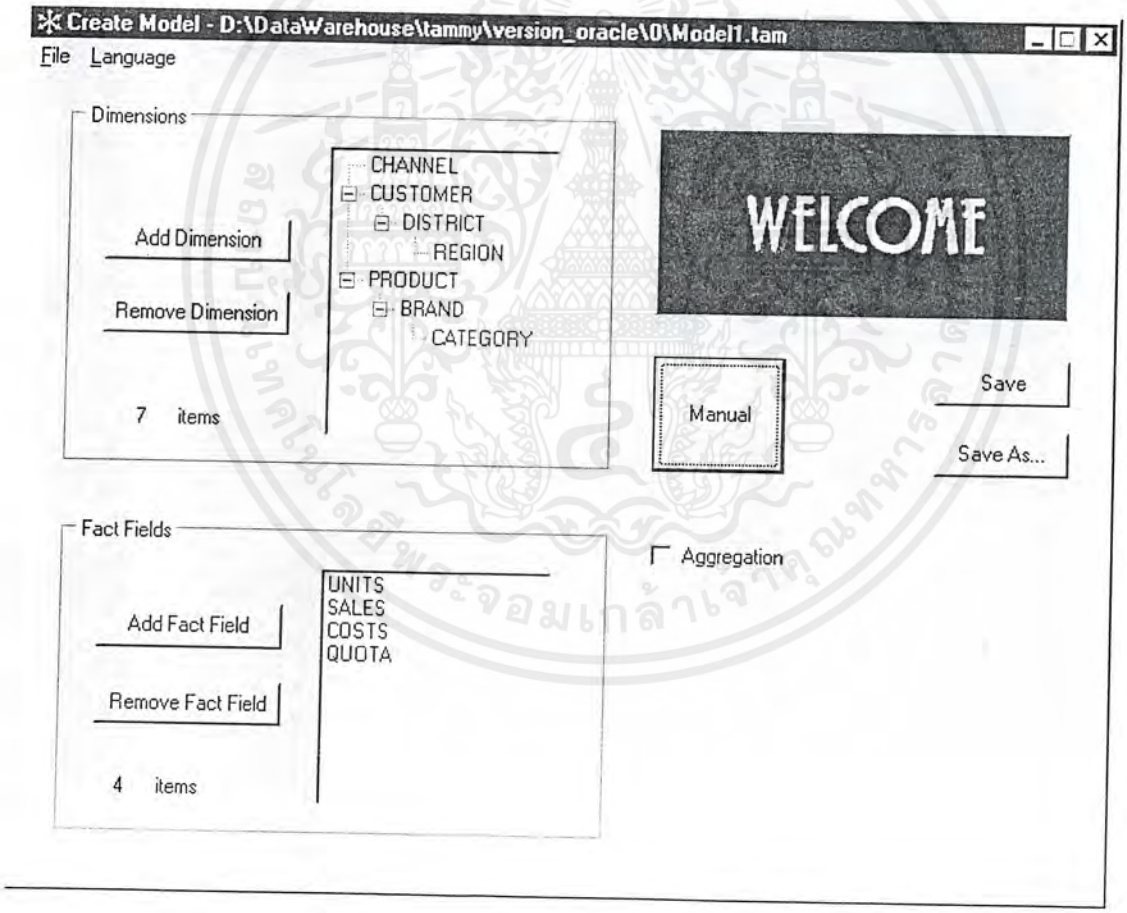
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- TMenuItem : เป็นคอมโพเนนท์ที่เก็บ Item ของ MainMenu ซึ่งจากโปรแกรมนี้ MenuItem ก็จะมี File, Graph และ Language นั่นเอง

- TRadioButton : เป็นคอมโพเนนท์ที่ใช้สำหรับสร้างระบบติดต่อกับผู้ใช้ โดยเป็นการให้ผู้ใช้สามารถเลือกตัวเลือกได้เพียงครั้งละตัว ซึ่งในที่นี้จะเป็นระบบติดต่อกับผู้ใช้โดยให้ผู้ใช้สามารถกำหนด Time frame ในกรณีที่ต้องการวิเคราะห์จากตารางที่ aggregate ไว้แล้วได้ ซึ่งจะให้เลือกตามรายเดือน หรือรายปี

- TGroupBox : เป็นคอมโพเนนท์ที่สร้างขึ้นเพื่อกำหนดกลุ่มของคอมโพเนนท์ที่มีความสัมพันธ์กัน ทำงานร่วมกัน นอกจากนี้แล้วยังใช้ในการตกแต่งฟอร์มเพื่อเพิ่มความสวยงามให้โปรแกรม

3. ฟอร์มการสร้างโมเดล เป็นฟอร์มที่จะปรากฏขึ้นเมื่อผู้ใช้เลือกเมนู File>Create Model โดยฟอร์มนี้จะให้ผู้ใช้ทำการสร้างโมเดลจากตารางที่มีอยู่ในระบบจัดการฐานข้อมูลอยู่แล้ว โดยผู้ใช้ต้องทำการเชื่อมต่อกับ Alias ที่ได้กำหนดไว้ ฟอร์มนี้จะมีลักษณะดังรูปที่ 6.14



รูปที่ 6.14 แสดงฟอร์มในการสร้างโมเดล

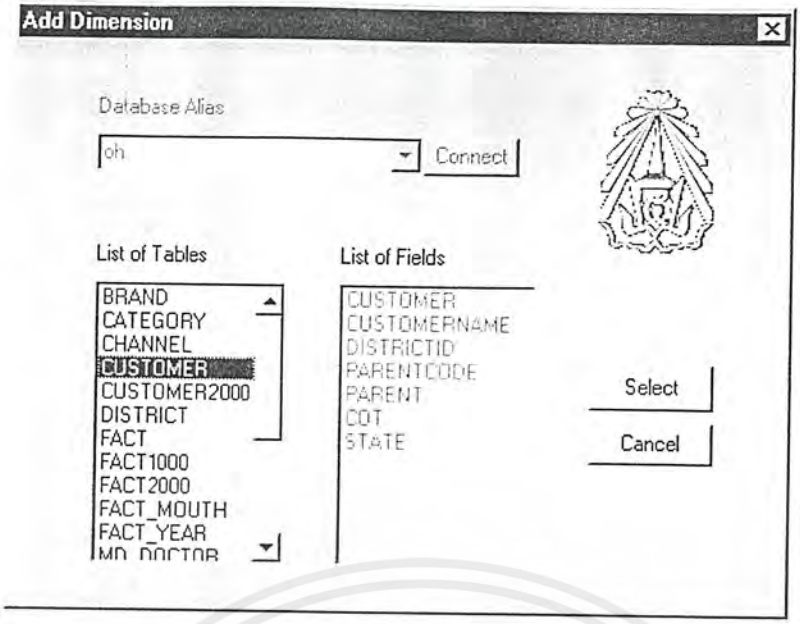
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมโพเนนท์ที่สำคัญของฟอร์มมีดังต่อไปนี้

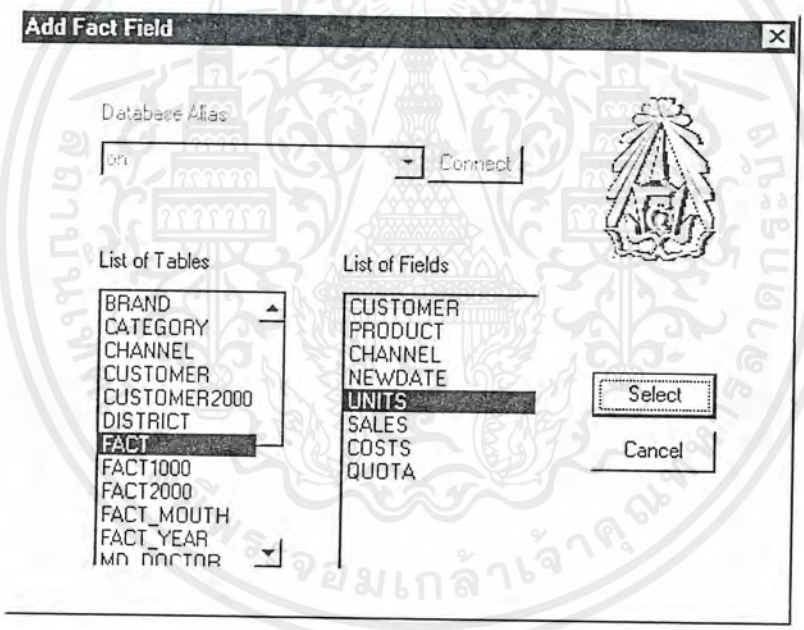
- TTreeView : เป็นคอมโพเนนท์ที่ใช้แสดงข้อมูลที่ความสัมพันธ์ในลักษณะของ Tree ที่ทำเช่นนี้เนื่องมาจากโปรแกรมที่ได้ทำการออกแบบขึ้นสามารถสร้างโมเดลที่รองรับฐานข้อมูลแบบสโนว์เฟลกได้ ซึ่งโปรแกรมจะทำการมองตารางเป็นลำดับขั้น โดยดูจากฟิลด์ที่มีชื่อเหมือนกันก็จะเป็นตารางที่สัมพันธ์กัน
- TPanel : เป็นคอมโพเนนท์ที่ใช้ในการตกแต่งฟอร์ม และใช้ในการจัดวางคอมโพเนนท์ให้อยู่ภายใน Panel ให้เป็นระเบียบ
- TGroupBox :เป็นคอมโพเนนท์ที่สร้างขึ้นเพื่อกำหนดกลุ่มของคอมโพเนนท์ที่มีความสัมพันธ์ ทำงานร่วมกัน และใช้ตกแต่งฟอร์มให้มีความสวยงาม
- TButton : เป็นคอมโพเนนท์ที่ใช้ในการติดต่อกับผู้ใช้งาน ซึ่งจะเป็นปุ่มให้คลิก และเมื่อถูกคลิกก็จะเกิดอีเวนต์ OnClick ขึ้น ซึ่งในโปรแกรมนี้จะมีคอมโพเนนท์ Button อยู่หลายอันด้วยกัน เช่น Button Add Dimension, Add Fact Field, Remove Dimension, Remove Fact Field ฯลฯ
- TMainMenu : ใช้สำหรับแสดงคำสั่งการทำงานที่มีอยู่ในโปรแกรม
- TOpenDialog : เป็นไดอะลอกที่ใช้สำหรับเลือกไฟล์ซึ่งจะแสดงรายชื่อไฟล์และโฟล์เดอร์ปรากฏขึ้นมา โดยสามารถเรียกใช้ OpenDialog ได้จากเมดถอด Execute
- TSaveDialog : จะปรากฏหน้าต่างที่ใช้ในการบันทึกไฟล์
- TCheckBox : เป็นคอมโพเนนท์ที่ใช้สำหรับสร้างระบบติดต่อผู้ใช้ โดยผู้ใช้สามารถเลือกตัวเลือกได้หลายๆ ตัวเลือก (เหมือนเป็น Option ที่จะเลือกหรือไม่เลือกก็ได้) ในที่นี้จะเป็นระบบติดต่อกับผู้ใช้โดยให้ผู้ใช้ระบุว่าจะมีการเลือกตารางที่ทำ aggregate ไว้แล้วหรือไม่
- GIFImage : เป็นคอมโพเนนท์ที่ใช้ในการแสดงรูปภาพสำหรับไฟล์ที่มีนามสกุล Gif ทำให้สามารถแสดงภาพเคลื่อนไหวได้

4. ฟอร์มในการเพิ่มตารางไดเมนชัน และฟิลด์ของตารางแฟ็ก เป็นฟอร์มที่สร้างขึ้นมาจากการคลิกที่ปุ่ม Add Dimension และ Add Fact Field โดยที่ฟอร์มนี้จะตัวติดต่อกับระบบจัดการฐานข้อมูลให้แล้วเลือกชื่อตารางต่างๆ ที่มีอยู่ในฐานข้อมูลนั้นขึ้นมาแสดงเพื่อให้ผู้ใช้ได้เลือก โดยรูปที่ 6.15 เป็นรูปที่แสดงการเพิ่มตารางไดเมนชัน ส่วนรูปที่ 6.16 จะแสดงการเพิ่มตัววัดทางธุรกิจหรือ ฟิลด์ในตารางแฟ็กนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.15 แสดงฟอร์มในการเพิ่มตารางไดเมนชัน



รูปที่ 6.16 แสดงฟอร์มในการเพิ่มฟิลด์ในตารางแฟ็ก

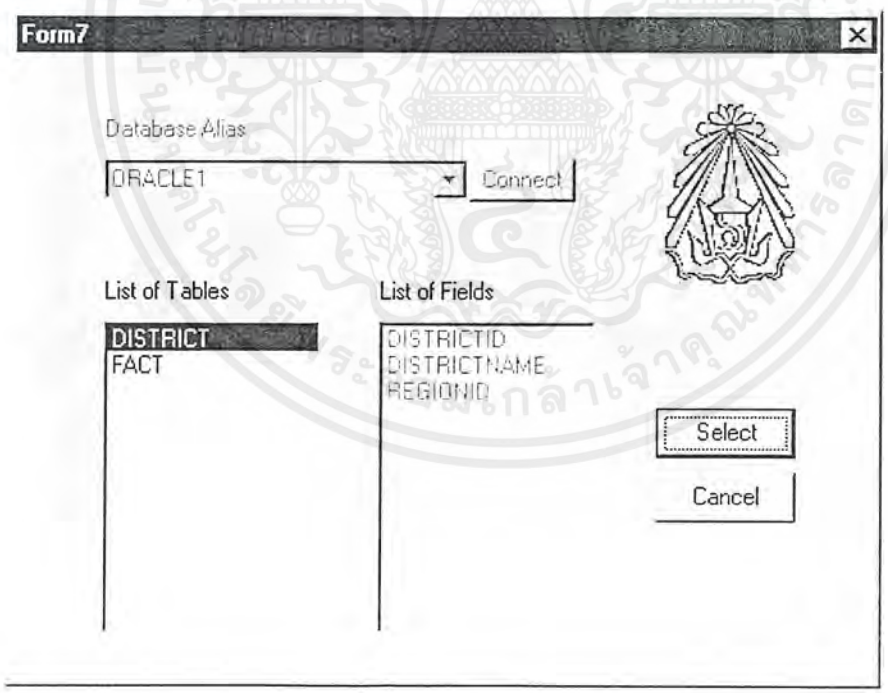
คอมโพเนนท์ที่สำคัญมีดังนี้

- TComboBox : ใช้เป็นตัว List ชื่อ Alias ของฐานข้อมูลเพื่อให้ผู้ใช้สามารถติดต่อกับฐานข้อมูลที่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- TListBox : ใช้ในการแสดง ชื่อตาราง และชื่อฟิลด์ทั้งหมดภายในตารางที่ถูกเลือก โดยจะใช้ TQuery เป็นตัวดึงข้อมูลออกมาจากฐานข้อมูล
- TButton : เป็นปุ่มสำหรับการ Select, Cancel และ Connect
- TLabel : เป็นการแสดงชื่อของคอมโพเนนท์ต่างๆ ในฟอร์ม เช่น List of Tables
- TQuery : เป็นตัวเชื่อมต่อกับฐานข้อมูลโดยใช้ภาษา SQL ในการดึงข้อมูลขึ้นมา
- TDataSource : เป็นตัวกลางในเชื่อมต่อระหว่างฐานข้อมูลกับ TQuery
- GIFImage : เป็นคอมโพเนนท์ที่ใช้ในการแสดงรูปภาพสำหรับไฟล์ที่มีนามสกุล Gif ทำให้สามารถแสดงภาพเคลื่อนไหวได้

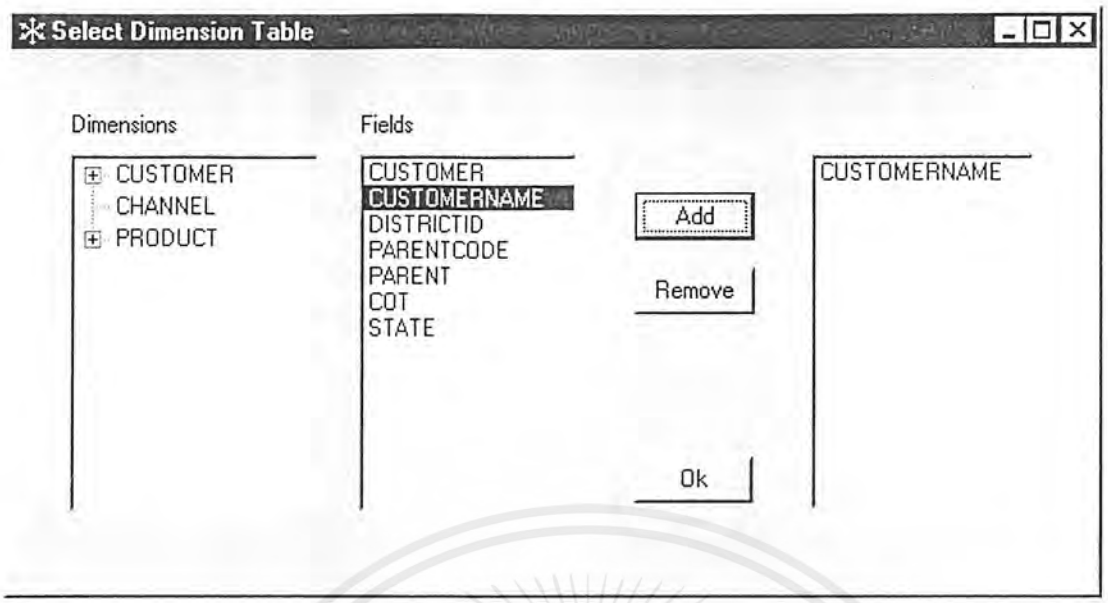
5. ฟอร์มสร้างตารางสโนว์เฟลกสำหรับแต่ละไดเมนชัน เป็นฟอร์มที่มีคอมโพเนนท์เหมือนกับฟอร์มในการเพิ่มตารางไดเมนชั่น และฟอร์มในการเพิ่มฟิลด์ในตารางแฟ็กทุกประการ เพียงแต่เปลี่ยนชื่อให้เป็น Add Snowflake Table to - เท่านั้น เนื่องจากโปรแกรมนี้รองรับฐานข้อมูลที่มีลักษณะเป็นแบบสโนว์เฟลก จึงจำเป็นจะต้องมีฟอร์มนี้ขึ้นมา เพื่อทำให้ผู้ใช้สะดวกในการใช้งานมากยิ่งขึ้น โดยเมื่อมีการดับเบิ้ลคลิกที่ชื่อของตารางไดเมนชันใด ก็จะเข้าสู่ฟอร์มนี้ และจะปรากฏตารางย่อยซึ่งมีคีย์ที่มีชื่อเหมือนกันกับตารางหลัก จึงทำให้ผู้ใช้สามารถสร้างตารางย่อย ได้อย่างสะดวกและง่ายดาย ส่วนตารางย่อยที่แตกย่อยลงไปอีก โปรแกรมจะเป็นตัวหาให้โดยอัตโนมัติ ตัวอย่างเช่นรูปที่ 6.17 เป็นการดับเบิ้ลคลิกที่ Customer และตารางย่อยที่มีชื่อคีย์เหมือนกับตาราง Customer ก็จะมีตารางแฟ็ก และตาราง District เท่านั้น



รูปที่ 6.17 แสดงฟอร์มในการสร้างตารางสโนว์เฟลกสำหรับแต่ละไดเมนชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ฟอร์มที่ใช้ในการเลือกตารางไดเมนชันเพื่อวิเคราะห์ข้อมูล



รูปที่ 6.18 แสดงฟอร์มในการเลือกตารางไดเมนชันเพื่อการวิเคราะห์ข้อมูล

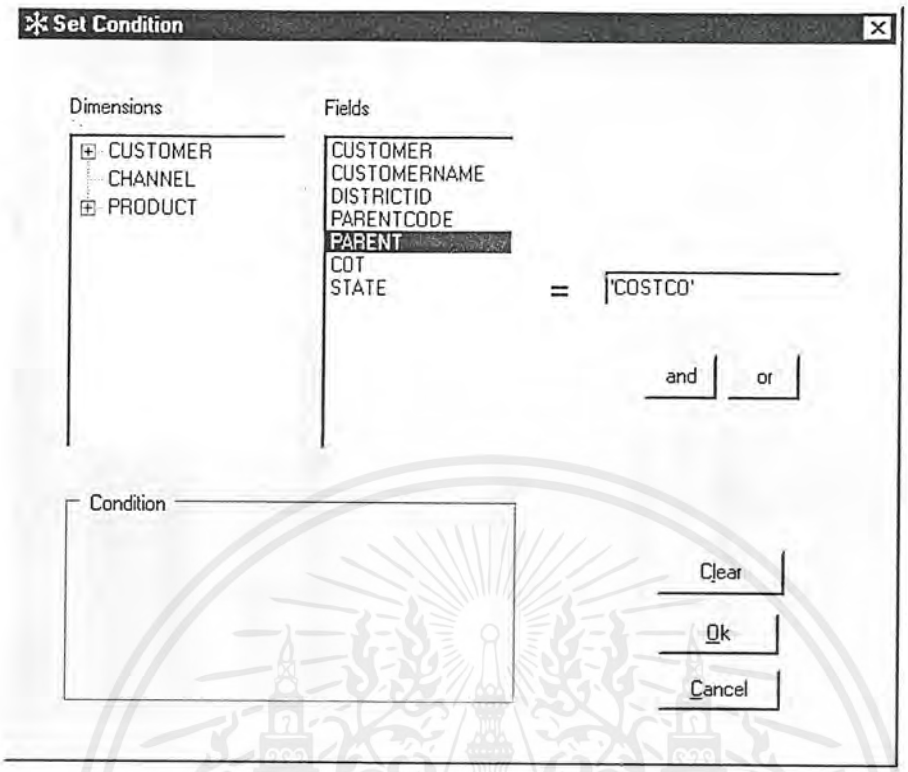
ฟอร์มนี้จะถูกเรียกใช้เมื่อมีการคลิกที่ปุ่ม View ของฟอร์มหลัก เพื่อเลือกฟิลด์ของตารางไดเมนชั่นที่จะใช้ในการวิเคราะห์ข้อมูล

คอมโพเนนท์ที่สำคัญมีดังต่อไปนี้

- TTreeView : เป็นคอมโพเนนท์ที่ใช้แสดงข้อมูลที่ความสัมพันธ์ในลักษณะของ Tree โดยจะแสดงชื่อของตารางไดเมนชันที่ได้มีการสร้างไว้แล้ว รวมทั้งตารางย่อยของแต่ละตารางที่สัมพันธ์กันด้วย
- TListBox : แสดงรายชื่อ ฟิลด์ทั้งหมดของตารางไดเมนชั่นที่มีการเลือก
- TButton : เป็นปุ่มที่ให้สำหรับ Add, Remove ชื่อฟิลด์ที่ได้เลือก และ ปุ่ม OK เมื่อตกลงที่จะใช้ชื่อฟิลด์เหล่านั้น
- TLabel : แสดงชื่อของ Dimensions และ Fields
- TQuery : เป็นคอมโพเนนท์ที่ใช้ในการดึงข้อมูลออกมาจากฐานข้อมูลโดยการป้อน SQL Statement เข้าไป เพื่อ List ชื่อฟิลด์ของตารางไดเมนชั่นที่เลือกแสดงใน ListBox
- TDataSource : เป็นตัวกลางในการเชื่อมต่อฐานข้อมูลกับ TQuery

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ฟอร์มสำหรับใส่เงื่อนไข เป็นฟอร์มที่จะเกิดขึ้นเมื่อมีการคลิกที่ปุ่ม Condition ในฟอร์มหลัก ซึ่งจะปรากฏฟอร์มมีลักษณะดังรูปที่ 6.19



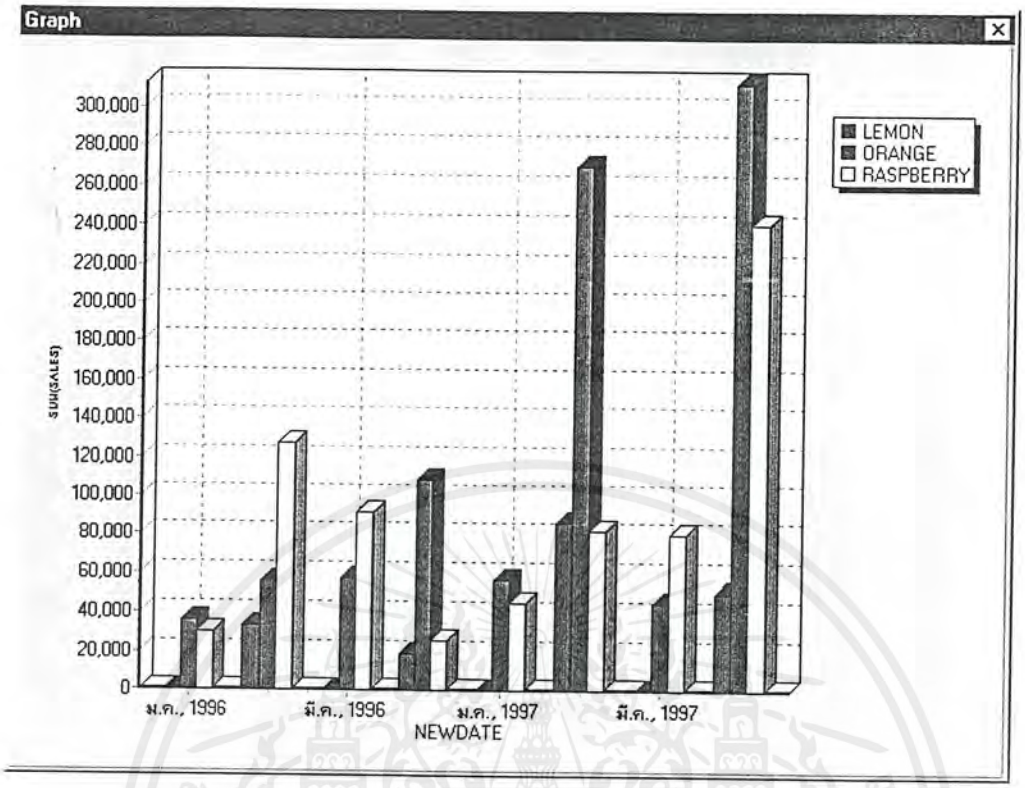
รูปที่ 6.19 แสดงฟอร์มในการใส่เงื่อนไขให้กับแต่ละฟิลด์ในตารางไดเมนชัน

คอมโพเนนท์ที่สำคัญมีดังนี้

- TTreeView : เป็นคอมโพเนนท์ที่ใช้แสดงข้อมูลที่ความสัมพันธ์ในลักษณะของ Tree โดยจะแสดงชื่อของตารางไดเมนชันที่ได้มีการสร้างไว้แล้ว รวมทั้งตารางย่อยของแต่ละตารางที่สัมพันธ์กันด้วย
- TListBox : แสดงรายชื่อ ฟิลด์ทั้งหมดของตารางไดเมนชั่นที่มีการเลือก
- TEdit : ใช้สำหรับใส่เงื่อนไขที่ต้องการลงไป โดยจะต้องมี '-' เนื่องจากข้อมูลในฟิลด์ส่วนใหญ่เป็น String
- TButton : ปุ่มที่ใช้สำหรับทำการ and, or เงื่อนไขต่างๆ ที่ผู้ใช้ต้องการ และปุ่ม Clear เมื่อต้องการลงเงื่อนไขที่มีอยู่ ปุ่ม OK เพื่อตกลงเลือกเงื่อนไขนั้นๆ และปุ่ม Cancel เมื่อต้องการยกเลิกการทำงานทั้งหมด
- TLabel : แสดงชื่อของ Dimensions และ Fields
- TQuery : เป็นคอมโพเนนท์ที่ใช้ในการดึงข้อมูลออกมาจากฐานข้อมูลโดยการป้อน SQL Statement เข้าไป เพื่อ List ชื่อฟิลด์ของตารางไดเมนชั่นที่เลือกแสดงใน ListBox
- TDataSource : เป็นตัวกลางในการเชื่อมต่อฐานข้อมูลกับ TQuery

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ฟอร์มสำหรับแสดงข้อมูลในรูปของกราฟ เพื่อให้ผู้ใช้สามารถเห็นแนวโน้มของข้อมูลได้ชัดเจนมากยิ่งขึ้น ดังรูปที่ 6.20



รูปที่ 6.20 แสดงฟอร์มสำหรับแสดงข้อมูลในรูปของกราฟ

คอมโพเนนท์ที่สำคัญมีดังนี้

- TDecisionGraph : เป็นคอมโพเนนท์ที่ใช้แสดงข้อมูลที่อยู่ในรูปของ crosstabs ให้อยู่ในรูปแบบของกราฟ แต่ละ Decision graph จะแสดงผลรวม ในรูปแบบของกราฟสำหรับ 1 หรือมากกว่า 1 ไดเมนชั่น โดย Decision graph จะแสดงฟิลด์ที่ได้มาจาก decision source ซึ่งจะเป็นแบบไดนามิก โดยข้อมูลจะสามารถเปลี่ยนแปลงได้เมื่อข้อมูลถูกเปิด ปิด แดร็ก และดรอป หรือมีการเปลี่ยน decision pivot

ข้อมูลของกราฟจะมาจาก dataset เช่น TdecisionQuery โดยปกติแล้วแถวแรกของไดเมนชั่นจะเป็นแกน x และคอลัมน์แรกของไดเมนชั่นก็จะเป็นแกน y

เราสามารถที่จะใช้ decision graph แทน หรือเพิ่มเติมจาก decision grid โดย decision grid และ decision graph จะใช้ decision source ตัวเดียวกันในการแสดงข้อมูลของไดเมนชั่น

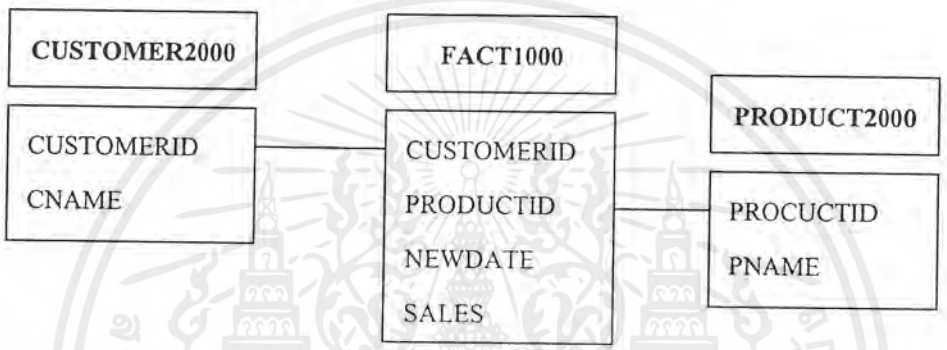
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
การทดสอบสมรรถนะของโปรแกรมประยุกต์

7.1 การทำ Aggregation fact table

เนื่องจากระบบฐานข้อมูลดาต้าแวร์เฮาส์นั้น จะประกอบด้วยเรคอร์ดจำนวนมาก ดังนั้นเมื่อมีการทำ query ก็จะต้องมีการ sum ข้อมูลภายในเรคอร์ดจำนวนมากเช่นกัน จึงได้มีการนำเสนอแนวความคิดในการทำ aggregate fact table เก็บเอาไว้ก่อน เพื่อเร่งให้การทำ query ในแต่ละครั้งสามารถทำได้รวดเร็วมากยิ่งขึ้น

โดยเราได้จำลองฐานข้อมูลดาต้าแวร์เฮาส์ที่มีขนาด 1,036,800 เรคอร์ด เพื่อดูว่าโปรแกรมที่เราได้ทำการพัฒนาขึ้นมานั้นสามารถทำงานในเวลาที่ผู้ใช้สามารถรับได้หรือไม่ โดยฐานข้อมูลที่ได้ทำการจำลองขึ้นมาจะมีลักษณะโครงสร้างดังรูปที่ 7.1



รูปที่ 7.1 โครงสร้างของฐานข้อมูลดาต้าแวร์เฮาส์ที่ได้ทำการจำลองขึ้นมา

เมื่อได้ทำการจำลองข้อมูลในตารางแฟ็กเป็นจำนวน 1,036,800 เรคอร์ด จากนั้นก็ทำการวิเคราะห์ข้อมูลโดยให้การ query ข้อมูล ต้องดึงข้อมูลจากตารางแฟ็กทุกๆ เรคอร์ด แล้วทำการจับเวลาที่ใช้ในการดึงข้อมูลขึ้นมาสรุปผล จากนั้นก็ลองทำการสร้างตารางแฟ็กใหม่โดยเป็นตารางที่มีการ aggregate ข้อมูลโดยให้มีการ sum ข้อมูลเป็นรายเดือน และรายปีเอาไว้ล่วงหน้า แล้วทำการ query ข้อมูลดู จับเวลาที่ใช้ในการสรุปผลข้อมูลดังกล่าว จะได้ผลดังต่อไปนี้

ชื่อ Fact table	จำนวนเรคอร์ด	เวลาที่ใช้ในการ query
Fact1000	1,036,800	50 นาที
Fact_Month	36,800	1 นาที
Fact_Year	3,200	15 วินาที

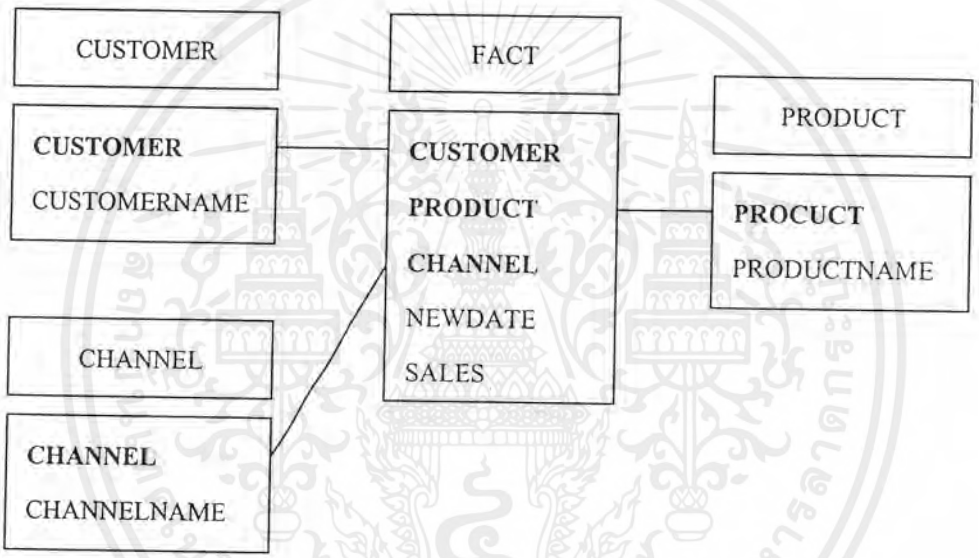
ตารางที่ 7.1 แสดงเวลาที่ใช้ในการ query ของแต่ละ fact table

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

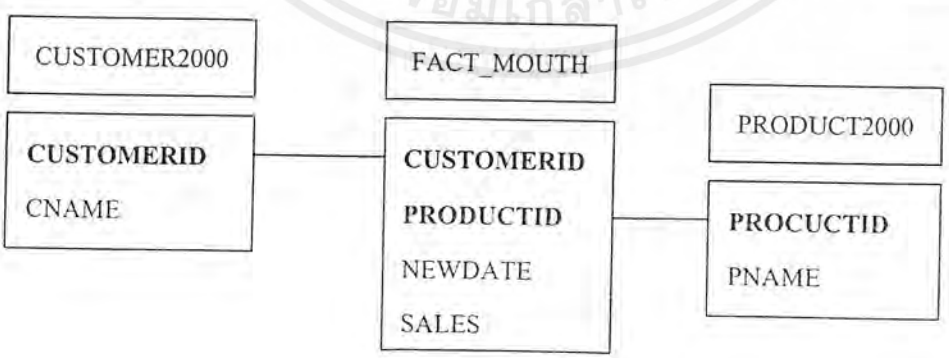
จากตารางที่ 7.1 จะเห็นว่า เมื่อมีการทำ aggregate fact table แล้วจะใช้เวลาในการ query น้อยลง เนื่องจากจำนวนเรคอร์ดที่มีใน fact table มีจำนวนลดลง ดังนั้นจึงสามารถสรุปได้ว่า ถ้าฐานข้อมูลมีขนาดใหญ่ มีจำนวนเรคอร์ดหลายแสนหลายล้านเรคอร์ด ผู้ใช้ควรจะมีการทำ aggregate fact table รวบรวมเก็บไว้ก่อน เพื่อเร่งให้การทำ query สามารถทำได้รวดเร็วมากยิ่งขึ้น

สำหรับอีกแนวทางหนึ่งก็คือผู้ใช้ต้องทำการสร้างอินเด็กซ์บนคอลัมน์ของตารางที่สร้างขึ้น ซึ่งวิธีการนี้จะเหมาะสมกับตารางที่มีข้อมูลหลายเรคอร์ดเช่นเดียวกัน ถ้าสร้างอินเด็กซ์บนตารางที่มีข้อมูลน้อยๆ จะทำให้เกิด overhead แทนที่จะทำให้ใช้เวลาน้อยลงจะกลับทำให้ใช้เวลาในการ query เพิ่มมากขึ้น และจะต้องพิจารณาด้วยว่าควรจะสร้างอินเด็กซ์บนคอลัมน์ไหน เพื่อให้การทำงานเกิดประสิทธิภาพสูงสุด

สำหรับฐานข้อมูลที่เราได้ทำการสร้างขึ้น จะทำการสร้างอินเด็กซ์บน primary key ของทุกๆ ตาราง ดังรูปที่ 7.2 ตัวหนาจะเป็นคอลัมน์ที่สร้างอินเด็กซ์ ซึ่งรูปดังกล่าวจะมีจำนวนเรคอร์ดเพียงเล็กน้อย หรือเป็นฐานข้อมูลที่มีขนาดเล็ก ส่วนรูปที่ 7.3 จะเป็นฐานข้อมูลดาต้าแวร์เฮาส์ที่มีขนาดใหญ่ มีจำนวนเรคอร์ดประมาณ 36,800 เรคอร์ด



รูปที่ 7.2 แสดงโครงสร้างสำหรับตารางที่ทำการสร้างอินเด็กซ์ที่มีจำนวนเรคอร์ดน้อย



รูปที่ 7.3 แสดงโครงสร้างสำหรับตารางที่ทำการสร้างอินเด็กซ์ที่มีจำนวนเรคอร์ดมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 การทดลองสร้างอินเด็กซ์บนฐานข้อมูลดาต้าแวร์เฮาส์ที่มีขนาดเล็ก

7.2.1 ทำการสร้างคำสั่ง SQL เพื่อดึงข้อมูลจากตารางในรูปที่ 7.1 ซึ่งเป็นฐานข้อมูลที่มีจำนวนเรคอร์ดเพียงเล็กน้อย และยังไม่ได้ทำการสร้างอินเด็กซ์ที่ Primary key

7.2.2 เก็บ log file ของการทำทรานเซคชั่นไว้ เพื่อ เปรียบเทียบ CPU time กับ log file ที่ทำการสร้างอินเด็กซ์บน primary key แล้ว

7.2.3 ทำการสร้างอินเด็กซ์บน primary key บนตารางในรูปที่ 7.1 จากนั้นก็สร้างคำสั่ง SQL เพื่อดึงข้อมูลจากตารางดังกล่าวขึ้นมา แล้วเก็บ log file ไว้

7.2.4 ทำการเปรียบเทียบ CPU time ระหว่าง log file ที่ยังไม่ได้สร้างอินเด็กซ์ กับ log file ที่สร้างอินเด็กซ์แล้ว

7.3 ผลทดลองสร้างอินเด็กซ์บนฐานข้อมูลดาต้าแวร์เฮาส์ที่มีขนาดเล็ก

เมื่อพิจารณาผลจาก log file ของการทำทรานเซคชั่นทั้ง 2 แบบ แล้วได้ผลดังนี้

7.3.1 Log file ที่เป็นผลจากการดึงข้อมูลจากฐานข้อมูลขนาดเล็ก และยังไม่มีการสร้างอินเด็กซ์

```
select customername,productname,channelname,newdate,sum(sales)
from customer,product,fact,channel
where customer.customer = fact.customer
and product.product = fact.product
and channel.channel = fact.channel
group by customername,productname,channelname,newdate

call     count   cpu    elapsed   disk   query   current   rows
-------------------------------------------------------------
Parse    3       0.03   0.12      0      0       0         0
Execute  3       0.00   0.00      0      0       0         0
Fetch    3       0.03   0.06      7      15      36        96
-------------------------------------------------------------
total    9       0.06   0.18      7      15      36        96
```

รูปที่ 7.4 แสดงผลใน log file เมื่อมีการดึงข้อมูลขึ้นมาจากตารางขนาดเล็กที่ยังไม่มีการสร้างอินเด็กซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3.2 Log file ที่เป็นผลจากการดึงข้อมูลจากฐานข้อมูลขนาดเล็ก และยังมีการสร้างอินเด็กซ์เรียบร้อยแล้ว

```
select customername,productname,channelname,newdate,sum(sales)
from customer,product,fact,channel
where customer.customer = fact.customer
and product.product = fact.product
and channel.channel = fact.channel
group by customername,productname,channelname,newdate
;

call     count       cpu    elapsed       disk      query    current        rows
------------------------------------------------------------------------------
Parse        4      0.04       0.13          0          0          0           0
Execute      4      0.00       0.02          0          0          0           0
Fetch        4      0.03       0.63         15        135         39         128
------------------------------------------------------------------------------
total       12      0.07       0.78         15        135         39         128
```

รูปที่ 7.5 แสดงผลใน log file เมื่อมีการดึงข้อมูลขึ้นมาจากตารางขนาดเล็กที่มีการสร้างอินเด็กซ์แล้ว

7.4 การทดลองสร้างอินเด็กซ์บนฐานข้อมูลดาต้าแวร์เฮาส์ที่มีขนาดใหญ่

7.4.1 ทำการสร้างคำสั่ง SQL เพื่อดึงข้อมูลจากตารางในรูปที่ 7.2 ซึ่งเป็นฐานข้อมูลที่มีจำนวนเรคอร์ดมาก และยังไม่ได้ทำการสร้างอินเด็กซ์ที่ Primary key

7.4.2 เก็บ log file ของการทำทรานแซคชั่นไว้ เพื่อ เปรียบเทียบ CPU time กับ log file ที่ทำการสร้างอินเด็กซ์บน primary key แล้ว

7.4.3 ทำการสร้างอินเด็กซ์บน primary key บนตารางในรูปที่ 7.2 จากนั้นก็สร้างคำสั่ง SQL เพื่อดึงข้อมูลจากตารางดังกล่าวขึ้นมา แล้วเก็บ log file ไว้

7.4.4 ทำการเปรียบเทียบ CPU time ระหว่าง log file ที่ยังไม่ได้สร้างอินเด็กซ์ กับ log file ที่สร้างอินเด็กซ์แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5 ผลทดลองสร้างอินเด็กซ์บนฐานข้อมูลดาต้าแวร์เฮาส์ที่มีขนาดใหญ่

เมื่อพิจารณาผลจาก log file ของการทำทรานเซคชั่นทั้ง 2 แบบ แล้วได้ผลดังนี้

7.5.1 Log file ที่เป็นผลจากการดึงข้อมูลจากฐานข้อมูลขนาดใหญ่ และยังไม่มีการสร้างอินเด็กซ์

```
select cname,pname,newdate,sum(sales)
from customer2000,product2000,fact_mouth
where customer2000.customerid = fact_mouth.customerid
and product2000.productid = fact_mouth.productid
group by cname,pname,newdate

call     count   cpu    elapsed    disk   query   current   rows
-------------------------------------------------------------------
Parse        1   0.01      0.01       0       0         0      0
Execute      2   0.54      0.72       0       0         6      0
Fetch      173   5.33      6.25     473     482      6494  36800
-------------------------------------------------------------------
total      176   5.88      6.98     473     482      6500  36800
```

รูปที่ 7.6 แสดงผลใน log file เมื่อมีการดึงข้อมูลขึ้นมาจากตารางขนาดใหญ่ที่ยังไม่มีการสร้างอินเด็กซ์

7.5.2 Log file ที่เป็นผลจากการดึงข้อมูลจากฐานข้อมูลขนาดใหญ่ และมีการสร้างอินเด็กซ์เรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
select cname,pname,newdate,sum(sales)
from customer2000,product2000,fact_mouth
where customer2000.customerid = fact_mouth.customerid
and product2000.productid = fact_mouth.productid
group by cname,pname,newdate

call     count       cpu    elapsed       disk      query    current        rows
-------------------------------------------------------------------------------
Parse        1      0.06       0.10          0          0          3           0
Execute      2      0.24       0.28          0          0          2           0
Fetch      173      5.10      40.53        793     110868       2227       36800
-------------------------------------------------------------------------------
total      176      5.40      40.91        793     110868       2232       36800
```

รูปที่ 7.7 แสดงผลใน log file เมื่อมีการดึงข้อมูลขึ้นมาจากตารางขนาดใหญ่ที่มีการสร้างอินเด็กซ์แล้ว

7.6 สรุปผลการทดลอง

7.6.1 สำหรับฐานข้อมูลขนาดเล็ก CPU time สำหรับการดึงข้อมูลจากตารางที่ยังไม่มีการสร้างอินเด็กซ์ จะใช้เวลาน้อยกว่า ตารางที่มีการสร้างอินเด็กซ์แล้ว เนื่องจากเกิด overhead สำหรับอินเด็กซ์ที่สร้างขึ้น ดังจะเห็นว่า CPU time ของรูปที่ 7.4 มีค่าเท่ากับ 0.06 และรูปที่ 7.5 มีค่าเท่ากับ 0.07

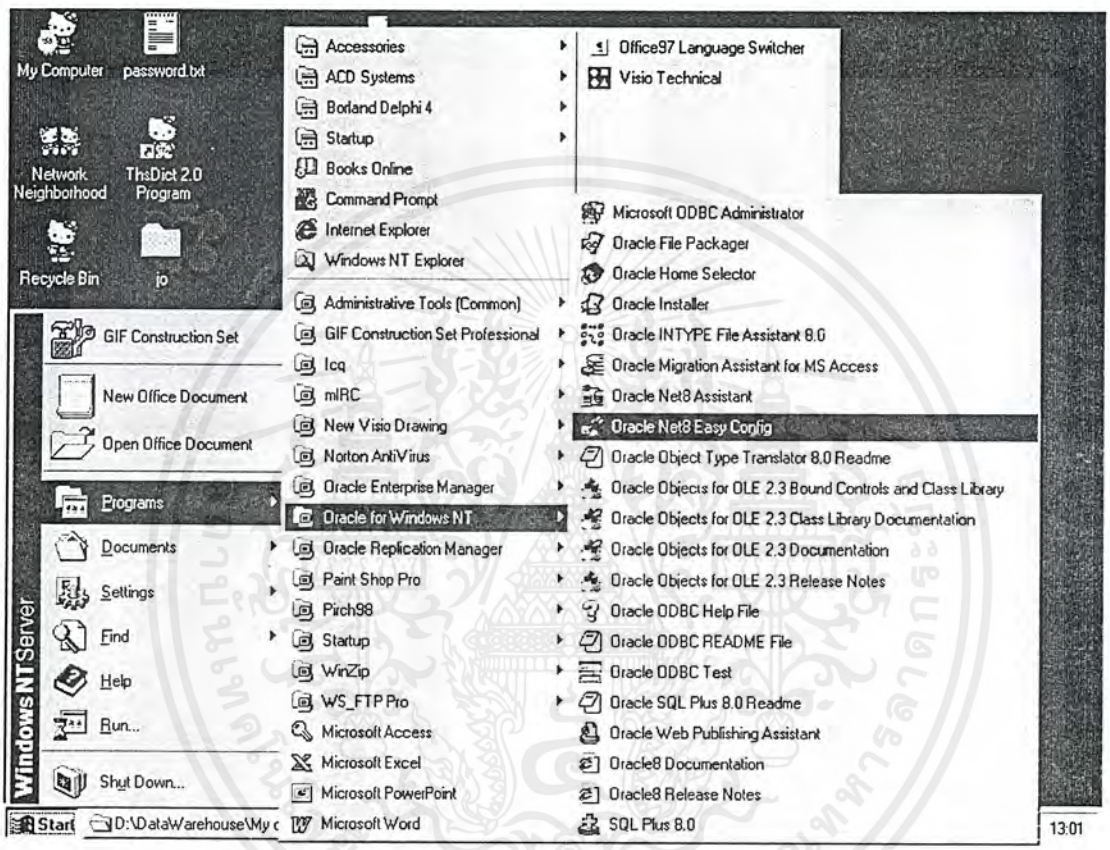
7.6.2 สำหรับฐานข้อมูลขนาดใหญ่ CPU time สำหรับการดึงข้อมูลจากตารางที่ยังไม่มีการสร้างอินเด็กซ์ จะใช้เวลามากกว่า ตารางที่มีการสร้างอินเด็กซ์แล้ว เนื่องจากเวลาที่ใช้ในการ scan ทั้งตารางจะใช้เวลามากกว่า การ scan ตามอินเด็กซ์ ดังจะเห็นว่า CPU time ของรูปที่ 7.6 มีค่าเท่ากับ 5.88 และรูปที่ 7.7 มีค่าเท่ากับ 5.40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
การสร้างเซอร์วิสเพื่อทำการติดต่อกับเซิฟเวอร์ของออราเคิล

เมื่อต้องการสร้างเซอร์วิสเพื่อเชื่อมต่อกับระบบจัดการฐานข้อมูลออราเคิลสามารถทำได้ดังต่อไปนี้

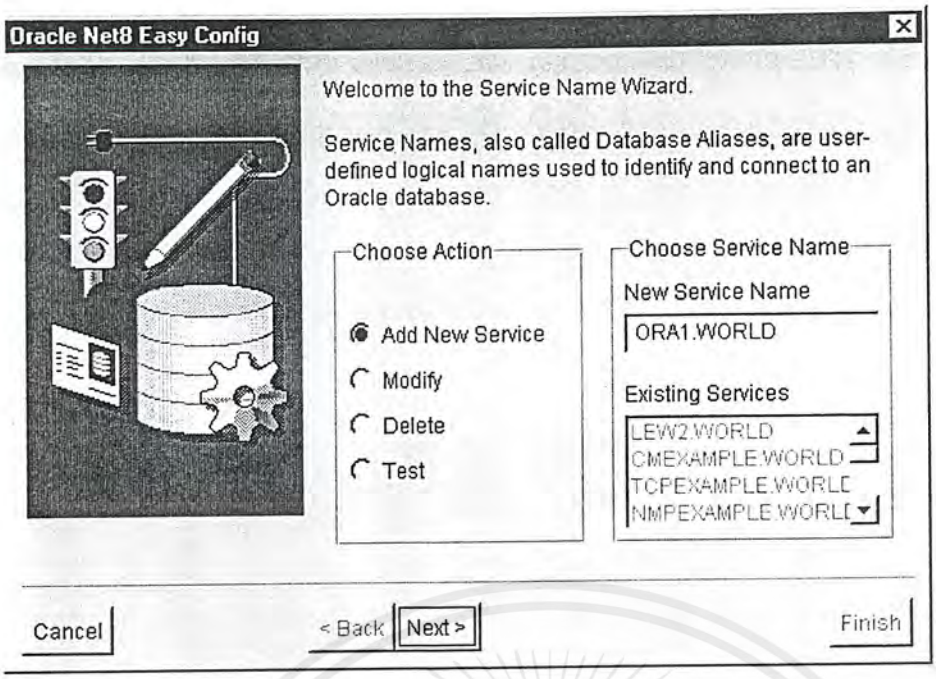
1. เปิดโปรแกรม Oracle Net8 Easy Config จาก Oracle for Windows NT ที่ Start ดังรูปที่ ก-1



รูปที่ ก-1 แสดงการเปิดโปรแกรม Oracle Net8 Easy Config

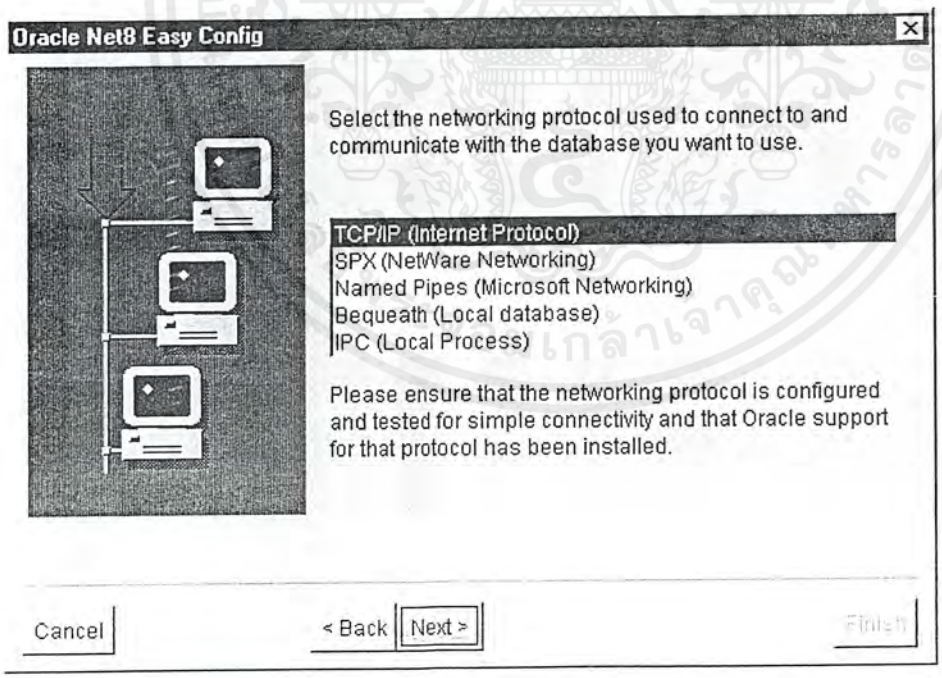
2. เมื่อทำการเปิดโปรแกรม Oracle Net8 Easy Config จะปรากฏหน้าจอดังรูปที่ ก-2 ซึ่งผู้ใช้จะต้องทำการใส่ชื่อเซอร์วิสที่ต้องการลงไปในช่อง New Service Name ในที่นี้ใส่ชื่อ "ORA1.WORLD"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-2 แสดงหน้าจอให้ใส่ชื่อเซอร์วิสที่ต้องการ

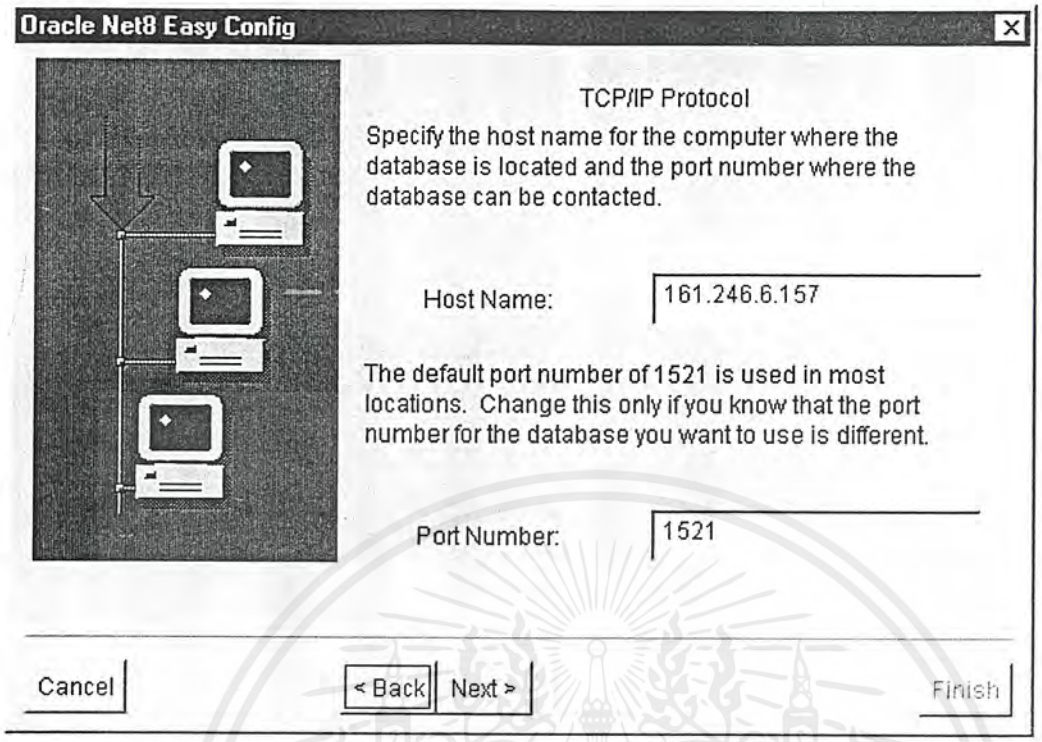
3. เมื่อคลิกปุ่ม Next> จะปรากฏหน้าจอดังรูปที่ ก-3 ซึ่งจะเป็นหน้าจอที่ให้ผู้ใช้เลือกโปรโตคอลที่ใช้ในการติดต่อกับระบบฐานข้อมูล



รูปที่ ก-3 แสดงหน้าจอให้เลือกโปรโตคอลในการเชื่อมต่อกับฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

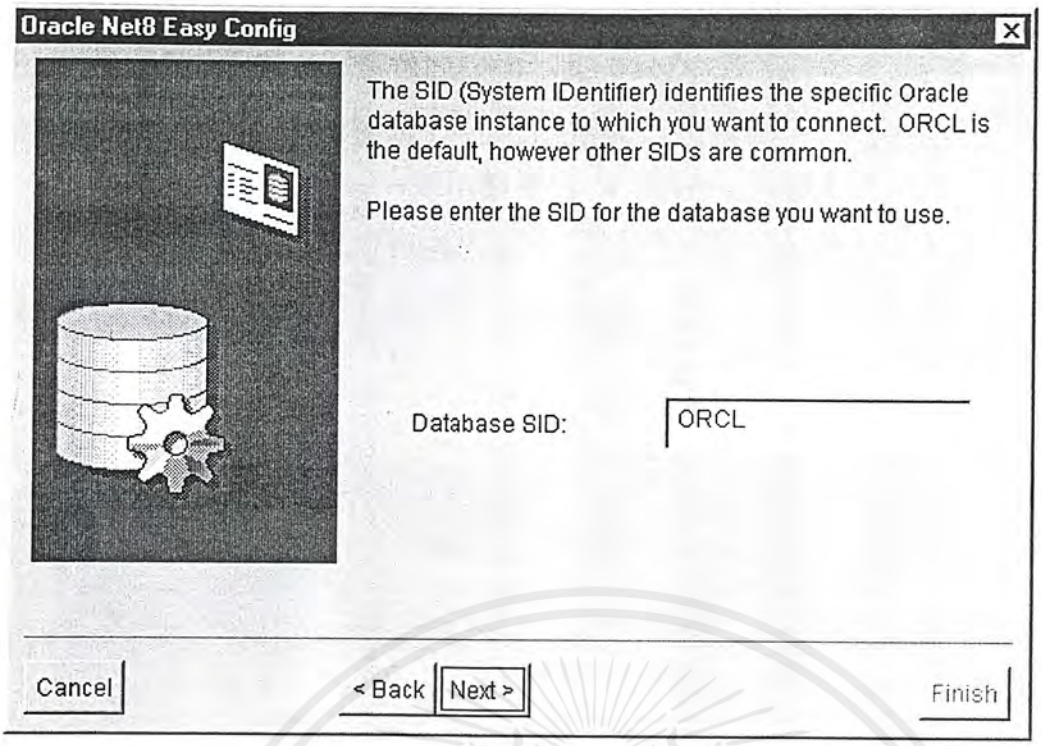
4. เมื่อคลิกที่ปุ่ม Next> จะปรากฏหน้าจอดังรูปที่ ก-4 ซึ่งจะเป็นหน้าจอที่ให้ผู้ใช้ใส่ IP ของเครื่องเซิฟเวอร์ออราเคิลที่จะติดต่อด้วย ในที่นี้ Host Name คือ 161.246.6.157



รูปที่ ก-4 แสดงหน้าจอให้ใส่ชื่อ Host Name ที่ต้องการติดต่อด้วย

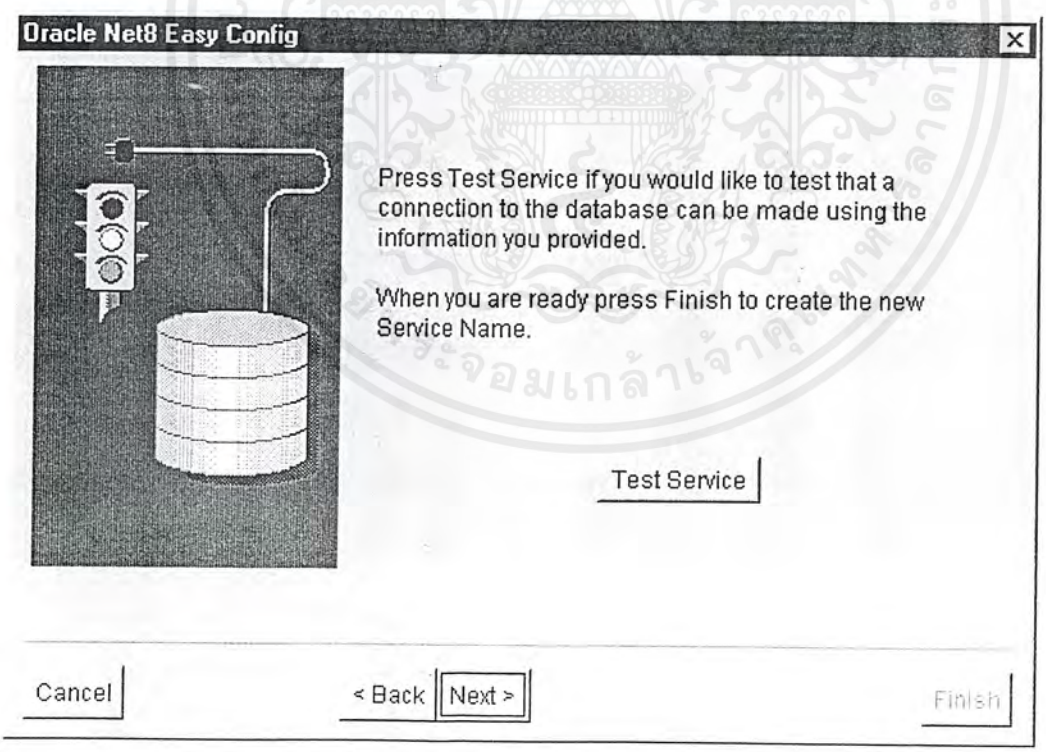
5. เมื่อคลิกปุ่ม Next> จะปรากฏหน้าจอดังรูปที่ ก-5 แล้วใส่ Database SID โดย Default จะเป็น ORCL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-5 แสดงหน้าจอให้ใส่ชื่อ Database SID

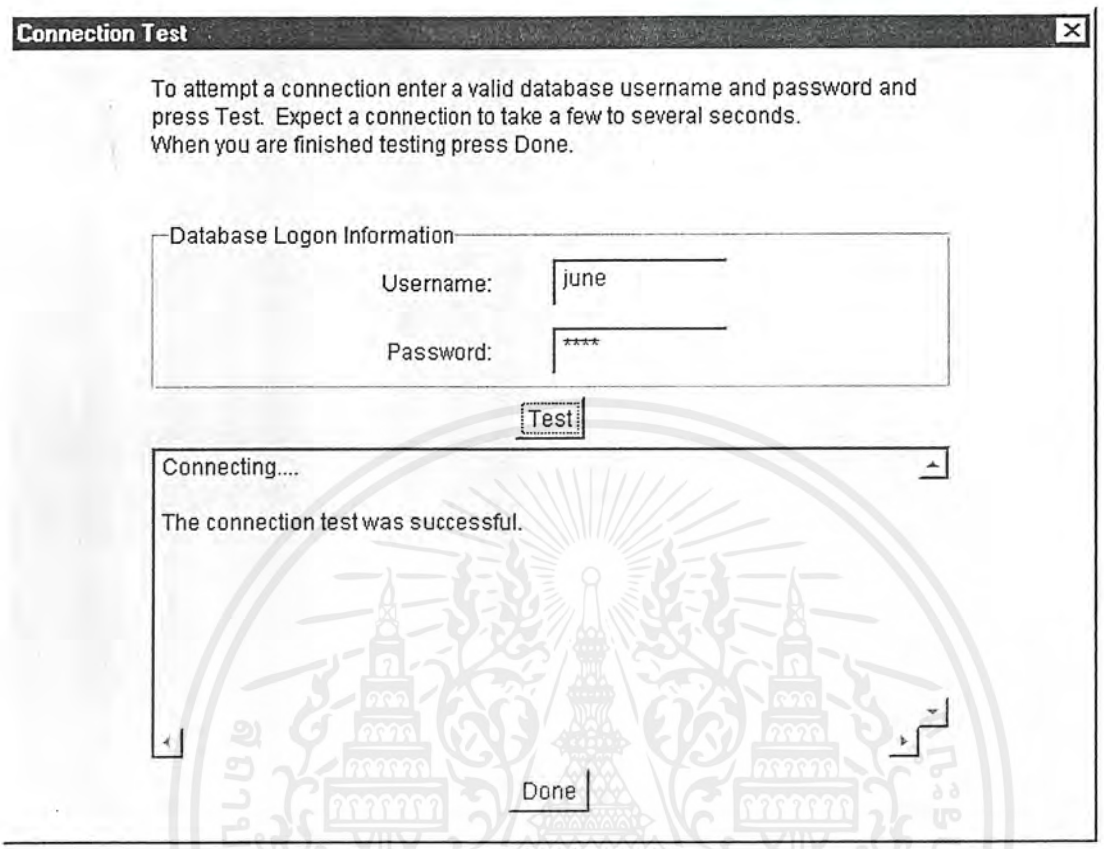
6. เมื่อคลิกที่ปุ่ม Next> จะปรากฏหน้าจอดังรูปที่ ก-6



รูปที่ ก-6 แสดงหน้าจอให้ทำการทดสอบเซอร์วิสที่ได้สร้างไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

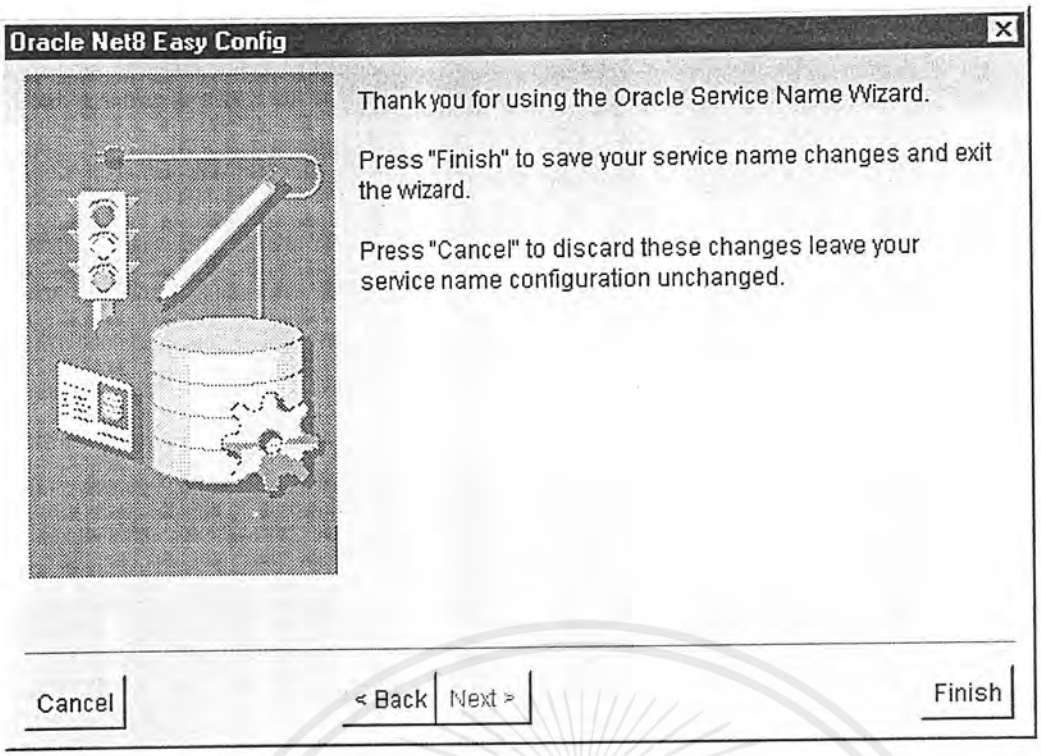
7. เมื่อคลิกที่ปุ่ม Test จะปรากฏหน้าจอดังรูปที่ ก-7 ซึ่งเป็นหน้าจอที่ให้ผู้ใช้ใส่ Username และ password ที่ได้สร้างไว้ที่เซิฟเวอร์ เพื่อทดสอบว่าสามารถติดต่อกับเครื่องเซิฟเวอร์ได้หรือไม่ ถ้าสามารถติดต่อได้ก็จะปรากฏหน้าจอดังรูป



รูปที่ ก-7 แสดงหน้าจอให้ใส่ Username และ password เพื่อทดสอบการเชื่อมต่อกับเซิฟเวอร์

8. เมื่อสามารถเชื่อมต่อได้แล้ว ทำการคลิกที่ปุ่ม Done ก็จะกลับมายังหน้าจอดังรูปที่ ก-8 ซึ่งจะเป็นหน้าจอที่บอกกับผู้ใช้ว่าจะเก็บเซอร์วิสที่สร้างนี้ไว้ หรือยกเลิกเซอร์วิสที่สร้างขึ้น โดยการคลิกที่ปุ่ม Finish และ Cancel ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-8 แสดงหน้าจอสุดท้ายของการสร้างเซอร์วิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

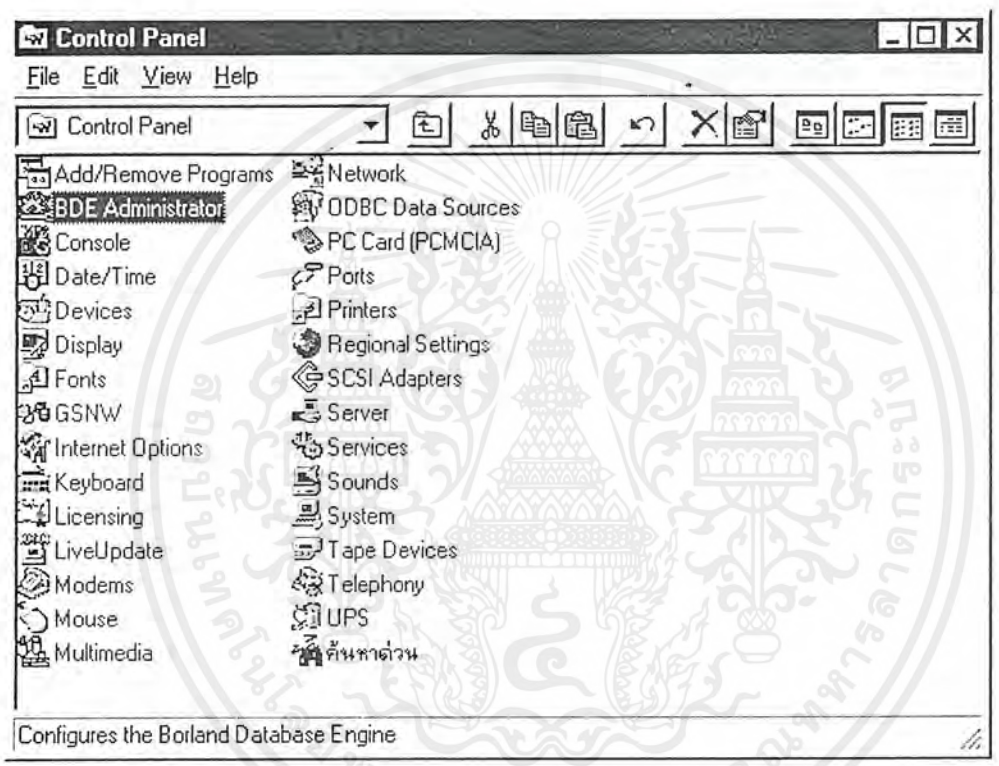
ภาคผนวก ข

วิธีการสร้าง BDE ในการเชื่อมต่อกับระบบจัดการฐานข้อมูลออราเคิล

BDE (Borland Database Engine) คือวิธีการเชื่อมต่อ (Driver) ระหว่างแอปพลิเคชั่นที่อยู่ฝั่งไคลเอนต์ (Client) กับฐานข้อมูลที่อยู่ฝั่งเซิฟเวอร์(Server)

โดยการกำหนดคุณสมบัติของ BDE ให้โปรแกรมประยุกต์ที่ทำการพัฒนาขึ้นสามารถเชื่อมต่อกับระบบจัดการฐานข้อมูลออราเคิลได้ ทำดังนี้

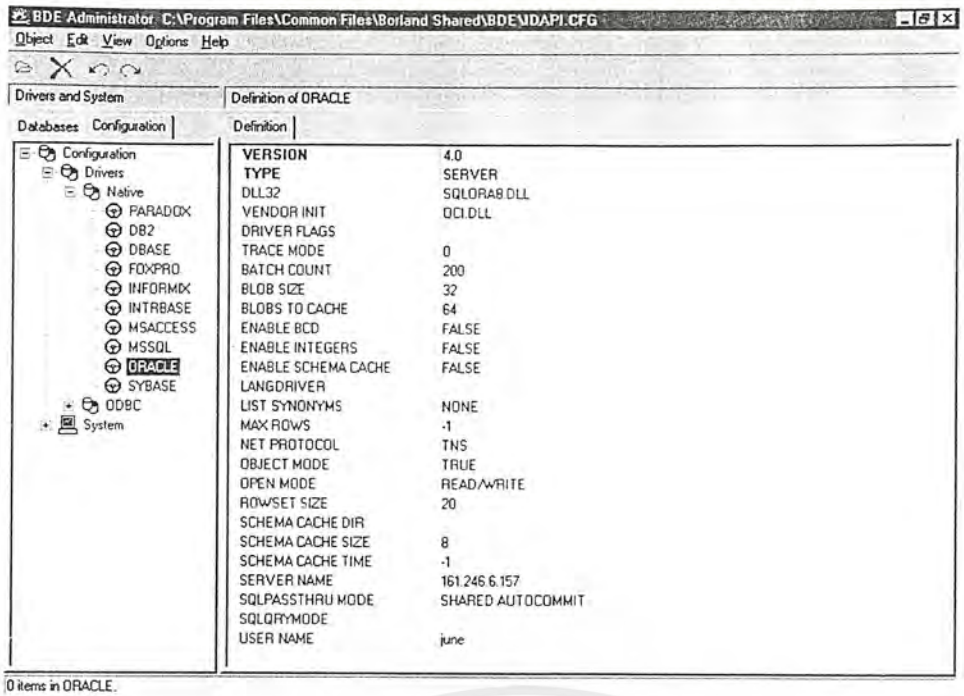
1. เปิด BDE Administrator จาก Control Panel ดังรูปที่ ข-1



รูปที่ ข-1 แสดงหน้าจอ Control Panel เพื่อเปิดโปรแกรม BDE Administrator

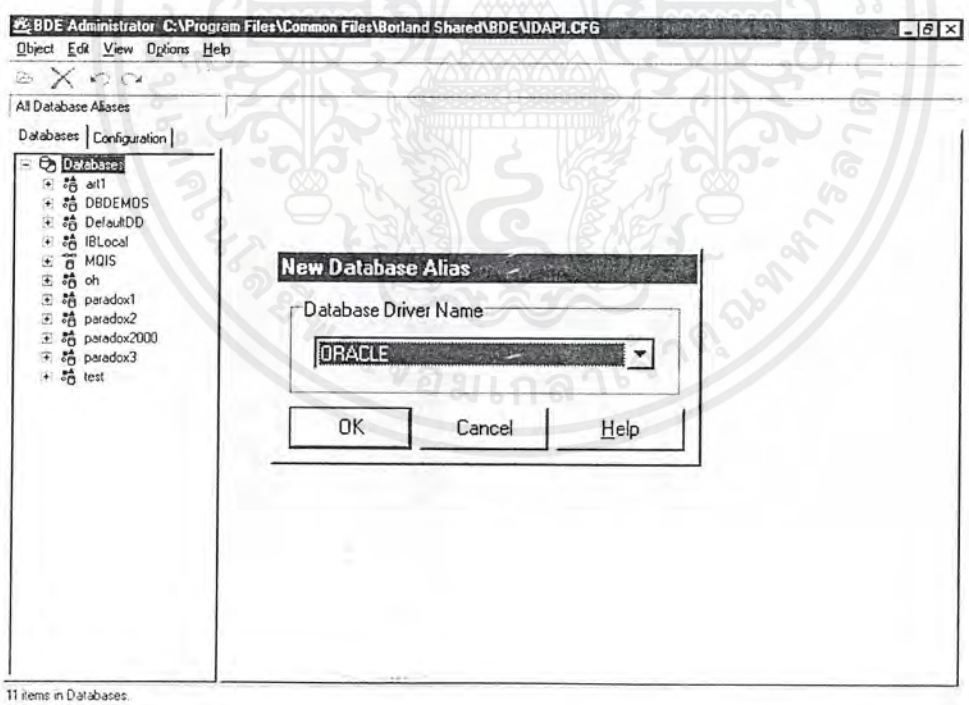
2. เมื่อเปิดโปรแกรมขึ้นมาแล้วจะปรากฏดังรูปที่ ข-2 ให้เลือกที่แทป Configuration > Drivers > Native > ORACLE แล้วเลือก DLL32 เป็น SQLORA8.DLL และเลือก VENDOR INIT เป็น OCI.DLL และก็ใส่ SERVER NAME หรือ IP ในที่นี้ใช้ IP เป็น 161.246.6.157

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-2 แสดงหน้าจอ Configuration ของออราเคิล

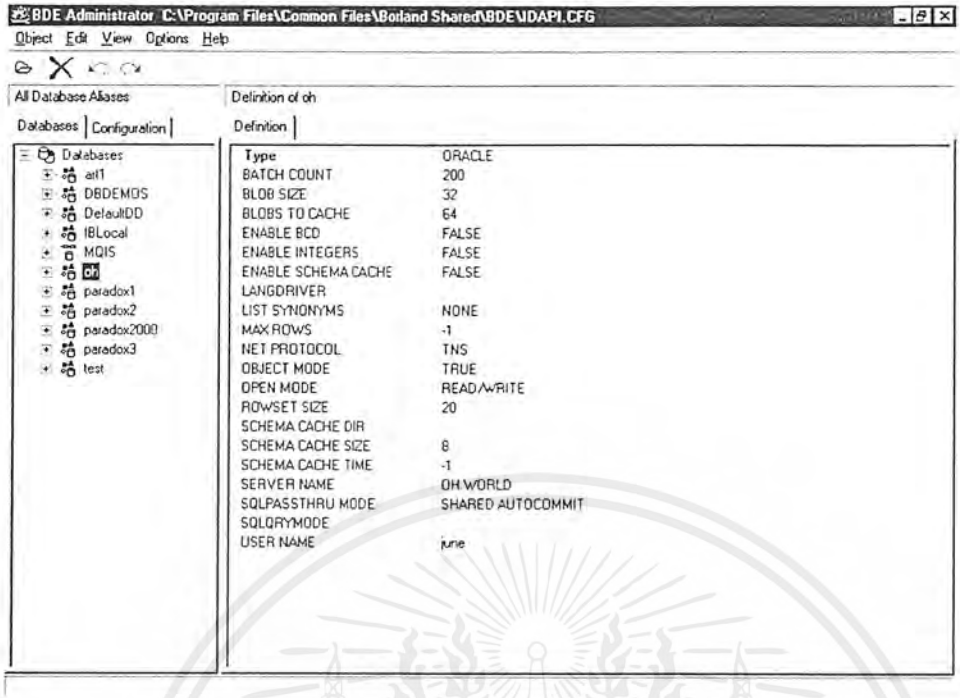
3. หลังจากเลือกเสร็จแล้วก็ Apply โดยการคลิกปุ่มขวาของเมาส์ จากนั้นก็เลือกที่แทป Database ดังรูปที่ ข-3 จากนั้นคลิกปุ่มขวาของเมาส์เพื่อเลือก New... ก็จะปรากฏหน้าจอของ New Database Alias แล้วจึงกำหนด Database Driver Name ให้เป็น ORACLE แล้วคลิก OK



รูปที่ ข-3 แสดงหน้าจอการเพิ่ม Alias ที่เป็นออราเคิล

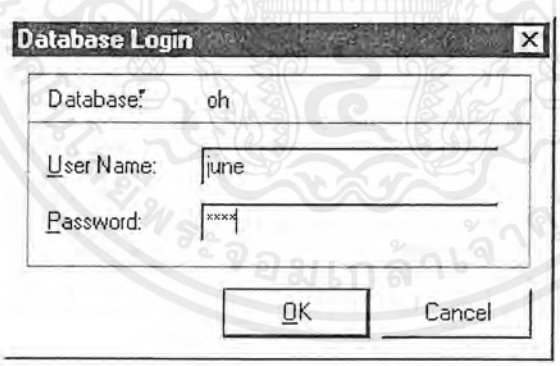
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. หลังจากสร้าง Database Alias ตัวใหม่ แล้วจะได้ชื่อ BDE ขึ้นมา จากนั้นก็คลิกขวาที่เมาส์ เลือก Apply แล้วเลือก SERVER NAME เป็นชื่อ Service ที่เราได้ทำการสร้างขึ้นมาจากภาคผนวก ก ดังรูปที่ ข-4



รูปที่ ข-4 แสดงหน้าจอการเลือก Service name ที่ได้สร้างขึ้นมา

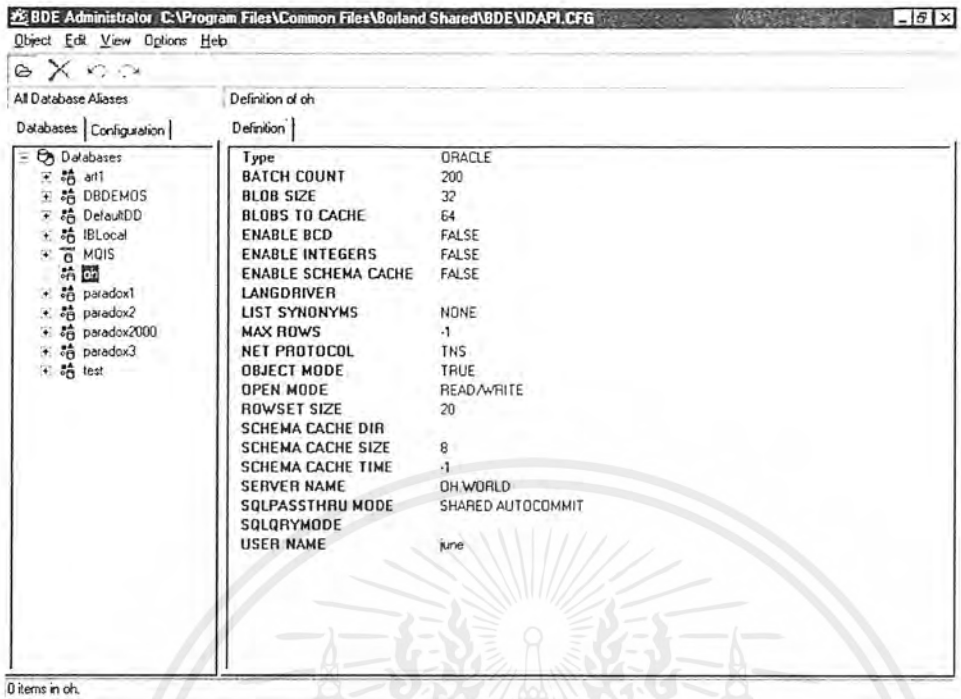
5. หลังจากนั้นทำการทดสอบ BDE ที่เราได้สร้างโดยการดับเบิลคลิก จะปรากฏหน้าจอดังรูปที่ ข-5



รูปที่ ข-5 แสดงหน้าจอที่ให้ผู้ใช้ใส่ User Name และ Password

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ถ้าสามารถเชื่อมต่อได้จะปรากฏหน้าจอดังรูปที่ ข-6



รูปที่ ข-6 แสดงหน้าจอเมื่อทำการเชื่อมต่อได้เรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค
คู่มือการใช้โปรแกรมประยุกต์

ความต้องการของระบบ : เครื่องที่จะติดตั้งควรจะมี

1. โปรแกรมเดลไฟล์ 4.0
2. ระบบจัดการฐานข้อมูลออราเคิล อาจจะเป็น ออราเคิลเซอร์เวอร์ หรือ ไคล์เอนท์ ก็ได้
3. ถ้าระบบจัดการฐานข้อมูลออราเคิลเป็นไคล์เอนท์ ก็ต้องมีการสร้างเซอร์วิส และต้องมีการสร้าง Alias ที่ Borland Database Engine (BDE) เพื่อให้โปรแกรมประยุกต์สามารถเชื่อมต่อกับฐานข้อมูลออราเคิลได้

โปรแกรมที่ได้ทำการออกแบบแบ่งออกเป็น 2 ส่วนดังนี้

1. ส่วนที่ใช้ในการสร้างโมเดล
2. ส่วนที่ใช้ในการวิเคราะห์ข้อมูล

โดยโปรแกรมที่ได้ทำการออกแบบนี้จะมีหน้าจอทั้งหมด 8 หน้าจอ ดังนี้

หน้าจอที่ 1 : เป็นหน้าจอก่อนเข้าสู่โปรแกรม เป็นหน้าจอที่แสดงรูปภาพ จะปรากฏฟอร์มนี้เมื่อเข้าสู่โปรแกรมประยุกต์ ผู้ใช้สามารถคลิกที่รูปภาพ เพื่อเข้าสู่โปรแกรม หรือคลิกที่ Exit เพื่อออกจากการใช้งานโปรแกรมได้ ดังรูปที่ ค-1

รูปที่ ค-1 แสดงรูปหน้าจอแรกที่ปรากฏเมื่อเข้าสู่โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอที่ 2 : เป็นหน้าจอการทำงานหลักของโปรแกรม ซึ่งจะมี Main menu อยู่ 3 เมนู นั่นคือ File โดยจะมีเมนูย่อยคือ Create Model, Load Model, Print, Exit Graph และ Language ซึ่งมีเมนูย่อยคือ Thai, English ดังรูปที่ ค-2

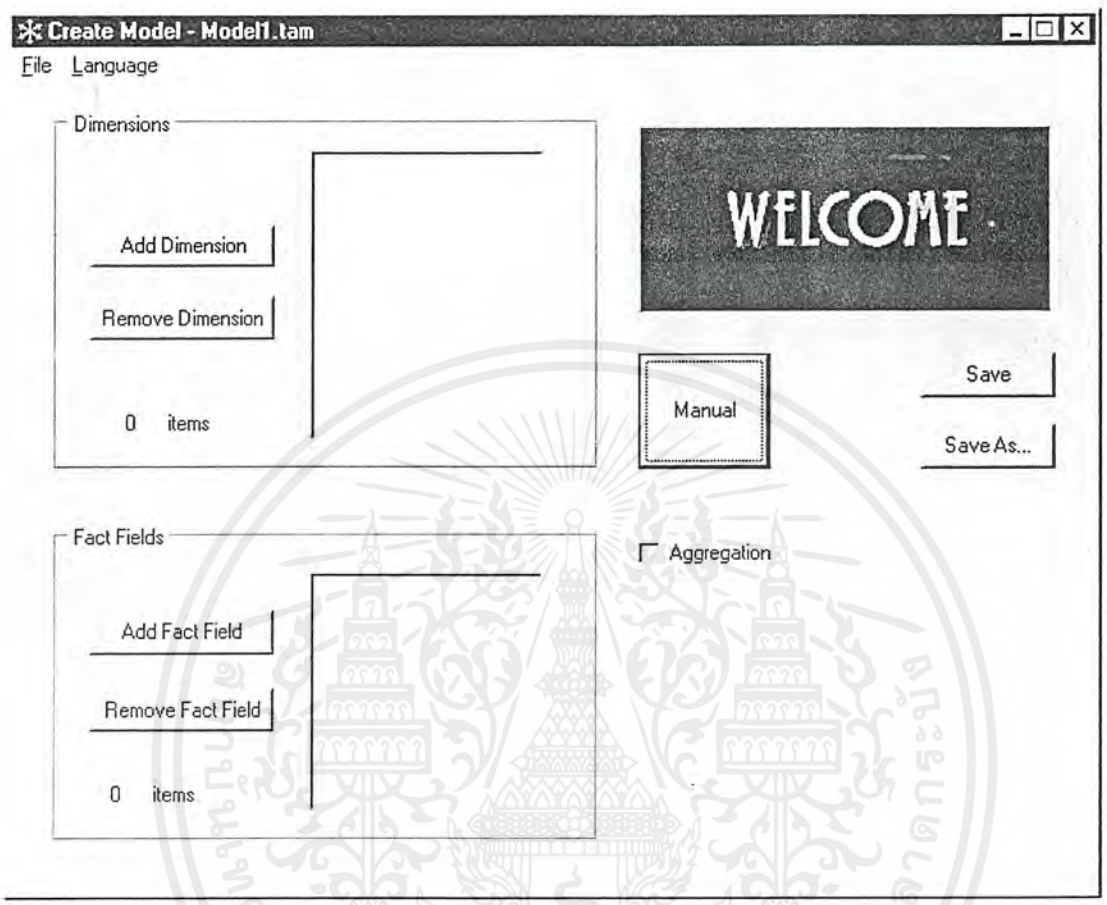
รูปที่ ค-2 แสดงรูปหน้าจอการทำงานหลักของโปรแกรม

หน้าจอที่ 3 : เมื่อผู้ใช้ทำการเลือกเมนู File > Create Model แล้วก็จะปรากฏหน้าจอดังรูปที่ ค-3 เพื่อให้ผู้ใช้สามารถสร้างไดเมนชั่น และตัววัดทางธุรกิจได้ตามความต้องการ

เมื่อเข้ามาสู่หน้าจอนี้จะมีเมนูให้เลือกนั่นคือ File ซึ่งมีเมนูย่อยคือ New..., Open..., Save, Save As..., Exit กับ Language ซึ่งมีเมนูย่อยคือ English และ Thai ซึ่งผู้ใช้สามารถที่จะโหลดโมเดลที่เคยสร้างเอาไว้แล้วมาแก้ไขใหม่ได้โดยเลือกเมนู File > Open... หรือจะทำการสร้างโมเดลใหม่ โดยเลือกเมนู File > New ซึ่งจะปรากฏไดอะลอกให้กรอกชื่อโมเดลที่จะสร้าง ซึ่งเมื่อจะทำการบันทึกโมเดลที่สร้างเสร็จแล้วก็จะบันทึกในชื่อที่ใส่นี้เลย หลังจากที่ผู้ใช้ได้ตั้งชื่อโมเดลที่ต้องการสร้างได้แล้ว ก็จะปรากฏหน้าจอตามรูปที่ ค-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ออกแบบไว้รองรับเมนูการทำงานแบบทั้งภาษาไทย และอังกฤษ โดยเลือกที่ Language > English ถ้าต้องการให้หน้าจอแสดงเป็นภาษาอังกฤษ และเลือกที่ Language > Thai ถ้าต้องการให้หน้าจอแสดงเป็นภาษาไทย โดยเมื่อเปิดโปรแกรมขึ้นมาครั้งแรก หน้าจอจะแสดงเป็นภาษาอังกฤษ (Default)

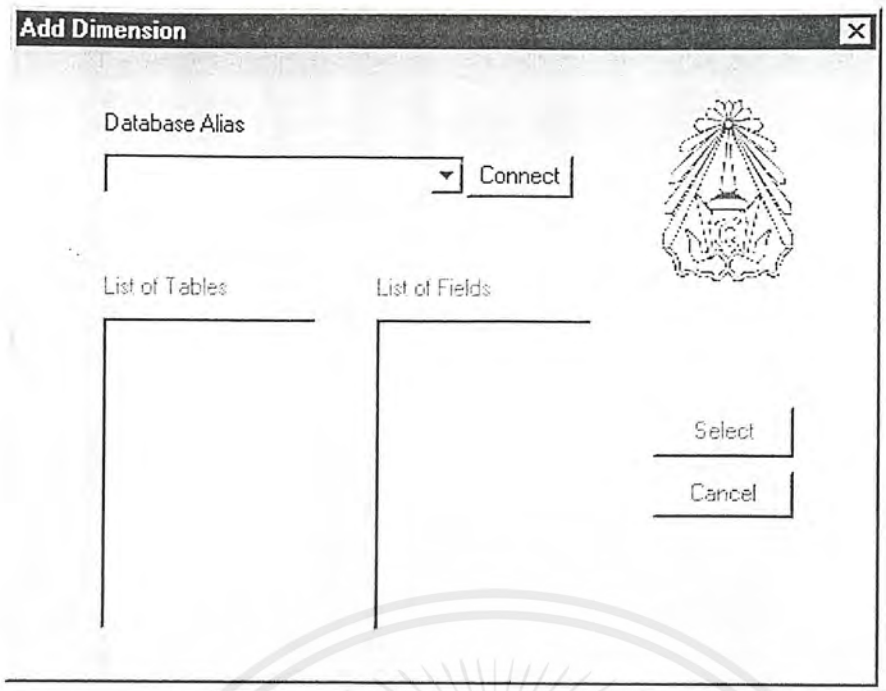


รูปที่ ค-3 แสดงรูปหน้าจอสำหรับสร้างโมเดล

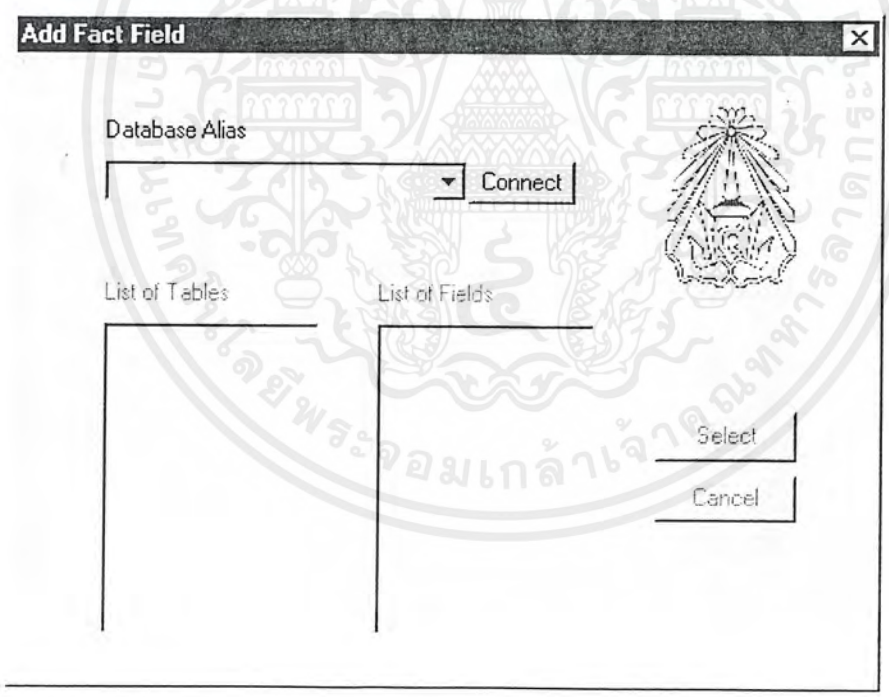
หน้าจอที่ 4 : เมื่อคลิกที่ปุ่ม Add dimension ซึ่งเป็นการเพิ่มตารางไดเมนชันให้กับโมเดลที่เราสร้าง จะปรากฏหน้าจอดังรูปที่ ค-4 ซึ่งเมื่อต้องการเพิ่มฟิลด์ของตารางแฟ็ก ก็คลิกที่ปุ่ม Add Fact Field และหน้าจอที่จะปรากฏขึ้นมาก็จะมีลักษณะเหมือนกันกับหน้าจอดังรูปที่ ค-4 แต่ชื่อฟอร์มจะเปลี่ยนจาก Add Dimension ไปเป็น Add Fact Field ดังรูปที่ ค-5

อย่างแรกที่ผู้ใช้จะพบเมื่อมาถึงหน้าจอดังกล่าว ก็คือการเลือก Database Alias ที่เราต้องการจะติดต่อด้วย เพื่อจะได้ดึงเอาตารางที่มีอยู่ทั้งหมดขึ้นมาเพื่อให้ผู้ใช้เลือกว่า ตารางไหนจะเป็นตารางไดเมนชัน และตารางไหนจะเป็นตารางแฟ็ก เมื่อทำการเลือกได้แล้วก็คลิกที่ปุ่ม Connect เพื่อเชื่อมต่อกับระบบฐานข้อมูล ก็จะปรากฏชื่อตารางทั้งหมดที่อยู่ในฐานข้อมูลที่เราได้เลือกเอาไว้ ดังรูปที่ ค-7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

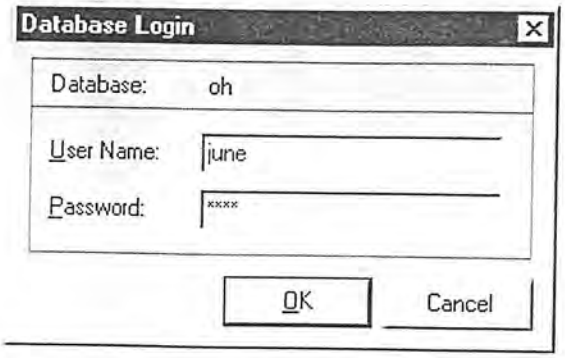


รูปที่ ค-4 แสดงรูปหน้าจอสำหรับการเพิ่มไดเมนชัน



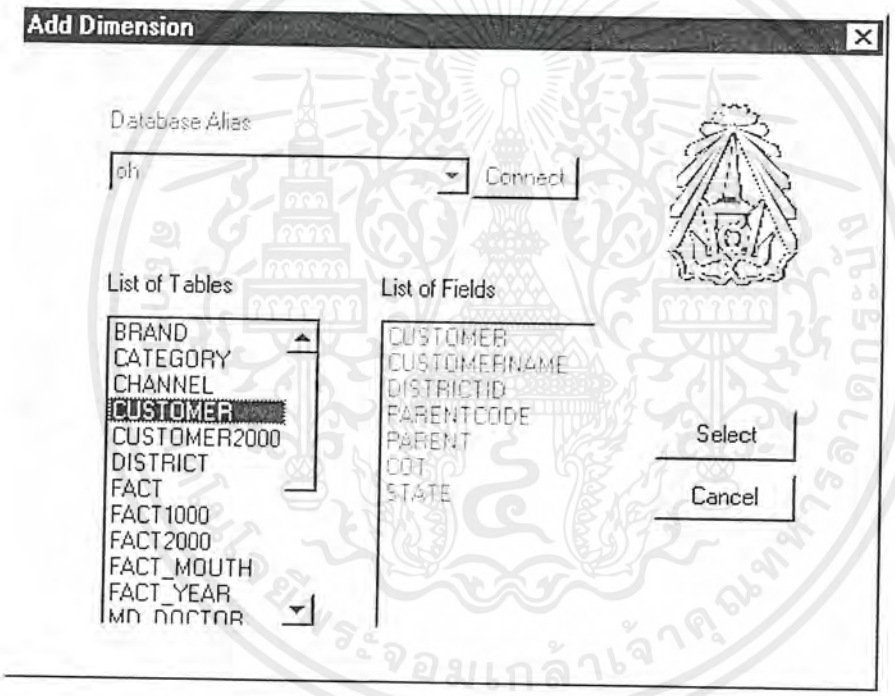
รูปที่ ค-5 แสดงรูปหน้าจอสำหรับการเพิ่มฟิลด์ในตารางแฟ็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค-6 แสดงรูปหน้าจอเมื่อมีการเชื่อมต่อกับฐานข้อมูลออกราเคิล

เมื่อได้ทำการเลือก Alias ที่สร้างไว้ใน BDE แล้ว ก็คลิกที่ปุ่ม Connect จะปรากฏหน้าจอดังรูปที่ ค-6 ซึ่งจะให้ผู้ใช้ใส่ User name และ Password จากนั้นคลิกปุ่ม OK แล้วจะปรากฏหน้าจอดังรูปที่ ค-7

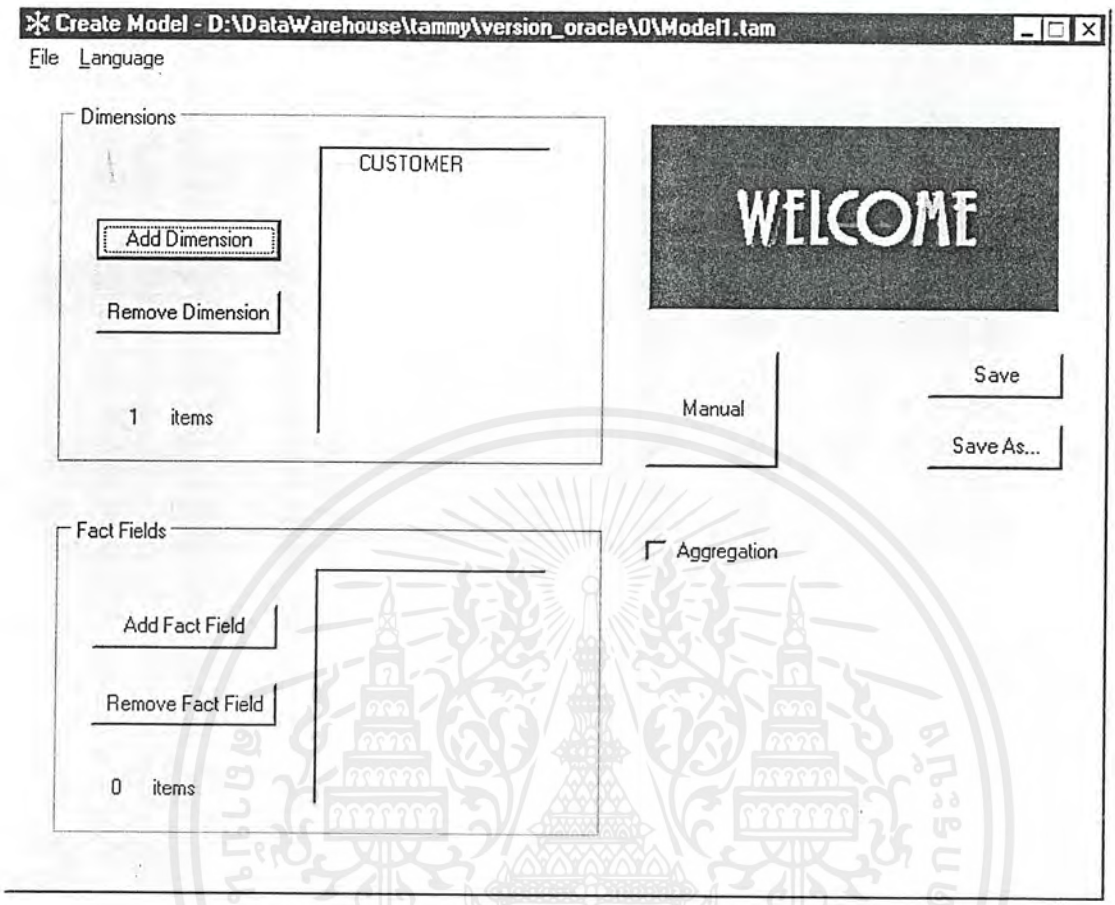


รูปที่ ค-7 แสดงรูปหน้าจอเมื่อทำการเชื่อมต่อกับฐานข้อมูล

จากรูปที่ ค-7 ผู้ใช้สามารถเลือกตารางที่ต้องการให้เป็นตารางไดเมนชันได้ จากตัวอย่าง ได้ทำการเลือกตาราง Customer จากนั้นก็คลิกที่ปุ่ม Select หน้าจอนี้ก็จะหายไป กลับเข้าสู่หน้าจอการสร้างโมเดล (Create Model) ดังรูปที่ ค-8 จะเห็นว่ามีไดเมนชัน Customer ปรากฏขึ้นมาในช่องไดเมนชัน

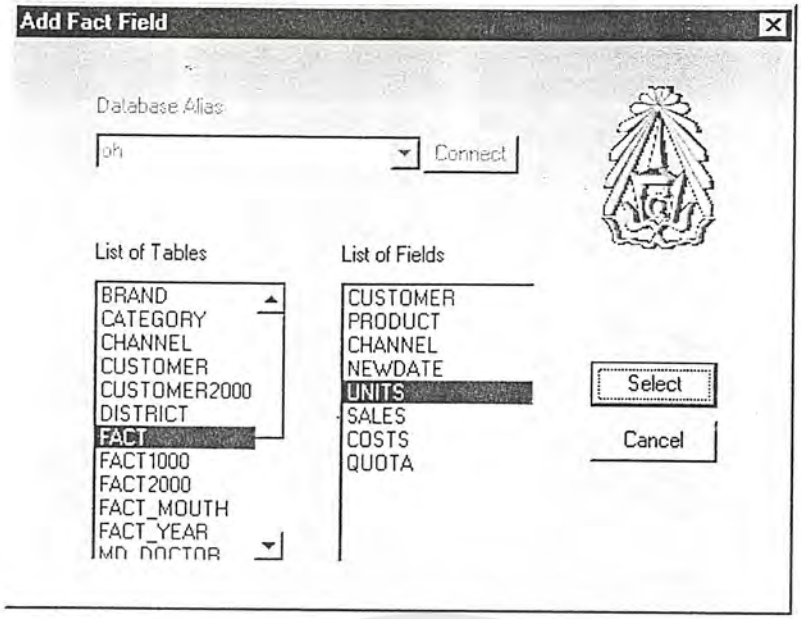
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการเพิ่มฟิลด์ของตารางแฟ็กก็ทำเช่นเดียวกันนี้ โดยเมื่อเชื่อมต่อกับฐานข้อมูล ก็เลือกตารางที่เป็นตารางแฟ็กในฐานข้อมูลที่เราได้ทำการออกแบบไว้ แล้วเลือกฟิลด์ที่ต้องการ จากนั้นคลิกที่ปุ่ม Select ก็จะปรากฏตัววัดที่หน้าจอของการสร้างโมเดลขึ้นมา ดังรูปที่ ค-9 และรูปที่ ค-10

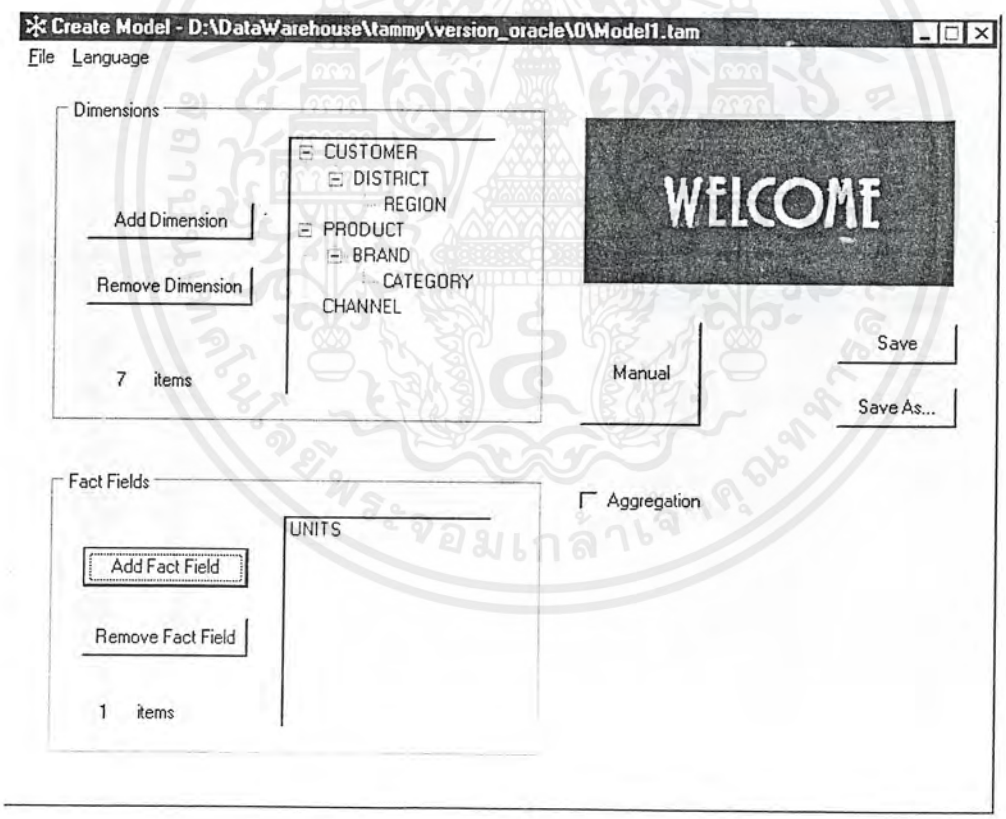


รูปที่ ค-8 แสดงรูปหน้าจอเมื่อมีการเลือกตารางไดเมนชันเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค-9 แสดงรูปหน้าจอเมื่อทำการเชื่อมต่อกับฐานข้อมูล ในการเพิ่มฟิลด์ของตารางแฟ็ก

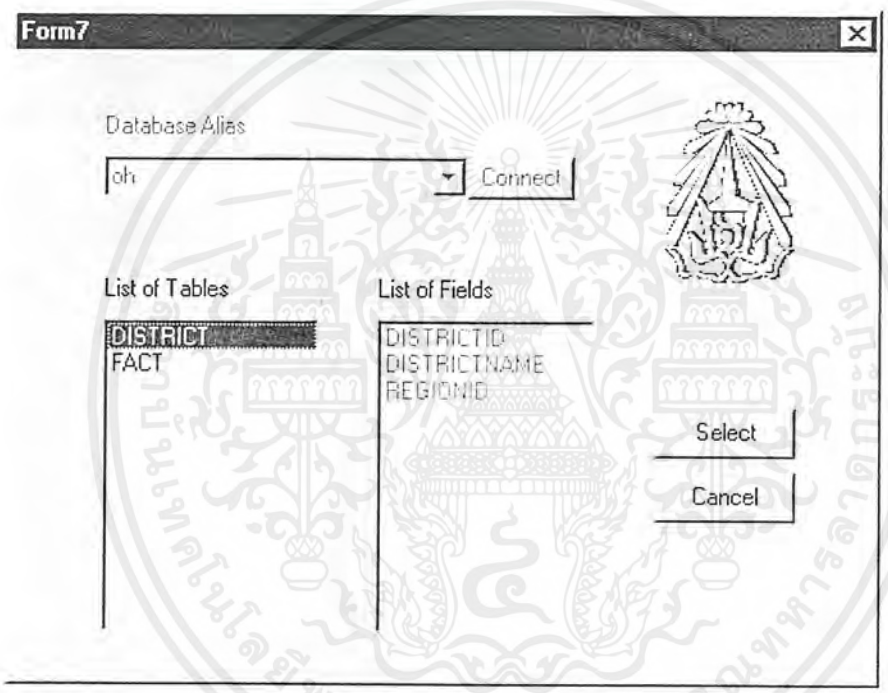


รูปที่ ค-10 แสดงรูปหน้าจอเมื่อมีการเลือกตารางไดเมนชัน และตัววัดเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

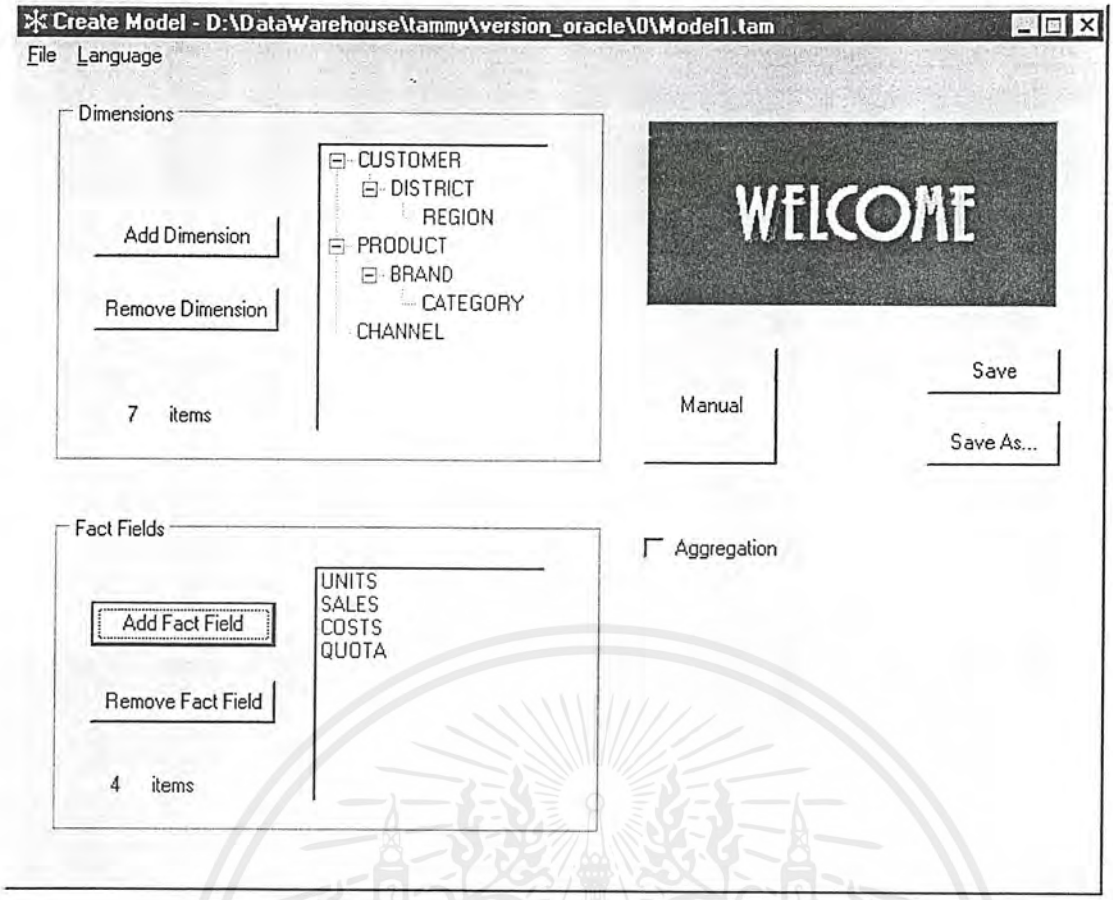
ซึ่งในรูปที่ ค-10 จะแสดงให้เห็นถึงลำดับชั้นของไดเมนชั่น หรือสโนว์เฟลกของไดเมนชัน โดยการสร้างไดเมนชันให้เป็นลำดับชั้นนี้ทำได้โดยการดับเบิลคลิกที่ชื่อไดเมนชันที่ได้สร้างเอาไว้แล้ว เช่นดับเบิลคลิกที่ Customer ก็จะปรากฏหน้าจอสำหรับสร้างตารางย่อย หรือตารางที่มีความสัมพันธ์กับตาราง Customer ดังรูปที่ ค-11 นั่นคือ ตาราง District นั้นมีความสัมพันธ์กับตาราง Customer โดยมีชื่อฟิลด์ที่เหมือนกันนั่นเอง สำหรับการสร้างสโนว์เฟลกนั้น ผู้ใช้สามารถดับเบิลคลิกเพียงครั้งเดียว นั่นคือลำดับชั้นแรกเพียงชั้นเดียว ส่วนชั้นต่อๆ มาโปรแกรมจะจัดการหาให้โดยอัตโนมัติ ทำให้ง่ายต่อการใช้งาน และผู้ใช้ไม่จำเป็นต้องทราบว่ามีตารางไหน เชื่อมต่อกับตารางไหน โดยโปรแกรมจะจัดการให้

ทำเช่นนี้ไปเรื่อยๆ สำหรับไดเมนชั่นที่มีการแตกออกเป็นตารางย่อย ก็จะได้ดังรูปที่ ค-10 ด้านบน จะเห็นว่าไดเมนชัน Channel นั้นไม่มีตารางย่อยที่มีความสัมพันธ์กับตัวเอง จึงไม่ต้องทำการสร้างสโนว์เฟลกให้



รูปที่ ค-11 แสดงรูปหน้าจอเมื่อมีการดับเบิลคลิกไดเมนชัน Customer เพื่อสร้างสโนว์เฟลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

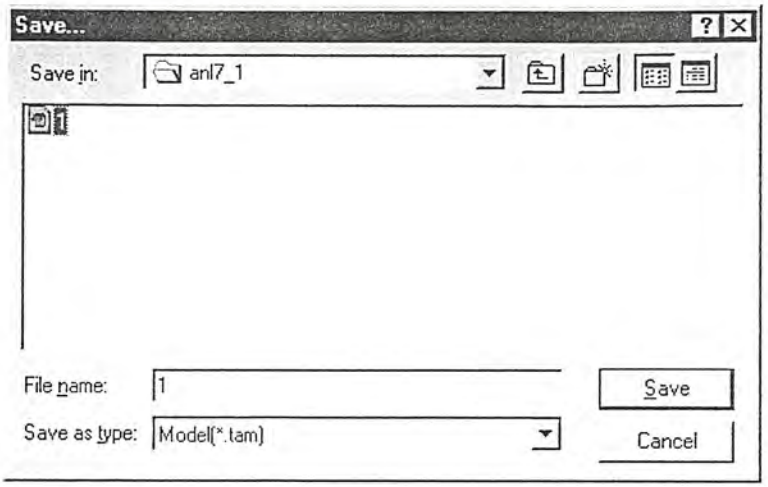


รูปที่ ค-12 แสดงรูปหน้าจอเมื่อมีการเลือกไดเมนชัน และตัววัดได้ตรงตามความต้องการแล้ว

เมื่อได้ทำการเพิ่มฟิลด์ของตารางแฟ็ก และตารางไดเมนชัน ได้ครบตามความต้องการแล้ว หน้าจอที่ได้จะมีลักษณะดังรูปที่ ค-12

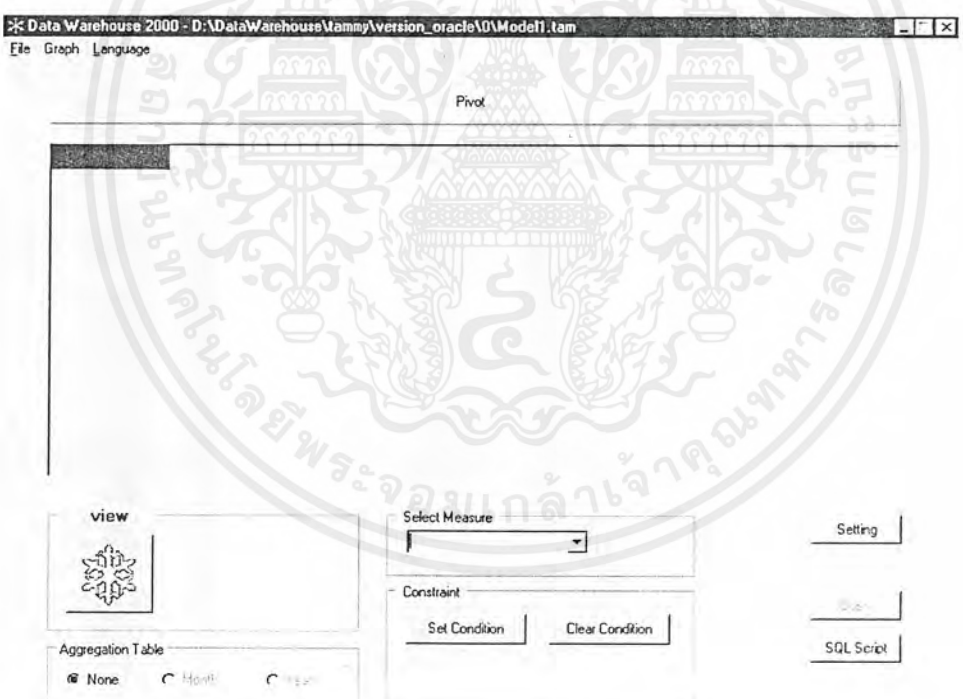
หลักจากที่สร้างโมเดลเสร็จเรียบร้อยผู้ใช้สามารถ Save เป็นอีกชื่อหนึ่งได้ โดยการกด Save As... หรือถ้าไม่ต้องการบันทึกเป็นอีกชื่อหนึ่ง ก็สามารถคลิกที่ปุ่ม Save เพื่อบันทึกชื่อเดิมที่ได้ตั้งไว้ในครั้งแรกที่ทำการสร้างโมเดลใหม่ แล้วจึงออกจากหน้าจอนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค-13 แสดงรูปหน้าจอเมื่อคลิกที่ปุ่ม Save หรือ Save As...

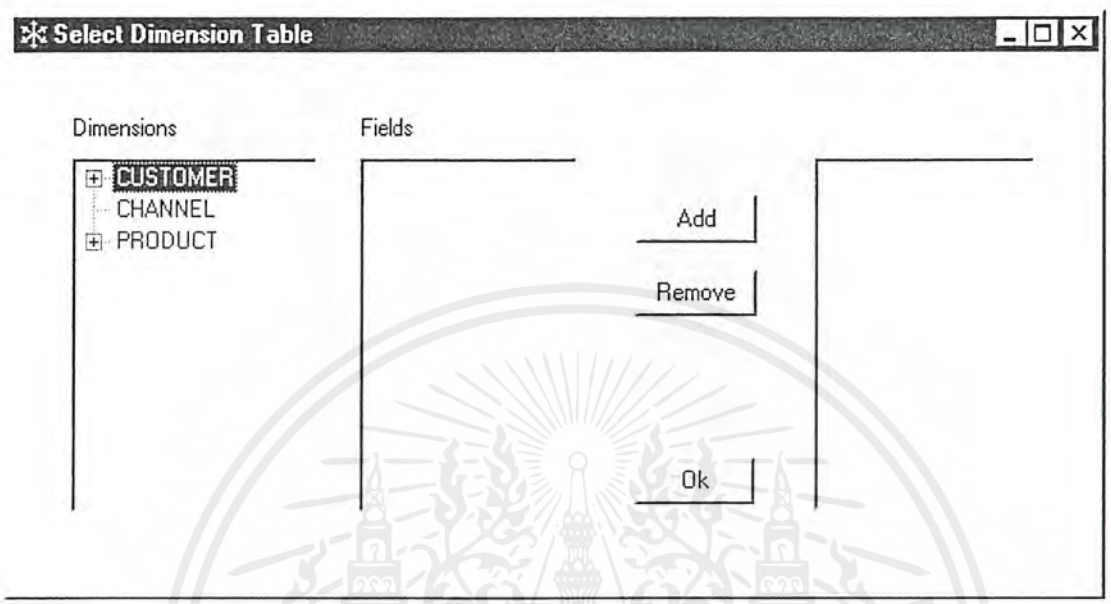
หน้าจอที่ 5 : เป็นหน้าจอสำหรับการวิเคราะห์โมเดลที่เคยสร้างเอาไว้แล้ว โดยการเลือกเมนู File > Load Model จากนั้นก็จะขึ้นเป็นไดอะลอก Open ขึ้นมา ผู้ใช้สามารถเลือกชื่อไฟล์ที่ต้องการจะนำมาวิเคราะห์ได้ เมื่อคลิกที่ปุ่ม Open แล้ว หน้าจอที่ปรากฏจะมีลักษณะ ดังรูปที่ ค-14 ชื่อไฟล์ที่ผู้ใช้เลือกจะปรากฏอยู่ที่มุมบนซ้ายของฟอร์มนี้



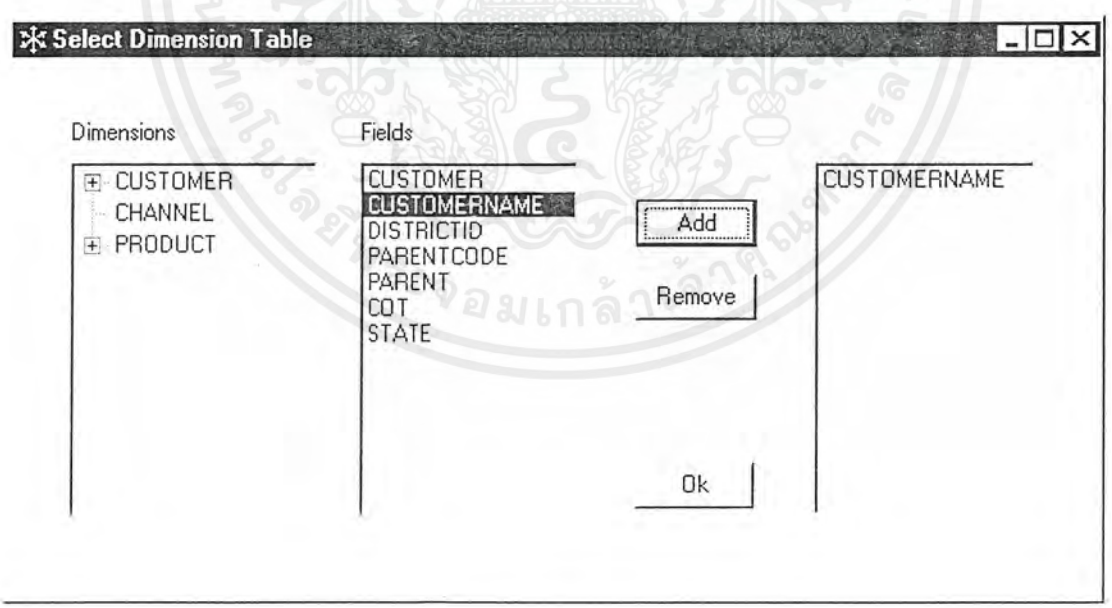
รูปที่ ค-14 แสดงรูปหน้าจอเมื่อมีการเลือกไฟล์ที่จะนำมาใช้ในการวิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอที่ 6 : หลังจากที่ปรากฏหน้าจอดังรูปที่ ค-14 เรียบร้อยแล้ว ผู้ใช้จะต้องคลิกที่ปุ่ม View เพื่อเลือกฟิลด์ของตารางไดเมนชันที่ต้องการนำมาใช้ในการวิเคราะห์ ก็จะปรากฏหน้าจอดังรูปที่ ค-15 จะเห็นว่า มีชื่อไดเมนชันที่เราได้สร้างเอาไว้ปรากฏอยู่ที่ Dimension จากนั้นผู้ใช้สามารถเลือกฟิลด์ที่ต้องการนำมาวิเคราะห์ได้ โดยการคลิกไดเมนชั่น Customer แล้วเลือกชื่อฟิลด์ที่จะให้แสดงข้อมูล แล้วคลิกที่ปุ่ม Add ชื่อฟิลด์นั้นก็จะไปปรากฏอยู่ใน Listbox ดังรูปที่ ค-16



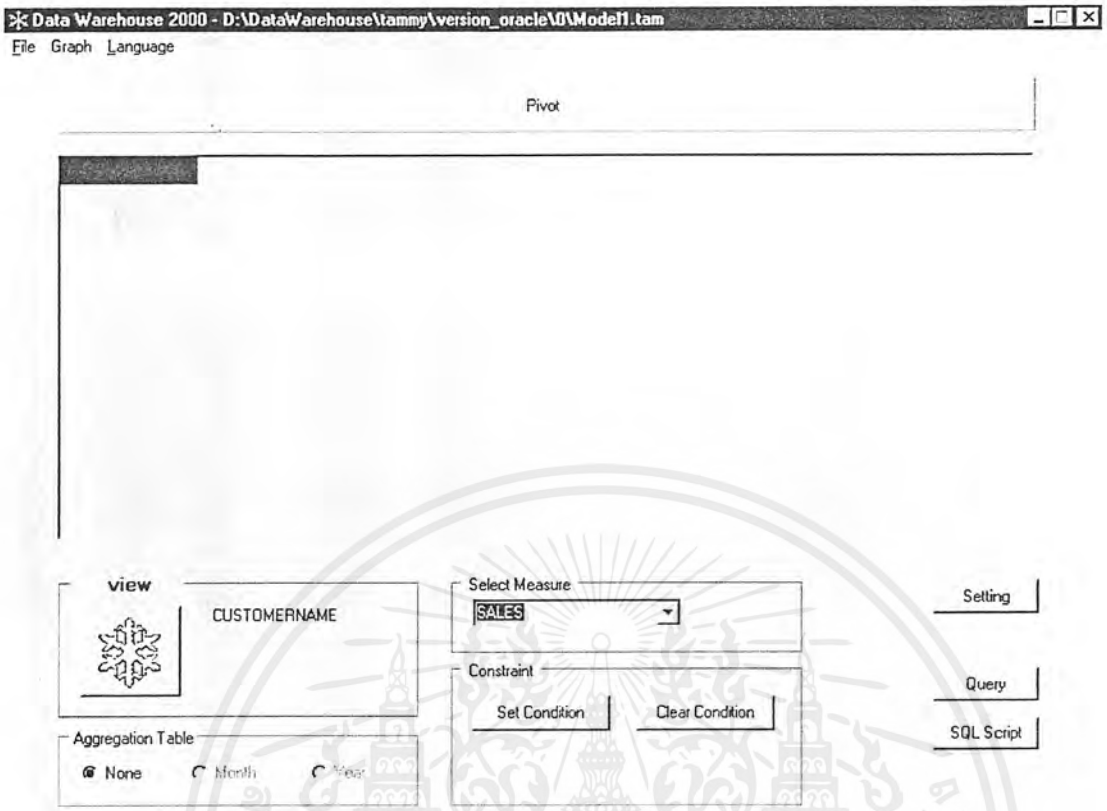
รูปที่ ค-15 แสดงรูปหน้าจอเมื่อมีการคลิกที่ปุ่ม View



รูปที่ ค-16 แสดงรูปหน้าจอเมื่อมีการเลือกฟิลด์ของตารางไดเมนชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

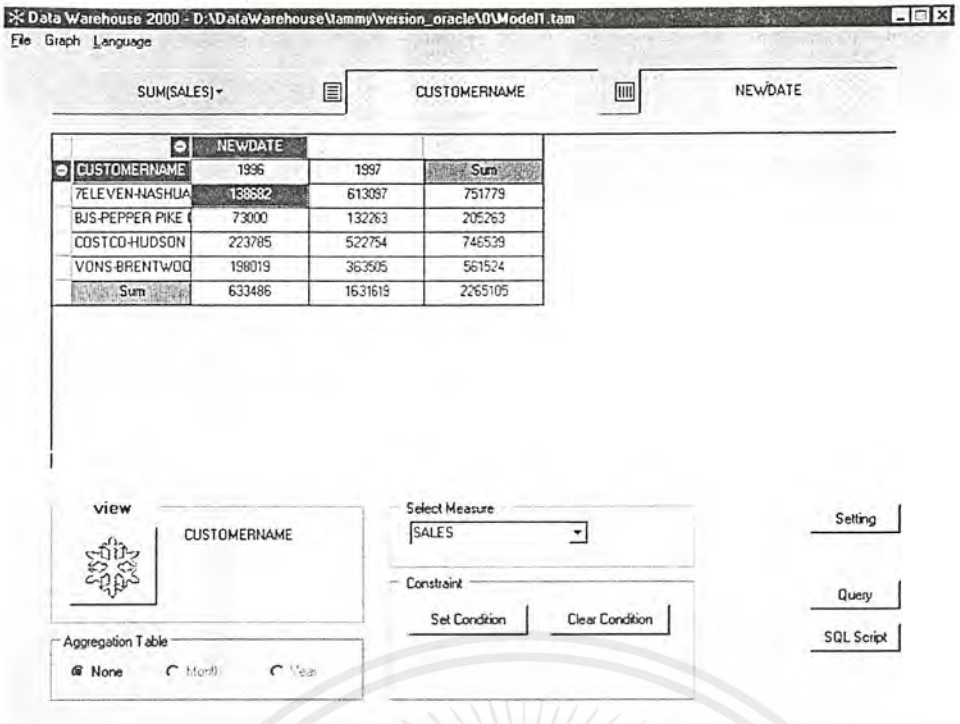
เมื่อเลือกฟิลด์เรียบร้อยแล้ว ก็คลิกที่ปุ่ม OK จะกลับมาสู่หน้าจอในรูปที่ ค-13 จากนั้นผู้ใช้จะต้องเลือก Measure ที่ต้องการใช้เป็นตัววัดทางธุรกิจ ดังตัวอย่างรูปที่ ค-16 จะเลือกตัววัดเป็น Sales



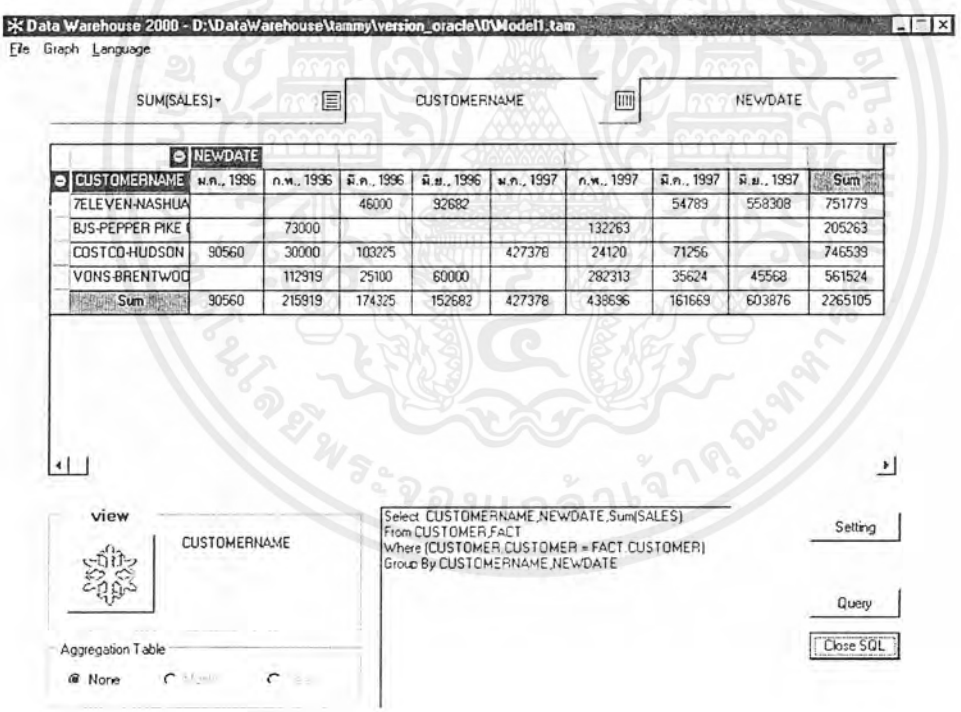
รูปที่ ค-17 แสดงรูปหน้าจอเมื่อมีการเลือกตัววัดทางธุรกิจ

เมื่อคลิกที่ปุ่ม Query จะแสดงข้อมูลออกมาทาง DecisionGrid เป็นดังรูปที่ ค-18 ผู้ใช้สามารถคลิกที่ปุ่ม Show SQL เพื่อดู SQL statement ที่ได้จากการเลือกฟิลด์ของไดเมนชัน และตัววัดได้ดังรูปที่ ค-19 ซึ่งจะเห็นได้ว่ารูปที่ ค-19 ต่างกับรูปที่ ค-18 ตรง time frame คือรูปที่ ค-18 จะสรุปข้อมูลตามรายปี ส่วนรูปที่ ค-19 เป็นการสรุปข้อมูลตามรายเดือน ซึ่งผู้ใช้สามารถกำหนด Time frame ได้เอง โดยการคลิกที่ปุ่ม Setting ก็จะปรากฏไดอะลอก ดังรูปที่ ค-20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

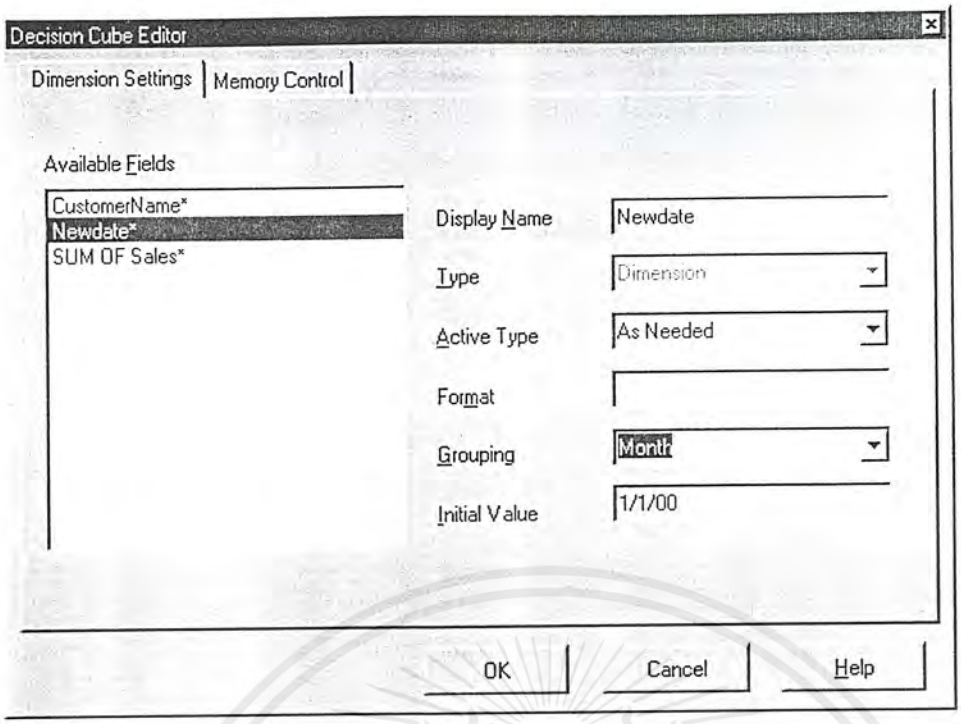


รูปที่ ค-18 แสดงรูปหน้าจอเมื่อมีการ Query ข้อมูลที่ได้ทำการเลือกไว้สรุปตามรายปี



รูปที่ ค-19 แสดงรูปหน้าจอเมื่อมีการเลือกที่ปุ่ม Show SQL เป็นข้อมูลที่สรุปตามรายเดือน

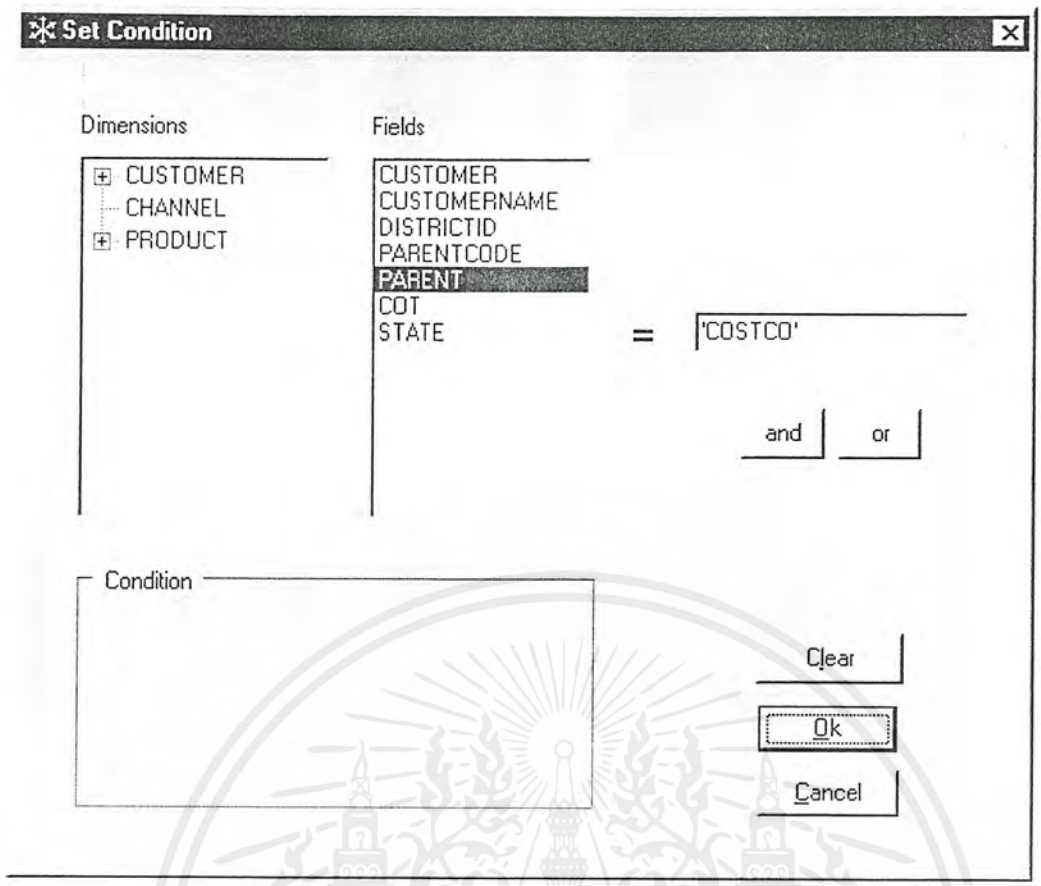
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค-20 แสดงหน้าจอการกำหนดค่า Time frame ให้สรุปข้อมูลเป็นรายเดือน

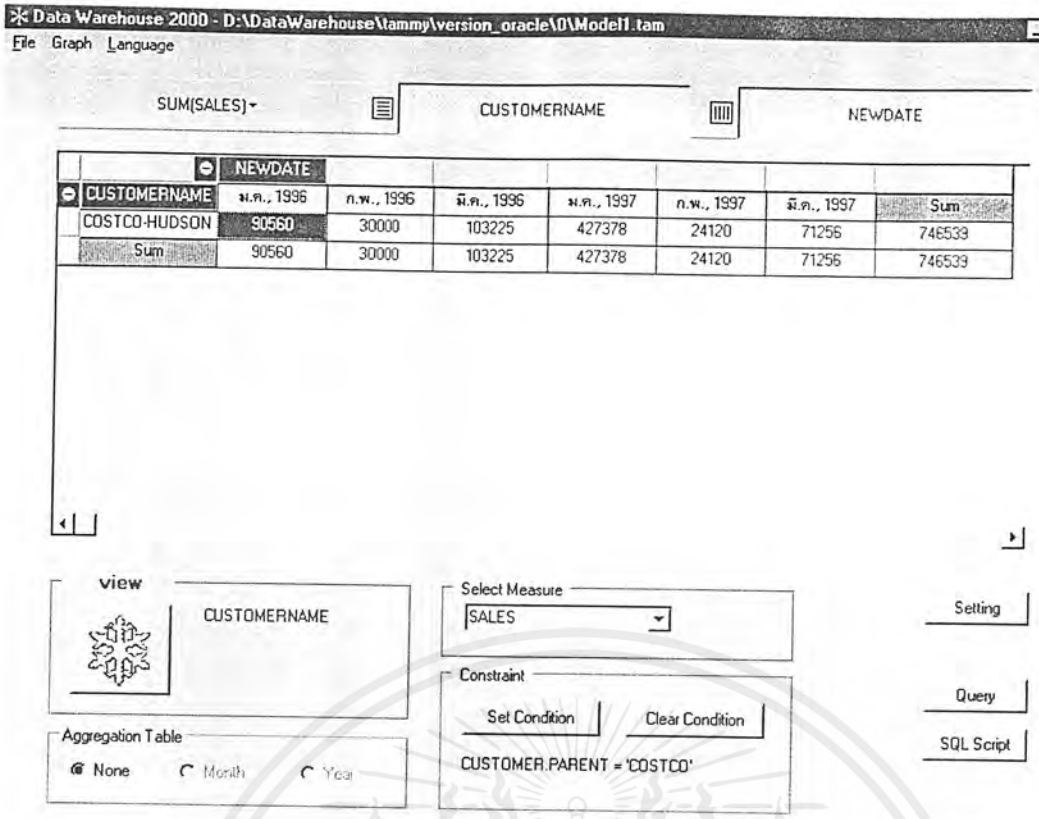
หน้าจอที่ 7 : เมื่อมีการคลิกที่ปุ่ม Condition หน้าจอนี้ก็จะปรากฏขึ้น เพื่อให้ผู้ใช้สามารถกำหนดเงื่อนไขในการวิเคราะห์ข้อมูลได้ โดยเงื่อนไขนี้จะเป็นเงื่อนไขสำหรับแต่ละแถวในตารางไดเมนชัน จากรูปที่ ค-21 เป็นการกำหนดเงื่อนไขให้กับฟิลด์ Parent ของตาราง Customer ให้มีค่าเท่ากับ 'COSTCO' เมื่อทำการกำหนดเงื่อนไขแล้วก็คลิกที่ปุ่ม OK เพื่อกลับสู่หน้าจอหลัก จากนั้น ก็คลิกที่ปุ่ม Query เพื่อทำการ query เงื่อนไขที่ได้กำหนดไว้ จะได้ดังรูปที่ ค-22 จากรูปจะเป็นว่าเป็นการแสดงยอดขายของ Parent = COSTCO ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค-21 แสดงหน้าจอสำหรับการกำหนดเงื่อนไขในฟิลด์ Parent ของไดเมนชัน Customer

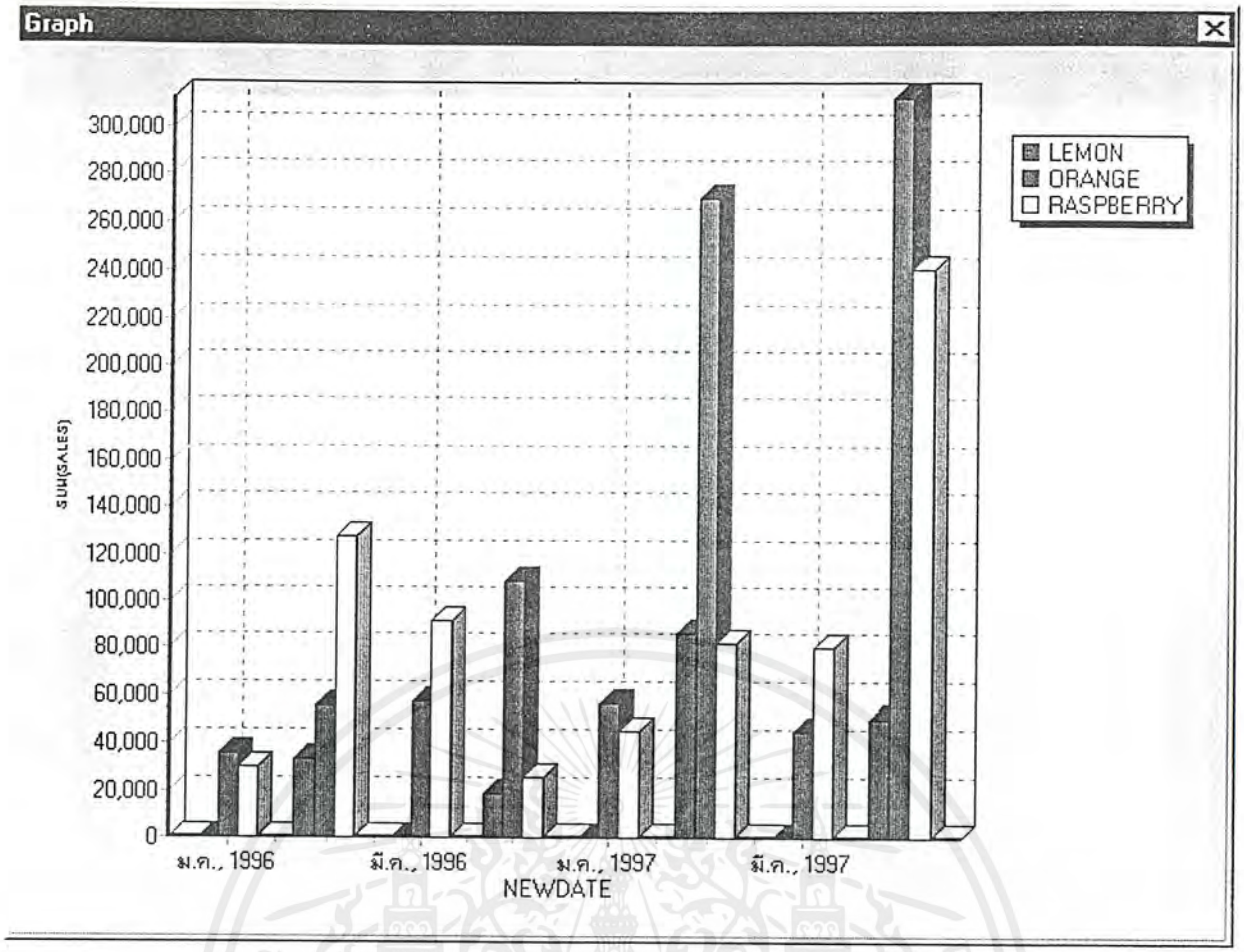
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค-22 แสดงหน้าจอสำหรับการ Query ตามเงื่อนไขที่ได้กำหนดไว้

หน้าจอที่ 8 : เมื่อผู้ใช้ต้องการดูแนวโน้มของข้อมูลที่จะวิเคราะห์ ก็สามารถทำได้โดยการคลิกที่เมนู Graph เพื่อให้ข้อมูลที่ต้องการวิเคราะห์นั้นแสดงอยู่ในรูปของกราฟ เพื่อให้เห็นถึงแนวโน้มที่ชัดเจนมากยิ่งขึ้น ดังรูปที่ ค-23 เป็นกราฟที่แสดงข้อมูลยอดขายที่สรุปตามรายเดือนสำหรับ Flavor ของสินค้าต่างๆ ดังรูปจะมี Lemon, Orange และ Raspberry

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

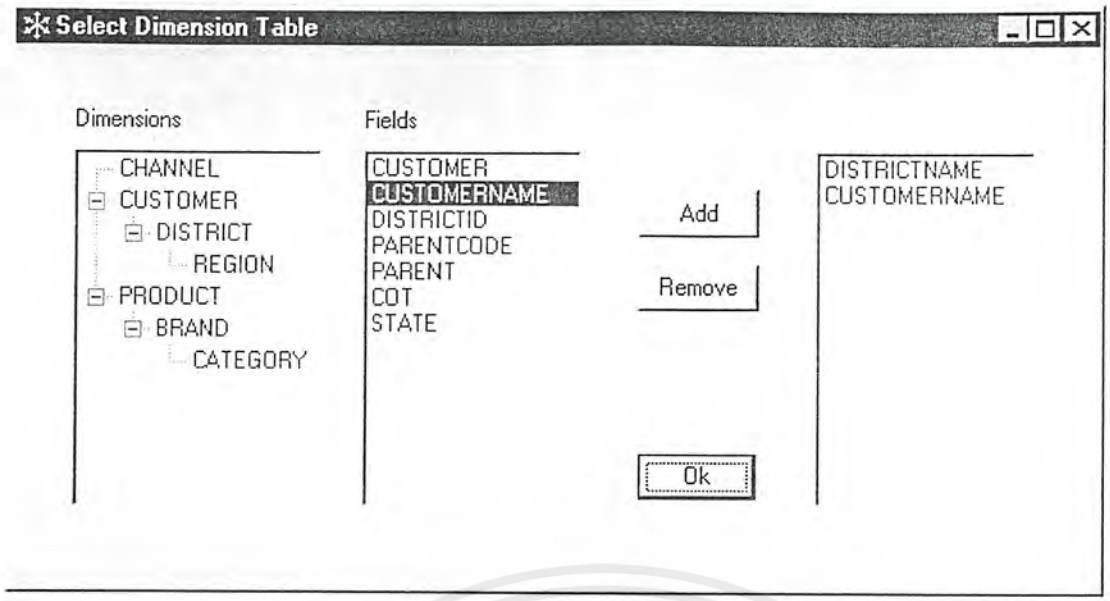


รูปที่ ค-23 แสดงกราฟของราคาขายสำหรับ Flavor ของสินค้าต่างๆ ในแต่ละเดือน

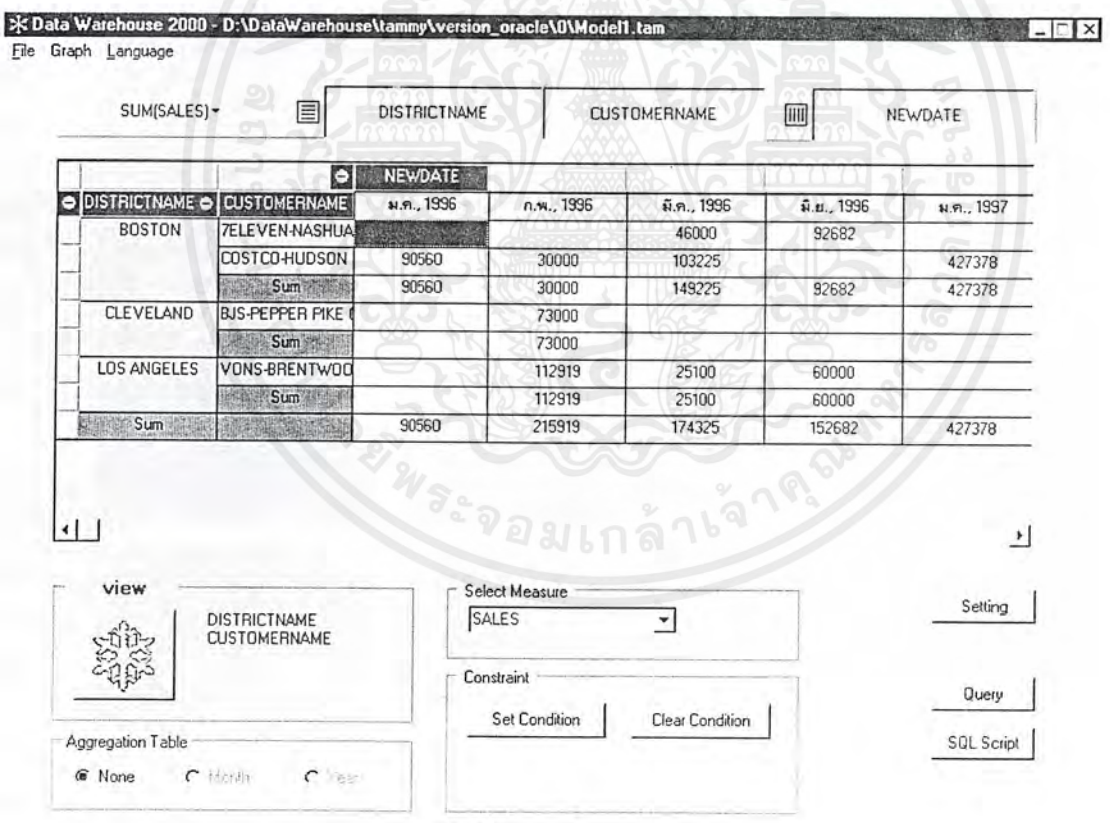
เมื่อผู้ใช้ต้องการ Drill Down ก็สามารถทำได้โดยการคลิกที่ปุ่ม View ก็จะปรากฏ หน้าจอดังรูปที่ ค-24 จะเห็นได้ว่า เราสามารถ drill down จาก DistrictName ให้เป็น CustomerName ได้ เมื่อเลือกแล้วก็ทำการวิเคราะห์โดยคลิกที่ปุ่ม Query จะได้หน้าจอดังรูปที่ ค-25

สำหรับการ Drill Up สามารถทำได้โดยการคลิกที่ปุ่ม View จะปรากฏหน้าจอดังรูปที่ ค-26 ผู้ใช้สามารถ drill up DistrictName โดยการเลือกลำดับชั้นที่สูงกว่า นั่นคือ RegionName เมื่อเลือกแล้วก็ทำการ Query จะปรากฏหน้าจอดังรูปที่ ค-27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

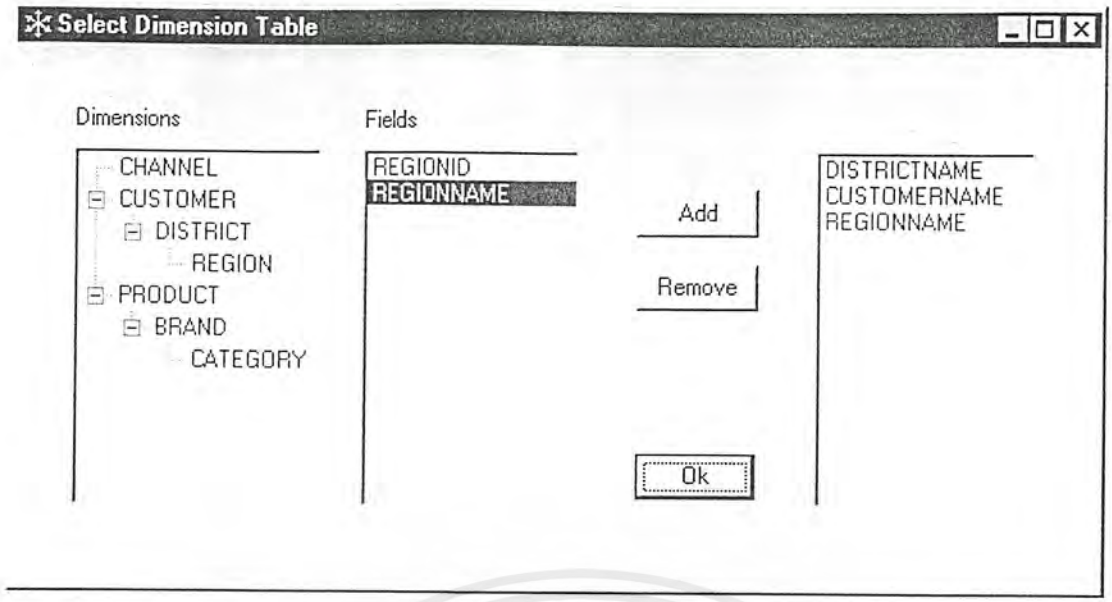


รูปที่ ค-24 การ Drill down จาก DistrictName ลงมาเป็น CustomerName

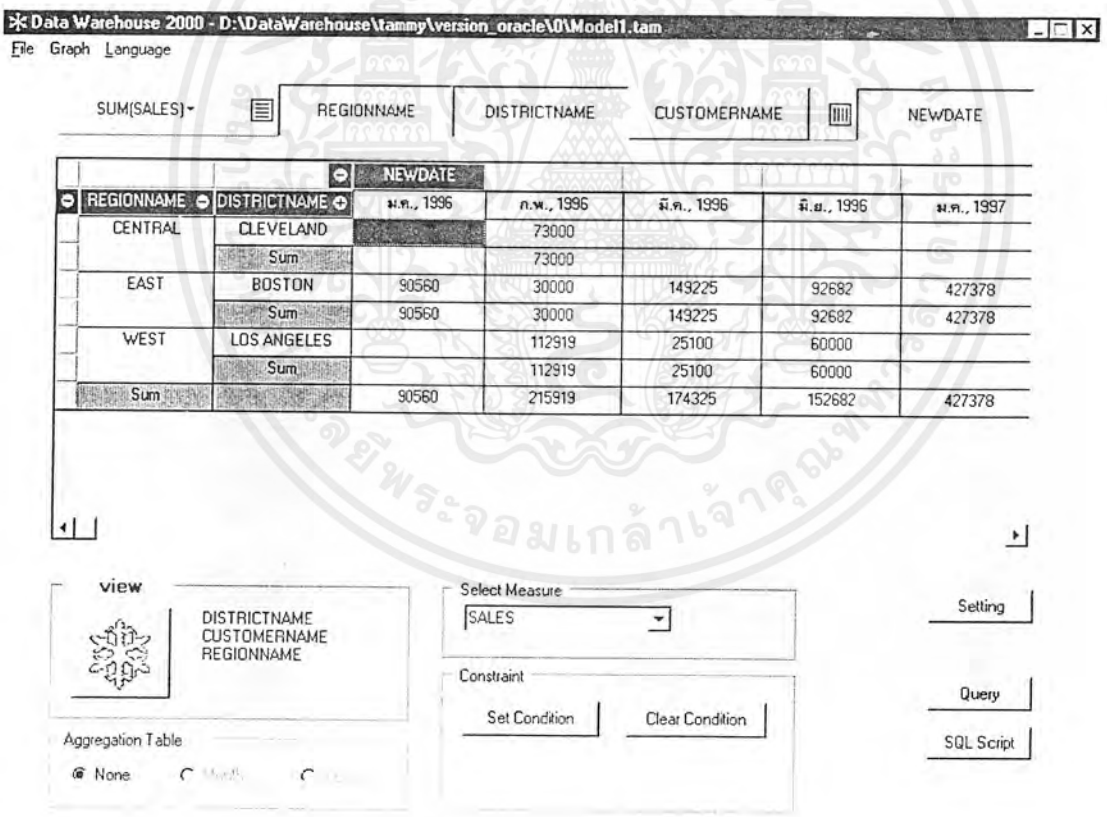


รูปที่ ค-25 แสดงหน้าจอการวิเคราะห์ข้อมูลที่ได้มาจากการ Drill down จาก DistrictName ไปเป็น CustomerName

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



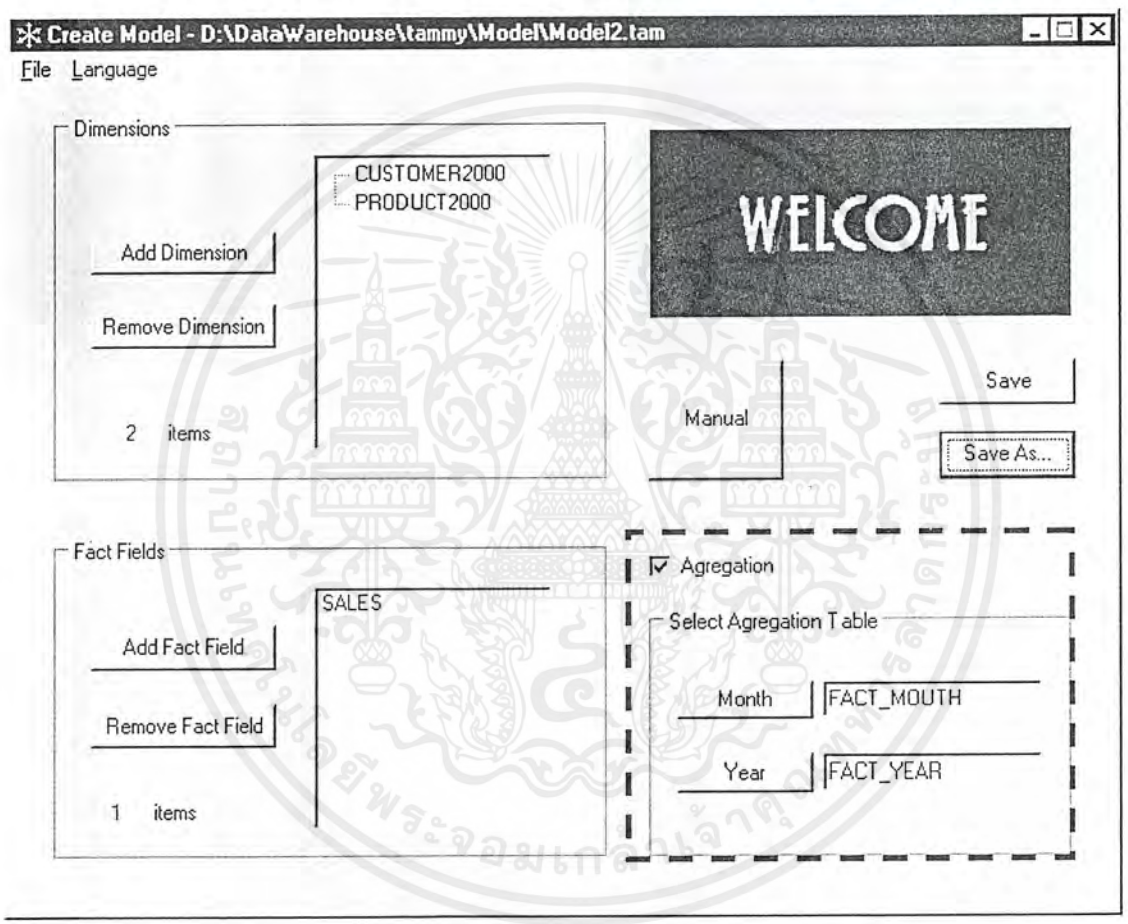
รูปที่ ค-26 การ Drill up จาก DistrictName ขึ้นไปเป็น RegionName



รูปที่ ค-27 แสดงหน้าจอการวิเคราะห์ข้อมูลที่ได้มาจากการ Drill up จาก BrandName ไปเป็น CategoryName

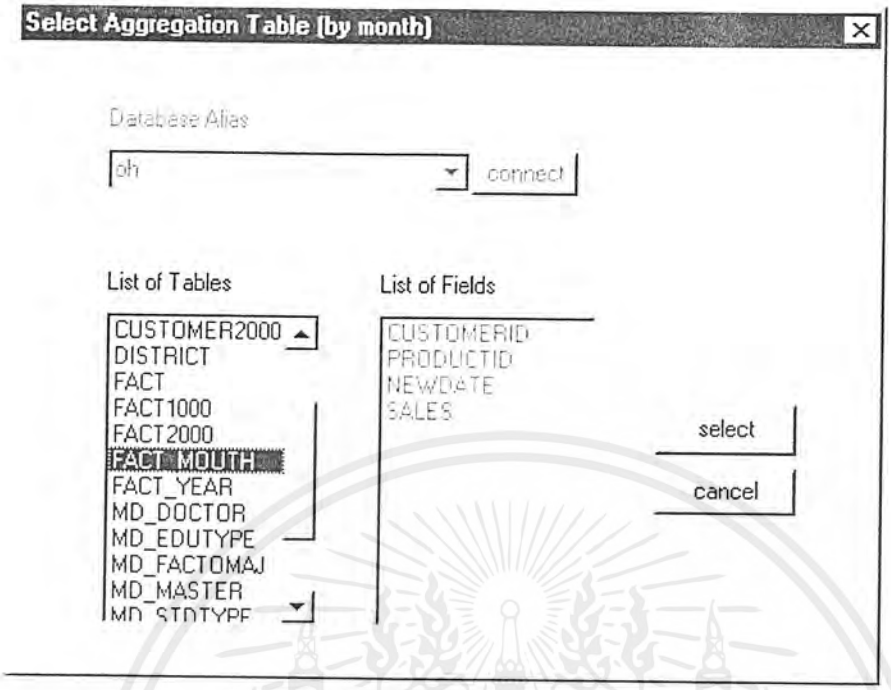
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการวิเคราะห์ข้อมูลขนาดใหญ่ที่มีจำนวนเรคอร์ดหลายแสนหลายล้านเรคอร์ดนั้นจำเป็นจะต้องมีการทำ Aggregation fact table เอาไว้ล่วงหน้า เพื่อเร่งการทำ query ให้มีความรวดเร็วมากยิ่งขึ้น ดังนั้น โปรแกรมประยุกต์ที่ได้ทำการพัฒนาขึ้นมาจึงมีส่วนที่ช่วยสนับสนุน aggregation fact table เอาไว้ โดยเมื่อผู้ใช้ต้องการที่จะเลือก aggregation fact table ก็สามารถเลือก option aggregation ดังรูปที่ ค-28 โดยโปรแกรมประยุกต์จะให้ผู้ใช้เลือก aggregation fact table ที่เป็นรายเดือน หรือเป็นรายปีได้จากการคลิกที่ปุ่ม Month หรือ Year ตามลำดับ และจะเข้าสู่หน้าจอที่ให้เลือก fact table ดังรูปที่ ค-29 และรูปที่ ค-30

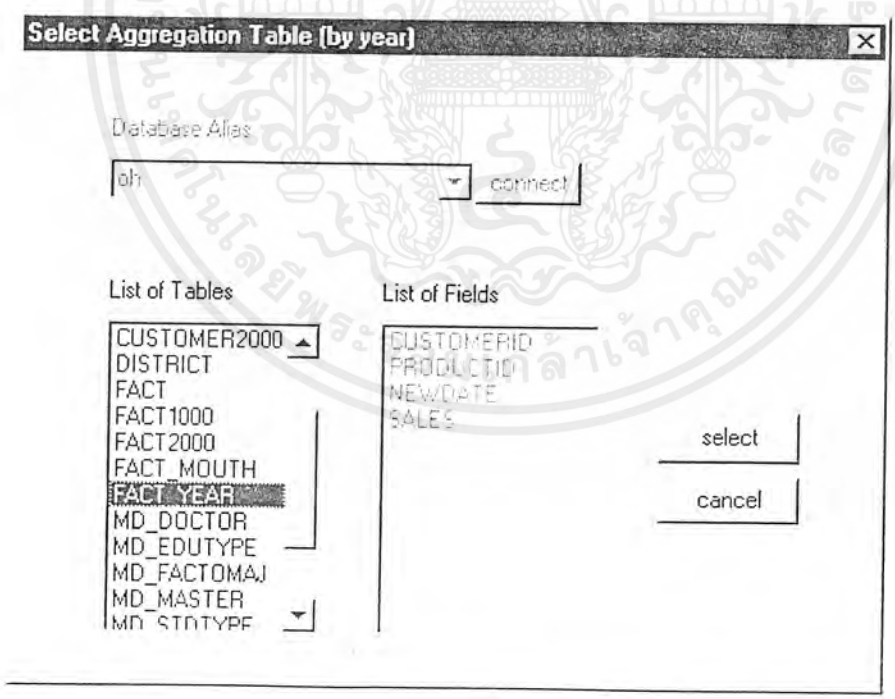


รูปที่ ค-28 แสดงหน้าจอในการเลือก option aggregation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



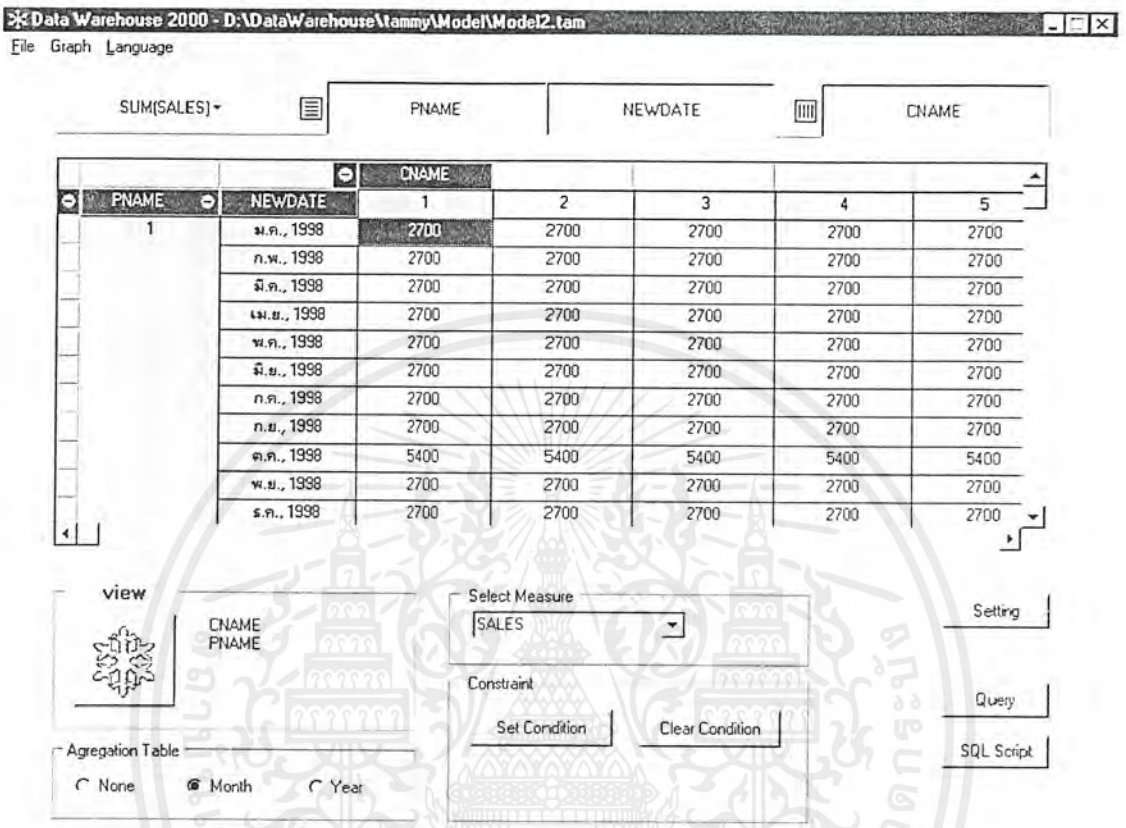
รูปที่ ค-29 แสดงหน้าจอสำหรับเลือก aggregation fact table สำหรับรายเดือน



รูปที่ ค-30 แสดงหน้าจอสำหรับเลือก aggregation fact table สำหรับรายปี

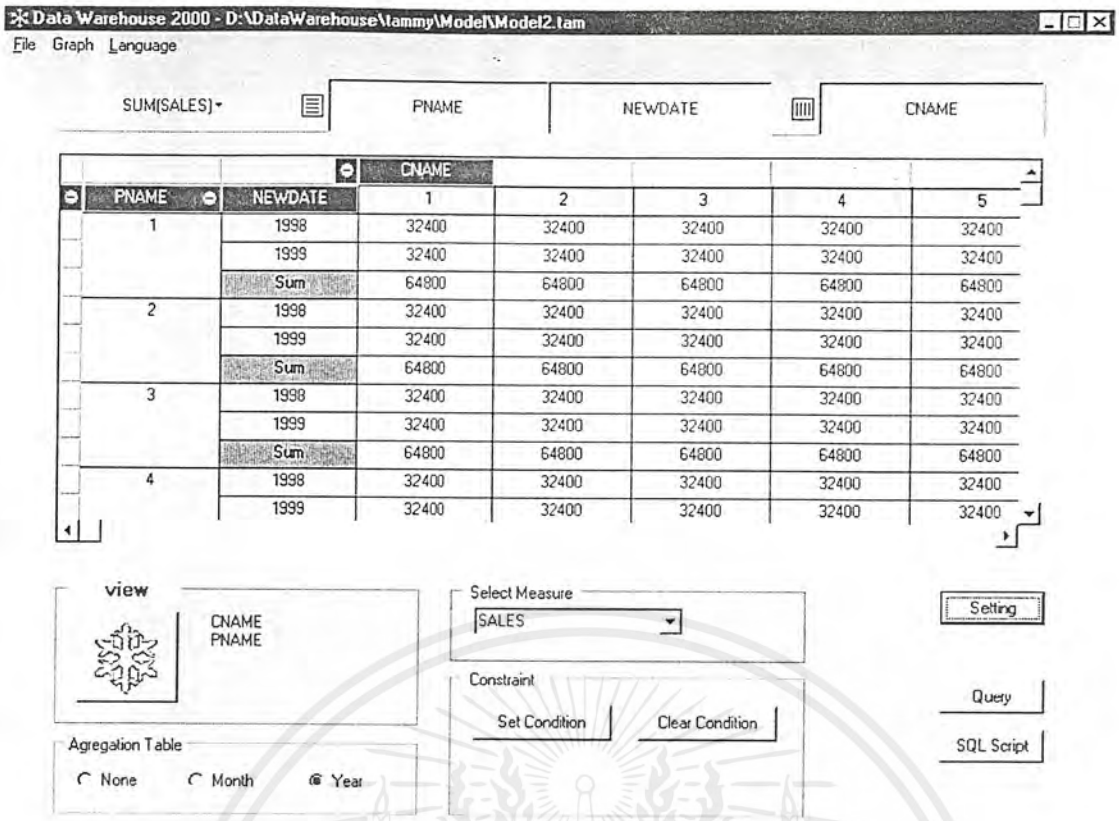
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเลือกเสร็จเรียบร้อยแล้ว ก็จะปรากฏหน้าจอดังรูปที่ ค-28 หลังจากนั้นก็ทำการบันทึกโมเดลที่ได้สร้างไว้ เพื่อนำไปวิเคราะห์ สำหรับหน้าจอของการวิเคราะห์ ก็จะมีส่วนของ Aggregation Table เพิ่มขึ้นมาทำให้ผู้ใช้สามารถเลือกได้ว่าจะดูตามรายเดือนหรือรายปี โดยเลือกจาก Month หรือ Year ตามลำดับ ดังรูปที่ ค-31 และรูปที่ ค-32



รูปที่ ค-31 แสดงหน้าจอสำหรับการวิเคราะห์ข้อมูลตามรายเดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค-32 แสดงหน้าจอสำหรับการวิเคราะห์ข้อมูลตามรายปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง
คู่มือสำหรับโปรแกรมเมอร์

ฟอร์มทั้งหมดที่ออกแบบสำหรับการใช้งาน

หน้าจอที่ใช้งานจะประกอบไปด้วยฟอร์มทั้งหมด 8 ฟอร์ม คือ

Form1 : เป็นฟอร์มแรกที่จะแสดงเมื่อทำการรันโปรแกรม

Form2 : เป็นฟอร์มที่ใช้ในการสร้างโมเดลและเปิดโมเดลที่ทำการสร้างไว้แล้วมาแก้ไข

Form3 : เป็นฟอร์มที่สำหรับทำการเพิ่มตารางไดเมนชั่น การเพิ่มตัววัด หรือการเพิ่มตาราง aggregation

Form4 : เป็นฟอร์มที่ใช้สำหรับการดึงโมเดลที่ได้สร้างไว้ขึ้นมาทำการวิเคราะห์

Form5 : เป็นฟอร์มที่ใช้ในการเลือกฟิลด์จากตารางไดเมนชั่นที่ต้องการให้เป็นมุมมองในการวิเคราะห์

Form6 : เป็นฟอร์มที่ใช้ในการกำหนดเงื่อนไขสำหรับ Query

Form7 : เป็นฟอร์มที่ใช้ในการเลือกตารางที่เป็น Snowflake

Form8 : เป็นฟอร์มที่ใช้ในการแสดงข้อมูลในลักษณะกราฟ

รายละเอียดโปรแกรมย่อย

Unit1

Procedure และ Function ต่างๆ

1. procedure TForm1.Image1Click(Sender: TObject);
   - เมื่อมีการคลิกที่รูป Image1 ก็จะหยุดการทำงานของ timer1
   - เข้าสู่ฟอร์ม 4
   - ฟอร์ม 1 ก็จะถูกปิดไป

2. procedure TForm1.Image3Click(Sender: TObject);
   - เมื่อมีการคลิกที่รูป Exit ก็จะหยุดการทำงานของ timer1
   - ปิดฟอร์ม 1 และจบการทำงานของแอปพลิเคชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. procedure TForm1.Timer1Timer(Sender: TObject);

- เมื่อเกิด timer แต่ละครั้งก็จะทำการเลื่อน label1 ไปทางซ้ายเรื่อยๆ และตรวจสอบดูว่า label1 เลยขอบจอหรือไม่ ถ้าเลยพ้นขอบจอด้านซ้ายไปแล้ว ก็ให้กลับไปเริ่มต้นใหม่ที่ขอบจอด้านขวา

Unit2

Type ต่างๆ

1. TAddStatus = (AddDimension,AddFact,AddAgreFactMonth,AddAgreFactYear);

ตัวแปรต่างๆ

1. AddStatus :TAddStatus;	ใช้ระบุว่าฟอร์ม 3 จะถูกใช้งานสำหรับการเพิ่มอะไร
2. SaveStr :string;	ใช้เก็บข้อมูลเกี่ยวกับโมเดลที่จะเก็บบันทึกลงไฟล์
3. SaveReady :boolean;	ใช้ตรวจสอบว่าเคยมีการบันทึกโมเดลไปแล้วหรือยัง
4. LoadReady :boolean;	ใช้ตรวจสอบว่าการโหลดโมเดลขึ้นมาสำเร็จหรือไม่
4. ModelDatabaseName :string;	ใช้เก็บชื่อฐานข้อมูลของโมเดล
5. ModelFileHandle : textfile;	ใช้เป็น File Handle ในการติดต่อกับไฟล์ (อ่าน/เขียน)
6. ModelFileName :string;	ใช้เก็บชื่อไฟล์ที่จะใช้ในการเก็บบันทึกข้อมูลของโมเดล
7. ModelTVFileName :string;	ใช้เก็บชื่อไฟล์ที่จะใช้ในการเก็บบันทึกข้อมูลของโมเดล
8. DimTVSelected :boolean;	ใช้ตรวจสอบว่ามีการคลิกที่ DimTV หรือไม่

Procedure และ Function ต่างๆ

1. procedure TForm2.New1Click(Sender: TObject);

- รับชื่อไฟล์จากผู้ใช้
- นำชื่อไฟล์มาแปลงเป็น ไฟล์.tam เก็บไว้ในตัวแปร ModelFileName
- นำชื่อไฟล์มาแปลงเป็น ไฟล์.tam~ เก็บไว้ในตัวแปร ModelTVFilename
- สร้างฟอร์ม 3 ขึ้นมาใหม่
- กำหนดค่าเริ่มต้นให้กับตัวแปรและคอมโพเนนต์ต่างๆ

2. procedure TForm2.BtAddDimClick(Sender: TObject); (การเพิ่มตารางไดเมนชั่น)

- เมื่อมีการคลิกปุ่ม Add Dimension ก็จะกำหนดสถานะการ Add ว่าขณะนี้เป็นการเพิ่มตารางได้เมนชั่น (AddDimension)
- แสดงฟอร์ม 3 ขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. procedure TForm2.BtRemDimClick(Sender: TObject); (การลบตารางไดเมนชั่น)
   - ตรวจสอบว่าถ้าใน DimTV มีข้อมูลอยู่และมีการเลือก node ใดๆใน DimTV แล้วก็ให้ลบ node นั้นออกจาก DimTV

4. procedure TForm2.BtAddFactClick(Sender: TObject); (การเพิ่มตัววัด)
   - เมื่อมีการคลิกปุ่ม AddFact ก็จะกำหนดสถานะการ Add ว่าขณะนี้เป็นการเพิ่มฟิลด์ที่เลือกจาก Fact Table (AddFact)
   - แสดงฟอร์ม 3 ขึ้นมา

5. procedure TForm2.BtRemFactClick(Sender: TObject); (การลบตัววัด)
   - ตรวจสอบว่าถ้าใน FactFieldListbox มีข้อมูลอยู่และมีการเลือก item ใดๆใน FactFieldListbox แล้วก็ให้ลบ item นั้นออกจาก FactFieldListbox และ FactTableListbox

6. procedure TForm2.BtSaveClick(Sender: TObject);
   - ถ้ามีการคลิกปุ่ม Save ให้ทำการตรวจสอบสถานะจากตัวแปร SaveReady ว่าโมเดลนั้นเคยมีการ Save ไปแล้วหรือยัง
   - ถ้ายังไม่เคยมีการ save ก็ให้ผู้ใช้เลือก path ที่ใช้เก็บบันทึกไฟล์ จากนั้นก็ให้ทำการ SaveModel ถ้าตรวจสอบว่าเคยมีการ save ไปแล้วก็ให้ทำการ SaveModel นั้นทับไฟล์เดิมได้เลย

7. procedure TForm2.SaveClick(Sender: TObject);
   - การ Save โดยเลือกที่เมนู ให้ทำการตรวจสอบสถานะจากตัวแปร SaveReady ว่าโมเดลนั้นเคยมีการ Save ไปแล้วหรือยัง
   - ถ้ายังไม่เคยมีการ save ก็ให้ผู้ใช้เลือก path ที่ใช้เก็บบันทึกไฟล์ จากนั้นก็ให้ทำการ SaveModel ถ้าตรวจสอบว่าเคยมีการ save ไปแล้วก็ให้ทำการ SaveModel นั้นทับไฟล์เดิมได้เลย

8. procedure TForm2.BtSaveAsClick(Sender: TObject);
   - เมื่อมีการคลิกที่ปุ่ม Save As จะให้ผู้ใช้เลือกชื่อไฟล์ใหม่และ path ที่จะทำการเก็บบันทึกไฟล์นั้น
   - ทำการ SaveModel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. procedure TForm2.SaveAS1Click(Sender: TObject);

- การ Save As โดยเลือกที่เมนู จะให้ผู้ใช้เลือกชื่อไฟล์ใหม่และ path ที่จะทำการเก็บบันทึกไฟล์นั้น
- ทำการ SaveModel

10. procedure TForm2.Exit1Click(Sender: TObject);

- เมื่อมีการเลือก Exit ที่เมนู ก็จะปิดฟอร์ม 3

11. procedure TForm2.SaveModel;

- ข้อมูลเกี่ยวกับโมเดลที่จะเก็บบันทึกลงไฟล์มีรูปแบบดังนี้ ซึ่งจะเก็บอยู่ในตัวแปร SaveStr

:DBName:
ค่าในตัวแปร ModelDatabaseName ← ชื่อฐานของมูลของโมเดล
:FactMonth:
ค่าใน EditMonth.Text ← ชื่อตาราง Aggregation แบบรายเดือน
:FactYear:
ค่าใน EditYear.Text ← ชื่อตาราง Aggregation แบบรายปี
:FactField:
Item ต่างๆ ใน FactFieldListbox ← รายการตัววัดต่างๆ ซึ่งเป็นฟิลด์ของ Fact Table
:FactTable:
Item ต่างๆ ใน FactTableListbox ← รายการ Fact Table ต่างๆ ที่สัมพันธ์กับตัววัดแต่ละตัว

- ทำการเก็บบันทึกข้อมูลในตัวแปร SaveStr ลงไฟล์ที่มีชื่อระบุอยู่ในตัวแปร ModelFileName
- ทำการเก็บบันทึกข้อมูลใน DimTV ซึ่งเป็นรายการของตารางไดเมนชั่นต่างๆ ลงไฟล์ที่มีชื่อระบุอยู่ในตัวแปร ModelTVFileName

12. procedure TForm2.LoadModel;

- ตรวจสอบว่ารูปแบบของไฟล์ถูกต้องหรือไม่ ถ้าถูกต้องก็ทำการโหลดโมเดล แต่ถ้าไม่ถูกต้องก็ไม่ต้องทำการโหลดโมเดล และกำหนดสถานะของการโหลดโมเดลไว้ในตัวแปร LoadReady ด้วย
- ทำการเปิดไฟล์ตามชื่อไฟล์ที่ระบุอยู่ในตัวแปร ModelFileName
  - อ่านชื่อฐานข้อมูลของโมเดลมาเก็บไว้ในตัวแปร ModelDatabaseName
  - อ่านชื่อตาราง Aggregation แบบรายเดือน มาเก็บไว้ใน EditMonth.Text

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- อ่านชื่อตาราง Aggregation แบบรายปี มาเก็บไว้ใน EditYear.Text
- อ่านรายการตัววัดต่างๆ ซึ่งเป็นฟิลด์ของ Fact Table มาเก็บไว้ในคอมโพเนนต์ FactFieldListbox
- อ่านรายการ Fact Table ต่างๆ ที่สัมพันธ์กับตัววัดแต่ละตัว มาเก็บไว้ในคอมโพเนนต์ FactTableListbox

- ทำการเปิดไฟล์ตามชื่อไฟล์ที่ระบุอยู่ในตัวแปร ModelTVFileName อ่านรายการของตารางไดเมนชั่นต่างๆ มาเก็บไว้ในคอมโพเนนต์ DimTV

13. procedure TForm2.Open1Click(Sender: TObject);

- ให้ผู้ใช้เลือกไฟล์
- นำชื่อ ไฟล์.tam มาเก็บไว้ในตัวแปร ModelFileName
- นำชื่อ ไฟล์.tam~ เก็บไว้ในตัวแปร ModelTVFilename
- ทำการโหลดโมเดลโดยเรียกโพรซีเดอร์ LoadModel
- ตรวจสอบสถานะของตัวแปร LoadReady ว่าโหลดโมเดลสำเร็จหรือไม่ ถ้าสำเร็จ ก็จะทำการกำหนดค่าเริ่มต้นต่างๆ ทำการสร้างฟอร์ม 3 ฟอร์ม 7 ขึ้นมาใหม่
- ถ้าการโหลดโมเดลไม่สำเร็จก็ให้แสดงข้อความเตือนว่ารูปแบบของไฟล์นั้นไม่ถูกต้อง

14. procedure TForm2.DimTVClick(Sender: TObject);

- ถ้ามีการคลิกที่ DimTV ก็ให้กำหนดสถานะของตัวแปร DivTVSelected เอาไว้เพื่อตรวจสอบว่ามีการเลือก node ใดๆ ใน DimTV หรือไม่

15. procedure TForm2.DimTVDblClick(Sender: TObject); (การเพิ่มตาราง Snowflake โดยอัตโนมัติ)

- ตรวจสอบดูว่าตารางไดเมนชั่นที่ต้องการเพิ่มตาราง Snowflake นั้น มีตารางที่เป็น Snowflake อยู่แล้วหรือยัง สามารถทำได้โดยการ ตรวจสอบว่า node ที่ถูกดับเบิ้ลคลิกใน DimTV นั้นมี children อยู่แล้วหรือยังถ้ามีอยู่แล้วก็จะไม่อนุญาตให้มีการเพิ่ม children อีก
- ให้ฟอร์ม 7 ทำการค้นหาตารางต่างๆ ที่มีความสัมพันธ์กับตารางที่เลือกใน DimTV โดยการเรียกโพรซีเดอร์ ListTable ซึ่งเป็นของฟอร์ม 7
- ถ้าตารางที่ค้นหามาได้ (ในฟอร์ม 7) มีเพียงตารางเดียว ก็ให้เพิ่มตารางนั้นเป็นตาราง Snowflake โดยอัตโนมัติ และทำการค้นหาเช่นนี้ต่อไปเรื่อยๆ
- ถ้าเมื่อใดตารางที่ค้นหามาได้ (ในฟอร์ม 7) มีมากกว่า 1 ตาราง ก็จะทำการแสดงฟอร์ม 7 นั้นขึ้นมาเพื่อให้ผู้ใช้ตัดสินในเลือกเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

16. procedure TForm2.manualClick(Sender: TObject);

- เมื่อมีการคลิกที่ปุ่ม Manual จะทำการตรวจสอบว่า node ใน DimTV ที่เลือกยังไม่มี children ก็ให้ให้ฟอร์ม 7 ทำการค้นหาตารางต่างๆ ที่มีความสัมพันธ์กับตารางที่เลือกใน DimTV โดยการเรียกโพรซีเดอร์ ListTable ซึ่งเป็นของฟอร์ม 7 แล้วให้แสดงฟอร์ม 7 ขึ้นมา

17. procedure TForm2.SwitchToThai;

- ตั้งค่า Caption ของคอมโพเนนต์ต่างๆ ของฟอร์ม 2 ให้เป็นคำอธิบายภาษาไทย

18. procedure TForm2.SwitchToEnglish;

- ตั้งค่า Caption ของคอมโพเนนต์ต่างๆ ของฟอร์ม 2 ให้เป็นคำอธิบายภาษาอังกฤษ

19. procedure TForm2.Thai1Click(Sender: TObject);

- ถ้ามีการเลือกภาษาเป็นภาษาไทยที่เมนู ก็ให้กำหนดค่าตัวแปร SwichMode ให้เป็น Thai จากนั้นก็เรียกโพรซีเดอร์ SwitchToThai ของทั้งฟอร์ม 2 และ ฟอร์ม 4

20. procedure TForm2.English1Click(Sender: TObject);

- ถ้ามีการเลือกภาษาเป็นภาษาอังกฤษที่เมนู ก็ให้กำหนดค่าตัวแปร SwichMode ให้เป็น English จากนั้นก็เรียกโพรซีเดอร์ SwitchToEnglish ของทั้งฟอร์ม 2 และ ฟอร์ม 4

21. procedure TForm2.FormShow(Sender: TObject);

- เมื่อแสดงฟอร์ม 2 ขึ้นมาต้องตรวจสอบขณะนั้นค่าในตัวแปร SwitchMode เป็นอะไร ถ้าอยู่ในโหมดภาษาอังกฤษก็ให้เรียกโพรซีเดอร์ SwitchToEnglish แต่ถ้าอยู่ในโหมดภาษาไทยก็ให้เรียกโพรซีเดอร์ SwitchToThai

22. procedure TForm2.AgreCheckBoxClick(Sender: TObject);

- ถ้ามีการเลือกที่ AgreCheckbox ก็ให้แสดงคอมโพเนนท์ที่เกี่ยวข้องสำหรับการเลือก Aggregation Table ขึ้นมา

23. procedure TForm2.BtMonthClick(Sender: TObject);

- เมื่อมีการคลิกปุ่ม Month ก็จะกำหนดสถานะการ Add ว่าขณะนี้เป็นการเพิ่มตาราง aggregation แบบรายเดือน (AddAgreFactMonth)
- แสดงฟอร์ม 3 ขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

24. procedure TForm2.BtYearClick(Sender: TObject);

- เมื่อมีการคลิกปุ่ม Year ก็จะกำหนดสถานะการ Add ว่าขณะนี้เป็นการเพิ่มตาราง aggregation แบบรายปี (AddAgreFactYear)
- แสดงฟอร์ม 3 ขึ้นมา

Unit3

Procedure และ Function ต่างๆ

1. procedure TForm3.FormCreate(Sender: TObject);

- เมื่อฟอร์ม 3 ถูกสร้างขึ้นมาใหม่ก็จะนำชื่อฐานข้อมูลทั้งหมดที่ได้ทำการสร้าง Alias ไว้ใน DBE Administrator มาแสดงในคอมโพเนนต์ AliasCombo

2. procedure TForm3.BtConnectClick(Sender: TObject);

- นำชื่อ Alias ของฐานข้อมูลที่ถูกเลือกจากคอมโพเนนต์ AliasCombo ไปเก็บในตัวแปร ModelDatabaseName
- กำหนดการ Enable ของคอมโพเนนต์ โดยการเรียกโพรซีเดอร์ EnableComponent
- ทำการแสดงรายชื่อตารางทั้งหมดของฐานข้อมูล โดยการเรียกโพรซีเดอร์ ListTable

3. procedure TForm3.TableListBoxClick(Sender: TObject);

- เมื่อมีการคลิกที่ item ใดๆ ใน TableListbox ก็จะทำการดึงรายชื่อฟิลด์ของชื่อตารางที่ถูกเลือกใน TableListbox จากฐานข้อมูลที่ระบุในตัวแปร ModelDatabaseName แล้วนำมาแสดงไว้ในคอมโพเนนต์ FieldListbox

4. procedure TForm3.BtSelectClick(Sender: TObject);

- ตรวจสอบสถานะของตัวแปร AddStatus ว่าขณะนั้นฟอร์ม 3 ถูกใช้งานสำหรับการ Add อะไร
- กรณีที่เป็นการเพิ่มตารางไดเมนชั่น
  - ทำการตรวจสอบว่าใน TableListbox มีข้อมูลอยู่ และมีการเลือกที่ item ใดๆ หรือไม่
  - ตรวจสอบว่าตารางไดเมนชั่นที่เลือกซ้ำกับตารางไดเมนชั่นที่เคยเลือกไปแล้วหรือไม่ ซึ่งสามารถทำได้โดยการตรวจสอบว่า item ของ TableListbox ที่ถูกเลือก ซ้ำกับ node ใดๆ ใน DimTV ของฟอร์ม 2 หรือไม่ ถ้าไม่ซ้ำก็ทำการ Add item ของ TableListbox ที่ถูกเลือกลงใน DimTV ของฟอร์ม 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ปิดฟอร์ม 3

- กรณีที่เป็นการเพิ่มตัววัด
  - ทำการตรวจสอบว่าใน FieldListbox มีข้อมูลอยู่ และมีการเลือกที่ item ใดๆ หรือไม่
  - ตรวจสอบว่าตัววัดที่เลือกซ้ำกับตัววัดที่เคยเลือกไปแล้วหรือไม่ ซึ่งสามารถทำได้โดยการตรวจสอบว่า item ของ FieldListbox ที่ถูกเลือก ซ้ำกับ item ใดๆ ใน FactFieldListbox ของฟอร์ม 2 หรือไม่ ถ้าไม่ซ้ำก็ทำการ Add item ของ TableListbox, FieldListbox ของฟอร์ม 3 ที่ถูกเลือกลงใน TableListbox, FactFieldListbox ของฟอร์ม 2 ตามลำดับ
  - ปิดฟอร์ม 3
- กรณีที่เป็นการเลือกตาราง Aggregation แบบรายเดือน
  - ทำการตรวจสอบว่าใน TableListbox มีข้อมูลอยู่ และมีการเลือกที่ item ใดๆ หรือไม่
  - นำชื่อ item ของ TableListbox ที่ถูกเลือกไปเก็บลงในคอมโพเนนต์ EditMonth ของฟอร์ม 2
  - ปิดฟอร์ม 3
- กรณีที่เป็นการเลือกตาราง Aggregation แบบรายปี
  - ทำการตรวจสอบว่าใน TableListbox มีข้อมูลอยู่ และมีการเลือกที่ item ใดๆ หรือไม่
  - นำชื่อ item ของ TableListbox ที่ถูกเลือกไปเก็บลงในคอมโพเนนต์ EditYear ของฟอร์ม 2
  - ปิดฟอร์ม 3

5. procedure TForm3.BtCancelClick(Sender: TObject);
   - ปิดฟอร์ม 3

6. procedure TForm3.ListTable;
   - ทำการดึงรายชื่อตารางทั้งหมดจากฐานข้อมูลที่ระบุในตัวแปร ModelDatabaseName มาแสดงไว้ในคอมโพเนนต์ TableListbox

7. procedure TForm3.EnableComponent;
   - กำหนดการ Enable ของคอมโพเนนต์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Unti4

Type ต่างๆ

1. Joinnodeptr = ^Joinnode;
2. Joinnode = record — เป็นข้อมูลชนิดเรคคอร์ดเก็บชื่อฟิลด์ที่ใช้ในการ join กัน ระหว่างตารางสองตารางในโมเดล
   - son :string;
   - father :string;
   - field :string;
   - next :Joinnodeptr;

   end;
3. TSwitchMode = (Thai,English);

ตัวแปรต่างๆ

1. DatabaseName :string; — ใช้เก็บชื่อฐานข้อมูลของโมเดลที่เรียกขึ้นมาวิเคราะห์
2. HeadTableJoinField :Joinnodeptr — เป็นพอยเตอร์ที่ชี้ไปยังเรคคอร์ดแรกของ link list ที่เก็บข้อมูลการ join กันของตารางทั้งหมดในโมเดล
3. TailTableJoinField :Joinnodeptr; — เป็นพอยเตอร์ที่ชี้ไปยังเรคคอร์ดสุดท้ายของ link list ที่เก็บข้อมูลการ join กันของตารางทั้งหมดในโมเดล
4. ShowSQL :boolean; — ใช้ในการตรวจสอบสถานะของการแสดงคำสั่ง SQL
5. SwitchMode :TSwitchMode; — ใช้ในการตรวจสอบว่าการแสดงผลอยู่ในโหมดภาษาอะไร

Procedure และ Function ต่างๆ

1. procedure TForm4.CreateModel1Click(Sender: TObject);
   - เมื่อมีการเลือก Create Model ที่เมนูหลักก็ให้สร้างฟอร์ม 2 ขึ้นมาใหม่
   - แสดงฟอร์ม 2

2. procedure TForm4.Exit1Click(Sender: TObject);
   - เมื่อมีการเลือก Exit ที่เมนูหลักก็ให้ปิดฟอร์ม 4 เป็นการจบการทำงาน

3. procedure TForm4.LoadModel1Click(Sender: TObject);
   - เรียกโพรซีเดอร์ ClearComponent เพื่อกำหนดค่าเริ่มต้นให้กับคอมโพเนนต์ต่างๆ
   - ให้ผู้ใช้เลือกไฟล์
   - นำชื่อ ไฟล์.tam มาเก็บไว้ในตัวแปร ModelFileName

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- นำชื่อ ไฟล์.tam~ เก็บไว้ในตัวแปร ModelTVFilename
- ทำการโหลดโมเดลโดยเรียกโพรซีเดอร์ LoadModel
- ตรวจสอบสถานะของตัวแปร LoadReady ว่าโหลดโมเดลสำเร็จหรือไม่ ถ้าสำเร็จ ก็จะทำการกำหนดค่าเริ่มต้นต่างๆ
- ถ้าการโหลดโมเดลไม่สำเร็จก็ให้แสดงข้อความเตือนว่ารูปแบบของไฟล์นั้นไม่ถูกต้อง

4. procedure TForm4.LoadModel;

- ตรวจสอบว่ารูปแบบของไฟล์ถูกต้องหรือไม่ ถ้าถูกต้องก็ทำการโหลดโมเดล แต่ถ้าไม่ถูกต้องก็ไม่ต้องทำการโหลดโมเดล และกำหนดสถานะของการโหลดโมเดลไว้ในตัวแปร LoadReady ด้วย
- ทำการเปิดไฟล์ตามชื่อไฟล์ที่ระบุอยู่ในตัวแปร ModelFileName
  - อ่านชื่อฐานข้อมูลของโมเดลมาเก็บไว้ในตัวแปร DatabaseName
  - อ่านชื่อตาราง Aggregation แบบรายเดือน มาเก็บไว้ใน EditMonth.Text
  - อ่านชื่อตาราง Aggregation แบบรายปี มาเก็บไว้ใน EditYear.Text
  - อ่านรายการตัววัดต่างๆ ซึ่งเป็นฟิลด์ของ Fact Table มาเก็บไว้ในคอมโพเนนต์ FactFieldCombobox และ FactFieldListbox
  - อ่านรายการ Fact Table ต่างๆ ที่สัมพันธ์กับตัววัดแต่ละตัว มาเก็บไว้ในคอมโพเนนต์ FactTableListbox
- ทำการเปิดไฟล์ตามชื่อไฟล์ที่ระบุอยู่ในตัวแปร ModelTVFileName อ่านรายการของตารางไดเมนชั่นต่างๆ มาเก็บไว้ในคอมโพเนนต์ DimTV ของฟอร์ม 5 และคอมโพเนนต์ DimTV ของฟอร์ม 6 ตามลำดับ
- สร้าง link list สำหรับเก็บข้อมูลการ join กันของตารางทั้งหมดในโมเดล โดยการเรียกโพรซีเดอร์ CollectTableJoinField

5. procedure TForm4.CollectTableJoinField;

- หาว่าตารางที่มีทั้งหมดในโมเดลต้อง join กับตารางอื่นที่มีชื่อว่าอะไรด้วยฟิลด์ชื่ออะไร ซึ่งสามารถทำได้โดยการหาว่าโหนดทุกๆ โหนดใน DimTV ของฟอร์ม 5 ต้อง join กับโหนดที่มีความสัมพันธ์กับตัวมันเอง (หมายความถึงโหนดที่มีความสัมพันธ์แบบพ่อลูกกัน) ด้วยฟิลด์ชื่ออะไร ซึ่งฟิลด์ที่ใช้ในการ join กันจะต้องมีชื่อที่เหมือนกัน
- จะทำการเก็บความสัมพันธ์แบบไม่ให้มีความซ้ำซ้อน โดยจะคิดเฉพาะโหนดลูกว่าต้อง join กับโหนดพ่อของมันด้วยฟิลด์ชื่ออะไร แต่จะไม่มีการเก็บว่าโหนดพ่อจะต้อง join กับโหนดลูกของมันด้วยฟิลด์ชื่ออะไรอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สำหรับโหนดที่อยู่ใน Level 0 จะไม่มีโหนดพ่อ ซึ่งหมายถึงตารางไดเมนชั่นนั้นจะมีการ join กันกับตาราง Fact โดยตรง เราจึงเก็บโหนดพ่อของโหนดที่อยู่ใน Level 0 เป็นสตริงว่าง เพื่อเอาไว้ตรวจสอบว่าถ้าโหนดพ่อของตารางใดเป็นสตริงว่าง ก็ให้ join ตารางนั้นกับตาราง fact
- สำหรับการค้นหาชื่อฟิลด์ที่เหมือนกันของตารางสองตารางใดๆ สามารถทำได้โดยการเรียกโพรซีเดอร์ FindJoinField โดยการส่งพารามิเตอร์เป็นชื่อตารางสองตารางใดๆ
- ข้อมูลการ join กันของตารางทั้งหมดในโมเดลจะเก็บในลักษณะของ Link list โดยมีเรคคอร์ดแรกชี้โดยตัวแปรพอยเตอร์ HeadTableJoinField และเรคคอร์ดสุดท้ายชี้โดยตัวแปรพอยเตอร์ TailTableJoinField
- นำข้อมูลทั้งหมดใน link list มาแสดงไว้ในคอมโพเนนต์ TableJoinFieldListBox เพื่อใช้สำหรับการดีบักโปรแกรม แต่ไม่ต้องแสดงให้ผู้ใช้เห็น

6. function TForm4.FindJoinField(son, father: string): string;
   - จะรับชื่อตารางสองตารางใดๆ ผ่านพารามิเตอร์ชื่อ son และ father และส่งข้อมูลกลับเป็นชื่อฟิลด์ที่เหมือนกันของตารางสองตารางนั้น โดยมีชนิดของข้อมูลเป็นสตริง
   - การค้นหาสามารถทำโดยการหาว่าฟิลด์ทุกฟิลด์ของตารางที่เป็นโหนดลูกมีฟิลด์ใดที่มีชื่อซ้ำกับฟิลด์ของตารางที่เป็นโหนดพ่อบ้าง ก็ให้ส่งข้อมูลกลับเป็นชื่อฟิลด์นั้น แต่ถ้าการค้นหาไม่เจอฟิลด์ที่มีชื่อเหมือนกันเลย ก็ให้ส่งข้อมูลกลับเป็นสตริงว่าง

7. procedure TForm4.BtQueryClick(Sender: TObject);
   - เมื่อมีการคลิกปุ่ม Query ก็จะทำการสร้างคำสั่ง SQL ขึ้นมา ซึ่งจะเก็บไว้ในตัวแปร RunScript จากนั้นก็จะให้คอมโพเนนต์ Query1 ทำการดึงข้อมูลจากฐานข้อมูลที่ระบุในตัวแปร Database ด้วยคำสั่ง SQL ที่ระบุในตัวแปร RunScript
   - ในการสร้างคำสั่ง SQL จะมีการเรียกใช้ฟังก์ชั่นต่างๆ เช่น
     - ListField จัดหารายการฟิลด์ที่ผู้ใช้เลือกขึ้นมาวิเคราะห์
     - Timekey จัดหาฟิลด์ของตาราง fact ที่มีชนิดข้อมูลเป็นเวลา
     - ListTable จัดหารายการตารางที่ต้องนำมาใช้งาน
     - ListJoin จัดหา clause ที่เป็นเงื่อนไขการ join ของตารางต่างๆ
   - เมื่อ Query1 ดึงข้อมูลขึ้นมาแล้วก็จะป้อนข้อมูลให้กับคอมโพเนนต์ DecisionCube1 ของฟอร์ม 4 และคอมโพเนนต์ DecisionGraph1 ของฟอร์ม 8 จากนั้นจะแสดงข้อมูลออกทางคอมโพเนนต์ DecisionGrid1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. function TForm4.CurrentFactTable: string;

- ทำการค้นหาว่าตาราง Fact ที่จะถูกใช้งานในขณะนั้นคือตารางใด ซึ่งอาจเป็นตารางที่ไม่ได้มีการทำ Aggregation เอาไว้ (มีโอกาสเป็นตารางใดก็ได้ขึ้นอยู่กับว่าผู้ใช้เลือกฟิลด์ชื่ออะไร เพราะตัววัดอาจมาจากตาราง fact ได้มากกว่า 1 ตาราง) หรืออาจเป็นตารางที่มีการทำ Aggregation แบบรายเดือนโดยการตรวจสอบจากคอมโพเนนต์ RdoMonth หรืออาจเป็นตารางที่มีการทำ Aggregation แบบรายปีโดยการตรวจสอบจากคอมโพเนนต์ RdoYear
- ฟังก์ชั่นนี้จะส่งค่ากลับเป็นชื่อของตาราง Fact ที่ถูกใช้งานในขณะนั้น

9. function TForm4.ListField: string;

- จัดหารายการฟิลด์ที่ผู้ใช้เลือกขึ้นมาวิเคราะห์ โดยนำมาจาก Link list ที่เก็บข้อมูลฟิลด์และตารางต่างๆที่ผู้ใช้เลือก ซึ่งมีตัวแปรพอยเตอร์ HeadSelectedField ชี้ที่เรคคอร์ดแรก
- ชื่อฟิลด์แต่ละฟิลด์จะถูกคั่นด้วย ' , '
- ฟังก์ชั่นนี้จะส่งค่ากลับเป็นรายการของฟิลด์ต่างๆ ซึ่งมีชนิดของข้อมูลเป็นสตริง

10. function TForm4.ListTable: string;

- จัดหารายการตารางที่ต้องนำมาใช้งาน โดยนำมาจาก item ต่างๆ ในคอมโพเนนต์ TableUsedListbox
- ตรวจสอบว่าถ้าผู้ใช้มีการกำหนด Condition หรือไม่โดยตรวจสอบจากตัวแปร SetCondition ถ้าผู้ใช้มีการกำหนด Condition ก็ให้เพิ่มตารางซึ่งนำมาจาก item ต่างๆในคอมโพเนนต์ ConditionTableListbox เพิ่มเข้าไปด้วย
- ชื่อตารางแต่ละตารางจะถูกคั่นด้วย ' , '
- ฟังก์ชั่นนี้จะส่งค่ากลับเป็นรายการของตารางต่างๆ ซึ่งมีชนิดของข้อมูลเป็นสตริง

11. function TForm4.ListJoin: string;

- จะจัดหา clause ที่เป็นเงื่อนไขการ join ของตารางต่างๆ ทำได้โดยพิจารณาจากตารางทั้งหมดที่ต้องนำมาใช้ ซึ่งจะนำมาจาก item ต่างๆในคอมโพเนนต์ TableUsedListBox จากนั้นก็ทำการค้นหาตารางที่ต้องนำมา join ด้วยและฟิลด์ที่ต้องใช้ในการ join ระหว่างสองตารางนั้นได้จาก Link list ที่เก็บข้อมูลการ join กันของตารางทั้งหมดในโมเดลซึ่งมีตัวแปรพอยเตอร์ HeadTableJoinField ชี้อยู่ที่เรคคอร์ดแรก
- ทำการตรวจสอบว่าผู้ใช้มีการกำหนด Condition หรือไม่โดยตรวจสอบจากตัวแปร SetCondition ถ้าผู้ใช้มีการกำหนด Condition ก็ให้เพิ่มเงื่อนไขการ join ของตารางซึ่งนำมาจาก item ต่างๆในคอมโพเนนต์ ConditionTableListbox เพิ่มเข้าไปด้วย
- เงื่อนไขการ join ของตารางแต่ละคู่จะถูกคั่นด้วย ' and '

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฟังก์ชั่นนี้จะส่งค่ากลับเป็น clause ที่เป็นเงื่อนไขการ join ของตารางต่างๆ ซึ่งมีชนิดของข้อมูลเป็นสตริง

12. function TForm4.TimeKey: string;

- จะทำการตรวจสอบทุกฟิลด์ของตาราง fact ว่าฟิลด์ใดมีชนิดของข้อมูลเป็นเวลา (ftDate หรือ ftDateTime) ก็จะทำการส่งค่ากลับเป็นชื่อฟิลด์ของตาราง fact ซึ่งมาชนิดของข้อมูลเป็นสตริง

13. procedure TForm4.BtSetTimeClick(Sender: TObject);

- ตรวจสอบมีการดึงข้อมูลขึ้นมาจากฐานข้อมูลและนำมาใส่ให้กับคอมโพเนนต์ Decisioncube1 แล้วหรือยัง โดยดูจากคำสั่ง SQL ในคอมโพเนนต์ Listbox1 ว่ามีหรือไม่ ถ้ามีก็ให้เรียกเมดถอด ShowCubeDialog ของคอมโพเนนต์ Decisioncube1 เพื่อให้ผู้ใช้สามารถกำหนด Time Frame ของข้อมูลที่ดึงขึ้นมาวิเคราะห์ได้

14. procedure TForm4.BtSQLClick(Sender: TObject);

- ตรวจสอบว่าขณะนั้นมีการแสดงคำสั่ง SQL ในคอมโพเนนต์ Listbox1 อยู่หรือไม่โดยดูได้จากตัวแปร ShowSQL ถ้าไม่มีการแสดงก็ให้แสดงขึ้นมา แต่ถ้ากำลังแสดงคำสั่ง SQL อยู่ก็ให้ปิดการแสดงนั้น

15. procedure TForm4.BtConditionClick(Sender: TObject);

- กำหนดสถานะของตัวแปร EditCondition
- แสดงฟอร์ม 6 ขึ้นมา

16. procedure TForm4.BTClearConClick(Sender: TObject);

- เรียกเมดถอด BtClearClick ของฟอร์ม 6 เพื่อทำการลบคอนดิชั่นที่ได้เคยกำหนดไว้ทั้งหมด

17. procedure TForm4.clearComponent;

- สร้างฟอร์ม 5 และฟอร์ม 6 ขึ้นมาใหม่
- กำหนดค่าเริ่มต้นให้กับคอมโพเนนต์ต่างๆ

18. procedure TForm4.FactFieldComboBoxChange(Sender: TObject);

- ตรวจสอบว่าถ้าผู้ใช้มีการเลือกฟิลด์จากไดเมนชั่น (ดูจากคอมโพเนนต์ DimLabel) และเลือกตัววัดที่จะนำมาวิเคราะห์เรียบร้อยแล้ว (ดูจากคอมโพเนนต์ FactFieldComboBox) ก็ให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Enable ปุ่ม Query ได้ แต่ถ้ายังไม่มีการเลือกอันใดอันหนึ่งก็จะไม่อนุญาติให้ผู้ใช้กดปุ่ม Query ได้

19. procedure TForm4.BitBtViewClick(Sender: TObject);

- เมื่อผู้ใช้มีการคลิกที่ปุ่ม View ก็จะแสดงฟอร์ม 5 ขึ้นมา

20. procedure TForm4.Print1Click(Sender: TObject);

- เมื่อผู้ใช้มีการคลิกเลือก print ที่เมนูหลัก ก็จะทำการพิมพ์ข้อมูลในคอมโพเนนต์ DecisionGrid1 ออกมา

21. procedure TForm4.Graph1Click(Sender: TObject);

- เมื่อผู้ใช้มีการคลิกเลือก Graph ที่เมนูหลัก ก็จะแสดงฟอร์ม 8 ขึ้นมา

22. procedure TForm4.English1Click(Sender: TObject);

- ถ้ามีการเลือกภาษาเป็นภาษาอังกฤษที่เมนู ก็ให้กำหนดค่าตัวแปร SwichMode ให้เป็น English จากนั้นก็เรียกโพรซีเดอร์ SwitchToEnglish ของทั้งฟอร์ม 4 และ ฟอร์ม 2

23. procedure TForm4.Thai1Click(Sender: TObject);

- ถ้ามีการเลือกภาษาเป็นภาษาไทยที่เมนู ก็ให้กำหนดค่าตัวแปร SwichMode ให้เป็น Thai จากนั้นก็เรียกโพรซีเดอร์ SwitchToThai ของทั้งฟอร์ม 4 และ ฟอร์ม 2

24. procedure TForm4.SwitchToEnglish;

- ตั้งค่า Caption ของคอมโพเนนต์ต่างๆ ของฟอร์ม 4 ฟอร์ม 5 ฟอร์ม 6 และฟอร์ม 8 ให้เป็นคำอธิบายภาษาอังกฤษ

25. procedure TForm4.SwitchToThai;

- ตั้งค่า Caption ของคอมโพเนนต์ต่างๆ ของฟอร์ม 4 ฟอร์ม 5 ฟอร์ม 6 และฟอร์ม 8 ให้เป็นคำอธิบายภาษาไทย

26. procedure TForm4.FormCreate(Sender: TObject);

- กำหนดการแสดงผลเริ่มต้นให้อยู่ในโหมดภาษาอังกฤษ โดยการกำหนดค่าให้กับตัวแปร Switchmode เป็น English

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Unti5

Type ต่างๆ

1. nodeptr = ^node;
2. node = record — เป็นข้อมูลชนิดเรคคอร์ดเก็บฟิลด์ที่ผู้ใช้เลือกและเก็บตารางต่างๆที่ต้องนำมาใช้ในการเข้าถึงฟิลด์นั้น

```
    field :string;
    Table :array[0..9] of string; {allow 10 level}
    level  :integer;
    next :nodeptr;
end;
```

ตัวแปรต่างๆ

1. HeadSelectedField :nodeptr; — เป็นพอยเตอร์ที่ชี้ไปยังเรคคอร์ดแรกของ link list ที่เก็บข้อมูลรายการฟิลด์ที่ผู้ใช้เลือกและตารางต่างๆที่ต้องนำมาใช้ในการเข้าถึงฟิลด์ทั้งหมดที่เลือก
2. TailSelectedField :nodeptr; — เป็นพอยเตอร์ที่ชี้ไปยังเรคคอร์ดสุดท้ายของ link list ที่เก็บข้อมูลรายการฟิลด์ที่ผู้ใช้เลือกและตารางต่างๆที่ต้องนำมาใช้ในการเข้าถึงฟิลด์ทั้งหมดที่เลือก

Procedure และ Function ต่างๆ

1. procedure TForm5.DimTVClick(Sender: TObject);
   - เมื่อมีการคลิกที่โหนดใดๆ ใน DimTV ก็จะทำการดึงรายชื่อฟิลด์ของชื่อตารางที่ถูกเลือกใน DimTV จากฐานข้อมูลที่ระบุในตัวแปร DatabaseName แล้วนำมาแสดงไว้ในคอมโพเนนต์ FieldDimListbox

2. procedure TForm5.DimTVKeyPress(Sender: TObject; var Key: Char); {dup}
   - เมื่อมีการเลือกที่โหนดใดๆ ใน DimTV และทำการกด Enter บนคีย์บอร์ด ก็จะทำการดึงรายชื่อฟิลด์ของชื่อตารางที่ถูกเลือกใน DimTV จากฐานข้อมูลที่ระบุในตัวแปร DatabaseName แล้วนำมาแสดงไว้ในคอมโพเนนต์ FieldDimListbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. procedure TForm5.BtAddClick(Sender: TObject);

- จะทำการตรวจสอบว่าถ้ามีการเลือกชื่อฟิลด์ซึ่งเป็น item ในคอมโพเนนต์ FieldDimListbox และชื่อฟิลด์นั้นจะต้องไม่ซ้ำกับที่เลือกไปแล้วนั่นก็คือจะไม่ซ้ำกับ item ที่มีอยู่ในคอมโพเนนต์ FieldSelectedlistbox ก็จะไม่มีการ add เพิ่มเข้าไป
- ในการ add ฟิลด์ที่เลือกแต่ละครั้งเราจะมีการเก็บข้อมูลของฟิลด์ทั้งหมดที่เลือกรวมทั้งตารางต่างๆ ที่ต้องใช้ในการเข้าถึงฟิลด์เหล่านั้น ควบคู่กันไปด้วย โดยจะทำการเก็บข้อมูลเป็น Link list ของเรคคอร์ดที่แสดงชื่อฟิลด์และตารางต่างๆที่ต้องนำมาใช้ในการเข้าถึงฟิลด์ที่ถูกเลือก โดยจะมีตัวแปรพอยเตอร์ HeadSelectedField ชี้ที่เรคคอร์ดแรกของ Link list และตัวแปรพอยเตอร์ TailSelectedField จะชี้อยู่ที่เรคคอร์ดสุดท้ายของ Link list
- ทุกครั้งที่ผู้ใช้มีการเลือกฟิลด์และคลิกปุ่ม add แต่ละเรคคอร์ดของชื่อฟิลด์และรายการตารางจะถูกสร้างขึ้นโดยการเรียกโพซีเดอร์ getnode จากนั้นเรคคอร์ดนั้นก็จะถูกนำไปต่อท้ายใน Link list

4. procedure TForm5.BtRemClick(Sender: TObject);

- ทำการตรวจสอบว่าผู้ใช้มีการเลือกชื่อฟิลด์ซึ่งเป็น item ในคอมโพเนนต์ FieldDimListbox หรือไม่
- ทำการตรวจสอบ index ของ item ของ FieldDimListbox ที่ถูกเลือก จากนั้นก็ทำการลบเรคคอร์ดใน Link list ที่เก็บข้อมูลของฟิลด์ทั้งหมดที่เลือกรวมทั้งตารางต่างๆ ที่ต้องใช้ในการเข้าถึงฟิลด์ โดยจะลบเรคคอร์ดลำดับที่ตรงกับ index ของ item ของ FieldDimListbox ที่ถูกเลือก
- ลบ item ของคอมโพเนนต์ FieldSelectedlistbox ที่ถูกเลือกนั้นทิ้งไป
- กำหนดค่าให้กับตัวแปร SelectDimChange เพื่อเอาไว้แสดงว่ามีการเปลี่ยนแปลงการเลือกตารางไดเมนชั่น

5. procedure TForm5.BtOkClick(Sender: TObject);

- นำรายการชื่อตารางทั้งหมดที่ต้องเรียกใช้งานจาก Link list ที่เก็บข้อมูลของฟิลด์ทั้งหมดที่เลือกรวมทั้งตารางต่างๆ ที่ต้องใช้ในการเข้าถึงฟิลด์ ไปเก็บในคอมโพเนนต์ TableUsedlistbox โดยเริ่มต้นที่เรคคอร์ดแรกซึ่งถูกชี้โดยตัวแปรพอยเตอร์ HeadSelectedField
- การนำชื่อตารางไปเก็บในคอมโพเนนต์ TableUsedlistbox จะเก็บรายชื่อตารางที่ต้องใช้ แต่ถ้าชื่อตารางใดที่นำมาจาก link list มีชื่อซ้ำกัน ก็จะใส่เพียงชื่อเดียว
- ตรวจสอบว่าถ้าผู้ใช้มีการเลือกฟิลด์จากไดเมนชั่น (ดูจากคอมโพเนนต์ DimLabel ของฟอร์ม 4) และเลือกตัววัดที่จะนำมาวิเคราะห์เรียบร้อยแล้ว (ดูจากคอมโพเนนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FactFieldComboBox ของฟอร์ม 4) ก็ให้ Enable ปุ่ม Query ได้ แต่ถ้ายังไม่มีการเลือกอันใดอันหนึ่งก็จะไม่อนุญาติให้ผู้ใช้กดปุ่ม Query ได้

- ตรวจสอบการเปลี่ยนแปลงการเลือกตารางไดเมนชั่น จากตัวแปร SelectDimChange ถ้ามีการเปลี่ยนแปลง ก็จะเรียกโพซีเดอร์ BtClearClick ของฟอร์ม 6 ก็คือถ้าผู้ใช้มีการเลือกไดเมนชั่นเปลี่ยนไปก็จะทำการลบ condition เดิมที่เคยตั้งไว้

6. function TForm5.getnode: nodeptr;

- จะทำการเก็บชื่อฟิลด์และตารางต่างๆที่ต้องนำมาใช้ในการเข้าถึงฟิลด์ที่ถูกเลือกนั้นๆ ในลักษณะที่เป็นเรคคอร์ด
- การสร้างเรคคอร์ดจะมีการเก็บชื่อฟิลด์โดยนำมาจากชื่อ item ของคอมโพเนนต์ FieldDimListbox และเก็บอาเรย์ของชื่อตาราง โดยอาเรย์ของชื่อตารางจะเริ่มเก็บจากชื่อโหนดที่ถูกเลือกรวมทั้งโหนดพ่อทั้งหมดของโหนดที่ถูกเลือกของ DimTV นั้นไปจนถึงโหนดที่เป็น root เนื่องจากการที่จะเลือกฟิลด์นั้นๆ ได้จำเป็นต้อง join ตารางของฟิลด์นั้นในเรื่อยๆ จนถึงตารางที่อยู่ใน level 0
- ฟังก์ชั่นนี้จะมีการส่งค่ากลับเป็นข้อมูลชนิดพอยเตอร์ซึ่งชี้ไปยังเรคคอร์ดที่สร้างได้นั้น

7. sprocedure TForm5.FormCreate(Sender: TObject);

- กำหนดค่าให้กับตัวแปร SelectDimChange เพื่อเอาไว้ตรวจสอบการเปลี่ยนแปลงการเลือกตารางไดเมนชั่น

Unti6

ตัวแปรต่างๆ

1. StrCondition :string; เก็บสตริงของเงื่อนไขซึ่งผู้ใช้กำหนด
2. setcondition :boolean; ใช้ตรวจสอบว่ามีการกำหนด Condition เรียบร้อยหรือยัง
3. EditCondition :boolean; ใช้ตรวจสอบว่ามีการแก้ไข Condition แล้วหรือไม่
4. SelectDimChange :boolean; ใช้ตรวจสอบการเปลี่ยนแปลงการเลือกตารางไดเมนชั่น

Procedure และ Function ต่างๆ

1. procedure TForm6.DimTVClick(Sender: TObject):

- เมื่อมีการคลิกที่โหนดใดๆ ใน DimTV ก็จะทำการดึงรายชื่อฟิลด์ของชื่อตารางที่ถูกเลือกใน DimTV จากฐานข้อมูลที่ระบุในตัวแปร DatabaseName แล้วนำมาแสดงไว้ในคอมโพเนนต์ FieldDimListbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. procedure TForm6.BtAndClick(Sender: TObject);
   - ตรวจสอบค่าตัวแปร EditCondition เพื่อดูว่ามีกำแก้ไข Condition แล้วหรือยัง ถ้ายังแสดงว่าเป็นการกำหนด คอนดิชั่น นี้เป็นเงื่อนไขแรก ก็ให้ทำการลบข้อมูล Condition เดิมในตัวแปร StrCondition เสียก่อน
   - นำชื่อโหนดใน DimTV ชื่อ item ใน FieldDimListbox และค่าในคอมโพเนนต์ Edit1 มาสร้างเป็นเงื่อนไขภาษา SQL ลงท้ายด้วย ' and ' เก็บไว้ในตัวแปร StrCondition
   - เรียกโพซีเดอร์ CollectConditionTable เพื่อเก็บรายชื่อตารางที่ต้องเรียกใช้งานในการเข้าถึงฟิลด์ที่ผู้ใช้กำหนด condition ลงในคอมโพเนนต์ ConditionTablelistbox
   - ลบตัวอักษรในคอมโพเนนต์ Edit1
   - กำหนดค่าให้กับตัวแปร EditCondition เพื่อแสดงว่ามีกำแก้ไข Condition แล้ว
3. procedure TForm6.BtOrClick(Sender: TObject);
   - ตรวจสอบค่าตัวแปร EditCondition เพื่อดูว่ามีกำแก้ไข Condition แล้วหรือยัง ถ้ายังแสดงว่าเป็นการกำหนด คอนดิชั่น นี้เป็นเงื่อนไขแรก ก็ให้ทำการลบข้อมูล Condition เดิมในตัวแปร StrCondition เสียก่อน
   - นำชื่อโหนดใน DimTV ชื่อ item ใน FieldDimListbox และค่าในคอมโพเนนต์ Edit1 มาสร้างเป็นเงื่อนไขภาษา SQL ลงท้ายด้วย ' or ' เก็บไว้ในตัวแปร StrCondition
   - เรียกโพซีเดอร์ CollectConditionTable เพื่อเก็บรายชื่อตารางที่ต้องเรียกใช้งานในการเข้าถึงฟิลด์ที่ผู้ใช้กำหนด condition ลงในคอมโพเนนต์ ConditionTablelistbox
   - ลบตัวอักษรในคอมโพเนนต์ Edit1
   - กำหนดค่าให้กับตัวแปร EditCondition เพื่อแสดงว่ามีการแก้ไข Condition แล้ว

4. procedure TForm6.BtOkClick(Sender: TObject);
   - ตรวจสอบค่าตัวแปร EditCondition เพื่อดูว่ามีกำแก้ไข Condition แล้วหรือยัง ถ้ายังแสดงว่าเป็นการกำหนด คอนดิชั่น นี้เป็นเงื่อนไขแรก ก็ให้ทำการลบข้อมูล Condition เดิมในตัวแปร StrCondition เสียก่อน
   - นำชื่อโหนดใน DimTV ชื่อ item ใน FieldDimListbox และค่าในคอมโพเนนต์ Edit1 มาสร้างเป็นเงื่อนไขภาษา SQL เก็บไว้ในตัวแปร StrCondition
   - เรียกโพซีเดอร์ CollectConditionTable เพื่อเก็บรายชื่อตารางที่ต้องเรียกใช้งานในการเข้าถึงฟิลด์ที่ผู้ใช้กำหนด condition ลงในคอมโพเนนต์ ConditionTablelistbox
   - ลบตัวอักษรในคอมโพเนนต์ Edit1
   - กำหนดค่าให้กับตัวแปร SetCondition เพื่อแสดงว่ามีการกำหนด Condition แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. procedure TForm6.BtClearClick(Sender: TObject);
   - ลบข้อมูลเกี่ยวกับการกำหนด Condition ในคอมโพเนนต์ต่างๆ
   - กำหนดค่าเริ่มต้นให้กับตัวแปรต่างๆ

6. procedure TForm6.BtCancelClick(Sender: TObject);
   - ปิดฟอร์ม 6

7. procedure TForm6.CollectConditionTable;
   - เมื่อผู้ใช้เลือกฟิลด์และกำหนด Condition.ให้กับฟิลด์นั้นๆ เราจะทำการเก็บรายชื่อตารางที่ต้องเรียกใช้งานในการเข้าถึงฟิลด์ที่ผู้ใช้กำหนด condition ลงในคอมโพเนนต์ ConditionTablelistbox
   - การเก็บรายชื่อตารางต่างๆ สามารถทำได้โดยเก็บจากชื่อโหนดที่ถูกเลือกรวมทั้งโหนดพ่อทั้งหมดของโหนดที่ถูกเลือกของ DimTV นั้นไปจนถึงโหนดที่เป็น root เนื่องจากการที่จะเลือกฟิลด์นั้นๆ ได้จำเป็นต้อง join ตารางของฟิลด์นั้นในเรื่อยๆ จนถึงตารางที่อยู่ใน level 0
   - การนำชื่อตารางไปเก็บในคอมโพเนนต์ ConditionTablelistbox จะเก็บรายชื่อตารางที่ต้องใช้ แต่จะมีการตรวจสอบด้วยว่าชื่อตารางที่ใส่ลงไปจะต้องไม่ซ้ำกัน และต้องไม่ซ้ำกับ item ในคอมโพเนนต์ ConditionTablelistbox ด้วย

Unit7

Procedure และ Function ต่างๆ

1. procedure TForm7.BtCancelClick(Sender: TObject);
   - ปิดฟอร์ม 7

2. procedure TForm7.BtSelectClick(Sender: TObject);
   - ทำการตรวจสอบว่าใน TableListbox มีข้อมูลอยู่ และมีการเลือกที่ item ใดๆ หรือไม่
   - ทำการ Add item ของ TableListbox ที่ถูกเลือก ให้เป็น node ลูกของ node ที่ถูกเลือกของ DimTV ในฟอร์ม 2
   - ปิดฟอร์ม 7

3. procedure TForm7.ListTable;
   - นำชื่อของฐานข้อมูลที่เก็บในตัวแปร ModelDatabaseName ในเก็บไว้ในตัวแปร Database เพื่อส่งไปใช้งานในฟอร์ม 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตรวจสอบดูว่าตารางในฟอร์ม 2 ที่ต้องการจะเพิ่มตารางที่เป็น Snowflake มีความสัมพันธ์กับตารางใดในฐานข้อมูลบ้าง ซึ่งสามารถทำได้โดยเปรียบเทียบว่า node ที่ถูกเลือกของ DimTV ในฟอร์ม 2 กับทุกๆ item ของ TableListbox ในฟอร์ม 3 คู่ใดที่มีชื่อฟิลด์ที่เหมือนกันบ้าง
- สำหรับการตรวจสอบชื่อฟิลด์ที่ตรงกันของตารางสองตารางใดสามารถทำได้โดยการเรียกฟังก์ชั่น FindJoinField ของฟอร์ม 4 โดยการส่งชื่อตารางสองตารางใดๆเข้าไปเป็นพารามิเตอร์

4. procedure TForm7.TableListBoxClick(Sender: TObject);
   - เมื่อมีการคลิกที่ item ใดๆ ใน TableListbox ก็จะทำการดึงรายชื่อฟิลด์ของชื่อตารางที่ถูกเลือกใน TableListbox จากฐานข้อมูลที่ระบุในตัวแปร ModelDatabaseName แล้วนำมาแสดงไว้ในคอมโพเนนต์ FieldListbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] C.J.Date(1986) : "*An Introduction of Database System*", 6th edition, Addison Wesley,Massachusetts, p.2-74, 1986

[2] James R. Groff and Paul N. Weinberg : "*Lan Times Guide to SQL*", Osborne McGraw – Hill. California. 1994

[3] Raph Kimbol (1996) : "*The Data Warehouse Toolkit : Practical Techniques For Building Dimensional Data Warehouse*", John Wiley & Sons,Inc., New York. 1996

[4] *IT.Soft.* ปีที่ 6, ฉบับที่ 64 (กรกฎาคม 2540):120-138. กองบรรณาธิการ "Data Warehouse เปิดผนึกคลังข้อมูลอัจฉริยะ."

[5] กนก กุสุมาลย์นุกูล และไกรวุฒิ มั่นเสถียรสิน (1999) : "*คู่มือการเขียนโปรแกรม Delphi4*" . บริษัท ซัคเซส มีเดีย จำกัด 1999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้