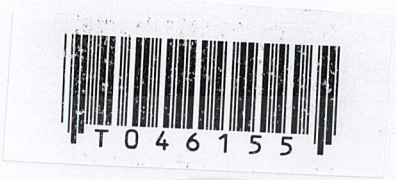
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การศึกษาและพัฒนาระบบดาต้าแวร์เฮาส์

DATA WAREHOUSE



โดย

นายกสิภพ ประศาสน์ศรีสุภาพ

นายเกียรติชุมพล สุทธิศิริกุล

อาจารย์ที่ปรึกษา

ดร. วรวัฒน์ ลิ้มโภคา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เลขหมู่.....
เลขทะเบียน.....46155
วัน, เดือน, ปี.....20 มี.ค. 2546

.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2544

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การศึกษาและพัฒนาระบบดาต้าแวร์เฮาส์

DATA WAREHOUSE

ผู้จัดทำ

1. นายกสิภพ ประศาสน์ศรีสุภาพ รหัสประจำตัว 41014018
2. นายเกียรติชุมพล สุทธิศิริกุล รหัสประจำตัว 41014047



อาจารย์ที่ปรึกษา

(ดร. วรวัฒน์ ลิ้มโภคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การศึกษาและพัฒนาระบบดาต้าแวร์เฮาส์

นายกสิภพ ประศาสน์ศรีสุภาพ 41014018
นายเกียรติชุมพล สุทธิศิริกุล 41014047
ดร. วรวัฒน์ ลิ้มโภคา อาจารย์ที่ปรึกษา
ปีการศึกษา 2544

บทคัดย่อ

โครงงานนี้เป็นการศึกษาระบบฐานข้อมูลดาต้าแวร์เฮาส์ (Data Warehouse) ซึ่งเป็นระบบฐานข้อมูลอีกรูปแบบหนึ่งที่แตกต่างไปจากฐานข้อมูลทั่วไป (Relational Database) ระบบฐานข้อมูล ดาต้าแวร์เฮาส์มีการจัดเก็บข้อมูลในลักษณะมัลติไดเมนชันแนล (Multidimensional) โครงงานนี้ใช้ Microsoft SQL Server 2000 Analysis Service มาทดลองสร้างระบบดาต้าแวร์เฮาส์ โดยได้ศึกษาการจัดเก็บข้อมูลในลักษณะต่างๆกัน คือ MOLAP , ROLAP และ HOLAP และได้ทดสอบเปรียบเทียบประสิทธิภาพการจัดเก็บข้อมูลทั้ง 3 แบบ นอกจากนี้ยังพัฒนาโปรแกรมประยุกต์ไคลเอ็นต์ขึ้นมาติดต่อ Analysis Service ซึ่งพัฒนาขึ้นทั้งที่ทำงานบนแลนและบนเว็บ เพื่อให้ผู้ใช้เข้าถึงฐานข้อมูลดาต้าแวร์เฮาส์ได้สะดวกและยืดหยุ่นมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Warehouse

Kasiphop Prasatsisuparp 41014018
Kiatchumpol Suttisirikul 41014047
Dr. Worrawat Limpoka Advisor

Abstract

This project concerns the study of Data Warehouse. It is another type of database which differs from Relational Database. Data Warehouse is stored in the form of Multidimensional Database. Microsoft Server 2000 is applied to study Data Warehouse by storaging mode of Analysis Service in different forms : MOLAP, ROLAP and HOLAP, then comparing the storage's efficiency of those three forms. Moreover, Client application is developed to connect Analysis Service both on LAN and WEB so that Data Warehouse's user can access more convenient and flexible.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดีด้วยความช่วยเหลือ ร่วมมือและสนับสนุนจากหลายฝ่ายเป็นอย่างดี จึงใคร่ขอขอบพระคุณบุคคลดังต่อไปนี้

1. บิดา มารดาและพี่น้องทุกๆ คน ผู้ส่งเสริมด้านการศึกษา เป็นกำลังใจและคอยให้ความสนับสนุนในทุกๆ ด้านมาโดยตลอด
2. ดร. วรวัฒน์ ลิ้มโภคา อาจารย์ที่ปรึกษาในการทำปริญญานิพนธ์ที่กรุณาให้คำแนะนำและชี้แนะแนวทางในการดำเนินงาน ให้ความเอาใจใส่และช่วยเหลือเสมอมา
3. อาจารย์ทุกๆ ท่านที่ประสิทธิ์ประสาทวิชาความรู้ต่างๆ ตลอดระยะเวลาสี่ปีที่ผ่านมา
4. เพื่อนๆ 4D ทุกคน สำหรับคำแนะนำ คอยให้ความช่วยเหลือทุกครั้งที่เกิดปัญหา

สุดท้ายขอมอบคุณความดีจากปริญญานิพนธ์ฉบับนี้ให้แด่ คุณบิดา มารดาและครูอาจารย์ผู้มีพระคุณทุกท่าน

กสิภพ ประศาสน์ศรีสุภาพ
เกียรติชุมพล สุทธิศิริกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญภาพ	IX
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของการทำโครงงาน	1
1.3 ขอบเขตของโครงงาน	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีเบื้องต้น	4
2.1 นิยามของดาต้าแวร์เฮาส์	4
2.2 ลักษณะที่สำคัญของดาต้าแวร์เฮาส์	4
2.3 คุณสมบัติของดาต้าแวร์เฮาส์	7
2.4 ความแตกต่างของดาต้าแวร์เฮาส์กับระบบฐานข้อมูลประจำวัน	7
2.5 การสร้างข้อมูล (Data Model)	8
2.6 ความแตกต่างของลักษณะของข้อมูลที่ใช้ในงานโอเปอเรชันและดาต้าแวร์เฮาส์	9
2.7 ดาต้าแวร์เฮาส์และ OLAP	10
บทที่ 3 โครงสร้างและการออกแบบของดาต้าแวร์เฮาส์	11
3.1 Dimensional Modeling	11
3.2 การออกแบบข้อมูล (Data Modeling)	14
3.3 โครงสร้างฐานข้อมูล (Database Schema)	16
3.3.1 สกีมาแบบดาว (Star Schema)	16
3.3.2 สกีมาแบบสโนว์เฟลก(Snowflake Schema)	17
3.4 การรวมค่าข้อมูล (Aggregation)	18
3.5 Crosstabulation หรือ Crosstab	19
3.6 ขั้นตอนการออกแบบดาต้าแวร์เฮาส์	20
บทที่ 4 การใช้สร้างระบบฐานข้อมูลดาต้าแวร์เฮาส์	21
4.1 Microsoft Analysis Service	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้าที่

4.2 การสร้างมัลติไดเมนชันแนลโมเดล (Multidimensional Model) บน Analysis Service 22

4.2.1 การติดต่อแหล่งข้อมูล (Data Source) 22

4.2.2 เริ่มสร้างคิวบ์ (Building Cube) 25

4.2.3 การออกแบบการจัดเก็บข้อมูลและการประมวลผลคิวบ์ 29

4.2.4 การดูข้อมูลรายละเอียดของคิวบ์ (Viewing Metadata for Cube) 31

4.2.5 การดูข้อมูลในคิวบ์ (Browse Cube) 32

บทที่ 5 โหมดการเก็บข้อมูล (Storage Mode) 34

5.1 การเก็บข้อมูล 34

5.2 วิธีการเก็บข้อมูลของ Analysis Server 34

5.3 ความเหมาะสมในการใช้งานการเก็บค่าทั้งสามแบบ 35

5.4 การทดสอบการรวมค่าไดเมนชัน 36

5.4.1 การทดสอบการรวมค่าไดเมนชันเดียว 36

5.4.2 การทดสอบการรวมค่าไดเมนชันของ 2 ไดเมนชัน 39

บทที่ 6 Microsoft OLAP Client Architecture 42

6.1 สถาปัตยกรรมไคลเอ็นต์ภายใน Analysis Service 42

6.2 PivotTable Service 42

6.2.1 ลักษณะเด่นของ PivotTable Service 43

6.2.2 คอมโพเนนท์ที่ใช้ใน PivotTable Service 44

6.2.3 การติดตั้งและลงทะเบียนคอมโพเนนท์ 44

6.2.4 ลักษณะการเชื่อมต่อของ PivotTable Service 45

6.2.5 การทำงานของไคลเอ็นต์ใน PivotTable Service 49

บทที่ 7 การออกแบบและพัฒนาโปรแกรมประยุกต์ทำงานบนแลน (LAN) 53

7.1 จุดประสงค์ในการพัฒนาโปรแกรม 53

7.2 ทูลส์ที่นำมาใช้ในการพัฒนาโปรแกรม 53

7.3 การออกแบบโปรแกรมประยุกต์ 53

7.4 คอมโพเนนต์หลักที่ใช้ในการพัฒนาโปรแกรม 54

7.5 การใช้งานโปรแกรมประยุกต์ 55

บทที่ 8 การออกแบบและพัฒนาโปรแกรมประยุกต์ที่ทำงานผ่านเว็บบราวเซอร์ (Web Browser) 59

8.1 จุดประสงค์ในการพัฒนาโปรแกรม 59

8.2 ทูลส์ที่นำมาใช้ในการพัฒนาโปรแกรม 59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้าที่

8.3 การติดตั้งโปรแกรมประยุกต์ที่ทำงานผ่านเว็บ 59
8.4 การใช้งานโปรแกรมประยุกต์ผ่านเว็บ 60
บทที่ 9 การทดสอบประสิทธิภาพการทำงานของโปรแกรมประยุกต์
และการเก็บข้อมูลของ Analysis Service 74
9.1 การทดสอบประสิทธิภาพของโปรแกรมประยุกต์ 74
9.1.1 ทดสอบการเพิ่มจำนวนเรคอร์ดบนตารางแฟ็ค 75
9.1.2 ทดสอบการเพิ่มจำนวนไดเมนชัน 79
9.1.3 ทดสอบการเพิ่มจำนวนการรวมค่า 81
9.2 การทดสอบประสิทธิภาพการเก็บข้อมูลแบบ MOLAP, ROLAP และ HOLAP 84
ภาคผนวก ก Multidimensional Expression (MDX) 94
ภาคผนวก ข คู่มือสำหรับโปรแกรมเมอร์ 107
บรรณานุกรม 115

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้าที่

ตารางที่ 2-1 แสดงความแตกต่างของข้อมูลในโอเปอเรชันกับข้อมูลในดาต้าแวร์เฮาส์ 9
ตารางที่ 3-1 แสดงตัวอย่าง ตารางแฟ็ค (fact table) เป็นตัวอย่างของตารางแฟ็คของ Food Mart 2000 12
ตารางที่ 3-2 แสดงการ Drill-down และ Drill-up 15
ตารางที่ 3-3 แสดงการตัดข้อมูล (Slicing) 15
ตารางที่ 3-4 แสดงการหมุนข้อมูล (Dicing) 16
ตารางที่ 3-5 แสดงการตัดข้อมูลจากข้อมูลในตารางที่ 3-4 16
ตารางที่ 5-1 แสดงการเปรียบเทียบจำนวนสมาชิกที่เกิดขึ้นจริงกับสมาชิกในทางทฤษฎีของไดเมนชันเดียว 39
ตารางที่ 5-2 แสดงข้อมูลการเปรียบเทียบจำนวนสมาชิกที่มีอยู่จริงกับสมาชิกทางทฤษฎีของสองไดเมนชัน 41
ตารางที่ 6-1 แสดง file set ของ PivotTable Service 44
ตารางที่ 6-2 แสดงไฟล์ที่จำเป็นสำหรับแต่ละงาน 44
ตารางที่ 6-3 แสดงไฟล์ที่จะติดตั้งโดย Ptslite.exe 45
ตารางที่ 9-1แสดงจำนวนเรคอร์ดในตารางแฟ็คและไดเมนชันของการเพิ่มเรเคอร์ดในตารางแฟ็ค 75
ตารางที่ 9-2 แสดงผลการทดสอบระยะเวลาคิวรีของ 1 ล้านเรคอร์ด 76
ตารางที่ 9-3 แสดงผลการทดสอบระยะเวลาคิวรีของ 2 ล้านเรคอร์ด 76
ตารางที่ 9-4 แสดงผลการทดสอบระยะเวลาคิวรีของ 3 ล้านเรคอร์ด 77
ตารางที่ 9-5 แสดงผลการทดสอบระยะเวลาคิวรีของ 4 ล้านเรคอร์ด 77
ตารางที่ 9-6 แสดงผลการทดสอบระยะเวลาคิวรีของ 5 ล้านเรคอร์ด 78
ตารางที่ 9-7 แสดงจำนวนเรคอร์ดในตารางแฟ็คและตารางไดเมนชันของการเพิ่มจำนวนไดเมนชัน 79
ตารางที่ 9-8 ผลการทดสอบระยะเวลาการคิวรีของ 1 ไดเมนชัน 79
ตารางที่ 9-9 ผลการทดสอบระยะเวลาการคิวรีของ 2 ไดเมนชัน 80
ตารางที่ 9-10 ผลการทดสอบระยะเวลาการคิวรีของ 3 ไดเมนชัน 80
ตารางที่ 9-11 แสดงจำนวนเรคอร์ดในตารางแฟ็คและตารางไดเมนชันของการเพิ่มการรวมค่า 81
ตารางที่ 9-12 ผลการทดสอบระยะเวลาการคิวรีของ MOLAP ด้วยการรวมค่าค่าต่างๆ 81
ตารางที่ 9-13 ผลการทดสอบระยะเวลาการคิวรีของ HOLAP ด้วยการรวมค่าค่าต่างๆ 82
ตารางที่ 9-14 ผลการทดสอบระยะเวลาการคิวรีของ ROLAP ด้วยการรวมค่าค่าต่างๆ 82
ตารางที่ 9-15 แสดงจำนวนเรคอร์ดของสกีมาแบบ 2 ไดเมนชัน 84
ตารางที่ 9-16 แสดงจำนวนเรคอร์ดของสกีมาแบบ 4 ไดเมนชัน 85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

หน้าที่

ตารางที่ 9-17 ผลทดสอบการเก็บข้อมูลแบบ MOLAP กับคิวบ์ 3 ล้านเรคอร์ด 2 ไดเมนชัน 87
ตารางที่ 9-18 ผลทดสอบการเก็บข้อมูลกับแบบ ROLAP คิวบ์ 3 ล้านเรคอร์ด 2 ไดเมนชัน 87
ตารางที่ 9-19 ผลทดสอบการเก็บข้อมูลแบบ HOLAP กับคิวบ์ 3 ล้านเรคอร์ด 2 ไดเมนชัน 88
ตารางที่ 9-20 ผลทดสอบการเก็บข้อมูลแบบ MOLAP กับคิวบ์ 3 ล้านเรคอร์ด 4 ไดเมนชัน 88
ตารางที่ 9-21 ผลทดสอบการเก็บข้อมูลแบบ ROLAP กับคิวบ์ 3 ล้านเรคอร์ด 4 ไดเมนชัน 89
ตารางที่ 9-22 ผลทดสอบการเก็บข้อมูลแบบ HOLAP กับคิวบ์ 3 ล้านเรคอร์ด 4 ไดเมนชัน 89
ตารางที่ 9-23 ตาราง Process Time ของกสีมาแบบ 2 ไดเมนชันที่อยู่ในหน่วยวินาที 90
ตารางที่ 9-24 ตาราง Process Time ของกสีมาแบบ 4 ไดเมนชันที่อยู่ในหน่วยวินาที 91
ตารางที่ ก-1 แสดงผลลัพธ์การคิวรีจากคิวบ์ TestCube 102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้าที่

รูปที่ 2-1 แสดงการเก็บข้อมูลตามหัวข้อและเก็บตามแอพพลิเคชัน 4
รูปที่ 2-2 แสดงการรวมข้อมูลก่อนเข้าสู่ดาต้าแวร์เฮาส์ 5
รูปที่ 2-3 แสดง non volatilily 6
รูปที่ 2-4 แสดงลักษณะของดาต้าแวร์เฮาส์ซึ่งเก็บข้อมูลโดยมีเวลาเป็นองค์ประกอบ 6
รูปที่ 2-5 แสดงความสัมพันธ์ระหว่างดาต้าแวร์เฮาส์และ OLAP 10
รูปที่ 3-1 แสดงตัวอย่าง Dimensional Modeling 11
รูปที่ 3-2 ลำดับขั้นในการออกแบบไดเมนชัน 13
รูปที่ 3-3 แสดงตารางไดเมนชัน 14
รูปที่ 3-4 แสดงโครงสร้างฐานข้อมูลที่เป็นสกีมาแบบดาว (Star Schema) 17
รูปที่ 3-5 แสดงโครงสร้างฐานข้อมูลที่เป็นสกีมาแบบสโนว์เฟลก (Snowflake Schema) 18
รูปที่ 3-6 แสดงการรวมค่า (Aggregation) 19
รูปที่ 3-7 แสดงข้อมูลในตารางแบบปกติ 19
รูปที่ 3-8 แสดงข้อมูลในตารางแบบ CrossTabs 19
รูปที่ 4-1 แสดงสถาปัตยกรรมภายในของ Analysis Service 21
รูปที่ 4-2 การติดตั้ง Data Source Name 22
รูปที่ 4-3 รูปแสดงหน้าจอโปรแกรม Anlysis Manager 23
รูปที่ 4-4 การเลือก Driver เพื่อใช้ในการติดต่อกับแหล่งข้อมูล 24
รูปที่ 4-5 การเลือก Data Source Name 25
รูปที่ 4-6 แสดงหน้าจอ Cube Wizard 26
รูปที่ 4-7 แสดงการเลือกชนิดของไดเมนชัน 27
รูปที่ 4-8 แสดงการเชื่อมต่อของ 2 ตารางเมื่อทำ Snowflake schema 28
รูปที่ 4-9 แสดงรูปสกีมาของคิวบ์ Sales 29
รูปที่ 4-10 เลือกวิธีการจัดเก็บข้อมูล 30
รูปที่ 4-11 แสดงการวัดประสิทธิภาพและขนาดของ Cube ที่สร้างขึ้นมา 31
รูปที่ 4-12 แสดงรายละเอียดของข้อมูล (Meta data) 31
รูปที่ 4-13 แสดงการเลือก Level เพื่อ Slice และ Dice ข้อมูล 32
รูปที่ 4-14 แสดงการ Drill down ข้อมูล 33
รูปที่ 5-1 แสดงการเก็บข้อมูลด้วยวิธีต่างๆ 35
รูปที่ 5-2 แสดงฐานข้อมูลก่อนถูกทำการรวมค่า (Aggregation) ด้วยการเก็บค่าแบบ ROLAP 36
รูปที่ 5-3 แสดงฐานข้อมูลหลังจากทำการรวมค่า (Aggregation) ด้วยการเก็บค่าแบบ ROLAP 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

หน้าที่

รูปที่ 5-4 แสดงตารางที่ถูกสร้างขึ้นหลังจากการรวมค่าเพื่อเก็บค่าที่สรุปค่าของตัววัดในระดับ Month 38

รูปที่ 5-5 สรุปค่าของตัววัดในระดับ Quarter 38

รูปที่ 5-6 สรุปค่าของตัววัดในระดับ Year 38

รูปที่ 5-7 แสดงตารางสรุปค่าของตัววัดในระดับ ALL 39

รูปที่ 5-8 รูปแสดงตารางการรวมค่าของสองไดเมนชัน 40

รูปที่ 6-1 แสดงโครงสร้างของไคลเอ็นต์ใน Analysis Service 42

รูปที่ 6-2 แสดงการเชื่อมต่อกับ Analysis 46

รูปที่ 6-3 แสดงการเชื่อมต่อโดยใช้ HTTP 47

รูปที่ 6-4 แสดงการเชื่อมต่อกับ OLE DB Provider 48

รูปที่ 6-5 แสดงการเชื่อมต่อกับ ไฟล์โลคอลคิวบ์ 49

รูปที่ 6-6 แสดงโครงสร้างลำดับขั้น 50

รูปที่ 7-1 แสดงฟอร์มเริ่มแรกของโปรแกรมประยุกต์ 53

รูปที่ 7-2 แสดงฟอร์มการทำงานหลักของโปรแกรม 54

รูปที่ 7-3 แสดงฟอร์ม Login 55

รูปที่ 7-4 แสดงการแสดงผลเริ่มแรกของฟอร์มหลัก 55

รูปที่ 7-5 การแสดงผลข้อมูลที่อยู่ในคิวบ์ของโปรแกรมประยุกต์ 56

รูปที่ 7-6 แสดงผลการ drill down 57

รูปที่ 7-7 แสดงภาพการทำ slice ข้อมูลของโปรแกรมประยุกต์ 57

รูปที่ 7-8 แสดงการสลับการแสดงผลระหว่างแถวกับคอลัมน์ 58

รูปที่ 8-1 สถาปัตยกรรมการทำงานของโปรแกรมประยุกต์ที่ทำงานผ่านเว็บบราวเซอร์ 59

รูปที่ 8-2 แสดงหน้าจอเลือกเซิร์ฟเวอร์ที่ต้องการติดต่อด้วย 61

รูปที่ 8-3 แสดงหน้าจอ เลือกฐานข้อมูล 61

รูปที่ 8-4 แสดงหน้าจอเลือกคิวบ์ 62

รูปที่ 8-5 แสดงหน้าจอการทำงานหลักของโปรแกรม 63

รูปที่ 8-6 แสดงการลากไดเมนชัน Education Level มาวางบนแกนแถว(Row)ของตารางแสดงผล 63

รูปที่ 8-7 แสดงผลลัพธ์หลังจากลากไดเมนชัน Education Level มาวาง 64

รูปที่ 8-8 แสดงการลากเลเวล [Gender].[Gender] มาวางในแกนแถวของตารางแสดงผล 64

รูปที่ 8-9 แสดงผลไดเมนชัน 2 ไดเมนชันบนแกนแถวของตารางแสดงผล 65

รูปที่ 8-10 แสดงการการลากไดเมนชัน Gender ออกจากตารางแสดงผล 66

รูปที่ 8-11 แสดงผลลัพธ์หลังจากดึงไดเมนชัน Gender ออกไปแล้ว 66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

หน้าที่

รูปที่ 8-12 แสดงหน้าจอ หลังจากลากไดเมนชัน Product มาวางยังส่วน Slice Dimension 68
รูปที่ 8-13 แสดงการกางสมาชิกเพื่อเลือกสมาชิกในระดับที่ลึกลงไป 69
รูปที่ 8-14 แสดงการทำ Drill down 69
รูปที่ 8-15 แสดงผลข้อมูลในรูปกราฟ โดยการคลิกที่ปุ่มกราฟ 70
รูปที่ 8-16 แสดงหน้าจอการทำกราฟแบบกำหนดข้อมูลเอง 71
รูปที่ 8-17 แสดงหน้าจอการเลือกข้อมูลที่จะแสดงบนแกนคอลัมน์ 71
รูปที่ 8-18 แสดงหน้าจอการเลือกข้อมูลที่จะแสดงบนแกนคอลัมน์ 71
รูปที่ 8-19 แสดงผลลัพธ์หลังจากการระบุข้อมูลบนแกนแถวและแกนคอลัมน์ 72
รูปที่ 8-20 แสดงกราฟที่ผู้ใช้ระบุข้อมูลบนแกน x และแกน y เอง 72
รูปที่ 8-21 แสดงผลข้อมูลในรูปแบบของ Microsoft Excel 73
รูปที่ 9-1 รูปแสดงเวลาในการคิวรีเมื่อมีจำนวนเรคอร์ดต่างๆ กัน 78
รูปที่ 9-2 รูปแสดงเวลาในการคิวรีเมื่อมีจำนวนไดเมนชันต่างๆ กัน 80
รูปที่ 9-3 รูปแสดงเวลาในการคิวรีเมื่อมีจำนวนการรวมค่าต่างๆ กัน 83
รูปที่ 9-4 แสดงสกีมาแบบ 2 ไดเมนชัน 85
รูปที่ 9-5 แสดงสกีมาแบบ 4 ไดเมนชัน 86
รูปที่ 9-6 กราฟ แสดงเวลาในการสร้างคิวบ์ต่อการเพิ่มของการรวมค่า (2ไดเมนชัน) 90
รูปที่ 9-7 กราฟแสดงเวลาในการสร้างคิวบ์ต่อการเพิ่มของการรวมค่า (4 ไดเมนชัน) 91
รูปที่ 9-8 รูปแสดงพื้นที่ที่ใช้บน Analysis Service (2ไดเมนชัน) 92
รูปที่ 9-9 รูปแสดงพื้นที่ที่ใช้บน Analysis Service (4 ไดเมนชัน) 92
รูปที่ ก-1 แสดงโครงสร้างของรีเลชันแนลดาต้าเบส (relational database) 94
รูปที่ ก-2 แสดงโครงสร้างของคิวบ์ 95
รูปที่ ก-3 แสดงส่วนของ member 99
รูปที่ ก-4 แสดงส่วนของ tuple 100
รูปที่ ก-5 แสดงส่วนของ tuple 101
รูปที่ ก-6 แสดงผลลัพธ์ของคำสั่ง Crossjoin 104
รูปที่ ก-7 แสดงผลลัพธ์การคิวรี 106
รูปที่ ก-8 แสดงผลลัพธ์การคิวรีเมื่อใช้ ToggleDrillState 106

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันนี้การใช้ข้อมูลเป็นสิ่งจำเป็นอย่างยิ่งผู้ที่สามารถนำข้อมูลมาใช้ได้อย่างมีประสิทธิภาพมากกว่าและรวดเร็วกว่าก็จะส่งผลให้ก่อให้เกิดข้อได้เปรียบเหนือผู้อื่น สิ่งที่จำเป็นต้องทำก่อนนำข้อมูลมาใช้นั่นคือการเก็บข้อมูล การเก็บข้อมูลจะมีการนำระบบฐานข้อมูลเข้ามาใช้ โดยระบบฐานข้อมูลที่ใช้นี้จะมีลักษณะเป็นระบบฐานข้อมูลสำหรับงานประจำวัน (OLTP) ลักษณะที่สำคัญของระบบฐานข้อมูลดังกล่าวคือ สนับสนุนให้ผู้ใช้หลายคนให้สามารถเข้ามาทำงานในเวลาเดียวกันได้ การเก็บข้อมูลไม่ได้เก็บสัมพันธ์กับเวลาโดยจะเก็บข้อมูลล่าสุดเสมอ มีโครงสร้างที่ซับซ้อน มักถูกนำไปใช้กับงานทรานแซคชัน (transaction) ถูกนำไปใช้กับงานที่ทำในลักษณะวันต่อวัน

การใช้ข้อมูลที่สำคัญอย่างหนึ่งสำหรับการดำเนินธุรกิจในทุกวันนี้คือ การเรียกใช้ข้อมูลเพื่อการตัดสินใจ โดยจำเป็นต้องทำได้อย่างรวดเร็วและมีประสิทธิภาพ ดังนั้นฐานข้อมูลประจำวันจึงไม่เหมาะสมกับการเรียกใช้ข้อมูลในลักษณะนี้ด้วยสาเหตุจากการคิวรี (query) เพื่อการตัดสินใจมักจะเป็นการคิวรีแบบเฉพาะกิจ (ad hoc query) ผู้เรียกใช้จำเป็นต้องมีความรู้ทางด้านเทคนิค การคิวรีต้องใช้คำสั่งที่ซับซ้อนซึ่งอาจจะทำให้ประสิทธิภาพของระบบต่ำลงมากหรือไม่อาจจะคาดการณ์ได้ ทำให้ไม่เหมาะกับการทำการวิเคราะห์แบบออนไลน์ (online analytical) การเก็บข้อมูลในระบบฐานข้อมูลประจำวันไม่ได้มีการเก็บข้อมูลย้อนหลัง (historical Data) แต่บางครั้งการตัดสินใจจำเป็นต้องใช้ข้อมูลที่ผ่านมาเพื่อช่วยในการคาดคะเนแนวโน้มที่จะเป็นไปได้ในอนาคต

ดังนั้นจึงได้ทำการศึกษาระบบฐานข้อมูลในรูปแบบอื่น ที่สามารถนำมาใช้สนับสนุนการตัดสินใจได้ ซึ่งก็คือระบบฐานข้อมูลดาต้าแวร์เฮาส์โดยที่ระบบฐานข้อมูลดาต้าแวร์เฮาส์นั้นได้ออกแบบมามีจุดประสงค์เพื่อจัดการกับข้อมูลและจัดสรรให้เกิดข้อมูลที่รองรับสำหรับการวิเคราะห์และตัดสินใจ โดยมีลักษณะดังนี้ คือ รวมข้อมูลจากหลายๆ แหล่งข้อมูล (Heterogeneous data source) ให้อยู่ในรูปแบบเดียวกัน มีการจัดการโครงสร้างของข้อมูลให้มีประสิทธิภาพต่อการคิวรีเพื่อการวิเคราะห์มากกว่าระบบฐานข้อมูลประจำวัน มีการเก็บข้อมูลย้อนหลังเพื่อช่วยให้การตัดสินใจมีความแม่นยำมากขึ้น

1.2 วัตถุประสงค์ของการทำโครงงาน

1.2.1 ศึกษาระบบฐานข้อมูลดาต้าแวร์เฮาส์ คืออะไร มีลักษณะอย่างไร นำมาใช้ประโยชน์ได้อย่างไร

1.2.2 ศึกษาวิธีการออกแบบระบบฐานข้อมูลดาต้าแวร์เฮาส์ โดยใช้โมเดลสกีมาแบบดาว (Star schema) และสโนว์เฟลกสกีมา (Snowflake schema)

1.2.3 ศึกษาระบบจัดการฐานข้อมูล Microsoft SQL Server 2000 และนำมาสร้างฐานข้อมูลดาต้าแวร์เฮาส์โดยใช้ฟังก์ชันการทำงานมีมาให้ใน Analysis Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.4 ศึกษาการเก็บข้อมูลในลักษณะต่างๆ ของระบบดาต้าแวร์เฮาส์ซึ่งได้แก่ MOLAP, ROLAP, HOLAP ว่าแต่ละแบบมีข้อดีข้อเสียอย่างไร จากนั้น ก็ทดลองเก็บข้อมูลในทุกรูปแบบแล้วนำผลลัพธ์ที่ได้มาเปรียบเทียบกัน

1.2.5 พัฒนาทูลส์ที่ใช้สำหรับคิวรีระบบฐานข้อมูลดาต้าแวร์เฮาส์

1.3 ขอบเขตของโครงงาน

โครงงานนี้จะมุ่งเน้นไปที่การศึกษาระบบฐานข้อมูลดาต้าแวร์เฮาส์ว่าคืออะไร มีประโยชน์อย่างไร และสามารถนำมาใช้ได้อย่างไร จากนั้นก็ทดลองสร้างระบบฐานข้อมูลดาต้าแวร์เฮาส์โดยใช้ฟังก์ชันการทำงานที่มีมาให้ใน Microsoft SQL Server 2000 โดยได้ทดลองเก็บข้อมูลดาต้าแวร์เฮาส์ในแบบต่างๆ คือ MOLAP , ROLAP และ HOLAP แล้วทำการวัดผลลัพธ์ในด้านต่างๆ แล้วนำมาเปรียบเทียบกัน

หลังจากนั้นได้ทำการพัฒนาโปรแกรมประยุกต์ทางฝั่งไคลเอ็นต์ขึ้นมา เพื่อให้ผู้ใช้สามารถเรียกคิวรีระบบฐานข้อมูลดาต้าแวร์เฮาส์ได้โดยสะดวก โดยโปรแกรมประยุกต์ที่ได้พัฒนาขึ้นมามีทั้งแบบที่ทำงานบนแลน (LAN) และทำงานผ่านเว็บบราวเซอร์

โดยโครงงานนี้ไม่ได้เน้นไปที่การออกแบบระบบฐานข้อมูลดาต้าแวร์เฮาส์ เราจะสมมุติว่าฐานข้อมูลดาต้าแวร์เฮาส์ได้ถูกออกแบบและสร้างเอาไว้เรียบร้อยแล้ว โปรแกรมประยุกต์ที่เราพัฒนาขึ้นมานี้ก็จะสามารถเรียกใช้ข้อมูลที่เตรียมไว้ได้เลย

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาและทำความเข้าใจเกี่ยวกับระบบฐานข้อมูลดาต้าแวร์เฮาส์ ประโยชน์ของฐานข้อมูลดาต้าแวร์เฮาส์ที่เหนือกว่าฐานข้อมูลประจำวัน และความเหมาะสมกับการนำไปใช้

1.4.2 ศึกษาทฤษฎีที่เกี่ยวข้องกับระบบฐานข้อมูลดาต้าแวร์เฮาส์ โครงสร้างของข้อมูล รูปแบบการเก็บข้อมูล รวมไปถึงโมเดลการออกแบบระบบดาต้าแวร์เฮาส์

1.4.3 ศึกษาวิธีการใช้งาน วิธีการติดตั้ง ของซอฟต์แวร์ SQL Server 2000 และศึกษาฟังก์ชันการทำงานที่ใช้สำหรับทำดาต้าแวร์เฮาส์บนโปรดักส์นี้

1.4.4 ศึกษารายละเอียดของการเก็บข้อมูลในรูปแบบ MOLAP , ROLAP , HOLAP แล้วทดลองเก็บข้อมูลในลักษณะต่างๆ จากนั้นนำผลลัพธ์ที่ได้มาเปรียบเทียบกันไม่ว่าจะเป็นด้านเวลาตอบสนองคิวรี (Query time) เวลาในการสร้าง (Process time) เนื้อที่ที่ใช้ในการเก็บ (Space)

1.4.5 พัฒนาโปรแกรมประยุกต์ทางฝั่งไคลเอ็นต์ที่ทำงานบนแลนขึ้นมา โดยได้เลือกใช้ Microsoft Visual Basic 6.0

1.4.6 ขยายขนาดของฐานข้อมูลเพื่อวัดประสิทธิภาพของโปรแกรมประยุกต์ที่พัฒนาขึ้นมา โดยเขียนโปรแกรมเพื่อเพิ่มข้อมูลลงในฐานข้อมูล จากนั้นวัดเวลาตอบสนองของโปรแกรม

1.4.7 พัฒนาโปรแกรมบนเว็บ เพื่อให้ผู้ใช้สามารถคิวรีข้อมูลจากระยะไกลได้ โดยพยายามทำให้สามารถใช้งานได้ง่ายและมีฟังก์ชันการทำงานที่ใกล้เคียงกับโปรแกรมประยุกต์บนแลน นอกจากนั้นยังพยายามทำให้มีการใช้งานที่ยืดหยุ่นมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ปัจจุบันนี้จะเห็นได้ว่าหลายๆ องค์กรมักจะมีการทำระบบดาต้าแวร์เฮาส์ของแต่ละองค์กร เนื่องจากว่าการตัดสินใจในทางธุรกิจเป็นที่สำคัญและจำเป็นต้องทำให้ได้อย่างรวดเร็วและถูกต้อง การที่จะทำเช่นนั้นได้จำเป็นต้องมีข้อมูลสนับสนุน ซึ่งระบบฐานข้อมูลดาต้าแวร์เฮาส์สามารถนำมาใช้ในส่วนของการเตรียมข้อมูลสนับสนุนการตัดสินใจได้ และในโครงงานนี้ได้อธิบายถึงทฤษฎีและลักษณะของระบบฐานข้อมูลดาต้าแวร์เฮาส์ซึ่งคาดว่าจะสามารถนำไปใช้ให้เป็นประโยชน์ นอกจากนี้โครงงานนี้ยังได้มีการพัฒนาโปรแกรมประยุกต์ทางฝั่งไคลเอ็นต์ขึ้นมา เพื่อที่ว่าจะได้สามารถนำไปใช้ได้กับระบบดาต้าแวร์เฮาส์อื่นๆ ที่สร้างบน Microsoft SQL Server ได้อย่างสะดวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อกจรศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีเบื้องต้น

2.1 นิยามของดาต้าแวร์เฮาส์

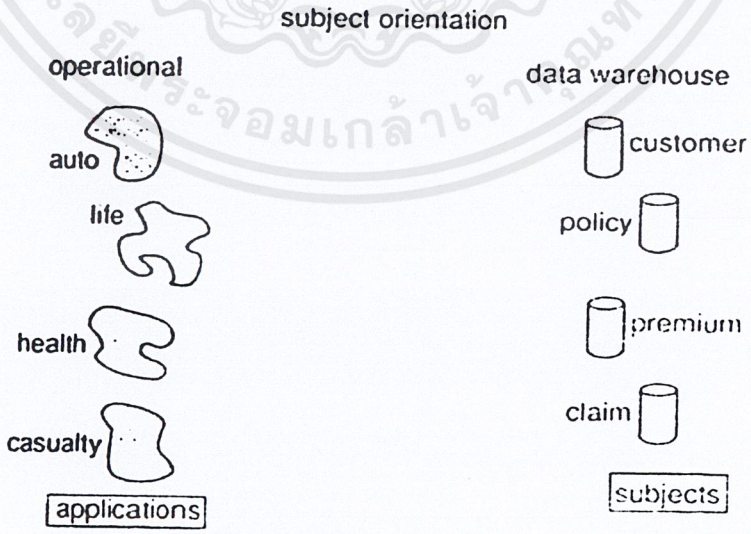
ดาต้าแวร์เฮาส์คือ การรวบรวมข้อมูลจากฐานข้อมูลของส่วนปฏิบัติงาน (Operational Database) หลายๆ รูปแบบหรืออาจจะมาจากแหล่งข้อมูลที่สำคัญและจำเป็นอื่นๆ มาทำการแปลงหรือสรุปให้อยู่ในรูปของฐานข้อมูลที่มีรูปแบบเหมาะสมต่อการใช้ในการวิเคราะห์ การเก็บรวบรวมและการนำกลับมาใช้ทำให้ได้เป็นแหล่งรวมของข้อมูลที่อยู่ในความสนใจของผู้ใช้ เพื่อใช้ประกอบในการตัดสินใจ ใช้เป็นข้อมูลทางธุรกิจ การวางแผน หรือเป็นข้อมูลสำหรับผู้บริหาร ซึ่งจะช่วยให้สามารถทำการตัดสินใจได้อย่างถูกต้องโดยง่าย และรองรับข้อมูลจำนวนมากได้ เป็นการนำเสนอแนวทางในการเข้าถึงข้อมูลในองค์กรได้อย่างมีประสิทธิภาพ

2.2 ลักษณะที่สำคัญของดาต้าแวร์เฮาส์

ดาต้าแวร์เฮาส์เป็นพื้นฐานของระบบสนับสนุนการตัดสินใจ นั่นคือดาต้าแวร์เฮาส์เป็นฐานข้อมูลที่ช่วยในกระบวนการตัดสินใจ โดยฐานข้อมูลดังกล่าวจะต้องมีลักษณะดังต่อไปนี้

2.2.1 เก็บข้อมูลตามหัวข้อ (Subject Oriented)

ตามปกติถ้าเป็นงานในระดับโอเปอเรชัน (Operational) ข้อมูลจะถูกจัดเก็บมาจากแต่ละแอพพลิเคชันขององค์กร เช่น ถ้าเป็นข้อมูลของบริษัทประกันภัย ข้อมูลที่ได้จากแอพพลิเคชันจะเก็บตามประวัติ ข้อมูลสุขภาพ การดำเนินชีวิต ฯลฯ ในขณะที่หัวข้อหลักในการเก็บข้อมูลขององค์กรประกอบด้วย ข้อมูลลูกค้านโยบายสิทธิพิเศษและการเอาประกันเป็นต้น แสดงความแตกต่างนี้ได้ดังรูป

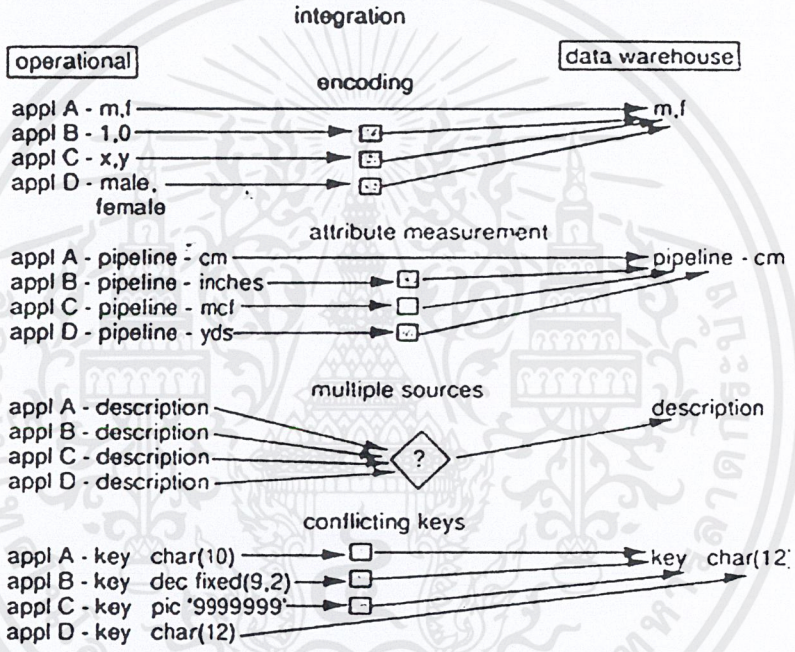


รูปที่ 2-1 แสดงการเก็บข้อมูลตามหัวข้อและเก็บตามแอพพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 มีการรวบรวมข้อมูลเข้าด้วยกัน (Integrated)

การรวบรวมข้อมูลเข้าด้วยกันเป็นลักษณะหนึ่งที่สำคัญมากของดาต้าแวร์เฮาส์ แสดงได้ดังรูปที่ 2-2 โดยการรวบรวมข้อมูลนี้เกิดขึ้นในขณะที่มีการส่งข้อมูลจากงานโอเปอเรชัน (ซึ่งการจัดเก็บข้อมูลทำตามแต่ละแอพพลิเคชัน) มายังส่วนของดาต้าแวร์เฮาส์ เนื่องจากการออกแบบงานในระดับแอพพลิเคชันได้มีขึ้นอยู่ตลอดเวลา ทำให้ขาดการปรับเปลี่ยนรหัสข้อมูล ชื่อข้อมูลที่ใช้ แอททริบิวต์ (Attribute) ที่ใช้งาน หน่วยการวัดและอื่นๆ ของข้อมูลในแต่ละแอพพลิเคชันให้สอดคล้องกัน ทำให้ข้อมูลที่เข้ามายังดาต้าแวร์เฮาส์มีความสับสนจากแต่ละแอพพลิเคชันที่เข้ามา ตัวอย่างเช่น รหัสของเพศที่เข้ามายังดาต้าแวร์เฮาส์อาจเข้ามาในรูปแบบของ m/f หรือ 1/0 หรือ x/y และอื่นๆ ได้อีกมากมาย รหัสดังกล่าวจึงจำเป็นต้องถูกปรับเปลี่ยนให้อยู่ในรูปแบบเดียวกันก่อนเก็บในดาต้าแวร์เฮาส์

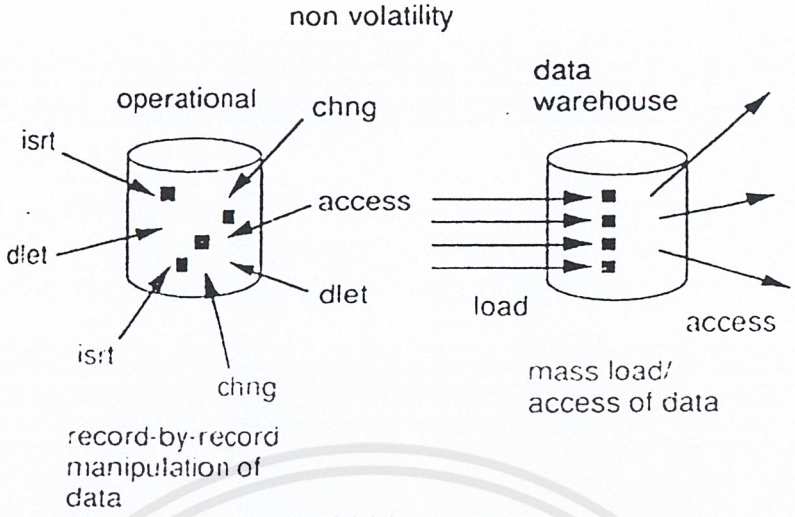


รูปที่ 2-2 แสดงการรวมข้อมูลก่อนเข้าสู่ดาต้าแวร์เฮาส์

2.2.3 ข้อมูลจะไม่เปลี่ยนแปลงได้ง่ายๆ (Nonvolatile)

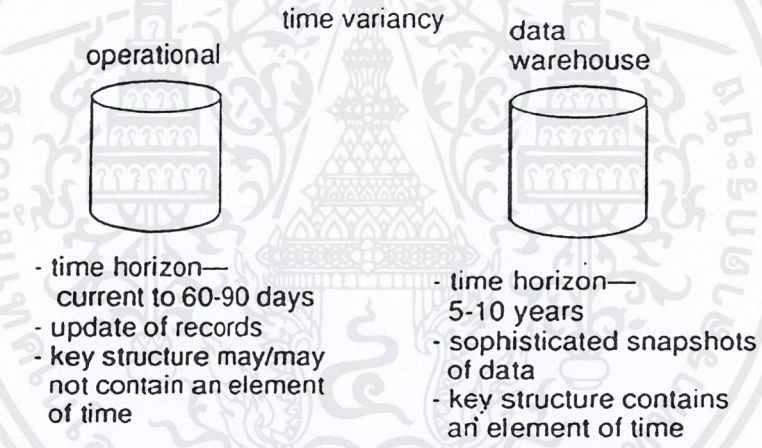
ลักษณะของข้อมูลในดาต้าแวร์เฮาส์จะไม่มีการเปลี่ยนแปลงบ่อยๆ ในขณะที่ข้อมูลในส่วนของโอเปอเรชันจะมีการเข้าถึงข้อมูลและจัดการกับข้อมูลนั้นได้ครั้งละ 1 เรคอร์ด การอัพเดตจะเกิดขึ้นที่ตัวข้อมูลแต่ในดาต้าแวร์เฮาส์จะมีรูปแบบการจัดเก็บข้อมูลที่แตกต่างออกไป เนื่องจากข้อมูลจะถูกโหลดเข้ามาและถูกเข้ามาใช้งานได้ แต่การอัพเดตข้อมูล (การเปลี่ยนแปลงที่ตัวเนื้อข้อมูลจริงๆ) จะไม่เกิดขึ้นภายในดาต้าแวร์เฮาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-3 แสดง non volatilily

2.2.4 มีเวลาเป็นองค์ประกอบ (Time variant)



รูปที่ 2-4 แสดงลักษณะของดาต้าแวร์เฮาส์ซึ่งเก็บข้อมูลโดยมีเวลาเป็นองค์ประกอบ

ตามรูปที่ 2-4 แสดงให้เห็นว่าเวลาที่ใช้ในการเก็บข้อมูลของดาต้าแวร์เฮาส์ จะนานกว่าที่เก็บในส่วนของงานโอเปอเรชัน ซึ่งโดยทั่วไปแล้วภายในดาต้าแวร์เฮาส์จะเก็บข้อมูลอยู่ในช่วง 5-10 ปี ในขณะที่ฐานข้อมูลของงานโอเปอเรชันจะเก็บค่าข้อมูลในขณะนั้น (current value) ในทางกลับกันดาต้าแวร์เฮาส์จะเก็บข้อมูลในหลายๆ ช่วงเวลา

โครงสร้างของข้อมูลในงานโอเปอเรชันไม่จำเป็นจะต้องประกอบด้วยค่าที่เกี่ยวกับเวลา เช่น ปี เดือน วัน ในขณะที่โครงสร้างของดาต้าแวร์เฮาส์จะต้องประกอบด้วยเวลาเป็นแกนหนึ่งเสมอ คือจะต้องมีองค์ประกอบของเวลาเข้ามารวมอยู่ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 คุณสมบัติของดาต้าแวร์เฮาส์

ดาต้าแวร์เฮาส์ได้รับการออกแบบให้เหมาะกับงานวิเคราะห์ข้อมูลในลักษณะ ข้อมูลรายปี รายสัปดาห์ หรือรายวัน ตัวอย่างเช่น ระบบวิเคราะห์ยอดขาย ยอดเงินเอาประกัน ซึ่งต้องอาศัยข้อมูลในการวิเคราะห์และวางแผนในการบริหารนั้น โดยดาต้าแวร์เฮาส์จะต้องมีคุณสมบัติดังนี้

2.3.1 โครงสร้างข้อมูลต้องสื่อให้ผู้ใช้เข้าใจ ซึ่งต้องทำการรวบรวมข้อมูลจากตารางต่างๆ ในหลายๆ ที่หรืออาจต้องทำการสรุปข้อมูลเตรียมไว้ก่อน

2.3.2 ข้อมูลในฐานข้อมูลค่อนข้างจะคงที่ การเปลี่ยนแปลงของฐานข้อมูลจะเกิดขึ้นเฉพาะในขณะที่ทำการถ่ายโอนข้อมูลลงในดาต้าแวร์เฮาส์

2.3.3 ระบบจะต้องสามารถรองรับข้อมูลจำนวนมากได้

2.3.4 จะต้องมีการสำรองข้อมูล (Back Up) โดยจะทำทั้งหมดหรือบางส่วนก็ได้

โดยกิจกรรมที่เกิดในคลังข้อมูลสามารถแบ่งได้เป็น 2 ส่วนหลักคือ ส่วนแรกคือระบบสนับสนุนการตัดสินใจ (Decision Support Systems :DSS) ที่มีการเข้าถึงข้อมูลจากผู้ใช้ (end user) และส่วนที่สองคือการทำงานของระบบที่ต้องทำการถ่ายเทและจัดการข้อมูลภายในดาต้าแวร์เฮาส์เอง

2.4 ความแตกต่างของดาต้าแวร์เฮาส์กับระบบฐานข้อมูลประจำวัน

ดาต้าแวร์เฮาส์และระบบฐานข้อมูลประจำวันนั้นมีความแตกต่างกันมากมาย ดังต่อไปนี้

2.4.1 ความถูกต้อง (Consistency)

ทั้งงานโอเปอเรชันและดาต้าแวร์เฮาส์ ต่างก็ให้ความสำคัญในเรื่องความถูกต้องของข้อมูล (data consistency) สำหรับงานโอเปอเรชันที่มีการทำทรานแซคชันจำนวนมากๆนั้น สิ่งที่ต้องการคือ การทำทรานแซคชันให้ครบไม่มีสูญหาย ดังนั้นจึงมีความจำเป็นที่ผู้ส่งและผู้รับจะต้องรับรู้และตรวจสอบอยู่ตลอดเวลาว่าขณะนี้มีการทำทรานแซคชันเกิดขึ้นหรือไม่

สำหรับดาต้าแวร์เฮาส์จะไม่สนใจการทำทรานแซคชันแต่ละครั้ง แต่จะสนใจว่า การโหลดข้อมูลใหม่เข้ามานั้นทำเสร็จแล้วหรือยัง นั่นคือสนใจว่าโหลดข้อมูลเข้ามาทั้งหมดและข้อมูลถูกต้องหรือไม่

2.4.2 ทรานแซคชัน (Transaction)

สำหรับระบบงานโอเปอเรชันนั้นในแต่ละวันอาจมีการทำทรานแซคชันเป็นหมื่นเป็นแสนครั้ง ซึ่งการทำทรานแซคชันแต่ละครั้งจะใช้ข้อมูลเพียงเล็กน้อยเท่านั้น อาจเป็นการใช้ข้อมูลเพียง 1 เรคอร์ด

สำหรับดาต้าแวร์เฮาส์ แต่ละวันจะทำเพียงแค่ 1 ทรานแซคชัน ซึ่งการทำทรานแซคชันนี้อาจต้องใช้ข้อมูลเป็นล้านเรคอร์ดเลยก็ได้ ดังนั้นเราจึงเรียกกระบวนการนี้ว่าเป็น production data load แทนสิ่งที่เราสนใจในกระบวนการนี้มีเพียงแค่ production data load เท่านั้น ถ้าการทำ production data load ถูกทำให้หยุดกลางคัน ระบบจะทำการเอาข้อมูลที่เคยมีมาก่อนที่จะทำ production data load มาเขียนทับลงทันที

2.4.3 ระดับของผู้ใช้งาน

สำหรับงานออนไลน์ทรานแซคชัน (OLTP) นั้น ผู้ใช้คือ ผู้ที่ทำงานอยู่ในองค์กรนั้น อาจจะเป็นเจ้าหน้าที่ที่ทำหน้าที่รับออร์เดอร์, เจ้าหน้าที่ที่ทำหน้าที่ตรวจสอบและดูแลเงิน, เจ้าหน้าที่ที่ทำหน้าที่คอยบริการลูกค้าใหม่, เจ้าหน้าที่ที่ทำหน้าที่ป้อนข้อมูลเข้าไป ซึ่งส่วนใหญ่ผู้ใช้เหล่านี้ในช่วงเวลาหนึ่งจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานกับ 1 แอคเคาท์ (account) เท่านั้น ผู้ใช้งานออนไลน์ทรานแซคชันนี้มักจะทำงานที่มีลักษณะเป็นงานซ้ำ ๆ เดิม ๆ รายงานส่วนใหญ่ที่ได้จากการทำงานบนระบบออนไลน์ทรานแซคชันนี้มักจะมีลักษณะเป็นรายลิสต์ของทั้งตารางเลย

สำหรับดาต้าแวร์เฮาส์ผู้ใช้คือผู้ที่ทำหน้าที่คอยดูแลการทำงานของพนักงานในองค์กรเท่านั้น ลักษณะการทำงานเช่น คอยนับจำนวนออร์เดอร์ใหม่ ๆ , หาเหตุว่าทำไมลูกค้าจึงไม่พอใจ, คอยตรวจดูว่ามีข้อมูลอะไรใหม่ ๆ เข้ามาบ้าง, คอยตรวจสอบและแก้ไขข้อผิดพลาดของข้อมูล ผู้ใช้ของดาต้าแวร์เฮาส์จะไม่ทำงานทีละ 1 แอคเคาท์ แต่จะพิจารณาจากแอคเคาท์ทั้งหมดแล้วหาคำตอบที่ต้องการออกมา (ชุดคำตอบ (answer set) ขนาดเล็ก ๆ) และคำถามที่ให้กับดาต้าแวร์เฮาส์ก็สามารถเปลี่ยนแปลงได้เรื่อยๆ ซึ่งอาจไม่ใช่คำถามเดิมที่เคยถามก็ได้

2.4.4 ความเกี่ยวพันกับเวลา (Time Dimension)

ระบบโอเปอเรชันจะทำงานอย่างรวดเร็วและทำทรานแซคชันอย่างสม่ำเสมอ การวัดเวลาใช้หน่วยเป็นนาทีและวินาที สถานะของข้อมูลต่าง ๆ มีการเปลี่ยนแปลงอยู่ตลอดเวลา และความสัมพันธ์ระหว่างเอนติตี้ (entity) ต่าง ๆ ก็เปลี่ยนแปลงไปด้วย

ฐานข้อมูลในงานออนไลน์ทรานแซคชันจะขาดการสนับสนุนหรือการอ้างอิงจากข้อมูลในอดีต แต่ทว่าในดาต้าแวร์เฮาส์มักจะมีคำถามที่ถามถึงการวิเคราะห์ข้อมูลในอดีตด้วย และแม้ว่าเราสามารถเก็บข้อมูลอดีตไว้ในงานออนไลน์ทรานแซคชันได้ แต่ก็เป็นภาระหนักของระบบในการทำให้มองเห็นภาพในอดีต และจะทำให้ยากในการทำทรานแซคชัน (ข้อมูลมีมากขึ้น ก็ต้องใช้เวลามากขึ้นด้วย)

2.5 การสร้างข้อมูล (Data Model)

ความแตกต่างที่สำคัญที่สุดของระบบโอเปอเรชันและดาต้าแวร์เฮาส์ คือ การสร้างข้อมูลหรือดาต้าโมเดล (Data Model) นั่นเอง

งานโอเปอเรชันจะใช้ Entity Relation Modeling คือ การกำจัดความซ้ำซ้อนให้หมดไป เพื่อการทำทรานแซคชันจะได้สามารถทำได้ง่ายและรวดเร็วยิ่งขึ้น สามารถทำทรานแซคชันได้โดยเข้าถึงข้อมูลเพียงตัวเดียวเท่านั้น การทำ entity relation modeling นั้นจะใช้วิธีแยกข้อมูล (Normalized) ออกเป็นตารางเล็ก ๆ และแต่ละตารางสามารถเชื่อมต่อไปยังตารางอื่นได้

ข้อสังเกตเกี่ยวกับดาต้าโมเดลของงานโอเปอเรชันซึ่งจะใช้ entity relation diagram มีดังนี้

- ไดอะแกรมนี้มีลักษณะสมมาตร (symmetric) หมายถึง ทุกๆ ตารางเหมือนกันหมด คือ ไดอะแกรมนี้ไม่สามารถบอกได้ว่าตารางไหนสำคัญกว่าหรือใหญ่กว่า และไม่สามารถบอกได้ว่า ตารางใดบรรจุตัววัดที่เป็นชนิดตัวเลข (numerical measurement) ทางธุรกิจ ซึ่งไดอะแกรมรูปแบบนี้ ผู้ใช้ทำความเข้าใจและจดจำได้ยาก
- ถ้ามี 2 ตารางในไดอะแกรมที่ต้องการที่จะรวมกันก็มีหลายวิธีที่จะทำได้และไม่ว่าจะวิธีไหนก็จะได้คำตอบเดียวกัน ซึ่งเป็นเรื่องยากในการเลือกว่าจะใช้เส้นทาง (path) ใดในการเชื่อม (link) ระหว่างตารางที่ต่างกันเพราะถึงแม้ท้ายสุดจะได้คำตอบเดียวกัน แต่ในระหว่างการทำอินเนอร์จอยน์ (Inner join) นั้นจะมีการใช้ดาต้าอิลิเมนท์ (data element) ไม่เหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับดาต้าโมเดลของดาต้าแวร์เฮาส์จะใช้ไดเมนแนลโมเดล (Dimensional Model) หรือ สตาร์จอยน์สกีมา (Star Join Schema) ซึ่งเป็นชื่อที่นักออกแบบฐานข้อมูลใช้กันมานานแล้ว เนื่องจากสกีมามีรูปร่างคล้ายดาว ซึ่งมีตารางหลัก 1 ตาราง อยู่ตรงกลางและมีตารางเล็ก ๆ ที่มีความสัมพันธ์กับตารางหลักอยู่รอบ ๆ ซึ่งตารางหลักนี้เป็นตารางเดียวที่ใช้การเชื่อมต่อหลายจุ (Multiple join) เพื่อเชื่อมต่อกับตารางอื่น ๆ แต่ตารางอื่น ๆ ที่อยู่รอบ ๆ นั้นจะมีแค่การเชื่อมต่อจุดเดียว (Single join) เพื่อเชื่อมเข้ากับตารางหลักเท่านั้น โดยตารางหลักเรียกว่าตารางแฟ็ค (Fact table) ส่วนตารางอื่น ๆ จะเรียกว่าตารางไดเมนชัน (Dimension table)

2.6 ความแตกต่างของลักษณะของข้อมูลที่ใช้ในงานโอเปอเรชันและดาต้าแวร์เฮาส์

ความแตกต่างของลักษณะของข้อมูลที่ใช้ในงานโอเปอเรชันและดาต้าแวร์เฮาส์ สามารถแสดงเปรียบเทียบได้ดังตารางที่ 2-1

ข้อมูลในงานโอเปอเรชัน	ข้อมูลในดาต้าแวร์เฮาส์
- เก็บรายละเอียดของข้อมูลตามแต่ละเรคอร์ดที่นำข้อมูลเข้า - สามารถทำการอัพเดตข้อมูลได้ - เข้าถึงข้อมูลครั้งละ 1 เรคอร์ด - ข้อมูลส่วนใหญ่ขึ้นกับงานของทรานแซคชันที่ทำ - ข้อมูลไม่มีความซ้ำซ้อน การเปลี่ยนแปลงข้อมูลแต่ละครั้งจะเขียนทับข้อมูลเดิม - โครงสร้างข้อมูลไม่มีการเปลี่ยนแปลง - ใช้งานกับฐานข้อมูลขนาดเล็กกว่า - สนับสนุนการปฏิบัติงานแบบวันต่อวัน - โปรแกรมที่ใช้งานขึ้นอยู่กับตัวแอพพลิเคชันที่ทำ - การใช้งานข้อมูลเกิดขึ้นตลอดเวลา	- เก็บเฉพาะข้อมูลสรุป - ไม่สามารถอัพเดตข้อมูลได้ ข้อมูลใหม่ที่เข้ามาจะเขียนต่อจากข้อมูลเดิม - เข้าถึงข้อมูลครั้งละกลุ่มข้อมูลหรือเป็นเซต - ข้อมูลจะถูกเก็บตามงานการวิเคราะห์ที่ต้องการใช้ - ข้อมูลมีความซ้ำซ้อนเพราะต้องเก็บข้อมูลเดียวกันในหลายช่วงเวลา - โครงสร้างข้อมูลมีความยืดหยุ่นตามการใช้งาน - ใช้งานกับฐานข้อมูลขนาดใหญ่ - จัดการข้อมูลได้ตามความต้องการใช้งาน - โปรแกรมที่ใช้งานขึ้นอยู่กับการวิเคราะห์ที่ต้องการ - การใช้งานข้อมูลเกิดขึ้นเป็นช่วงเวลา

ตารางที่ 2-1 แสดงความแตกต่างของข้อมูลในโอเปอเรชันกับข้อมูลในดาต้าแวร์เฮาส์

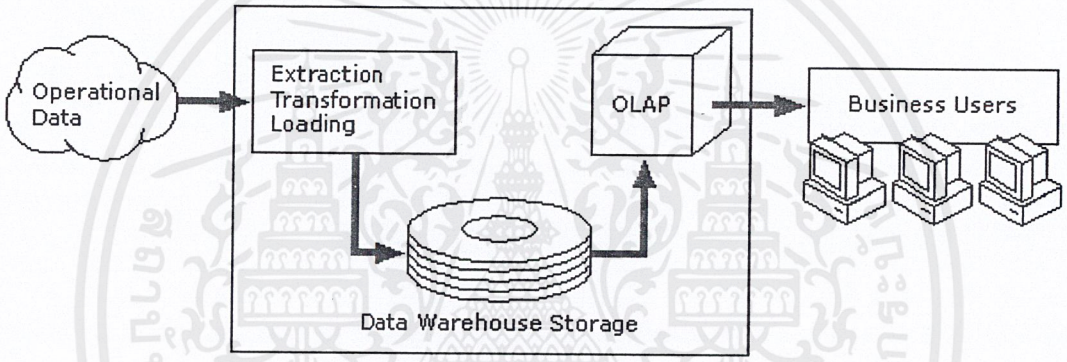
2.7 ดาต้าแวร์เฮาส์ และ OLAP

ดาต้าแวร์เฮาส์ คือ ระบบฐานข้อมูลที่บรรจุข้อมูลที่แสดงถึงประวัติของธุรกิจขององค์กรนั้นๆ ข้อมูลประวัติข้อมูลถูกใช้สำหรับวิเคราะห์และสนับสนุนการตัดสินใจใจหลายๆระดับเพื่อวางแผนและพัฒนาประสิทธิภาพขององค์กร ข้อมูลในดาต้าแวร์เฮาส์ถูกจัดการในการวิเคราะห์มากกว่าการทำแบบระบบ OnLine Transaction Processing (OLTP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยปกติแล้วดาต้าแวร์เฮาส์จะถูกใช้เป็นพื้นฐานสำหรับระบบสนับสนุนการตัดสินใจ (Decision Support System) เนื่องจากดาต้าแวร์เฮาส์ได้ถูกออกแบบมาเพื่อแก้ปัญหาที่เกิดขึ้นเมื่อเราใช้ฐานข้อมูลของงานโอเปอเรชันมาทำการวิเคราะห์

เทคโนโลยี OLAP (OnLine Analytical Processing) ทำให้เราสามารถนำข้อมูลไปใช้ได้อย่างมีประสิทธิภาพมากยิ่งขึ้นเหมาะสำหรับการวิเคราะห์แบบออนไลน์ (online analysis) ทำให้สามารถตอบคำถามที่มีความซับซ้อนได้อย่างรวดเร็ว เทคนิคมัลติไดเมนชันแนลโมเดล (Multidimensional data model) ของ OLAP และเทคนิคการรวมค่าข้อมูล (data aggregation) จะเข้ามาช่วยจัดการและสรุปค่าของข้อมูลที่มีปริมาณมากให้มีประสิทธิภาพยิ่งขึ้น ทำให้เราสามารถประเมินได้เร็วขึ้นโดยใช้การวิเคราะห์ออนไลน์และเครื่องมือที่เป็นกราฟฟิคที่สนับสนุนระบบ OLAP ทำให้เกิดความเร็วและความยืดหยุ่นที่สนับสนุนการวิเคราะห์แบบเรียลไทม์ (Real Time)



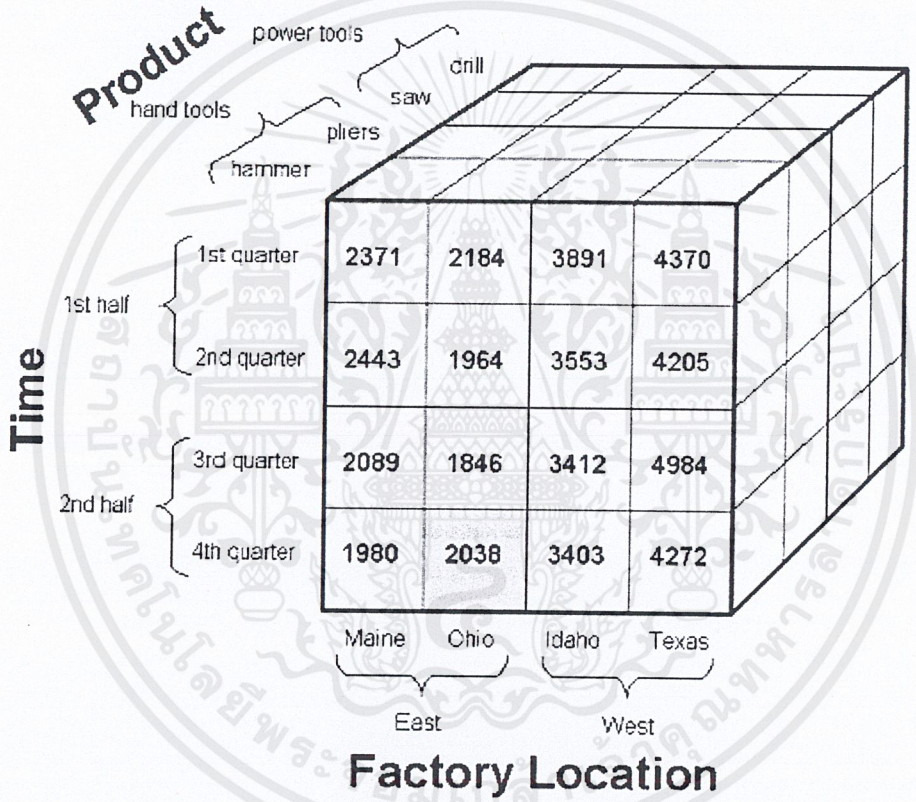
รูปที่ 2-5 แสดงความสัมพันธ์ระหว่างดาต้าแวร์เฮาส์และ OLAP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
โครงสร้างและการออกแบบของดาต้าแวร์เฮาส์

3.1 Dimensional Modeling

ในการออกแบบดาต้าแวร์เฮาส์นั้น เราจะใช้โมเดลการมองหลายมิติ (Dimensional Model) เพื่อเอาไปออกแบบดาต้าแวร์เฮาส์ โดยจะใช้โมเดลที่ทำให้เห็นง่ายๆจึงเลือกใช้โมเดล 3 มิติ (dimensions) เราจะมองภาพโมเดลนี้ได้เป็นลูกบาศก์หรือคิวบ์ (Cube) ซึ่งจะมองเป็นกี่มิติก็ได้



รูปที่ 3-1 แสดงตัวอย่าง Dimensional Modeling

เราสามารถใช้เทคนิคที่เรียกว่า Slice และ Dice ในการตัดข้อมูลมาวิเคราะห์ได้ เช่น ถ้าต้องการจะรู้ว่า hammer ที่ขายที่ Ohio ในช่วงเวลา 4th quarter ขายได้ทั้งหมดเป็นเงินเท่าไรก็สามารถ Slice และ Dice เข้าไปดูข้อมูลได้ ข้อมูลที่จะได้ คือ 2038 ดอลลาร์ เป็นต้น

โดยที่ Dimensional Modeling จะประกอบด้วย 3 ส่วน คือ ตารางแฟ็ค (Fact table) ,ไดเมนชัน (Dimension) และก็ตัววัด (Measure)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **แฟ็ค (Fact)**

แต่ละดาต้าแวร์เฮาส์จะมีตารางเดียวหรือหลายตารางของตารางแฟ็คก็ได้ แต่ละตารางแฟ็คจะมีตัววัดซึ่งไว้วัดค่าในการดำเนินการที่ธุรกิจนั้นๆต้องการ ตารางแฟ็คจะเก็บเรคอร์ดที่มีขนาดใหญ่บางครั้งเป็น 100 ล้านเรคอร์ด

ลักษณะของตารางแฟ็คนั้นจะบรรจุข้อมูลที่เป็นตัวเลขซึ่งสามารถนำมารวมกันได้ แต่ละตารางแฟ็คจะมีหลายอินเด็กซ์ที่จะเป็น foreign key ซึ่งเป็น primary key ของตารางไดเมนชันที่เชื่อมอยู่กับตารางแฟ็ค

Column	Description
Product_id	Foreign key for dimension table **product**.
Time_id	Foreign key for dimension table **time_by_day**.
Customer_id	Foreign key for dimension table **customer**.
Promotion_id	Foreign key for dimension table **promotion**.
Store_id	Foreign key for dimension table **store**.
Store_sales	Currency column containing the value of the sale.
Store_cost	Currency column containing the cost to the store of the sale.
Unit_sales	Numeric column containing the quantity sold.

ตารางที่ 3-1 แสดงตัวอย่าง ตารางแฟ็ค (fact table) เป็นตัวอย่างของตารางแฟ็คของ Food Mart 2000

ตารางแฟ็คจะทำหน้าที่เก็บข้อมูลที่มีความสัมพันธ์กัน จะประกอบไปด้วยตัววัดและตัวข้อมูลซึ่งแสดงถึงข้อมูลทางธุรกิจและการดำเนินการทางธุรกิจหรือเหตุการณ์ที่สามารถใช้ในการวิเคราะห์หรือใช้ประมวลผลทางธุรกิจ ใน OLAP Services ส่วนของแฟ็คจะอยู่ในรูปของตารางหลักของฐานข้อมูล OLAP ซึ่งจะเก็บข้อมูลที่เป็นตัวเลข ตารางเหล่านี้จะหมายถึงตารางแฟ็คและมักจะได้รับมาจากข้อมูลดิบในฐานข้อมูลเชิงปฏิบัติการ

ข้อมูลในตารางแฟ็ค ควรจะมีลักษณะ คือ เป็นตัวเลข (Numeric), มีค่าต่อเนื่อง (Continuously valued) และมาจากทุก ๆ ตาราง ไดเมนชัน (Additive)

- **Numeric** : เหตุผลที่ต้องเป็นตัวเลข ก็คือ ในความเป็นจริงคำถามทุกคำถามที่ทำกับตารางแฟ็คนั้น ระบบจัดการฐานข้อมูล (DBMS) จะใช้เรคอร์ดเป็นพัน ๆ หมื่นๆ หรือเป็นล้านๆ เรคอร์ด เพื่อสร้างเซตคำตอบขึ้นมา จำนวนเรคอร์ดที่มากมายนี้จะถูกบีบอัด เป็นจำนวนแถวเพียงไม่กี่แถว ในเซตคำตอบของผู้ใช้นั่นก็คือ การทำคำสั่ง Select list หรือ Built-in function : SUM ของ SQL นั่นเอง วิธีเดียวที่จะบีบอัดเรคอร์ดเหล่านี้ให้กลายเป็นเซตคำตอบได้ ก็คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

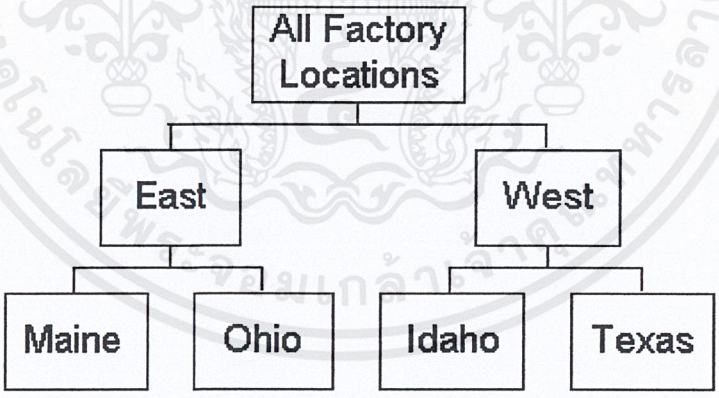
กลายเป็นเซตคำตอบได้ ก็คือ การบวกค่าของเรคอร์ดต่าง ๆ เข้าด้วยกัน ดังนั้นเรคอร์ดที่มีลักษณะเป็นตัวเลขและ Additive จะทำให้สร้างเซตคำตอบได้ง่าย

- **Continuously valued** : ค่าในตารางแฟ็คควรจะเป็นค่าที่มีความต่อเนื่อง เพื่อที่จะสามารถนำทางให้กับผู้ออกแบบฐานข้อมูล ให้สามารถแยกได้ว่าอะไรคือแฟ็ค อะไรคือแอททริบิวต์ (Attribute) ของตารางไดเมนชัน
- **Additive** : ค่าของตัววัดในตารางแฟ็ค จะเป็นค่า Additive แต่ก็จะมีค่าแฟ็คที่ไม่เป็น Additive ด้วย เป็นแฟ็คที่เป็น Semiadditive หรือ Nonadditive ซึ่ง Semiadditive คือ เกรน (grain) ที่ไม่ได้มาจากทุกตารางไดเมนชัน ซึ่งจะสามารถหาผลรวมตามไดเมนชันบางไดเมนชันเท่านั้น ส่วน Nonadditive คือ เกรนที่ไม่สามารถหาผลรวมตามไดเมนชันใด ๆ ได้

- **ไดเมนชัน (Dimension)**

ไดเมนชันคือ โครงสร้างของแอททริบิวท์ของลูกบาศก์ ไดเมนชันจะมีข้อมูลของลำดับชั้นที่ถูกจัดเก็บเพื่อเอาไว้อธิบายข้อมูลที่อยู่ในตารางแฟ็ค

เก็บคำอธิบายของแต่ละไดเมนชันของธุรกิจเอาไว้ ซึ่งคำอธิบายเหล่านี้จะช่วยในการอธิบายถึงสมาชิกในทุก ๆ ไดเมนชันและในตารางไดเมนชันจะประกอบด้วยหลาย ๆ แอททริบิวท์ ซึ่งแอททริบิวท์ที่ดีจะต้องเป็นตัวอักษร และแต่ละแอททริบิวท์ต้องแยกออกจากกัน แอททริบิวท์หล่านี้ถูกใช้เป็นแหล่งที่มาของข้อบังคับ (Constrains) และ row header ในเซตคำตอบของผู้ใช้



รูปที่ 3-2 ลำดับขั้นในการออกแบบไดเมนชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแสดงลำดับขั้นของไดเมนชัน Location โดยไดเมนชันนี้ถูกกำหนดมาจากการเลือก Region และ State จากตารางข้างล่าง

State_ID	Region	State
1	East	Maine
2	East	Ohio
3	West	Idaho
4	West	Texas

รูปที่ 3-3 แสดงตารางของไดเมนชัน

จากตารางนี้ State_ID จะถูกนำไปเป็น Foreign Key ของตารางแฟ็คที่เกี่ยวข้องต่อไป

- **ตัววัดค่า (measure)**

เป็นแอททริบิวท์ที่เป็นตัวเลขของแฟ็คแสดงประสิทธิภาพหรือพฤติกรรมของธุรกิจที่เกี่ยวข้องกับไดเมนชัน ซึ่งสมาชิกที่แท้จริง จะเรียกว่า ตัวแปร (Variables)

3.2 การออกแบบข้อมูล (Data Modeling)

การออกแบบข้อมูล เป็นขั้นตอนที่สำคัญในการสร้างดาต้าแวร์เฮาส์ เพราะการออกแบบจะทำให้สามารถมองระบบที่กำลังจะสร้างได้ชัดเจนยิ่งขึ้น

3.2.1 Data Cubes

ในดาต้าแวร์เฮาส์นั้น ข้อมูลจะถูกสร้างให้อยู่ในรูปแบบที่ง่ายต่อการมองและวิเคราะห์ โดยจะสร้างเป็นแบบลูกบาศก์หรือคิวบ์โดยส่วนใหญ่แล้วมักจะเป็น 3 มิติ หรือมากกว่า 3 มิติ (มากกว่า 3 มิติ จะเรียกว่า Hypercube)

3.2.2 โอเปอเรชัน (Operation) ที่ใช้ในการวิเคราะห์ข้อมูล

ผู้ออกแบบข้อมูลสามารถใช้ Dimensional Modeling ที่รองรับ OLAP และรองรับการตัดสินใจ โดยการวิเคราะห์ข้อมูลใน OLAP Services จะมีอยู่ 4 โอเปอเรชันด้วยกัน คือ Drill-down, Drill-up, Slicing และ Dicing

- Drill-down และ Drill-up

Drill-down และ Drill-up เป็นโอเปอเรชันที่ใช้สำหรับการมองข้อมูลที่เป็นระดับๆ ในไดเมนชัน โดยการ Drill-down จะทำให้เห็นรายละเอียดของข้อมูลในระดับที่ลึกลงไป ในขณะที่การ Drill-up นั้นจะเป็นการดูรายละเอียดของข้อมูลในระดับที่สูงขึ้น ดังแสดงในตารางที่ 3-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Quarter	Ford	Mercury
A B C-Autos	Q1 – 1998	120	89
	Q2 – 1998	123	91
	Q3 – 1998	101	75
	Q4 – 1998	99	73
Drill-Up ↑ Drill-Down ↓			
	Month	Ford	Mercury
A B C-Autos	Oct – 1998	40	31
	Nov – 1998	35	29
	Dec – 1998	24	13

ตารางที่ 3-2 แสดงการ Drill-down และ Drill-up

จากตารางที่ 3-2 เป็นข้อมูลแสดงการขายรถยนต์ของตัวแทนจำหน่าย ซึ่งจะแบ่งเป็น 4 ควอเตอร์ (Quarter) ด้วยกัน เมื่อทำการ Drill-down ในควอเตอร์ที่ 4 (Q4 – 1998) แล้วจะเห็นว่าจะเป็นการแสดงข้อมูลในระดับของเดือนใน Q4 (เดือนตุลาคม ถึง เดือนธันวาคม) ส่วนการ Drill-up จะแสดงผลตรงกันข้ามกับการ Drill-down นั่นเอง เช่น ถ้า Drill-up ข้อมูลในระดับเดือน ก็จะเห็นข้อมูลแสดงในระดับของควอเตอร์

- Slicing และ Dicing

การตัดข้อมูล (Slicing) เวลาที่ทำการวิเคราะห์ข้อมูลสามารถที่จะวิเคราะห์ข้อมูลได้เพียงบางช่วงหรือเฉพาะส่วนที่สนใจเท่านั้นส่วนการหมุนข้อมูล (Dicing) ก็สามารถหมุนหรือสับเปลี่ยนตำแหน่งของไดเมนชันได้ ดังตารางที่ 3-3, 3-4 และ 3-5

1998 Sales		Ford		Mercury	
	Quarter	Escort	Taurus	Tracer	Sable
Midwest	Detroit	21	39	13	33
	Cleveland	19	40	9	35
Northeast	Philadelphia	8	26	3	19
	New Jersey	9	19	5	12

ตารางที่ 3-3 แสดงการตัดข้อมูล (Slicing)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1998 Sales		1998 Ford and Mercury			
	Quarter	Q1	Q2	Q3	Q4
Midwest	Detroit	119	134	128	110
	Cleveland	104	124	120	101
Northeast	Philadelphia	99	110	109	89
	New Jersey	65	79	78	69

ตารางที่ 3-4 แสดงการหมุนข้อมูล (Dicing)

1998 Sales		1998 Ford Only			
	Quarter	Q1	Q2	Q3	Q4
Midwest	Detroit	73	80	73	70
	Cleveland	65	79	79	59
Northeast	Philadelphia	33	59	49	38
	New Jersey	36	45	49	38

ตารางที่ 3-5 แสดงการตัดข้อมูลจากข้อมูลในตารางที่ 3-4

จากตารางข้างต้น ประกอบไปด้วย 3 ไดเมนชัน คือ Geographic location, Time และ Product (รถยนต์) ในตารางที่ 3-3 เป็นการตัดข้อมูลการวิเคราะห์เฉพาะรถยนต์ยี่ห้อ Ford และ Mercury ใน 4 เมืองทางเขต Midwest และ Northeast ในปี 1998 ในตารางที่ 3-4 แสดงการหมุนข้อมูล โดย Time จะกลายเป็นไดเมนชันหลัก ส่วนในตารางที่ 3-5 เป็นการตัดข้อมูลจากตารางที่ 3-4 ที่เลือกเฉพาะรถยนต์ยี่ห้อ Ford ขึ้นมาวิเคราะห์เท่านั้น

3.3 โครงสร้างฐานข้อมูล (Database Schema)

โครงสร้างฐานข้อมูลหลัก ๆ จะมีอยู่ 2 รูปแบบ คือ สกีมาแบบดาว (Star Schema) และสกีมาแบบสโนว์เฟลก (Snowflake schema)

3.3.1 สกีมาแบบดาว

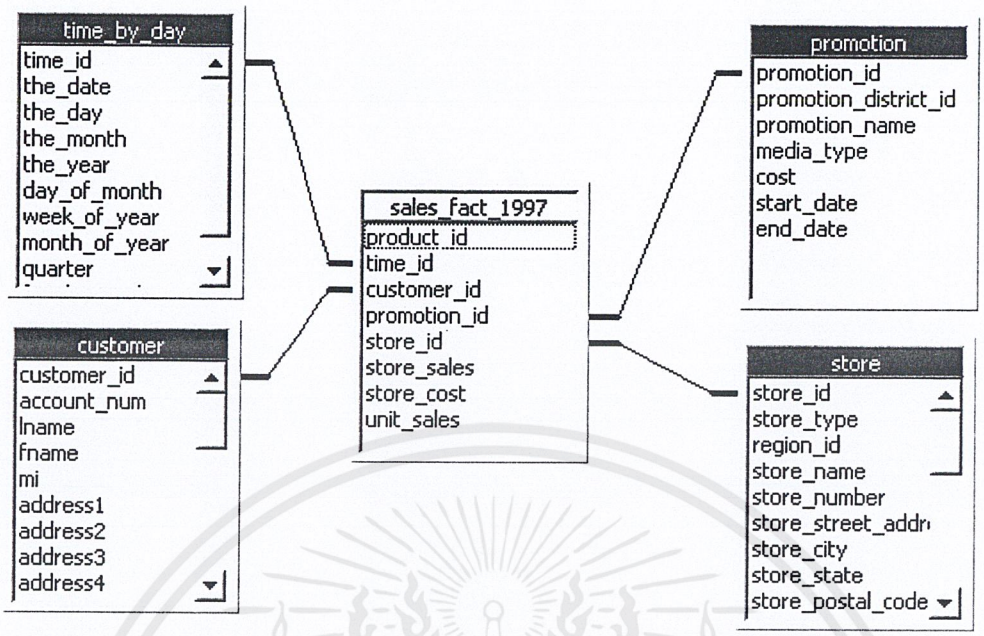
สกีมาแบบดาวเป็นวิธีหนึ่งของไดเมนชันแนลโมเดล ซึ่งเป็นชื่อที่ใช้กันมานานแล้ว เนื่องจากมีรูปร่างคล้ายกับดาว ซึ่งประกอบไปด้วยตารางใหญ่ 1 ตารางอยู่ตรงกลาง และมีตารางเล็ก ๆ ที่มีความสัมพันธ์กับตารางหลักอยู่รอบๆ ตารางหลักนี้จะเป็นตารางเดียวที่ใช้การเชื่อมต่อแบบหลายจุด (Multiple join) เพื่อเชื่อมต่อกับตารางอื่น ๆ แต่ตารางอื่น ๆ ที่อยู่รอบ ๆ จะมีการเชื่อมต่อเพียงแค่จุดเดียว (Single join) เพื่อเชื่อมเข้ากับตารางหลักเท่านั้น

ตารางหลัก จะเรียกว่า ตารางแฟ็ค (Fact Table)

ตารางอื่น ๆ จะเรียกว่า ตารางไดเมนชัน (Dimension Table)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



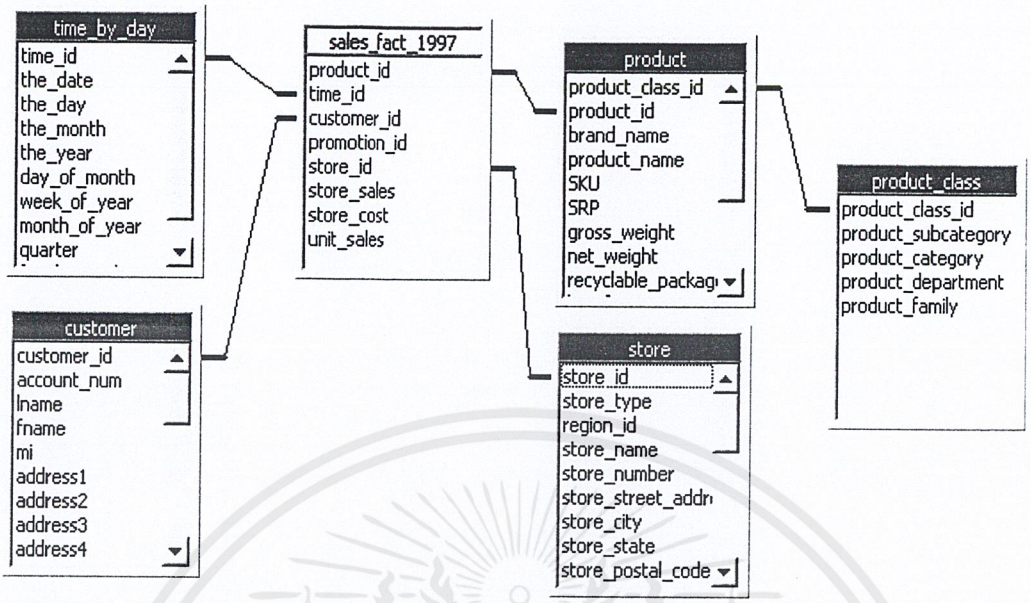
รูปที่ 3-4 แสดงโครงสร้างฐานข้อมูลที่เป็นสกีมาแบบดาว (Star Schema)

3.3.2 สกีมาแบบสโนว์เฟลก (Snowfake Schema)

ตารางไดเมนชันที่มีข้อมูลแบบเป็นลำดับขั้น (Hierarchy) อาจจะมีการแตกข้อมูลออกมาเป็นตารางย่อย ๆ เพื่อให้ผู้ออกแบบสามารถเข้าใจได้ง่ายยิ่งขึ้น ซึ่งเรียกโครงสร้างนี้ว่า สโนว์เฟลก (Snowflake) ความสัมพันธ์ที่เป็นแบบ many – to – one แต่ละอันจะถูกแยกออกเป็นตารางย่อย ๆ โดย Primary key ของตารางย่อย จะต้องมี Foreign key ของตารางที่ตัวเองไปเกาะอยู่ ดังแสดงในรูปที่ 3-5

การทำโครงสร้างแบบสโนว์เฟลก อาจทำให้ประสิทธิภาพในการขับเคลื่อนลดต่ำลง เนื่องจากต้องมีการรวมกันระหว่างตารางย่อยกับตารางไดเมนชันอีกทีหนึ่ง รวมทั้งต้องสร้าง SQL ที่มากขึ้น แต่อย่างไรก็ตาม สโนว์เฟลกก็เหมาะสำหรับฐานข้อมูลที่มีขนาดใหญ่ เช่น มีจำนวนหลายแสนเรคอร์ด เพราะมีส่วนช่วยลดความซ้ำซ้อนลงได้ รวมทั้งโครงสร้างแบบสโนว์เฟลกมีข้อดี เช่น เมื่อต้องการเรียกดูเฉพาะฟิลด์ (field) ในตารางไดเมนชัน ที่เป็นสโนว์เฟลก ก็ทำเพียงแค่ดึงข้อมูลจากตารางไดเมนชันนั้นเพียงตารางเดียวและทำการเชื่อมต่อกับตารางแฟ็คเท่านั้น ไม่ต้องทำการเชื่อมต่อระหว่างตารางไดเมนชันนั้นกับตารางย่อยของตัวมันเองอีกและจะไม่ทำให้เกิดปัญหาการอัพเดทโดยไม่ต้องทำการอัพเดทหลายที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
46155



รูปที่ 3-5 แสดงโครงสร้างฐานข้อมูลที่เป็นสกีมาแบบสโนว์เฟลก (Snowflake Schema)

3.4 การรวมค่าข้อมูล (Aggregation)

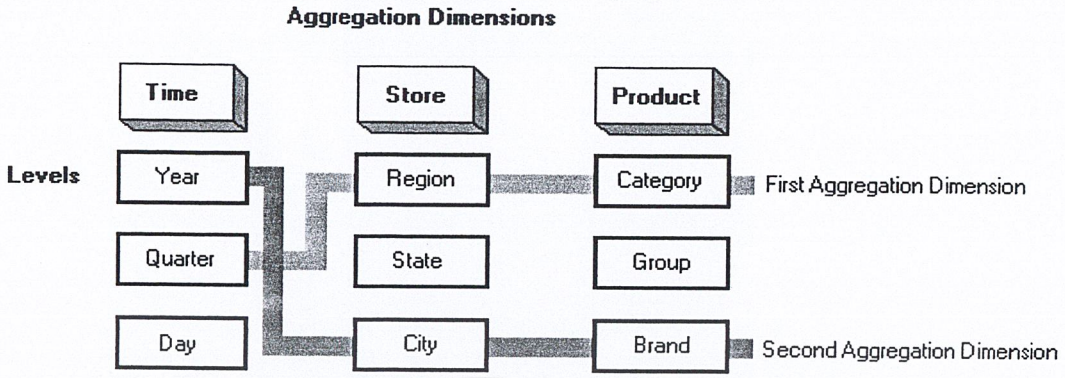
เนื่องจากข้อมูลพื้นฐานของไดเมนชันแนลดาต้าแวร์เฮาส์ (Dimensional Data Warehouse) จะประกอบด้วยเรคอร์ดจำนวนมาก ถ้านักวิเคราะห์ต้องการทำคิวรีโดยไม่มีการกำหนดข้อบังคับให้กับบางไดเมนชัน เนื่องจากในดาต้าแวร์เฮาส์จะมีเฉพาะแต่ข้อมูลพื้นฐานเท่านั้น (Based-level data) จึงทำให้เวลาคิวรีจะต้องทำการรวมข้อมูลภายในเรคอร์ดจำนวนมากมาย ซึ่งถ้าเป็นเช่นนี้จะทำให้การทำคิวรีเกิดความสิ้นเปลืองสูงมาก ดังนั้นจึงได้มีการทำการรวมค่าข้อมูลเอาไว้ล่วงหน้า เพื่อเร่งให้การทำคิวรีสามารถทำได้เร็วขึ้น มีประสิทธิภาพที่ดียิ่งขึ้น

การรวมค่าต้องมี fact table record ที่แสดงถึงข้อสรุปที่ได้จากตารางแฟ็ค ระดับพื้นฐาน (Based-level fact table) เป็นข้อมูลตัววัดมีขนาดเล็กที่สุด

การรวมค่ามีหลายชนิด และแต่ละชนิดจะมีตารางแฟ็คเป็นของตัวเอง ซึ่งตารางแฟ็คเหล่านี้จะถูกเรียกว่าเป็น Derivative fact table เพราะข้อมูลเหล่านั้นได้รับมาจากตารางแฟ็คระดับพื้นฐาน

เราสามารถสร้างการรวมค่าได้มากมายตามที่เราต้องการ ซึ่งในความเป็นจริงเราจะสร้างการรวมค่าไว้เฉพาะส่วนที่เราต้องการเท่านั้น ไม่จำเป็นต้องสร้างการรวบรวมทุก ๆ ฟิลด์ในแต่ละไดเมนชันและไม่จำเป็นต้องทำการรวมค่าในทุก ๆ ไดเมนชันด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-6 แสดงการรวมค่า (Aggregation)

3.5 Crosstabulation หรือ Crosstab

CrossTabs เป็นวิธีในการแสดงกลุ่มของข้อมูล เพื่อให้มีความสัมพันธ์กันและสามารถมองเห็นแนวโน้มได้ง่ายซึ่งฟิลด์ของตารางไดเมนชันจะอยู่ในรูปของ CrossTabs และค่าในแต่ละฟิลด์จะถูกแบ่งกลุ่มและถูกสรุปภายในไดแมนชัน ทำให้สามารถมองได้ง่ายกว่าแสดงแบบตารางปกติ

	Store Cost	Store Sales	Store Profit
Breakfast Foods	2,756.80	฿6,941.46	151.79%
Carousel	595.97	฿1,500.11	151.71%
Canned Products	1,317.13	฿3,314.52	151.65%
Baking Goods	15,370.61	฿38,670.41	151.59%
Alcoholic Beverages	5,576.79	฿14,029.08	151.56%
Health and Hygiene	12,972.99	฿32,571.86	151.07%
Snack Foods	26,963.34	฿67,609.82	150.75%
Baked Goods	6,564.09	฿16,455.43	150.69%
Beverages	11,069.53	฿27,748.53	150.67%
Frozen Foods	22,030.66	฿55,207.50	150.59%

รูปที่ 3-7 แสดงข้อมูลในตารางแบบปกติ

		MeasuresLevel	
- Country	+ State Province	Unit Sales	Store Cost
All Customers	All Customers Total	266,773.00	225,627.23
- Canada	Canada Total		
	+ BC		
+ Mexico	Mexico Total		
+ USA	USA Total	266,773.00	225,627.23

รูปที่ 3-8 แสดงข้อมูลในตารางแบบ CrossTabs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ขั้นตอนการออกแบบดาต้าแวร์เฮาส์

ในการออกแบบดาต้าแวร์เฮาส์ จะประกอบไปด้วย 4 ขั้นตอน คือ

3.6.1 เลือกกระบวนการทางธุรกิจ (Business Process) ที่ต้องการสร้าง

กระบวนการทางธุรกิจ เป็นกระบวนการหลักที่ต้องการทำในองค์กร ซึ่งกระบวนการนั้นมีระบบเดิมสนับสนุนอยู่ โดยข้อมูลในระบบนั้นสามารถนำมารวบรวม เพื่อทำเป็นดาต้าแวร์เฮาส์ได้ เช่น ใบสั่งของ (Order), ใบส่งของ (Invoice), สินค้าที่ส่งไป (Shipments),รายการสินค้า (Inventory) ฯลฯ

3.6.2 เลือกเกรน (Grain) ของกระบวนการทางธุรกิจ

เกรน : เป็นข้อมูลพื้นฐาน ข้อมูลที่เป็นอะตอมมิก (Atomic) ซึ่งถูกแสดงอยู่ในตารางแฟ็คสำหรับกระบวนการนี้ เกรนที่มีอยู่ทั่วไป เช่น ข้อมูลการทำทรานแซคชันในแต่ละครั้ง, ข้อมูลของการทำงานในแต่ละวันหรือสรุปในแต่ละวัน, ข้อมูลสรุปการทำงานในแต่ละเดือน

3.6.3 เลือกไดเมนชันที่จะถูกนำมาใช้กับแต่ละเรคอร์ดของตารางแฟ็ค

ไดเมนชันที่มีอยู่โดยทั่วไป เช่น Product, Customer, Time ซึ่งแต่ละไดเมนชันจะถูกอธิบายแยกกันในลักษณะของแอททริบิวท์และจะถูกอธิบายเป็นตัวหนังสือ

3.6.4 เลือกแฟ็คที่มีการวัดหรือมีการประมวลผลหรือมีการคำนวณไว้แล้ว (measured fact) ที่จะเก็บอยู่ในแต่ละเรคอร์ดของตารางแฟ็ค

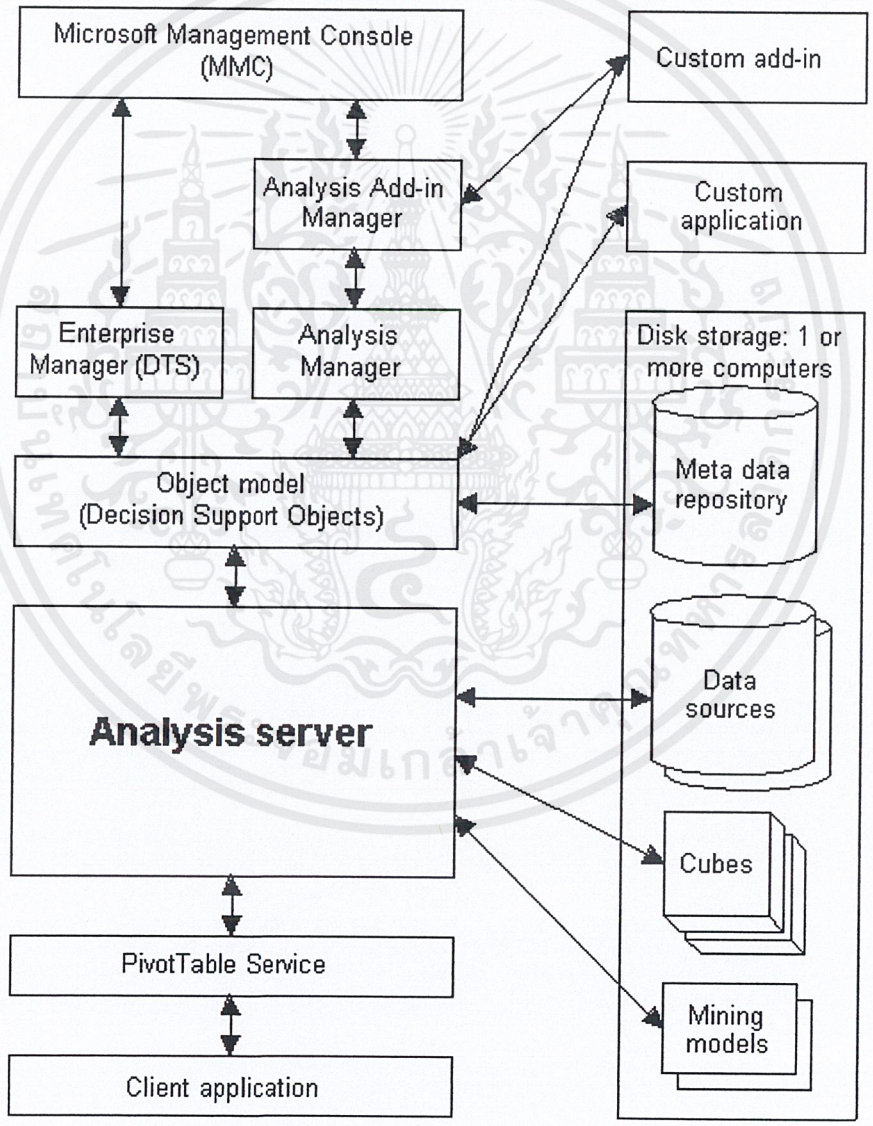
ตัววัดที่มีอยู่โดยทั่วไป เช่น ปริมาณต่าง ๆ ซึ่งจะมีลักษณะเป็นตัวเลข ได้แก่ ยอดขาย, จำนวนสินค้าที่ ขายได้, ค่าใช้จ่ายทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การใช้ทูลส์สร้างระบบฐานข้อมูลดาต้าแวร์เฮาส์

4.1 Microsoft Analysis Service

ทูลส์ที่เราเลือกนำมาสร้างระบบดาต้าแวร์เฮาส์คือ Microsoft SQL Server 2000 ซึ่งในตัวโปรดักส์นี้จะมีส่วนของ Analysis Service ซึ่งเป็นส่วนสำคัญที่ใช้สำหรับสร้างระบบดาต้าแวร์เฮาส์ โดยใน Analysis Service จะประกอบไปด้วยคอมโพเนนท์มากมายและมีสถาปัตยกรรมดังรูปที่ 4-1



รูปที่ 4-1 แสดงสถาปัตยกรรมภายในของ Analysis Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การสร้างมัลติไดเมนชันแนลโมเดล (Multidimensional Modal) บน Analysis Service

เราสามารถสร้างระบบดาต้าแวร์เฮาส์ได้โดยใช้โมเดลของมัลติไดเมนชันแนลโมเดล โครงสร้างหลักของข้อมูลที่เก็บในลักษณะมัลติไดเมนชัน คือ คิวบ์ การสร้างคิวบ์ใน Analysis Service จะมี วิซาร์ด (Wizard) มาช่วยทำให้เพิ่มความสะดวกสบายให้ กับผู้ออกแบบมายิ่งขึ้น สามารถทำได้ตามขั้นตอนดังนี้

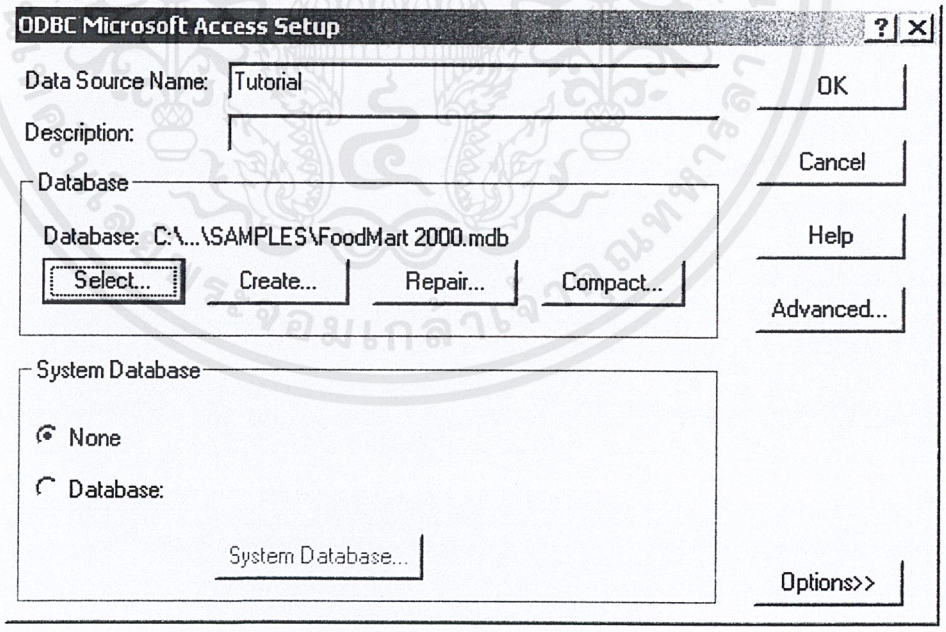
4.2.1 การติดต่อแหล่งข้อมูล (Data Source)

ก่อนเริ่มทำงานกับ Analysis Service อย่างแรกต้องตั้งค่าการเชื่อมต่อกับแหล่งข้อมูลใน ODBC Data Source Administrator

4.2.1.1 วิธีการติดตั้ง Data Source Name (DSN)

1. สำหรับ Window NT :คลิกที่ปุ่ม **Start** ชี้ไปที่ **Setting**, คลิกที่ **Control Panel** และ ดับเบิลคลิกที่ Data Sources (ODBC)

   สำหรับ Window 2000: คลิกที่ **Start**, ชี้ไปที่ **Setting**, คลิกที่ **Control Panel**, ดับเบิลคลิกที่ **Administrative Tools** และ ดับเบิลคลิกที่ **Data Source (ODBC)**
2. ที่แท็ป **System DSN** คลิก **Add**
3. เลือก Driver ของ Data Source นั้นๆ ในที่นี้เลือกของ **Microsoft Access Driver (*.mdb)** และ ก็คลิก **Finish**
4. ในช่อง **Data Source Name** ใส่ชื่อ Data Source Name ที่ต้องการ ในที่นี้จะใส่ **Tutorial** และใต้ **Database** คลิก **Select**



รูปที่ 4-2 การติดตั้ง Data Source Name

5. ในช่อง **Database** ให้คลิกปุ่มที่ **Select** ซึ่งจะปรากฏไดอะล็อกบ๊อกซ์ขึ้นมาให้เลือกตำแหน่งที่ Database นั้นอยู่ ในที่นี้จะเลือก **C:\Program Files\Microsoft Analysis Services\Samples** และคลิก **FoodMart 2000.mdb** คลิก **Ok**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

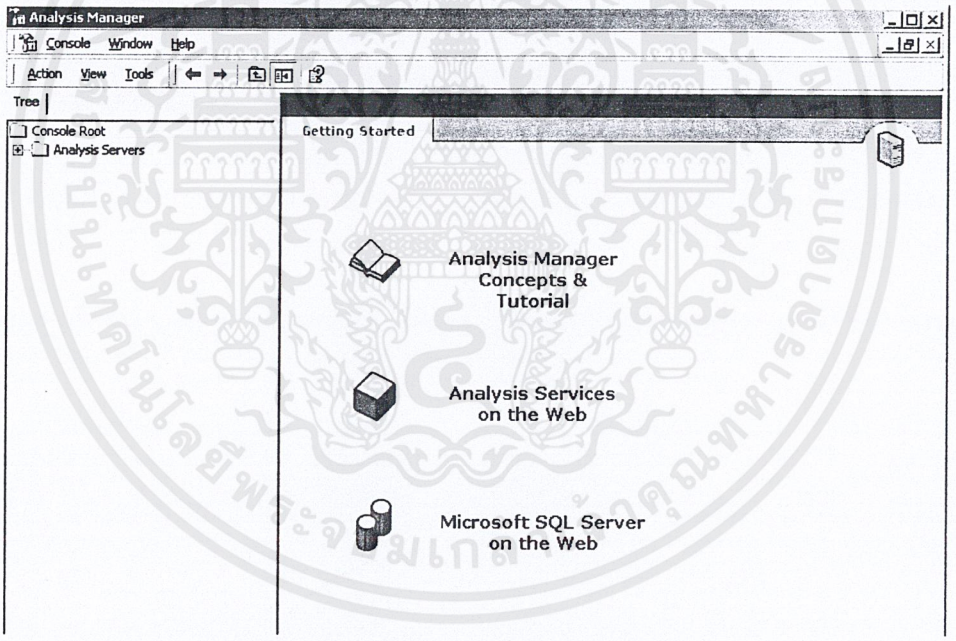
6. ใน **ODBC Microsoft Access Setup dialog box** คลิก **Ok**
7. ใน **ODBC Data Source Administrator dialog box** คลิก **Ok**

4.2.1.2 ติดตั้ง Database และ Datasource

ก่อนที่จะออกแบบคิวบ์ จะต้องติดตั้ง Database Structure ให้ติดต่อไปยัง Data source ที่ได้ติดตั้งไปแล้วก่อนหน้านี้ใน ODBC Data Source Administrator

วิธีการติดตั้ง Database Structure

1. เปิดโปรแกรม Analysis Manager ขึ้นมาและใน Analysis Manger Tree ให้ กดปุ่มขยาย **Analysis Servers**
2. คลิกที่ชื่อ Server ของที่เราต้องการจะทำงานด้วย
3. คลิกขวาและคลิก **New Database**
4. ใน **Database** Dialog Box ใน **Database name** box ให้ใส่ Database name ที่ต้องการ ในที่นี้จะใส่เป็น **Tutorial** และคลิก **Ok**
5. ใน Analysis Manager Tree pane, ให้ขยาย **server** และขยาย **Tutorial** database ที่เพิ่งสร้างไป



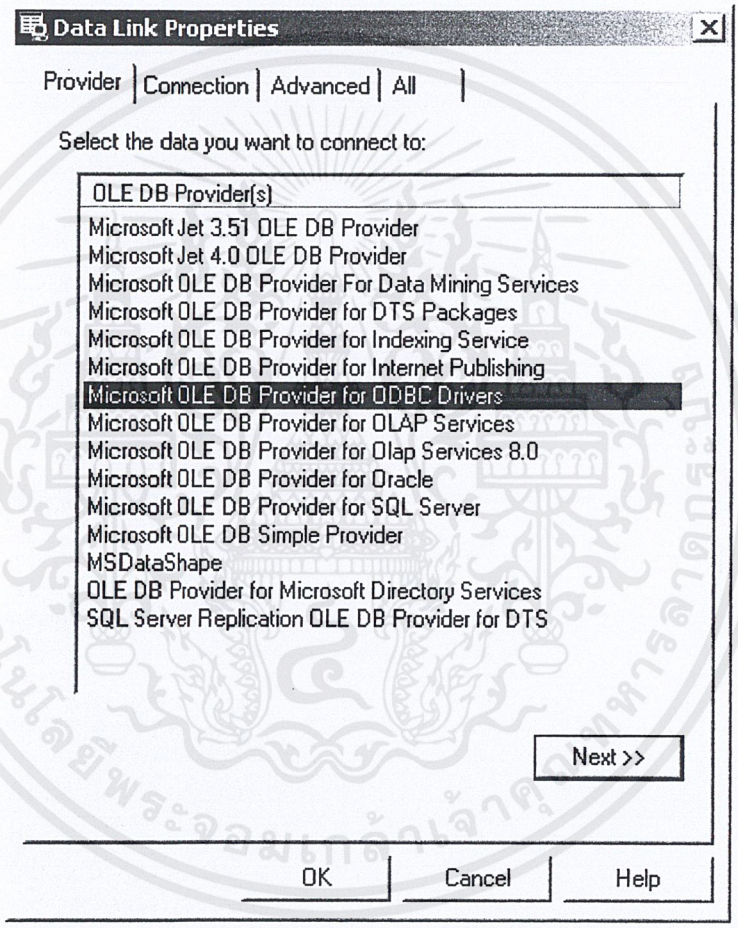
รูปที่ 4-3 รูปแสดงหน้าจอของโปรแกรม Analysis Manager

ต่อไปจะเป็นการสร้างการเชื่อมต่อไปยังข้อมูลตัวอย่าง ใน **Tutorial** data source การติดตั้ง data source ใน Analysis Manager ที่ทำไปนี้ ถูกเชื่อมต่อไปยังฐานข้อมูล ที่ system data source name (DSN) ที่ได้ทำการติดตั้งใน ODBC Data Source Administrator ไว้ล่วงหน้าก่อนแล้ว ข้อมูลทั้งหมดที่มาจากแหล่งข้อมูลนี้จะถูกนำมาสร้างคิวบ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.3 วิธีการติดตั้ง Data Source ให้กับ Analysis Manger

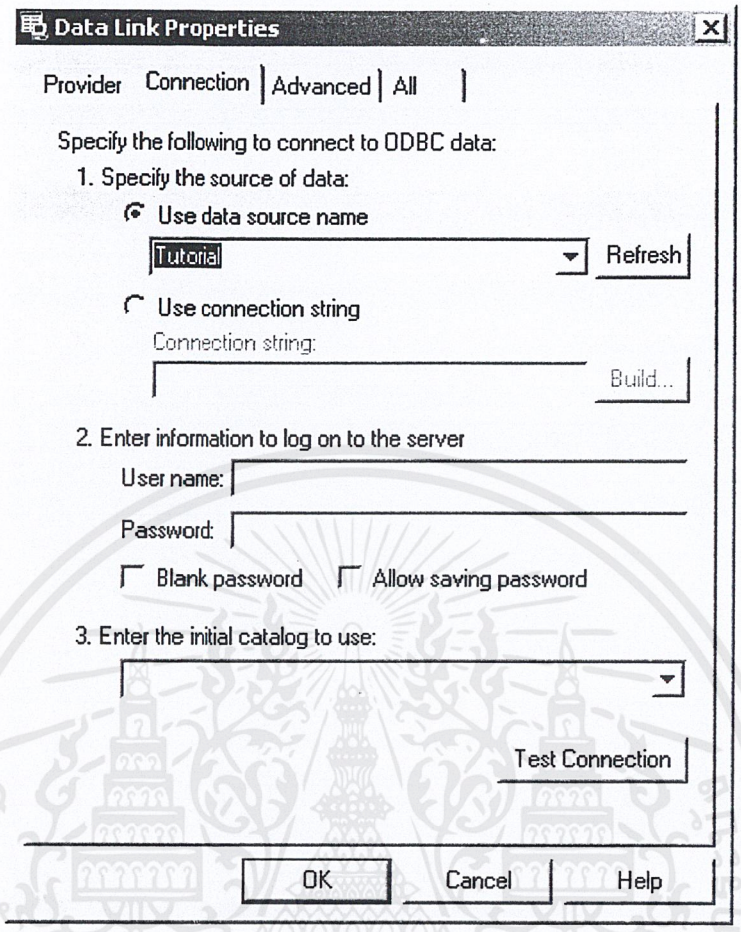
1. ใน **Analysis Manager** tree pane, คลิก **Data source** ใต้ **Tutorial database** และคลิกขวาเลือก **New Data Source**
2. ในหน้าต่าง **Data Link Properties** คลิกแท็ป **Provider** และคลิก **Microsoft OLE DB Provider for ODBC Drivers** ซึ่งถ้าเราใช้ฐานข้อมูลตัวอื่นที่ไม่ใช่ Microsoft Access ให้เลือกตัวอื่นตามแต่ละชนิดของฐานข้อมูล เช่น ถ้าฐานข้อมูลของเราเป็น Microsoft SQL Server ก็ให้เลือกที่ Microsoft OLE DB Provider for SQL Server



รูปที่ 4-4 การเลือก Driver เพื่อใช้ในการติดต่อกับแหล่งข้อมูล

3. คลิกแท็ป **Connection** และที่ **Use data source name** ให้คลิกเลือก **Tutorial**
4. คลิก **Test Connection** เพื่อตรวจสอบการเชื่อมต่อว่าสมบูรณ์หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-5 การเลือก Data Source Name

5. คลิก **OK** เพื่อปิดหน้า **Data Link Properties**

4.2.2 เริ่มสร้างคิวบ์ (Building Cube)

ในขั้นตอนนี้จะใช้ **Cube Wizard** เข้ามาช่วยสร้างคิวบ์ ซึ่งคิวบ์ที่จะสร้างนี้จะใช้เป็นดาต้าแวร์เฮาส์ของแผนกการตลาด เพื่อนำมาใช้ในการวิเคราะห์ค่ายอดขาย (Sales)

- เปิด **Cube Wizard** ใน Analysis Manager tree pane, ใต้ **Tutorial** database คลิกขวาที่โฟลเดอร์ Cube, คลิกที่ **New Cube** และก็คลิกที่ **Wizard**

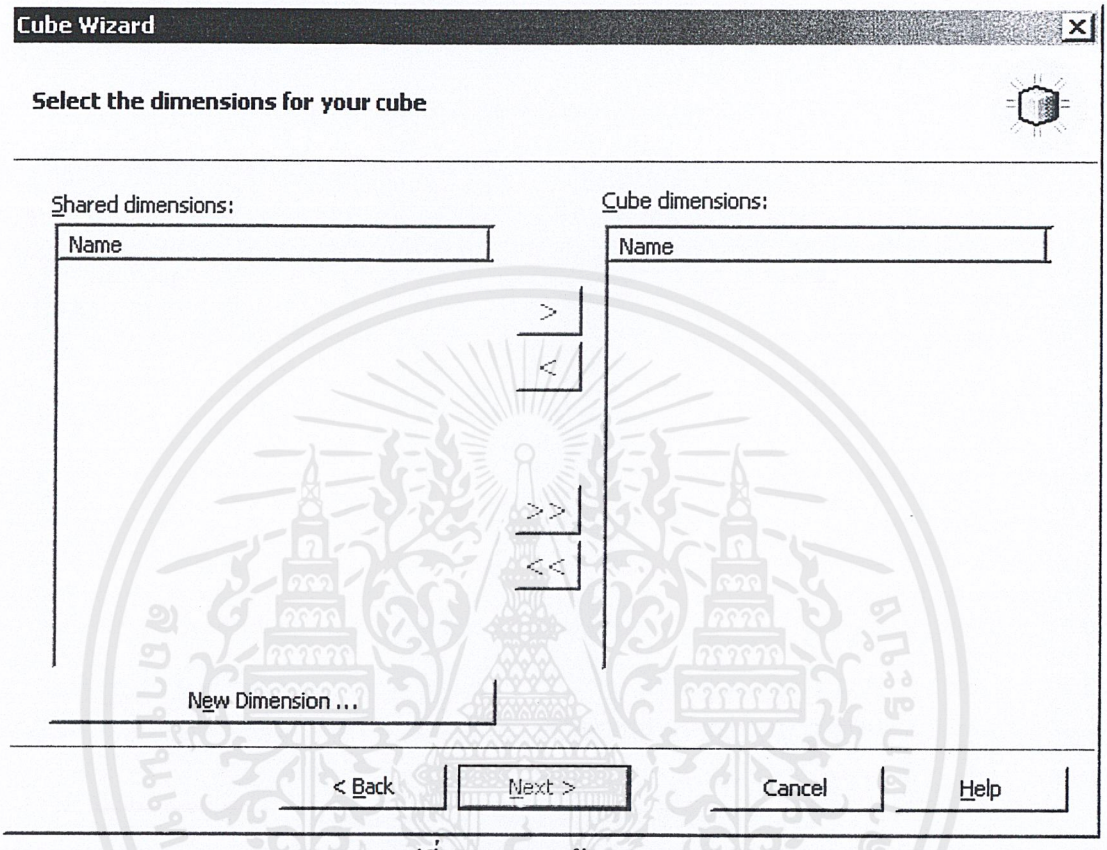
4.2.2.1 วิธีการสร้างตัววัด (Measures)

1. ในหน้าต่าง **Welcome** ของ Cube Wizard ให้คลิก **Next**
2. ในหน้าต่าง **Select a fact table from a data source** ให้ขยายที่ฐานข้อมูล **Tutorial**, และคลิกขวาที่ Cube folder คลิก **sale_fact_1998** (เลือกเป็น ตารางแฟ็ค)
3. สามารถ view ข้อมูลในตาราง **Sales_fact_1998** โดยการคลิก **Browse Data** หลังจากปิด Browse Data แล้วให้ คลิก **Next**
4. กำหนดตัววัดสำหรับคิวบ์ ใต้ **Fact Table numeric columns**, ดับเบิลคลิกเลือก **Store_sales, store_cost, unit sales** และ คลิก **Next.**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.2 วิธีสร้างไดเมนชัน Time

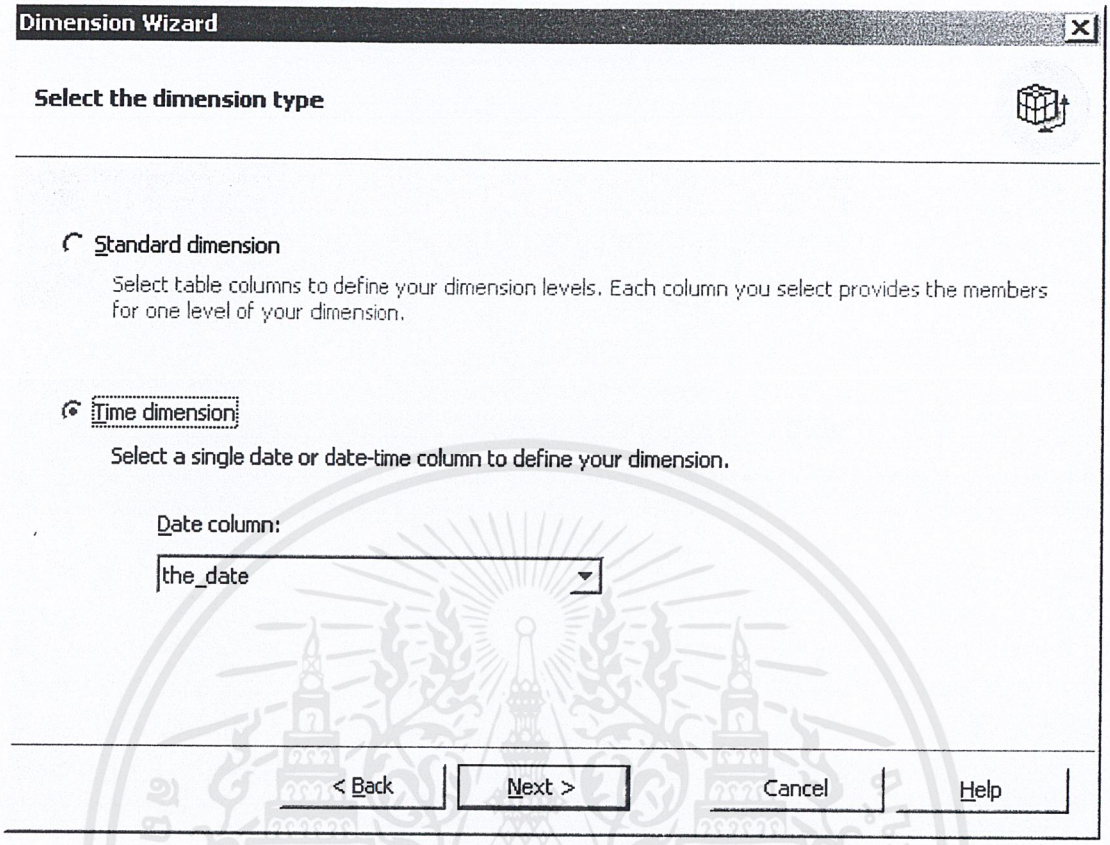
1. ในหน้าต่าง **Select the dimensions for your cube** ของ Cube Wizard , คลิก **New Dimension** ซึ่งมันจะไปเรียก **Dimension Wizard**



รูปที่ 4-6 แสดงหน้าจอ Cube Wizard

2. ที่ขั้นตอน Welcome คลิก **Next**
3. ที่ขั้นตอน **Choose how you want to create dimension** เลือก **Star Schema: A single dimension table,** และ คลิก **Next**
4. ที่ขั้นตอน **Select dimension table** เลือก **time_by_day** คลิก **Next**
5. ที่ขั้นตอน **Select the dimension type,** เลือก **Time dimension,** และ คลิก **Next**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



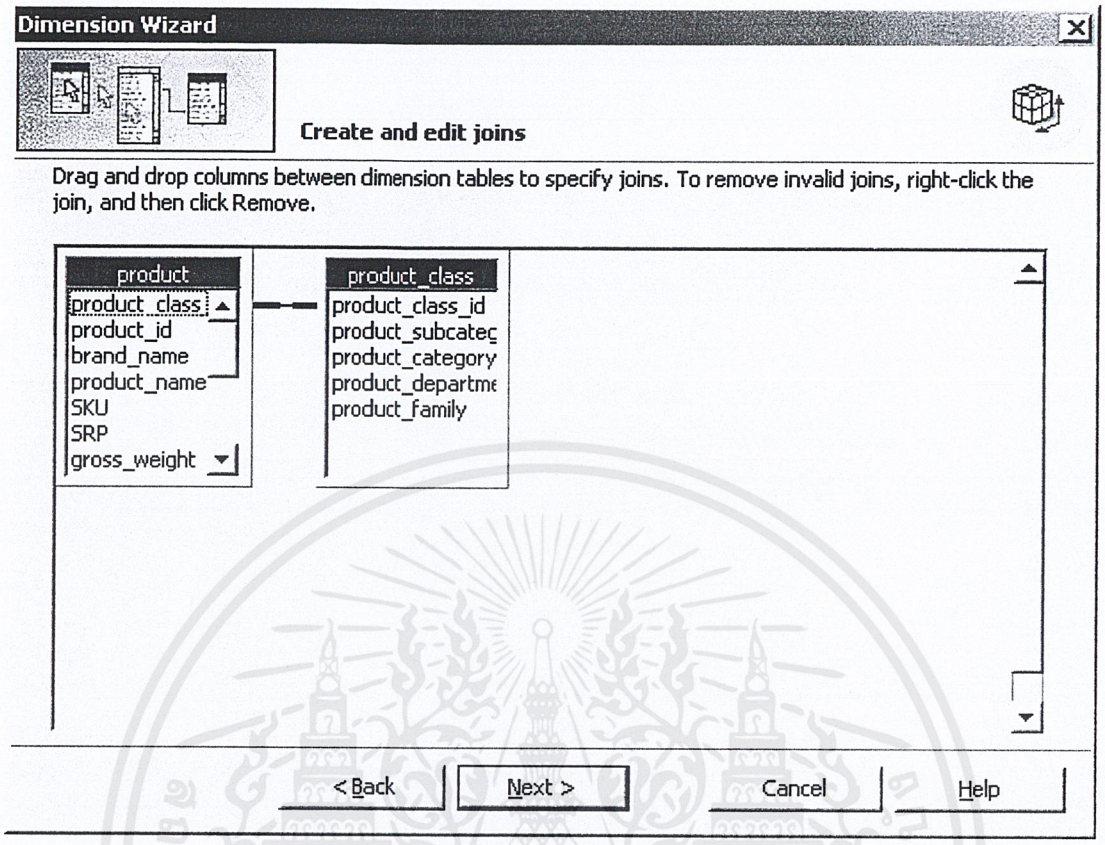
รูปที่ 4-7 แสดงการเลือกชนิดของไดเมนชัน

5. ถัดไป ทำการกำหนด Level สำหรับไดเมนชัน ที่ขั้นตอน **Create the time dimension levels** คลิก **select time levels**, เลือก **Year, Quarter, Month** และคลิก Next
6. ที่ขั้นตอน **Select advanced option** คลิก **Next**
7. ที่ Wizard สุดท้ายใส่ ช่องของ New dimension ให้ใส่ Time คลิก **finish** เพื่อกลับที่ **Cube Wizard**

4.2.2.3 วิธีสร้างไดเมนชัน Product

1. คลิกเลือกที่ **New Dimension** อีกครั้ง ที่ขั้นตอน **Welcome to the Dimension** คลิก **Next**
2. ที่ขั้นตอน **Choose how you want to create the dimension** เลือก **Snowflake schema:Multiple, related dimension tables** หลังจากนั้น คลิก **Next**
3. ที่ขั้นตอน **Select the dimension tables** ดับเบิลคลิก **product** และ **product_class** ย้ายไปยัง **Selected tables** หลังจากนั้นคลิก **Next**
4. สองตารางที่ได้เลือกไว้ที่ขั้นตอนก่อนหน้านี้ จะได้ถูกเชื่อต่อระหว่างกันและจะถูกแสดงไว้ที่ขั้นตอน **Create and edit joins ของ Dimension Wizard** หลังจากนั้นคลิก **Next**
5. กำหนด Level สำหรับ **Product Dimension** โดยให้ดับเบิลคลิกไปยังที่ **Product_category, Product_subcategory, Brand_name** ตามลำดับหลังจากนั้นคลิก **Next**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-8 แสดงการเชื่อมต่อของ 2 ตารางเมื่อทำ Snowflake schema

6. ที่ขั้นตอน **Specify the member key columns** คลิก **Next**
7. ที่ขั้นตอน **Select advanced options** คลิก **Next**
8. ที่ขั้นตอนสุดท้ายใส่ชื่อ **Product** ที่ **Dimension name** แล้วก็ให้เลือกช่อง **Share this dimension with other cubes** ออกไปจากนั้นคลิก **Finish**
9. คุณสามารถเห็น **Product** Dimension ที่ **Cube dimension** list

4.2.2.4 สร้างไดเมนชัน Store

ทำการสร้างไดเมนชัน Store เข้าไปที่ **New dimension** เลือก **A single Dimension table (flat of star schema)** จากนั้นเลือก **Store** ส่วน **Levels** ให้เลือกเป็น Store_country, store_state, store_city, และ **store_name columns** ตามลำดับ แล้วตั้งชื่อไดเมนชันนี้ว่า "Store" แล้วก็ให้เลือกช่อง Share this dimension with other cubes ออกไป เช่นกัน

4.2.2.5 สร้างไดเมนชัน Promotion

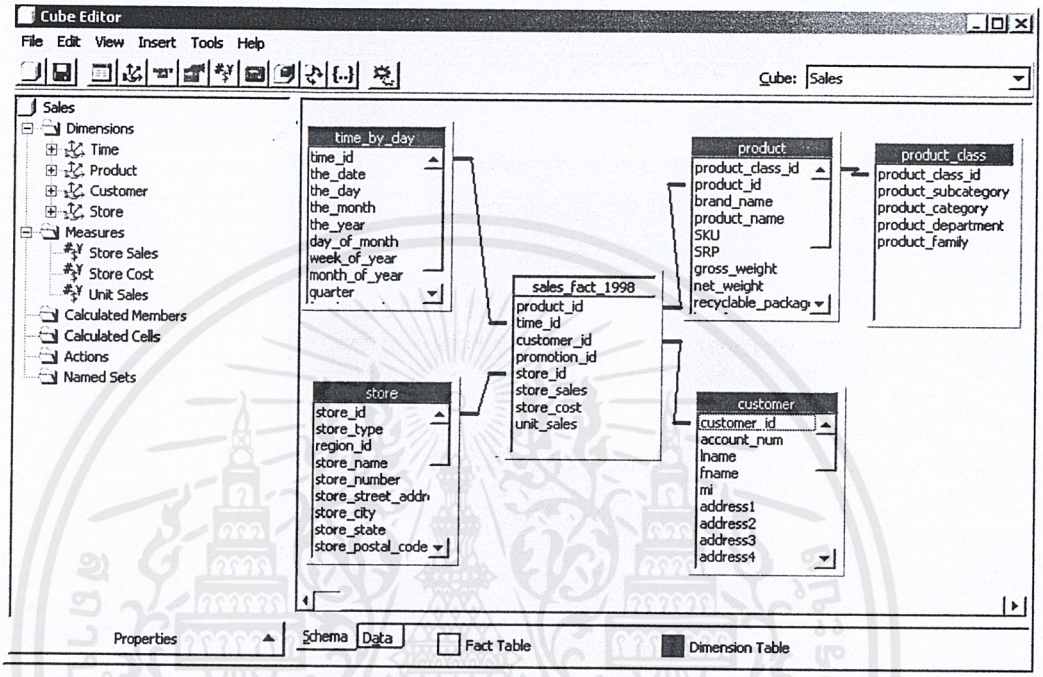
ทำการสร้างไดเมนชัน Promotion เข้าไปที่ **New dimension** เลือก **A Single Dimension table (flat of star schema)** จากนั้นให้ทำการเลือก **Promotion** ส่วน **Levels** ให้เลือกเป็น **Media_type** และ **promotion_name columns** ตามลำดับ แล้วตั้งชื่อไดเมนชันนี้ว่า "**Promotion**" แล้วก็เลือกช่อง **Share this dimension with other cubes** ออกไป เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.6 สร้างไดเมนชัน Customer

ทำเหมือนๆ ไดเมนชันอื่นๆ ก่อนหน้านี้ โดยเรียง Level ดังนี้ **Country, State province, city, name**

ทั้ง **Time, Product, Store** และ **Promotion Dimension** ที่เลือกมาทั้งหมดนั้น จะถูกเก็บอยู่ในสิสต์ของ Cube Dimension เป็นอันเสร็จสิ้นขั้นตอนของการสร้างคิวบ์ ให้ตั้งชื่อคิวบ์ นี้ว่า **"Sales"**



รูปที่ 4-9 แสดงรูปสกีมาของคิวบ์ Sales

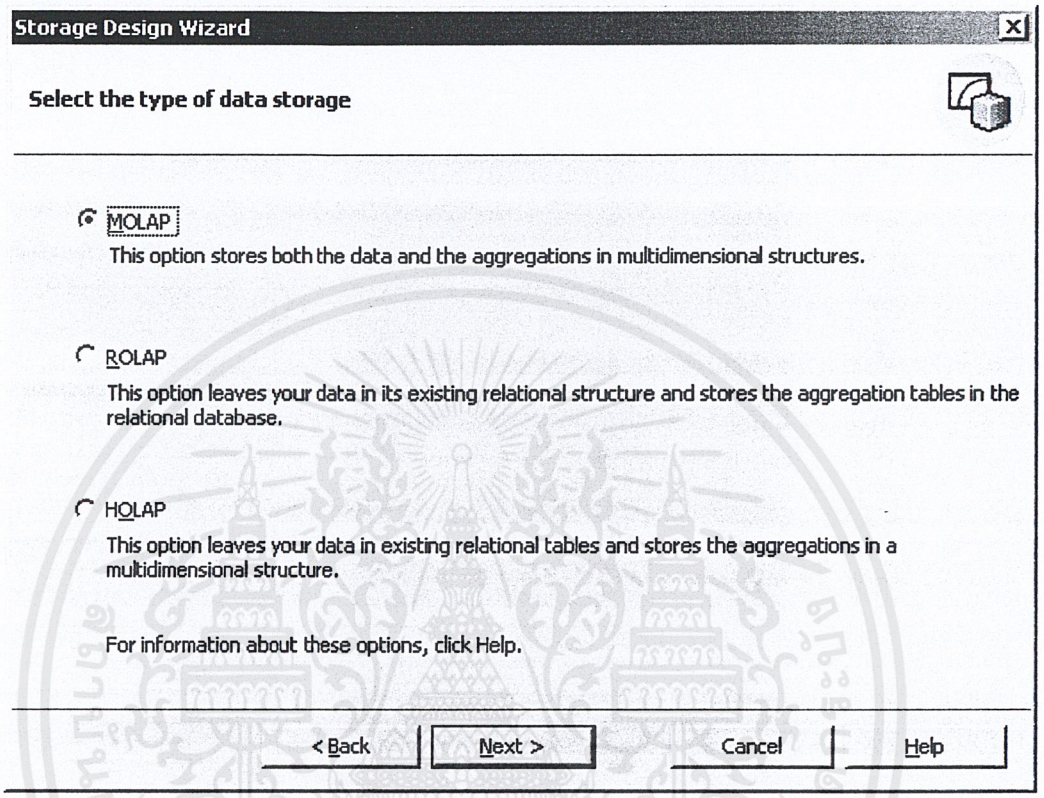
- การแก้ไขคิวบ์ โดยใช้ Cube editor wizard ในหน้าต่างที่แสดง Schema ของ Cube editor เราจะเห็นตารางแฟ็ค เป็นสีเหลือง ส่วนไดเมนชันที่มาเชื่อมต่อ จะเป็นสีน้ำเงิน และใน Cube editor tree view จะแสดงโครงสร้างของคิวบ์ในแบบลำดับขั้นต้นไม้ (Hierarchical Tree)

ใน Cube editor นี้ สามารถที่จะสร้างไดเมนชันที่ต้องการใช้เพิ่มได้ แต่ไดเมนชันที่สร้างเพิ่มนี้จะเป็น Private dimension คือ ใช้ได้เฉพาะในคิวบ์ที่กำลังใช้งานอยู่เท่านั้น ไม่สามารถใช้ร่วมกับคิวบ์ อันอื่นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การออกแบบการจัดเก็บข้อมูลและการประมวลผลคิวบ์ (Designing Storage and Processing Cube)

Analysis Services นั้น เราสามารถปรับปรุงประสิทธิภาพและเวลาที่ใช้ในการคิวรีได้ โดยใช้ **Storage Design Wizard** ระบุวิธีการจัดเก็บได้ 3 ชนิด คือ MOLAP, ROLAP หรือ HOLAP



รูปที่ 4-10 เลือกวิธีการจัดเก็บข้อมูล

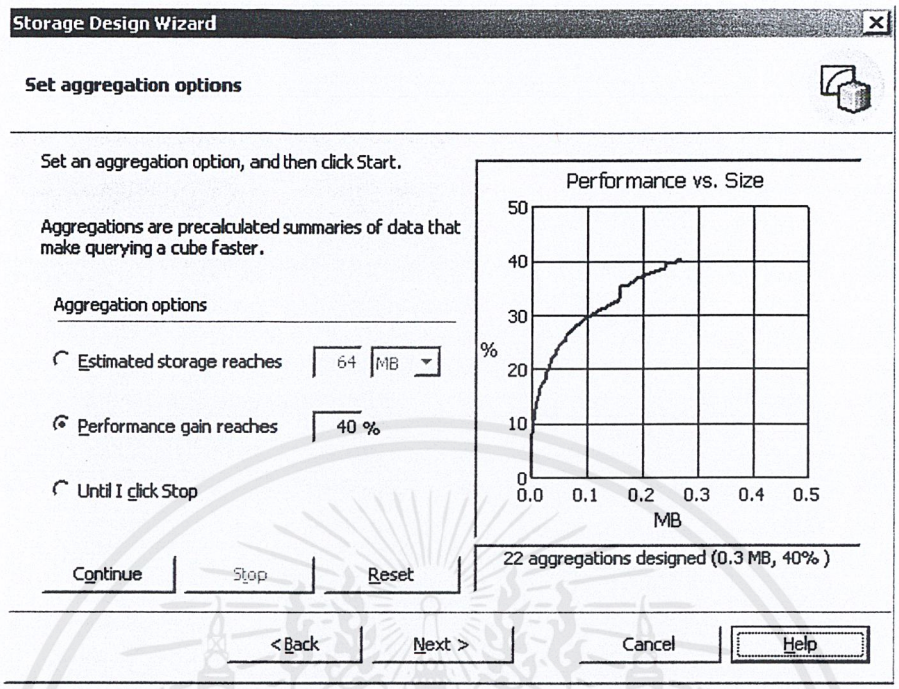
MOLAP เป็นโหมดที่มีการทำการรวมค่าข้อมูลและก๊อปปี้ข้อมูลไปเก็บไว้ในโครงสร้างแบบมัลติไดเมนชันแนลบนเครื่อง Analysis Server

ROLAP ใช้วิธีการเก็บข้อมูลที่เป็นการรวมค่าไว้ในตารางของรีเลชันแนลดาต้าเบส (Relational Database)

HOLAP เป็นการรวมวิธีของ MOLAP และ ROLAP เข้าด้วยกัน ส่วนที่เหมือน MOLAP คือจะเก็บข้อมูลที่เป็นการรวมค่าไว้ในโครงสร้างแบบมัลติไดเมนชันแนลบนเครื่อง Analysis Server แต่ข้อมูลจริงจะไม่ได้ถูกเก็บไว้ด้วย

หลังจากที่เลือกชนิดของการจัดเก็บข้อมูลแล้ว สามารถดูกราฟที่แสดงถึงประสิทธิภาพ (Performance) และขนาด (Size) ของคิวบ์ แสดงได้ดังรูปที่ 4-11

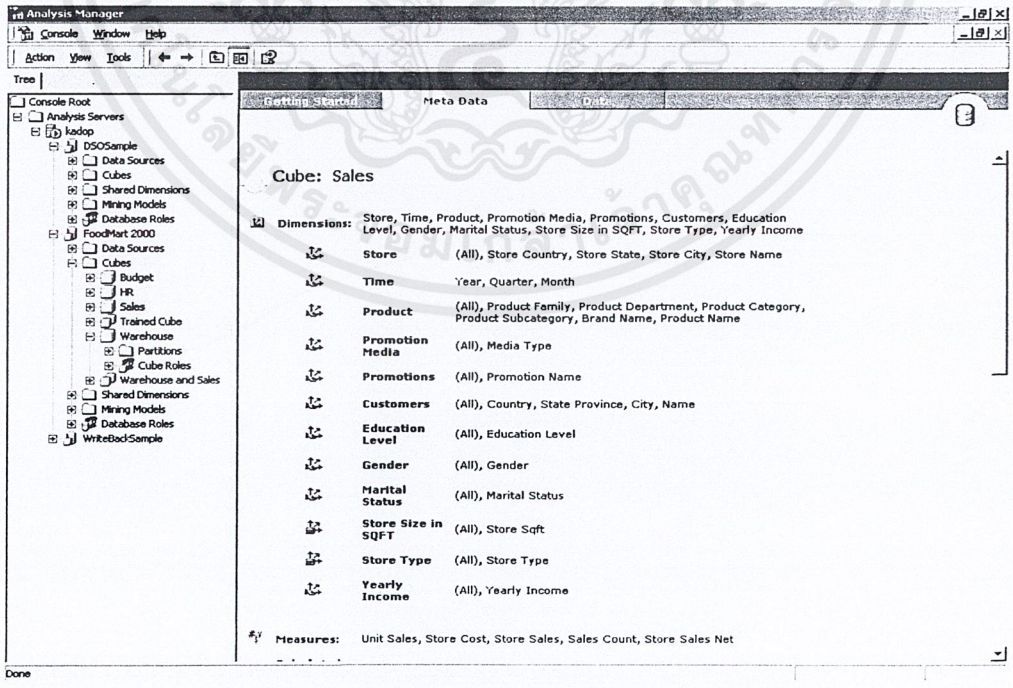
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-11 แสดงการวัดประสิทธิภาพและขนาดของ Cube ที่สร้างขึ้นมา

4.2.4 การดูข้อมูลรายละเอียดของคิวบ์ (Viewing Metadata for Cube)

Analysis Services สามารถดูรายละเอียดข้อมูลที่มีอยู่ในคิวบ์ หรือเมต้าดาต้า (Metadata) ได้แก่ Dimensions, Measures, Calculated Members, Source Tables, Processed, Type, Size และ Data Source

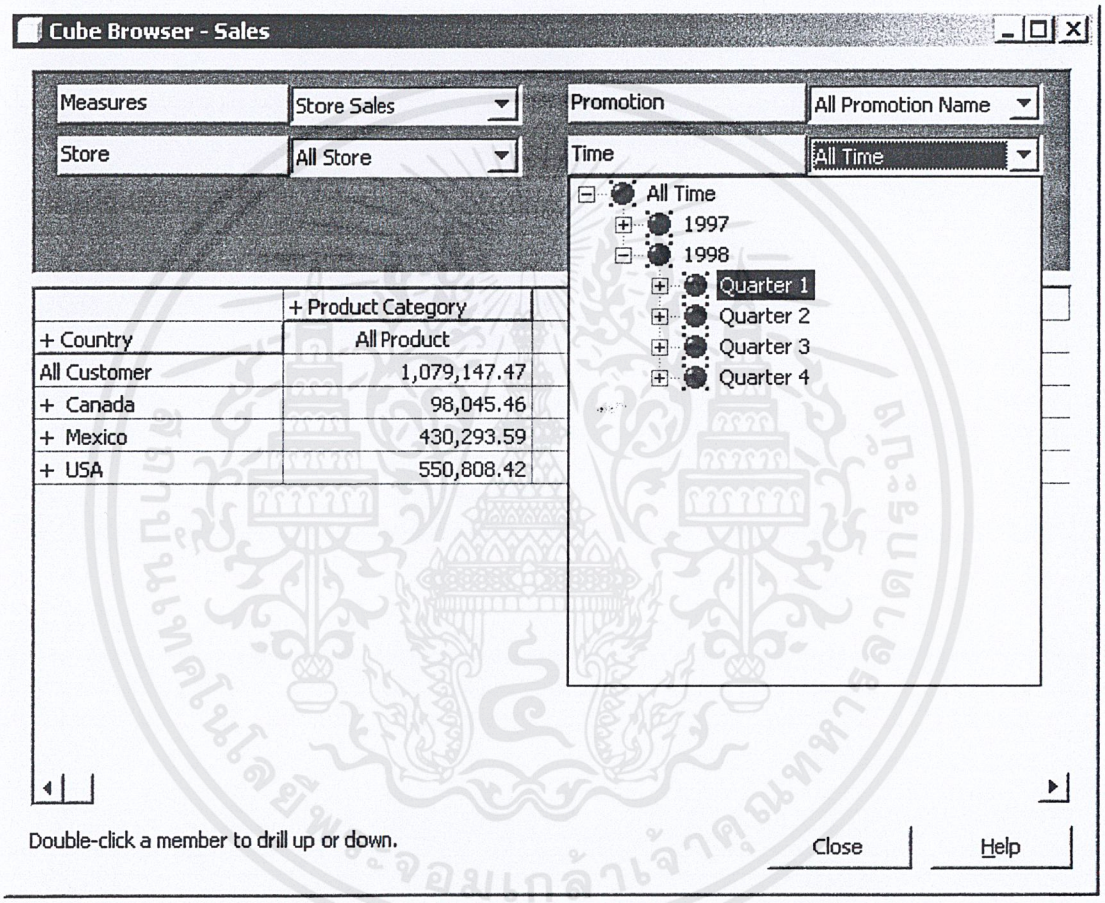


รูปที่ 4-12 แสดงรายละเอียดของข้อมูล (Meta data)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 การดูข้อมูลในคิวบ์ (Browse Cube)

สามารถใช้ Cube Browser เพื่อดูข้อมูลในลักษณะต่าง ๆ กันได้ สามารถใส่จำนวนไดเมนชันที่จะดูข้อมูลและสามารถที่จะ Drill down ในรายละเอียดที่มากขึ้นได้ด้วยที่หน้าต่าง Cube Browser และยังสามารถเพิ่มไดเมนชันที่ต้องการดู เข้าไปแทนที่ไดเมนชันที่แสดงอยู่ในกริด (Grid) ได้ เพียงแค่ลากไดเมนชันจากส่วนของพาเลต (Palette) ด้านบน ลงมาวางแทนไดเมนชันในกริด ส่วนไดเมนชันที่ถูกแทนที่ก็จะถูกนำขึ้นไปเก็บไว้ที่ Palette แทน ในส่วนของ Time Dimension เราสามารถเลือกดูลักษณะของเวลาได้ทั้งแบบ Year, Quarter หรือ Month



รูปที่ 4-13 แสดงการเลือก Level เพื่อ Slice และ Dice ข้อมูล

ส่วนการทำ Drill down ข้อมูล ก็สามารถทำได้ง่ายเช่นกัน เพียงแค่ดับเบิลคลิกที่เซลล์ ในกริดเพียงเท่านี้เราก็จะเห็นรายละเอียดของข้อมูลในระดับที่ลึกลงไป แสดงได้ดังรูปที่ 4-14 ส่วนการทำ Drill Up ข้อมูล ก็ทำตรงกันข้ามกับการ Drill down คือ ให้ดับเบิลคลิกที่เซลล์ที่มีเครื่องหมาย – อยู่ข้างหน้า เราก็จะเห็นข้อมูลในระดับที่สูงขึ้นมา 1 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Cube Browser - Sales

Measures	Store Sales	Promotion	All Promotion Name
Store	All Store	Time	Quarter 1

		+ Country	
- Product Category	+ Product Subcategory	All Customer	+ Canada
All Product	All Product Total	290,873.18	23,881.13
- Baking Goods	Baking Goods Total	8,103.52	708.65
	+ Cooking Oil	3,344.79	306.67
	+ Sauces	710.35	63.03
	+ Spices	2,462.64	215.34
	+ Sugar	1,585.74	123.61
+ Bathroom Products	Bathroom Products Total	6,805.34	609.34
+ Beer and Wine	Beer and Wine Total	7,614.09	781.04
+ Bread	Bread Total	8,340.32	688.55
+ Breakfast Foods	Breakfast Foods Total	8,452.72	652.49
+ Candles	Candles Total	792.53	91.02
+ Candy	Candy Total	7,618.12	518.07
+ Canned Anchovies	Canned Anchovies Total	1,165.30	148.23

Double-click a member to drill up or down.

Close Help

รูปที่ 4-14 แสดงการ Drill down ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
โหมดการเก็บข้อมูล (Storage Mode)

5.1 การเก็บข้อมูล

การคิวรีข้อมูลของ OLAP นั้นจะทำคิวรีโดยเสมือนว่าค่านั้นได้คิวรีจากคิวบ์ (cube)โดยตรง จากตัวอย่าง FoodMart 2000 สมมุติว่ามีตัววัด (measure) คือ unit sale, store sales และมีไดเมนชัน Time, Customer ถ้าต้องการจะคิวรี (Query) ข้อมูลในระดับชั้น (Level) ที่ต่ำสุด เช่น ต้องการหาค่า unit sale ใน San Francisco ที่เวลา 1997 จะรีเทิร์นค่า 88 ชิ้นกลับมา แต่ในเวลาเดียวกันถ้าคิวรีคิวบ์ที่ระดับรายละเอียดที่สูงกว่า ตัวอย่างเช่น unit sale ของประเทศอเมริกา (USA) จะรีเทิร์นค่า 267,730 ชิ้นราวกับว่าข้อมูลนั้นถูกเก็บอยู่ในคิวบ์โดยตรง

สมมุติฐานข้อมูลดาต้าแวร์เฮาส์ FoodMart 2000 มีไดเมนชันคือ Product, Customer, Time, Promotion และมีสินค้าทั้งหมด 300 รายการ มีประวัติลูกค้าที่สั่งซื้อของ 4 คน มีเวลาทั้งหมด 20 เดือน และมีการโปรโมทสินค้า 10 แบบ เพราะฉะนั้นในตารางแฟ็ค (Fact Table) จะมีจำนวนแถวทั้งหมดเท่ากับ 300*4*20*10 หรือ 240,000 แถวตามทฤษฎี แต่ในความเป็นจริงอาจมีการเก็บค่าในตารางแฟ็คไม่ถึงตามนั้นเนื่องจากข้อมูลบางส่วนที่เป็นข้อมูลที่ไม่มีค่าใดๆในตัววัด (measure) ในที่นี้ตารางแฟ็คมีจำนวนเรคอร์ดโดยประมาณเท่ากับ 10,000 เรคอร์ด (สมมุติขึ้นมาเพื่อเอามาเปรียบเทียบในภายหลัง)

เนื่องจากคิวบ์นั้นประกอบด้วยค่าที่ถูกรวม (Aggregate) ทุกๆค่าที่เป็นไปได้ในทุกระดับที่เป็นไปได้ในทุกๆไดเมนชัน ซึ่งทำให้คิวบ์ต้องมีค่ามากกว่า 240,000 แบบที่เป็นไปได้ เช่น สมมุติว่ามี 330 สมาชิกในไดเมนชัน Product , มี 10 สมาชิกในไดเมนชัน Customer , มี 24 สมาชิกในไดเมนชัน Time , 22 สมาชิกในไดเมนชัน Promotion และมีตัววัดอีก 3 ตัว จะเห็นได้ว่าจะต้องเก็บค่าถึง 330*10*24*22*3 หรือเท่ากับ 5,227,200 ค่าซึ่งประกอบด้วยค่าประมาณ 522 เท่าของจำนวนเรคอร์ดที่เก็บในตารางแฟ็ค การเกิดอย่างนี้เรียกว่า Data Explosion โดยประเด็นของคิวบ์แบบ OLAP คือจะจัดการกับ Data Explosion

การไม่ทำการรวมค่าทั้งหมดจะคำนวณข้อมูลขึ้นมาเมื่อต้องการ เป็นวิธีการที่ง่ายที่สุดที่แก้ปัญหา Data Explosion ในการทำคลังข้อมูลที่มีขนาดใหญ่มากๆ จะคิวรีสักครั้งจะเป็นการนำข้อมูลจำนวนมากๆจากระดับที่ต่ำกว่ามารวมกัน ซึ่งจะเร็วหรือไม่นั้นเป็นความสามารถของ OLAP ที่จะสร้างอัลกอลิทึมมาจัดการ หรือเป็นความสามารถของผู้ดูแลระบบที่จะตัดสินใจในการปรับแต่งระหว่างขนาดจริงของคิวบ์และความเร็วของคิวบ์ให้สมดุลกัน

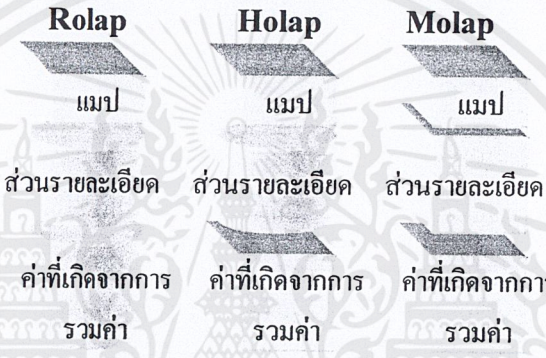
5.2 วิธีการเก็บข้อมูลของ Analysis Server

Analysis Service จะประกอบด้วย 3 ส่วนคือ แมป, ค่ารายละเอียด, ค่าจากการรวมค่า ซึ่งแมปจะเก็บข้อมูลของลำดับขั้นที่เกี่ยวกับสมาชิกของไดเมนชันทั้งหมดที่ถูกใช้ในคิวบ์ ส่วนรายละเอียด จะเป็นส่วนที่เป็นค่าล่างสุดของไดเมนชันนั้นๆ จะมีค่าเหมือนกับค่าที่อยู่ในตารางแฟ็ค ส่วนค่าจากการรวมจะเป็นค่าที่ถูกรวมผลจากระดับที่สูงขึ้นไปจากส่วนรายละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OLAP จะเก็บแมปของคิวบ์ไว้ใน Analysis เสมอแต่ Analysis จะให้เลือกวิธีการว่าจะเก็บข้อมูลส่วนรายละเอียดและส่วนค่าจากการรวมด้วยวิธีไหน โดยมีให้เลือก 3 แบบคือ

1. ROLAP (Relational OLAP) เป็นการเก็บข้อมูลแบบรีเลชันแนล (Relational OLAP) จะเก็บส่วนรายละเอียดและส่วนค่าจากการรวมในรีเลชันแนลดาต้าเบส
2. HOLAP (Hybrid OLAP) เป็นการเก็บข้อมูลแบบผสมระหว่างแบบ MOLAP และ ROLAP โดยจะเก็บข้อมูลในส่วนรายละเอียดไว้ในตารางรีเลชันแนล และส่วนค่าที่เกิดจากการรวมจะเก็บไว้ในคิวบ์
3. MOLAP (Multidimensional OLAP) เป็น OLAP ที่เก็บข้อมูลทั้งส่วนรายละเอียดและค่าจากส่วนรวมค่าภายในคิวบ์



รูปที่ 5-1 แสดงการเก็บข้อมูลด้วยวิธีต่างๆ

โปรแกรมประยุกต์ทางฝั่งไคลเอ็นต์จะไม่ต้องสนใจวิธีเก็บข้อมูลว่าเป็นแบบใด การเลือกว่าจะจัดเก็บแบบใดจะมีผลต่อประสิทธิภาพในการคิวรีเท่านั้น

เนื่องจากโปรแกรมทางไคลเอ็นต์ไม่สามารถกำหนดวิธีการเก็บข้อมูลได้ ดังนั้นผู้ดูแลระบบสามารถเปลี่ยนการจัดเก็บข้อมูลโดยไม่มีผลกระทบต่อโปรแกรมทางฝั่งไคลเอ็นต์

5.3 ความเหมาะสมในการใช้งานการเก็บค่าทั้งสามแบบ

MOLAP เป็นโหมดที่มีการรวมค่าข้อมูลและก๊อปปี้ข้อมูลไปเก็บไว้ในโครงสร้างแบบมัลติไดเมนชันแนลบนเครื่อง Analysis Server การเก็บข้อมูลในโหมด MOLAP นี้ทำให้เราสามารถคิวรีซึ่งทำให้เราได้รับคำตอบจากการคิวรีในระยะเวลาอันสั้น โดยขึ้นอยู่เปอร์เซ็นต์และการออกแบบส่วนที่เราทำการรวมค่า โดยทั่วไปแล้ว MOLAP เหมาะสมสำหรับงานที่ใช้เป็นประจำและจำเป็นที่จะต้องให้คำตอบได้อย่างรวดเร็ว

ROLAP ใช้วิธีการเก็บข้อมูลรวมค่าในตารางของรีเลชันแนลดาต้าเบส อย่างไรก็ตามเราสามารถสร้างข้อมูล โดยปราศจากการทำสร้างการรวมค่าในรีเลชันแนลดาต้าเบสก็ได้ เนื่องจาก ROLAP ไม่ได้มีการก๊อปปี้ข้อมูลมาเก็บเอาไว้ ทำให้เมื่อผลลัพธ์ไม่สามารถหาจากข้อมูลรวมค่าหรือไคลเอ็นต์แคช (client cache)ได้จำเป็นต้องเข้าไปหาข้อมูลยังตารางแฟ็ค ถ้าเก็บแบบ ROLAP เวลาตอบสนองในการคิวรีจะช้ากว่าทั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOLAP และ HOLAP , โดยทั่วไปแล้ว ROLAP จะใช้กับข้อมูลใหญ่ๆ ที่ไม่มีการคิวรีบ่อยๆ เช่น ข้อมูลที่เก่ามากๆ

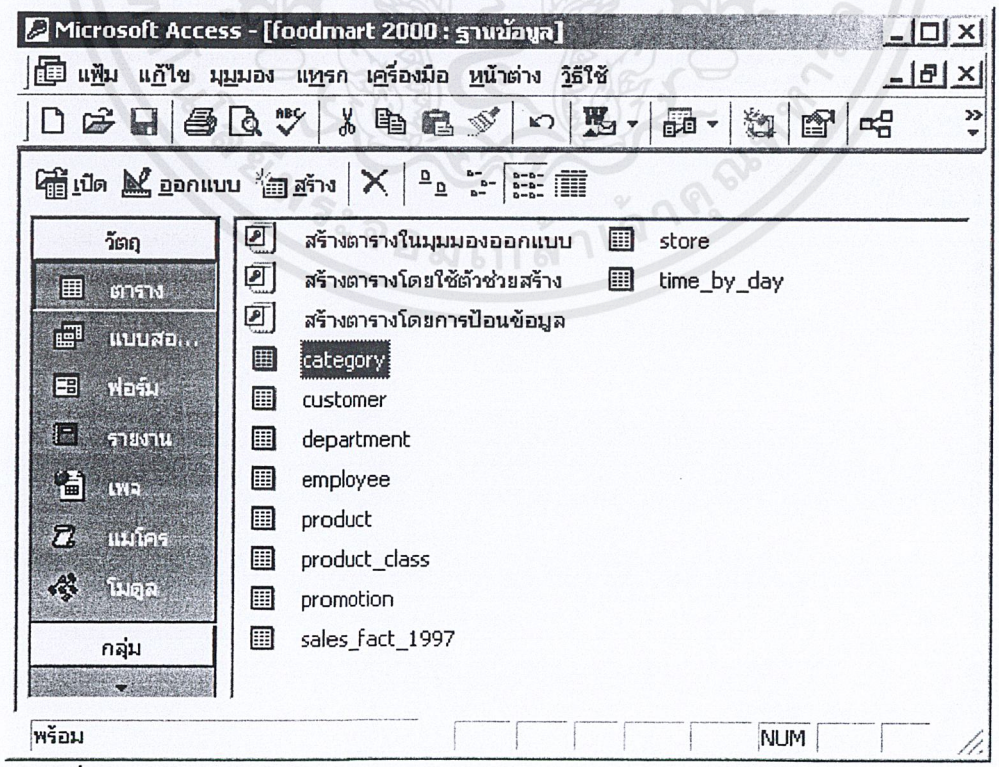
HOLAP เป็นการรวมวิธีของ MOLAP และ ROLAP เข้าด้วยกัน ส่วนที่เหมือน MOLAP คือจะเก็บข้อมูลการรวมค่าไว้ในโครงสร้างแบบมัลติไดเมนชันแนลบนเครื่อง analysis server แต่ข้อมูลจริงจะไม่ได้ถูกเก็บไว้ด้วย สำหรับการคิวรีที่ใช้ข้อมูลรวมค่า HOLAP จะทำงานเหมือนกับ MOLAP แต่ถ้าคิวรีจำเป็นต้องใช้ข้อมูลจริง (source data) เช่นการทำ drill down ข้อมูลในระดับล่างสุดที่เป็นอะตอมมิก ที่ไม่ได้ทำการรวมค่าไว้ HOLAP จึงต้องดึงข้อมูลจากรีเลชันแนลดาต้าเบสส่งผลให้ HOLAP ทำงานได้ไม่เร็วเท่ากับ MOLAP แต่อย่างไรก็ตาม การเก็บข้อมูลแบบ HOLAP จะมีข้อดีเหนือ MOLAP คือ ใช้เนื้อที่ในการเก็บน้อยกว่า MOLAP และยังมีเวลาตอบสนองดีกว่า ROLAP เมื่อทำงานกับข้อมูลประเภทที่เป็นการสรุปโดยทั่วไปแล้ว HOLAP จะเหมาะสมสำหรับการคิวรีข้อมูลขนาดใหญ่ที่ต้องการและต้องการเวลาในการตอบสนองที่รวดเร็ว

5.4 การทดสอบการรวมค่าไดเมนชัน

เพื่อให้เห็นชัดเจนว่าเกิดการรวมค่าอย่างไร จึงใช้ ROLAP เนื่องจากมีการเก็บข้อมูลการรวมค่าไว้ในรีเลชันแนลดาต้าเบสทำให้ง่ายต่อความเข้าใจและอีกเหตุผลคือการเก็บข้อมูลแบบมัลติไดเมนชันแนลนั้นไม่สามารถรู้ถึงอัลกอลิทึมภายในของมันได้ แหล่งข้อมูลที่นำมาทดสอบนั้นเป็นตัวอย่างจาก Analysis Service คือ FoodMart 2000.mdb

5.4.1 การทดสอบการรวมค่าไดเมนชันเดียว

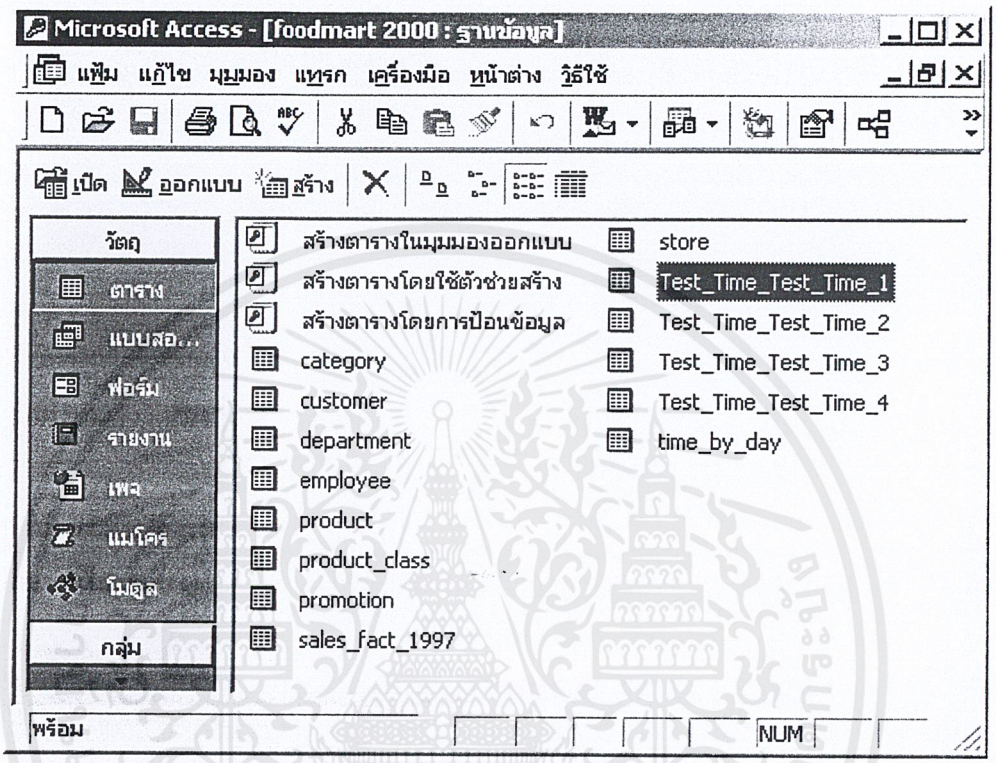
1. เปิดแหล่งข้อมูล FoodMart 2000.mdb เพื่อดูตารางข้างในประกอบด้วยตารางอะไรบ้าง



รูปที่ 5-2 แสดงฐานข้อมูลก่อนถูกทำการรวมค่า (Aggregation) ด้วยการเก็บค่าแบบ ROLAP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หลังจากนั้นสร้างคิวบ์โดยเลือกตารางแฟ็ค sales_face_1997, เลือกตัววัด unit sale และ store sale และเลือกไดเมนชัน Time แบบ Year, Quarter, Month และตั้งชื่อคิวบ์เป็น Test_Time
3. จากนั้นให้กำหนดการเก็บข้อมูลเป็นแบบ ROLAP
4. เมื่อกลับมาดูที่ฐานข้อมูล FoodMart 2000 จะเห็นว่ามันมีการเปลี่ยนแปลงดังรูป



รูปที่ 5-3 แสดงฐานข้อมูลหลังจากทำการรวมค่า (Aggregation) ด้วยการเก็บค่าแบบ ROLAP

5. สังเกตุว่าจะมีตารางมาใหม่ 4 ตารางได้แก่ Test_Time_Test_Time_1 , Test_Time_Test_Time_ 2 , Test_Time_Test_Time_3 , Test_Time_Test_Time_4

รูปที่ 5-4 แสดงให้เห็นการเก็บการรวมค่าในตาราง **Test_Time_Test_Time_4** โดยค่าที่รวมได้จะเก็บอยู่ในคอลัมน์ **SUM_StoreSale** และ **SUM_UnitSale**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test_Time_Test_Time_4 : ตาราง

Year_L2	Quarter_L3	Month_L4	SUM_StoreSa	SUM_UnitSale
1997	1	1	฿45,539.69	21628
1997	1	2	฿44,058.79	20957
1997	1	3	฿50,029.87	23706
1997	2	4	฿42,878.25	20179
1997	2	5	฿44,456.29	21081
1997	2	6	฿45,331.73	21350
1997	3	7	฿50,246.88	23763
1997	3	8	฿46,199.04	21697
1997	3	9	฿43,825.97	20388
1997	4	10	฿42,342.27	19958
1997	4	11	฿53,363.71	25270
1997	4	12	฿56,965.64	26796

ระเบียน: 1 จาก 12

รูปที่ 5-4 สรุปค่าของตัววัดในระดับ Month

Test_Time_Test_Time_3 เก็บคอลัมน์ของ Year และ Quarter ไว้เท่านั้น ในตารางนี้มีจำนวนแถวทั้งหมด 4 แถว เป็นการรวมค่าไว้สำหรับสมาชิกในระดับ Quarter

Test_Time_Test_Time_3 : ตาราง

Year_L2	Quarter_L3	SUM_StoreSa	SUM_UnitSale
1997	1	฿139,628.35	66291
1997	2	฿132,666.27	62610
1997	3	฿140,271.89	65848
1997	4	฿152,671.62	72024

ระเบียน: 1 จาก 4

รูปที่ 5-5 สรุปค่าของตัววัดในระดับ Quarter

Test_Time_Test_Time_2 เก็บคอลัมน์ของ Year ในตารางนี้มีจำนวนทั้งหมด 1 แถว เป็นการรวมค่าไว้สำหรับสมาชิกในระดับ Year

Test_Time_Test_Time_2 : ตาราง

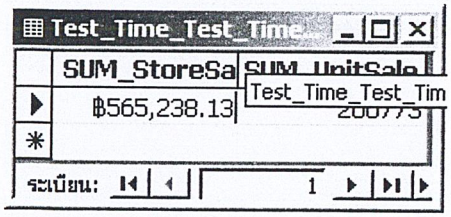
Year_L2	SUM_StoreSa	SUM_UnitSale
1997	฿565,238.13	266773

ระเบียน: 1 จาก 1

รูปที่ 5-6 สรุปค่าของตัววัดในระดับ Year

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test_Time_Test_Time_1 ประกอบด้วยฟิลด์ที่รวมค่าทั้งหมดของค่าตัววัดเท่านั้น เป็นค่าที่ตรงกับระดับ All ของคิวบ์



รูปที่ 5-7 แสดงตารางสรุปค่าของตัววัดในระดับ ALL

สังเกตว่าการรวมค่าในแต่ละระดับนั้นไม่ได้รวมค่าทุกๆ สมาชิกในระดับนั้นๆ แต่จะรวมค่าอะไรไว้บ้างขึ้นอยู่กับรูปแบบของข้อมูลเฉพาะที่อยู่ในตารางแฟ็ค

ระดับ	จำนวนสมาชิกที่เกิดขึ้นจริง	จำนวนสมาชิกในทางทฤษฎี	ตารางรวมค่าเลขที่
ALL(1)	1	1	1
Year(2)	1	2	2
Quarter(8)	4	8	3
Month(24)	12	24	4

ตารางที่ 5-1 แสดงการเปรียบเทียบจำนวนสมาชิกที่เกิดขึ้นจริงกับสมาชิกในทางทฤษฎีของไดเมนชันเดียว

จำนวนสมาชิกทางทฤษฎีคือจำนวนที่บอกถึงค่าที่เป็นไปได้ทั้งหมดที่สามารถอยู่ในตารางการรวมค่า เกิดจากแต่ละระดับของไดเมนชันคูณกัน จำนวนสมาชิกที่เกิดขึ้นจริงคือค่าที่อยู่ในตารางรวมค่าจริงๆ และคอลัมน์ตารางรวมค่าเลขที่บอกว่าค่าสมาชิกนั้นถูกเก็บที่ตารางใด เช่น
Test_Time_Test_Time_1 จะเป็นตารางเลขที่ 1

5.4.2 การทดสอบการรวมค่าไดเมนชันของ 2 ไดเมนชัน

การรวมค่าจะซับซ้อนมากขึ้นถ้ามีหลายไดเมนชันมาเกี่ยวข้องการสร้างคิวบ์ด้วย การทดลองนี้จะสร้างคิวบ์ด้วยสองไดเมนชันโดยใช้ Customer กับ Time ในตัวอย่าง FoodMart 2000 โดยจะกำหนด Customer และ Time จะมีระดับดังนี้ Country > State_Province > City และ Year > Quarter > Month ตามลำดับ เพราะฉะนั้นในฐานข้อมูล FoodMart จะต้องมีตารางอยู่เท่ากับจำนวนระดับของทุกไดเมนชันที่รวมกับ ระดับ [(ALL)] มาคูณกันซึ่งคือ 4*4 จะได้ 16 ตาราง
ในแต่ละระดับมีจำนวนสมาชิกดังนี้

ไดเมนชัน Customer

ALL มี สมาชิก 1 ตัว

Country มีสมาชิก 3 ตัว

State_Province มีสมาชิก 13 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

City มีสมาชิก 109 ตัว

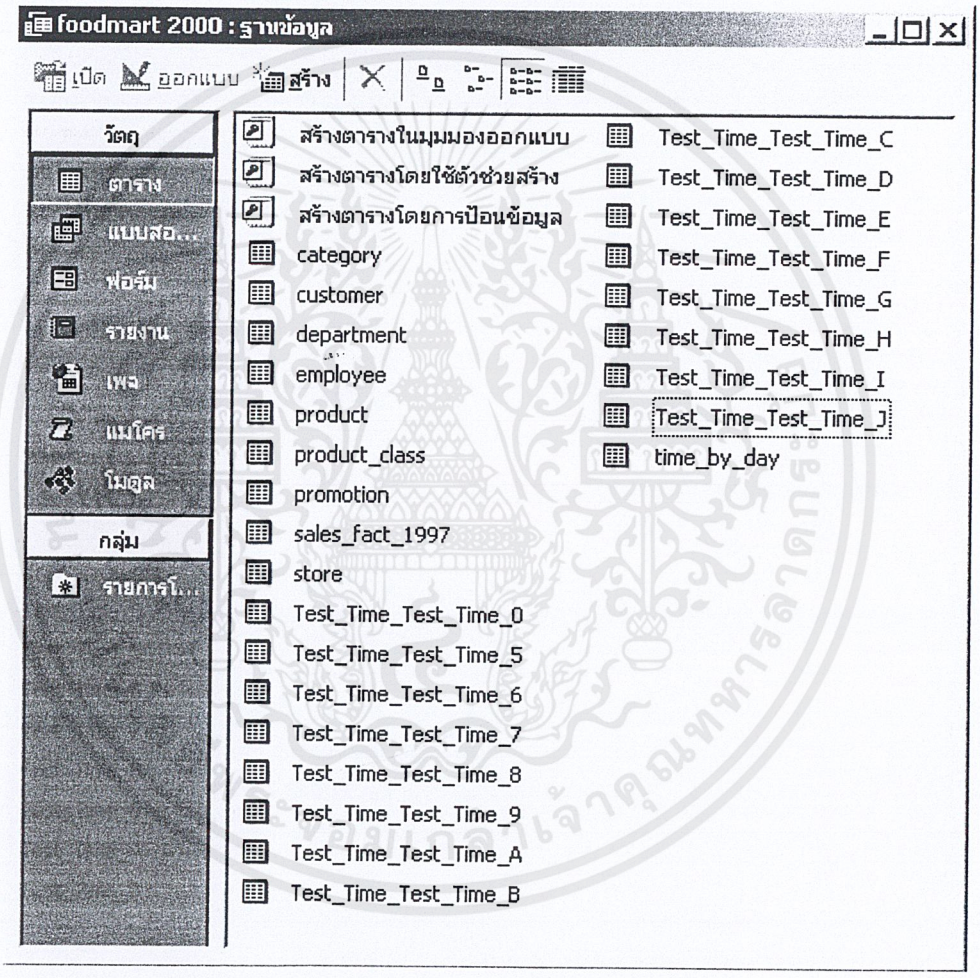
ไดเมนชัน Time

All มีสมาชิก 1 ตัว

Year มีสมาชิก 2 ตัว

Quarter มีสมาชิก 8

Month มีสมาชิก 24 ตัว



รูปที่ 5-8 รูปแสดงตารางการรวมค่าของสองไดเมนชัน

แต่ในตารางรวมค่าจะเก็บค่าไม่ครบทั้งหมดจะเก็บเพียงแค่บางส่วนที่มีในตารางแฟ็ค หรือตามการเลือกที่เกิดจากการปรับเปลี่ยนประสิทธิภาพในการเก็บข้อมูล (ตอนเลือกว่าจะให้มีจำนวนการรวมค่ากี่เปอร์เซ็นต์) ตารางที่ 5-2 แสดงจำนวนเรคอร์ดที่มีตามทฤษฎีและจำนวนเรคอร์ดที่เกิดขึ้นจริงในแต่ละตารางการรวมค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Level	ALL(1)	Country(3)	State Province(13)	City(109)
ALL(1)	1 / 1 [5]	1 / 3 [9]	3 / 13 [C]	78 / 109 [G]
Year(2)	1 /2 [6]	1 / 6 [0]	3 / 26 [D]	78 / 218 [H]
Quarter(8)	4 / 8 [7]	4 / 24 [A]	12 / 104 [E]	312 / 872 [I]
Month(24)	12 / 24 [8]	12 / 72 [B]	36 / 312 [F]	898 / 2616 [J]

ตารางที่ 5-2 แสดงข้อมูลการเปรียบเทียบจำนวนสมาชิกที่มีอยู่จริงกับสมาชิกทางทฤษฎีของสองไดเมนชัน

จากตารางที่ 5-2 ตัวเลขที่มีอยู่ในวงเล็บคือตัวเลขที่แสดงถึงจำนวนสมาชิกทั้งหมดที่มีในแต่ละระดับในไดเมนชันนั้นๆ ส่วนตัวเลขที่อยู่ในก้ามปู ([ ]) แสดงถึงค่าเหล่านี้อยู่ในตารางเลขที่เท่าไหร่ และ ตัวเลข / ตัวเลข หมายถึง จำนวนสมาชิกที่มีอยู่จริง / จำนวนสมาชิกทางทฤษฎี

สรุป

- การรวมค่าเกิดจากการนำค่าในระดับที่ต่ำกว่ามารวมกัน
- จำนวนการรวมค่า (number of Aggregation) ทั้งหมดที่เป็นไปได้ คือจำนวนของระดับในทุกๆ ไดเมนชันมาคูณกัน
- การเพิ่มไดเมนชันทำให้เกิดการเพิ่มจำนวนของการรวมค่าที่เป็นไปได้อย่างมากเนื่องจากเป็นการนำจำนวนระดับที่มีในไดเมนชันมาคูณ
- จำนวนค่าจริงๆที่เป็นไปได้จากการสรุปค่าในการรวมค่านั้นขึ้นอยู่กับรูปแบบของข้อมูลในตารางหลัก
- ไม่สามารถบอกได้ว่าตารางการรวมค่าที่ทำการเก็บแบบ ROLAP จะได้ตารางเลขที่เท่าไร

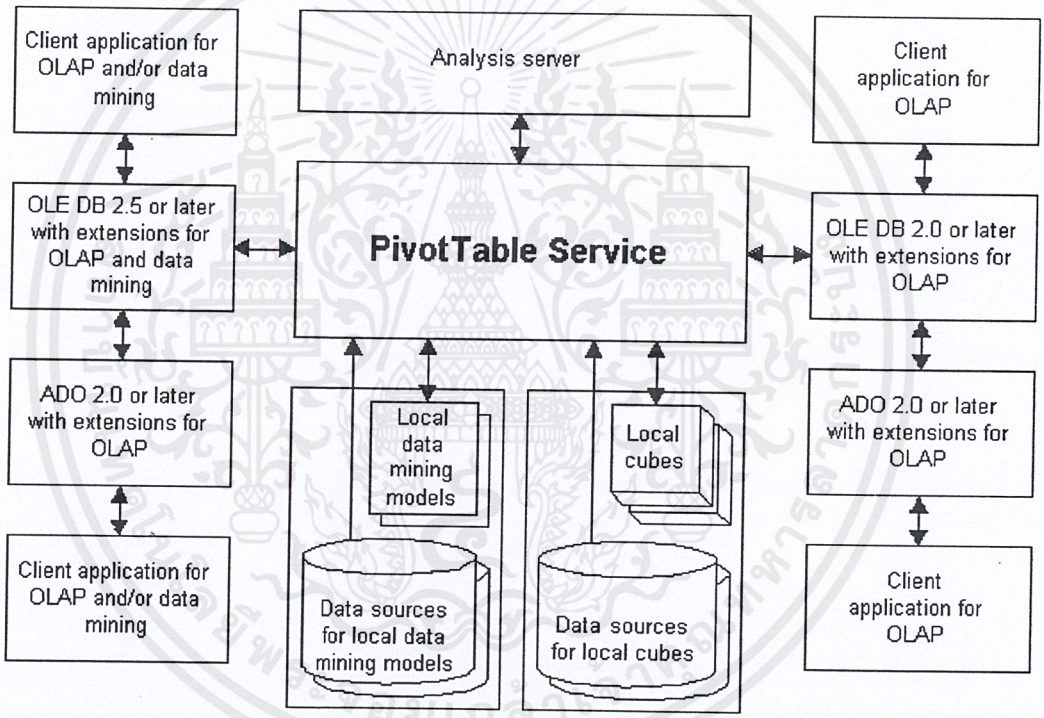
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

Microsoft OLAP Client Architecture

6.1 สถาปัตยกรรมไคลเอ็นต์ภายใน Analysis Service

หัวใจของสถาปัตยกรรมไคลเอ็นต์ คือ PivotTable Service โดยจะทำหน้าที่เป็นตัวติดต่อกับ Analysis server และจะสร้างอินเตอร์เฟสซึ่งจะเรียกใช้โดยโปรแกรมประยุกต์ไคลเอ็นต์ ในการเข้าถึงข้อมูล OLAP โปรแกรมประยุกต์ไคลเอ็นต์สามารถเรียกใช้ PivotTable Service ได้โดยเรียกผ่านอินเตอร์เฟสของ OLE DB เมื่อใช้โปรแกรมภาษา C++ หรือใช้ Microsoft ActiveX Data Object (ADO) เมื่อใช้โปรแกรมพวก Visual Basic ดังรูปที่ 6-1



รูปที่ 6-1 แสดงโครงสร้างของไคลเอ็นต์ใน Analysis Service

6.2 PivotTable Service

PivotTable Service คือ OLE DB Provider สำหรับข้อมูลมัลติไดเมนชันแนล นั่นคือ PivotTable Service จะสร้างฟังก์ชันของ OLE DB ให้แอพพลิเคชันเรียกใช้ในการเข้าถึงข้อมูลมัลติไดเมนชันแนลและนอกจากนี้ PivotTable Service ยังสามารถดึงข้อมูลที่เป็นตารางหรือข้อมูลที่เป็นมัลติไดเมนชันแนลเพื่อให้รองรับกับการใช้ SQL หรือ MDX (MultiDimensional eXpression)

PivotTable Sevice สามารถสร้างไฟล์โลคอลคิวบ์ (local cube) ที่เก็บข้อมูลจากคิวบ์บนเครื่องเซิร์ฟเวอร์หรือข้อมูลจากรีเลชันแนลดาต้าเบส OLEDB, โลคอลคิวบ์สามารถเก็บข้อมูลที่เป็นแฟ้มข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คิวบ์แบบมัลติไดเมนชันแนลบนเครื่องไคลเอ็นต์และยังสามารถใช้การวิเคราะห์แบบออฟไลน์ (Off-line) ได้ นั่นคือถ้าโลคอลมีการเก็บข้อมูลแบบ Multidimensional OLAP (MOLAP) ก็จะสามารถทำได้โดยไม่ต้องทำการเชื่อมต่อไปยัง Analysis Server

PivotTable Service ถูกออกแบบสำหรับการวิเคราะห์ข้อมูลทั้งแบบออนไลน์และออฟไลน์ เมื่อมีการร้องขอมาจากเครื่องไคลเอ็นต์ทาง Analysis Server ก็จะทำการประมวลผลการร้องขอเหล่านั้นและสร้างเซตข้อมูลจากส่วนของข้อมูลที่มีการรวมค่าไว้ก่อนหรือจากการประมวลผลการรวมค่าใหม่ เมื่อทำเสร็จ Analysis Server ก็จะส่งเซตข้อมูลกลับไปให้ยังเครื่องไคลเอ็นต์ที่ร้องขอมา

เราสามารถพัฒนาโปรแกรมประยุกต์ไคล์เอ็นต์ซึ่งเรียกใช้ PivotTable Service ได้หลายวิธี นั่นคือสามารถใช้ Microsoft ActiveX Data Objects Multidimensional (ADO MD) สำหรับการพัฒนาโปรแกรมด้วยภาษาที่มีลักษณะเป็น Component Object Model (COM) Automation เช่น Visual Basic หรือ ASP หรืออีกวิธีก็คือการใช้ OLE DB สำหรับการพัฒนาโปรแกรมด้วยภาษา C++

6.2.1 ลักษณะเด่นของ PivotTable Service

- ความสามารถในการปรับปรุงประสิทธิภาพของโปรแกรมประยุกต์ไคลเอ็นต์ในการเข้าถึง ข้อมูลของ OLAP
- PivotTable Service จะถูกรวมเข้าไปใน Microsoft Excel 2000 ซึ่งจะทำให้ Excel กลายเป็นทูลส์ที่มีประสิทธิภาพในการนำเสนอข้อมูล OLAP
- สามารถสร้างคิวบ์ของฝั่งไคลเอ็นต์ได้ ซึ่งยอมให้นำไปวิเคราะห์บนเครื่อง คอมพิวเตอร์เครื่องใดก็ได้ ในขณะที่ไม่ได้เชื่อมต่อกับระบบเครือข่ายและแหล่งข้อมูล (Data Source) หลัก, คิวบ์ของฝั่งไคลเอ็นต์มักจะเก็บไว้ในไฟล์ คิวบ์บนเครื่องไคลเอ็นต์ไฟล์เหล่านี้สามารถเปิดและอ่านได้โดย OLAP
- เนื่องจาก PivotTable Service ถูกติดตั้งบนเครื่องไคลเอ็นต์ เช่น เมื่อทำการติดตั้ง Microsoft Excel ผู้พัฒนาโปรแกรมก็สามารถสร้างแอพพลิเคชัน โดยการเขียนโปรแกรมด้วยภาษาใด ๆ ก็ตามที่ถนัด เพื่อให้เข้าใช้ข้อมูล OLAP หรือแหล่งข้อมูลรีเลชันแนล โดยใช้เทคโนโลยีของ OLE DB
- PivotTable Service ทำงานกับ Analysis Service เหมือนกับเป็นส่วนหนึ่งของสถาปัตยกรรมของเซอร์วิสเหล่านี้ดังนั้น PivotTable Service จึงยอมให้หลาย ๆไคลเอ็นต์สามารถเข้าใช้คิวบ์อันเดียวกันได้ โดยแบ่งการประมวลผลตามผู้ใช้
- สนับสนุน Multidimensional Expressions (MDX) ซึ่งเป็นภาษาที่ใช้ในการคิวรีและอนุญาตให้ทำการเข้าใช้คิวบ์ที่เก็บข้อมูลไดเมนชันและส่วนอื่น ๆ ได้โดยใช้ MDX Statement
- PivotTable Service อนุญาตให้ทำการดาวน์โหลดข้อมูลจากแหล่งข้อมูลและเก็บข้อมูลในโครงสร้างแบบมัลติไดเมนชันแนลบนโลคอลคอมพิวเตอร์เพื่อใช้สำหรับการวิเคราะห์แบบออฟไลน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.2 คอมโพเนนท์ที่ใช้ใน PivotTable Service

ใน PivotTable Service ได้มีการรวม Dynamic-Link Libraries (DLLs) เอาไว้ดังนั้นที่เครื่องไคลเอ็นต์ก็จะต้องมีไฟล์เหล่านี้ซึ่งจะต้องมีไฟล์ใดบ้างก็ขึ้นอยู่กับแอพพลิเคชันที่ทำต้องการใช้คุณสมบัติใดของ PivotTable Service

File set	Component files
1	Msolap80.dll, Msolui80.dll, Msolap80.rll, Olapuir.rll, and Microsoft® Data Access Components (MDAC)
2	File Set 1 plus Msmdcb80.dll, Msmdgd80.dll, and an appropriate OLE DB tabular data provider
3	File Set 1 plus Msdmine.dll, Msmdun80.dll, Msdmine.rll, and Msdmeng.dll

ตารางที่ 6-1 แสดง file set ของ PivotTable Service

Task	File set
Communicate with the Analysis server using TCP/IP or HTTP and read local cube files	1
Create and refresh local cubes	2
Read OLAP and relational data mining models	3

ตารางที่ 6-2 แสดงไฟล์ที่จำเป็นสำหรับแต่ละงาน

6.2.3 การติดตั้งและลงทะเบียนคอมโพเนนท์

เราสามารถติดตั้งและลงทะเบียนคอมโพเนนท์ได้อย่างสะดวกโดยใช้โปรแกรมติดตั้งที่มีมาให้ในแผ่น CD ของ SQL Server 2000 โดยจะอยู่ในไดเรคทอรี \Msolap\Install\Pts ซึ่งจะมีอยู่ 2 โปรแกรม คือ Ptslite.exe และ Ptsfull.exe โดย Ptslite.exe จะติดตั้งเฉพาะไฟล์ของ PivotTable Service แต่ถ้าเลือก Ptsfull.exe ก็จะติดตั้ง Microsoft Data Access Component (MDAC) ให้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ptslite.exe

เมื่อเรารัน Ptslite.exe โปรแกรมก็จะติดตั้งไฟล์เหล่านี้ให้

atl.dll	msdmeng.dll	msdmine.dll
msmdcb80.dll	mdmdgd80.dll	msolap80.dll
msolui80.dll	msmdcube.dll	msmdgdrv.dll
msolap.dll	msolapui.dll	msdmine.rll
msolap80.rll	olapuir.rll	Msvbvm60.dll
msmdun80.dll	msolapr.dll	

ตารางที่ 6-3 แสดงไฟล์ที่จะติดตั้งโดย Ptslite.exe

ถ้า Ptsfull.exe โปรแกรมจะติดตั้งไฟล์ที่ได้แสดงไว้ข้างต้น และจะติดตั้ง MDAC เพิ่มให้ด้วย

6.2.4 ลักษณะการเชื่อมต่อของ PivotTable Service

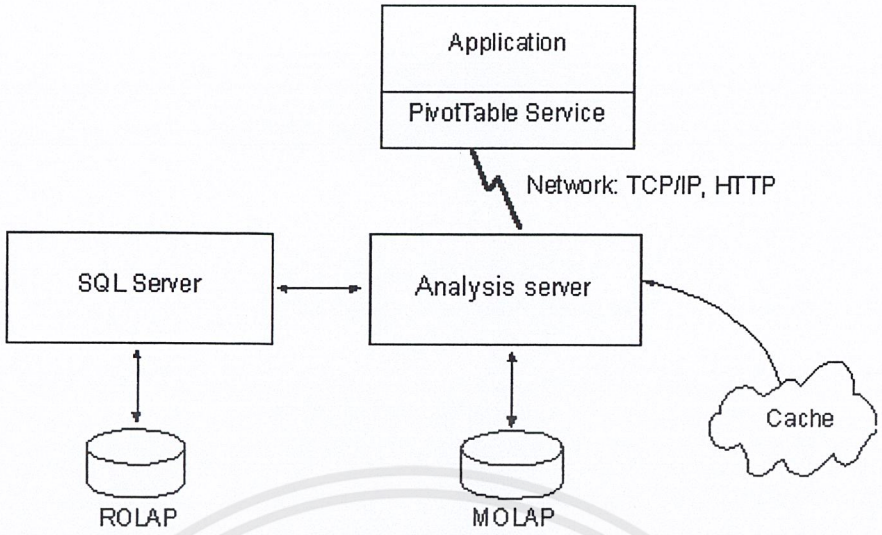
เราสามารถใช้ PivotTable Service เชื่อมต่อได้ 3 รูปแบบคือ

- การเชื่อมต่อกับ Microsoft SQL Server2000 Analysis Services
- เชื่อมต่อกับ OLE DB Provider
- เชื่อมต่อกับโลคอลคิวบ์

6.2.4.1 การเชื่อมต่อกับ SQL Server2000 Analysis Service

เราสามารถเชื่อมต่อกับ Analysis Server ได้หลายวิธี เช่น ใช้โพรโตคอล HTTP ในการเชื่อมต่อผ่านอินเตอร์เน็ต เมื่อเราใช้ PivotTable Service ในการแสดงข้อมูลจาก Analysis Service , PivotTable Service จะทำการติดต่อกับ Remote Analysis Server ผ่านทางเน็ตเวิร์คโดยใช้โพรโตคอล TCP/IP หรือ HTTP ดังรูป

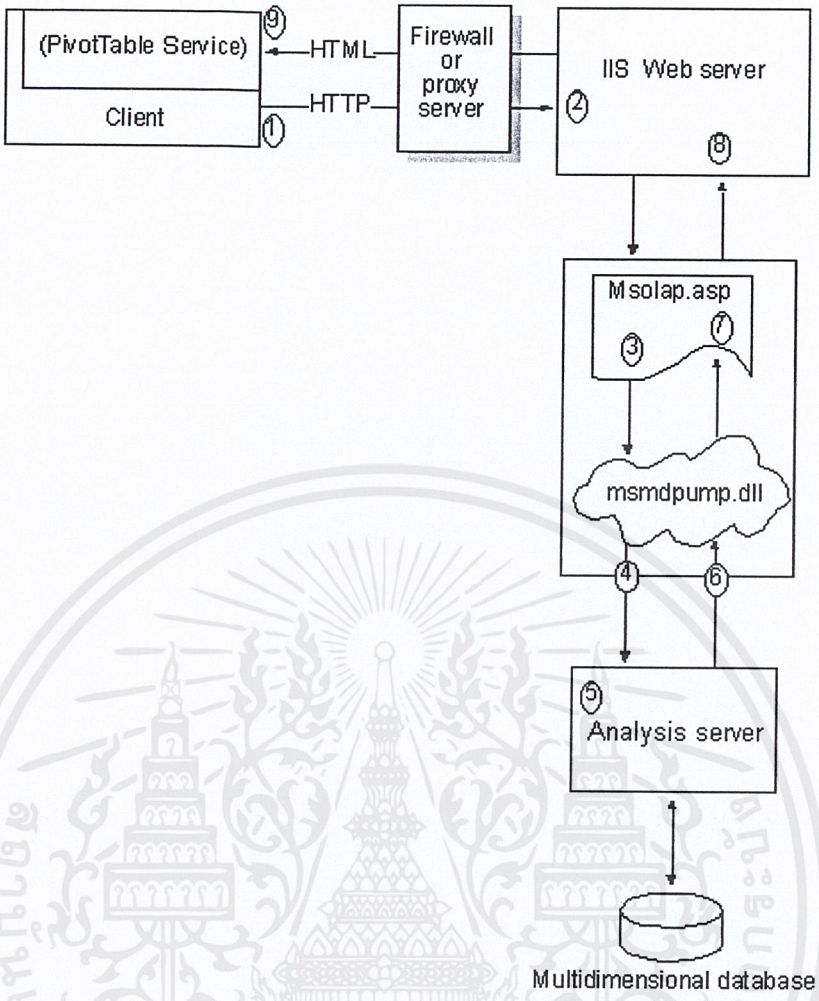
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-2 แสดงการเชื่อมต่อกับ Analysis server

การเชื่อมต่อโดยใช้โพรโตคอล HTTP ทำให้ผู้ใช้งานทั่วไปสามารถติดต่อกับ Analysis server โดยผ่าน Microsoft Internet Information Service (IIS) ได้ โดยการกำหนดคุณสมบัติของแหล่งข้อมูลใน connection string ให้เป็น HTTP หรือ HTTPS URL (ซึ่งจะมีระดับความปลอดภัยที่สูงกว่า) ลักษณะการเชื่อมต่อจะเป็นดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-3 แสดงการเชื่อมต่อโดยใช้ HTTP

ตัวอย่างของโปรแกรมที่ทำการติดต่อในรูปแบบนี้ แสดงได้ดัง

```
Dim cat as new ADOMD.Catalog
cat.ActiveConnection = "Provider = msolap;" & _
    " Datasource =" _ &
    " http://<URL>/;" & _
    " Initial Catalog = FoodMart 2000"
```

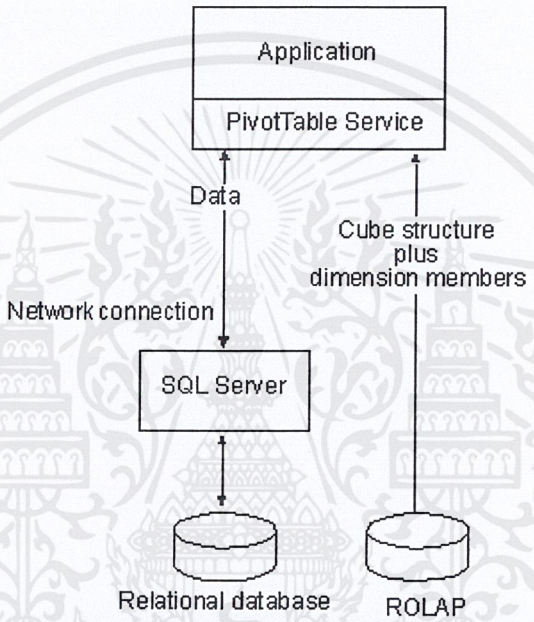
ถ้าต้องการให้มีการใช้ SSL (Secure Socket Layer) ด้วยก็สามารถทำได้ดังนี้

```
Dim cat as new ADOMD.Catalog
cat.ActiveConnection = "Provider = msolap; Datasource =" & _
  " https://<URL>/;" & _
  " Initial Catalog = FoodMart 2000"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.4.2 การเชื่อมต่อกับ OLE DB Provider

เมื่อต้องการทำงานกับโลคอลคิวบ์ที่เป็น Relational OLAP (ROLAP) เราต้องทำการเชื่อมต่อกับ Relational data provider โดยโลคอลคิวบ์ในลักษณะนี้จะเก็บนิยามโครงสร้างของคิวบ์แต่จะไม่ได้เก็บข้อมูลจริงๆ หรือข้อมูลที่ทำการคำนวณไว้ล่วงหน้า ดังนั้นเมื่อเราต้องการดึงข้อมูลจึงจำเป็นต้องใช้การติดต่อกับ tabular data provider โดยขบวนการเหล่านี้ไคลเอ็นต์แอพพลิชันไม่จำเป็นต้องรับรู้ ผลจากการทำงานในลักษณะนี้ทำให้เราจะได้คิวบ์ที่มีขนาดเล็กกว่าคิวบ์ซึ่งสร้างโดย MOLAP แต่ก็การทำงานก็จะมีประสิทธิภาพน้อยกว่า local cube แบบ MOLAP ด้วยเช่นกัน ลักษณะการเชื่อมต่อจะเป็นดังรูป

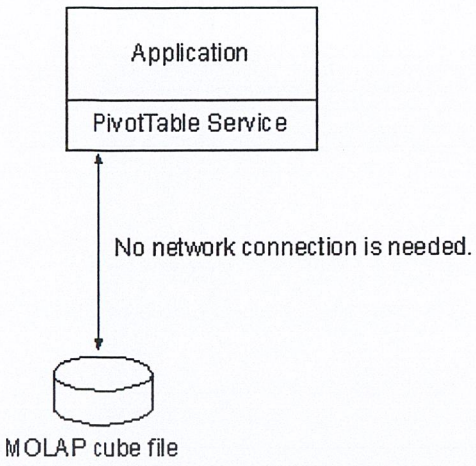


รูปที่ 6-4 แสดงการเชื่อมต่อกับ OLE DB Provider

6.2.4.3 การเชื่อมต่อกับ โลคอลคิวบ์ไฟล์

ในการแสดงข้อมูลจากคิวบ์ MOLAP, PivotTable Service จะเชื่อมต่อกับโลคอลคิวบ์ในแบบเดียวกับที่เชื่อมต่อกับแหล่งข้อมูลอื่น ๆ โดย PivotTable Service จะประมวลผลจากการคิวรี แล้วส่งค่าที่ได้คืนกลับไปยังโปรแกรมประยุกต์ ซึ่งโปรแกรมประยุกต์ทางฝั่งไคลเอ็นต์สามารถเข้าถึงไดเมนชัน, เลเวล (level), พร็อพเพอร์ตี้ ได้โดยที่ไม่ต้องทำการเชื่อมต่อกับรีโมทเซิร์ฟเวอร์ (Remote Server)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-5 แสดงการเชื่อมต่อกับ ไฟล์โลคอลคิวบ์

6.2.5 การทำงานของไคลเอ็นต์ใน PivotTable Service

- การเชื่อมต่อกับแหล่งข้อมูล

วิธีการเบื้องต้นในการเชื่อมต่อแหล่งข้อมูลคือการใช้ออปเจ็กต์ (Object) **Connection** หรือใช้คุณสมบัติ **ActiveConnection** ของออปเจ็กต์ **Catalog** โดยเราสามารถกำหนดพารามิเตอร์ของการเชื่อมต่อได้โดยกำหนดใน **Connection String**

ADO Connection Object

เมธอด (Method) **Open** ของออปเจ็กต์ **Connection** ได้รวมเอาพารามิเตอร์สำหรับใช้ในการเชื่อมต่อเข้าไปใน **connectionstring Property** เมื่อเมธอดนี้ถูกสั่งให้ทำงานก็จะทำการเชื่อมต่อกับแหล่งข้อมูลที่กำหนดในเมธอด Open : connect. Open connectionstring, UserId, Password , OpenOptions

ในการเชื่อมต่อกับ SQL Server 2000 Analysis Service นั้น **Datasource** ต้องตั้งชื่อหรือ IP address ของ Analysis server ที่ต้องการจะเชื่อมต่อ ส่วน **Provider** ต้องกำหนดเป็น "MSOLAP" ส่วน **Initial Catalog** ให้ระบุฐานข้อมูลบนเซิร์ฟเวอร์ที่ต้องการจะติดต่อ

- การแสดงโครงสร้างของข้อมูล (Retrieving Schema Information)

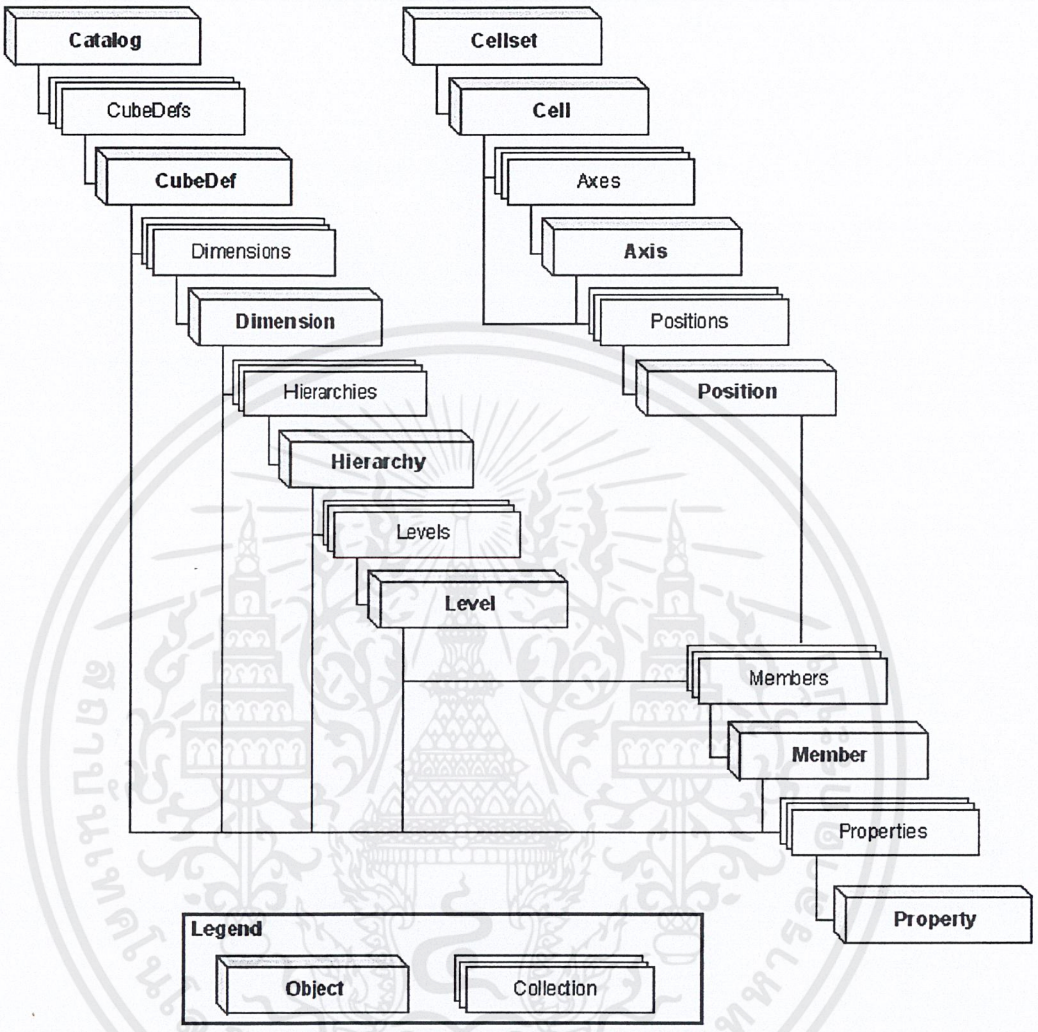
สามารถใช้ Microsoft ActiveX Data Objects (Multidimensional)(ADO MD), ADO หรือ OLE ในการแสดงโครงสร้างของสกีมา (schema rowsets) โดยใช้ PivotTable Service

ในการแสดงโครงสร้างของข้อมูลของคิวบ์จะใช้ออปเจ็กต์ **Cubedef** ใน ADO MD หรือใช้เมธอด **OpenSchema** ใน ADO , ออปเจกต์ **CubeDef** จะมีโครงสร้างคิวบ์เป็นแบบลำดับขั้น (hierarchy) จะมีบางข้อมูลเกี่ยวกับคิวบ์ที่ไม่ได้เก็บอยู่ในออปเจกต์ **CubeDef** เช่นการกำหนดแอคชัน (actions) และการจัดรูปแบบเซลล์ (cell formulas) ซึ่งจะต้องใช้เมธอด **OpenSchema** ในการแสดงข้อมูลเหล่านี้

ในการแสดงโครงสร้างของสกีมาจะใช้ ADO หรือ OLE DB สำหรับใน ADO จะใช้เมธอด **OpenSchema** ของออปเจ็กต์ **Connection** ส่วนใน OLE DB จะใช้อินเทอร์เฟส IDBchemaRowset COM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออปเจ็กต์ **CubeDef** จะเก็บข้อมูลไดเมนชันของโลคอลคิวบ์โดยใช้คอลเล็คชัน (collection) **Dimensions** ของตัวมันเอง ซึ่ง **Dimensions** มีรูปแบบเป็นคอลเล็คชันของ **Hierarchies** แสดงดังรูปที่ 6-6



รูปที่ 6-6 แสดงโครงสร้างลำดับขั้น

- การแสดงข้อมูล (Retrieving Data)

เราสามารถแสดงข้อมูลทำได้โดยใช้ออปเจ็กต์ **Cellset** ของ ADOMD หรือใช้ **Recordset** ของ ADODB เราสามารถใช้ ADOMD ในการแสดงผลลัพธ์ของการคิวรีด้วย MDX จากโลคอลคิวบ์โดยใช้ออปเจ็กต์ **Cellset** ได้ ส่วนการแสดงผลลัพธ์ที่มีลักษณะเป็นตารางจะใช้ ออปเจ็กต์ **Command** และ **Recordset**

<u>ตัวอย่างการใช้ Cellset</u>

จะใช้ออปเจ็กต์ **Connection** ในการเชื่อมต่อกับ Analysis Server, ค่าของพร็อพเพอร์ตี้ **Source** ของ **Cellset** จะอยู่ในรูปแบบของ MDX query, พร็อพเพอร์ตี้ **ActiveConnection** ของ **Cellset** จะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดเหมือนกับพร็อพเพอร์ตี้ **ActiveConnection** ของออปเจ็กต์ **Connection** และมีการใช้เมธอด Open ในการแสดงผลลัพธ์ที่ได้

ออปเจ็กต์ **Cellset** จะเก็บคอลเล็คชันที่เรียกว่า **Axes** ซึ่งจะใช้อธิบายในแต่ละแกน จะมีออปเจ็กต์ **Axis** 1 ตัวในคอลเล็คชันนี้สำหรับแต่ละไดเมนชันที่ต้องการ แต่ละออปเจ็กต์ **Axis** จะเก็บคอลเล็คชันของ **Positions** ซึ่งเก็บข้อมูลในแต่ละแถว (row), คอลัมน์ (column), หน้า (page) และเซตของผลลัพธ์

```
Dim conn As New ADODB.Connection
Dim cst As New ADOMD.Cellset
Dim axs As ADOMD.Axis
Dim pos As ADOMD.Position
Dim iCol As Integer, cCol As Integer
Dim iRow As Integer, cRow As Integer
Dim nFixedCols As Integer, nFixedRows As Integer

  'Set up the connection to the server.
  conn.ConnectionString = "Datasource=LocalHost; Provider=msolap; Initial Catalog=FoodMart
2000;"
  conn.Open
  Set cst.ActiveConnection = conn ' You must use Set.
  cst.Source = "Select CrossJoin [Product].[Product Family].Members, " & _
    "[Promotion Media].Members) on rows," & _
    "[Measures].Members on Columns " & _
    "From Sales"
  cst.Open

  'Set up the FlexGrid control.
  MSFlexGrid1.Clear
  nFixedCols = 2
  nFixedRows = 1
  cCol = cst.Axes(0).Positions.Count
  MSFlexGrid1.Cols = cCol + nFixedCols
  cRow = cst.Axes(1).Positions.Count
  MSFlexGrid1.Rows = cRow + nFixedRows
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MSFlexGrid1.FixedCols = nFixedCols
MSFlexGrid1.FixedRows = nFixedRows
   MSFlexGrid1.MergeCol(0) = True
MSFlexGrid1.MergeCol(1) = True

'Add column headers.
iCol = 2
For Each pos In cst.Axes(0).Positions
   'The caption for each member is used as the header.
   MSFlexGrid1.TextMatrix(0, iCol) = pos.Members(0).Caption
   iCol = iCol + 1
Next

'Add row headers.
iRow = 1
For Each pos In cst.Axes(1).Positions
   'The CrossJoin function in MDX indicates that this axis will have two members per position.
   MSFlexGrid1.TextMatrix(iRow, 0) = pos.Members(0).Caption
   MSFlexGrid1.TextMatrix(iRow, 1) = pos.Members(1).Caption
   iRow = iRow + 1
Next

'Iterate through the cellset array values.
For iCol = 0 To cCol - 1
   For iRow = 0 To cRow - 1
      ' Retrieve each value with the default method of the cst object.
      MSFlexGrid1.TextMatrix(iRow + nFixedRows, iCol + nFixedCols) = cst(iCol, iRow).Value
   Next
Next
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

การออกแบบและพัฒนาโปรแกรมประยุกต์ทำงานบนแลน (LAN)

โปรแกรมประยุกต์ที่ต้องการพัฒนาขึ้นมาจะมีลักษณะ รันอยู่บนเครื่องฝั่งไคลเอ็นต์โดยมีหน้าที่การทำงานของโปรแกรมคือ สามารถเรียกดูและใช้งานข้อมูลที่อยู่ภายในคิวบ์ซึ่งอยู่ที่เครื่อง Analysis Server ได้

7.1 จุดประสงค์ในการพัฒนาโปรแกรม

7.1.1 เพื่อให้ผู้ใช้สามารถเรียกใช้งานระบบฐานข้อมูลดาต้าแวร์เฮาส์ที่อยู่บนฝั่งซิร์ฟเวอร์จากเครื่องของตนเองซึ่งไม่ได้ลงโปรแกรม Analysis Service ได้

7.1.2 เพื่อทดสอบประสิทธิภาพและเวลาตอบสนอง (response time) เมื่อเรียกใช้งานระบบฐานข้อมูลดาต้าแวร์เฮาส์ผ่านทางเครื่องไคลเอ็นต์

7.2 ทูลส์ที่นำมาใช้ในการพัฒนาโปรแกรม

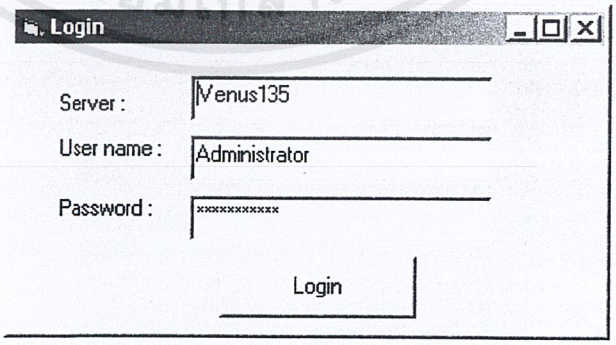
ในการพัฒนาโปรแกรมประยุกต์นี้ ได้เลือกใช้ Visual Basic เวอร์ชัน 6.0 มาเป็นทูลส์ในการพัฒนา เนื่องจากเป็นทูลส์ที่ใช้งานง่ายและเนื่องจากเป็นผลิตภัณฑ์ของ Microsoft เองด้วยจึงมีการเข้ากันได้ดีกับ Microsoft SQL2000 Server Analysis Service และอีกประการหนึ่งที่เลือกใช้ทูลส์ตัวนี้คือเป็นทูลส์ที่มีความโดดเด่นทางด้านคอมโพเนนต์ด้วย

7.3 การออกแบบโปรแกรมประยุกต์

โดยหลักการของดาต้าแวร์เฮาส์แล้ว การเรียกใช้งานจะมุ่งเน้นไปที่การเรียกดูข้อมูลเพื่อการวิเคราะห์เพื่อให้ได้มาซึ่งคำตอบที่ต้องการจะนำไปใช้ตัดสินใจอย่างรวดเร็ว ดังนั้นโปรแกรมที่พัฒนาจึงไม่ได้มุ่งไปที่การอัพเดตข้อมูล และได้ออกแบบมาอย่างเรียบง่ายเพื่อให้ง่ายต่อการทำความเข้าใจและใช้งานได้อย่างสะดวก

โปรแกรมที่ได้ทำการออกแบบขึ้นมาจะมีฟอร์มอยู่ด้วยกัน 2 ฟอร์ม คือ

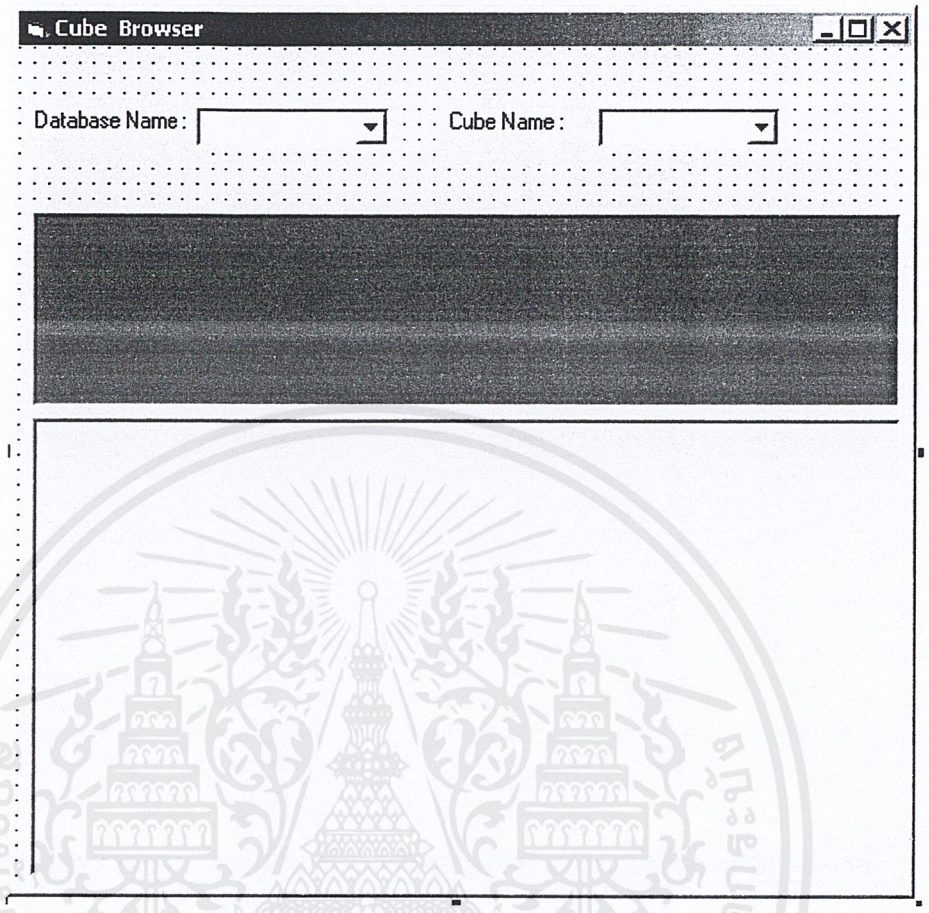
- ฟอร์มที่ใช้การติดต่อกับ Analysis Server



รูปที่ 7-1 แสดงฟอร์มเริ่มแรกของโปรแกรมประยุกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฟอร์มสำหรับเรียกดูและใช้งานข้อมูล



รูปที่ 7-2 แสดงฟอร์มการทำงานหลักของโปรแกรม

ฟอร์มนี้จะประกอบไปด้วยคอมโบบ็อกซ์ (Combo Box) 2 ช่อง และมีเพลน (Plain) สำหรับแสดงผลอีก 2 เพลน โดยเพลนด้านบนจะแสดงรายละเอียดของไดเมนชันต่างๆ ที่มีในคิวบ์ที่ผู้ใช้เลือก ส่วนในเพลนด้านล่างจะใช้สำหรับแสดงผลลัพธ์จากการคิวรีของผู้ใช้

7.4 คอมโพเนนต์หลักที่ใช้ในการพัฒนาโปรแกรม

คอมโพเนนต์ที่ใช้พัฒนาโปรแกรมประกอบไปด้วย

Button ปุ่มกด เมื่อมีการกดปุ่มโปรแกรมจะไปทำงานตามขั้นตอนที่กำหนดไว้

Text Box เป็นช่องใช้สำหรับให้ผู้ใช้กรอกชื่อเซิร์ฟเวอร์ที่ต้องการติดต่อ

Combo Box จะเป็นลักษณะของ drop down list ข้อมูลที่เป็นไปได้ทั้งหมดประกอบอยู่ในนี้แล้ว ผู้ใช้เพียงแต่เลือกอันใดอันหนึ่งเท่านั้น

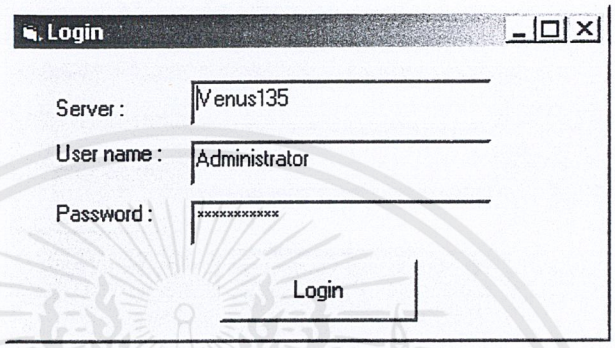
Cube Browser เป็นคอมโพเนนต์ที่ไม่ใช่คอมโพเนนต์พื้นฐานของวิชวลเบสิค โดยเมื่อเราต้องการจะใช้งานเราจะต้องเพิ่มคอมโพเนนต์นี้เข้ามาก่อน ซึ่งคอมโพเนนต์นี้จะมีให้หลังจากเราได้ลงโปรแกรม Microsoft SQL2000 Analysis Service, Cube Browser เป็นคอมโพเนนต์ที่ทำให้เราสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนาโปรแกรมได้อย่างง่าย สะดวกและรวดเร็ว เนื่องจากเราเพียงแค่กำหนดพารามิเตอร์ที่จำเป็นในการเชื่อมต่อกับ Analysis Server แค่นั้นก็พอ หลังจากนั้นตัวคอมโพเนนต์จะไปเชื่อมต่อและดึงข้อมูลจาก Analysis Server มาให้ ซึ่งลักษณะการทำงานของโปรแกรมจะเหมือนกันเราทำการดูข้อมูลในคิวบ์ ในโปรแกรม Analysis manager เลยสามารถ drill up, drill down, slice และเลือกไดเมนชันที่ต้องการให้แสดงผลได้

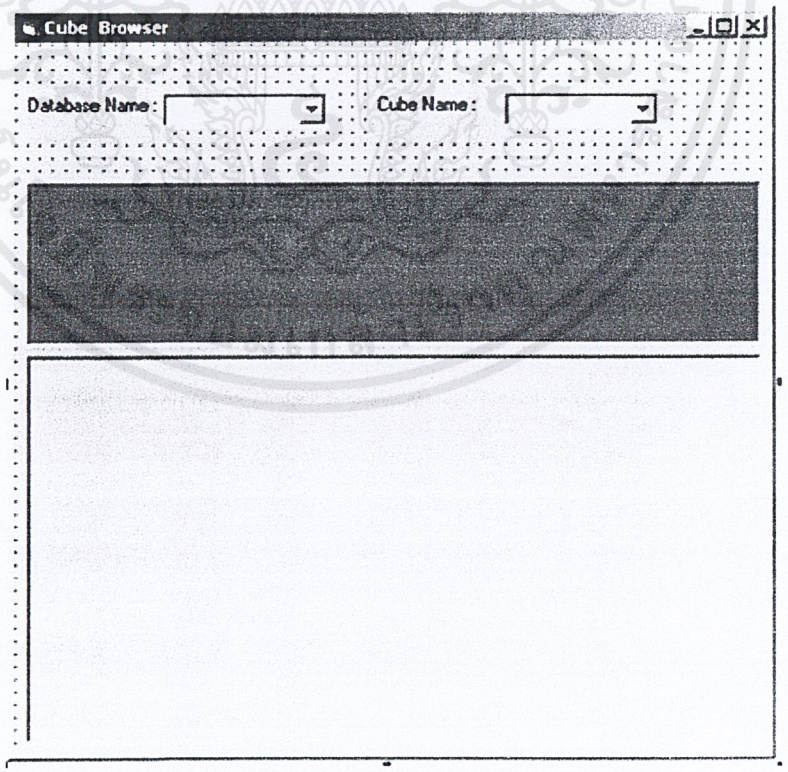
7.5 การใช้งานโปรแกรมประยุกต์

เริ่มแรกเมื่อรันโปรแกรมจะปรากฏฟอร์มดังรูปขึ้นมา



รูปที่ 7-3 แสดงฟอร์ม Login

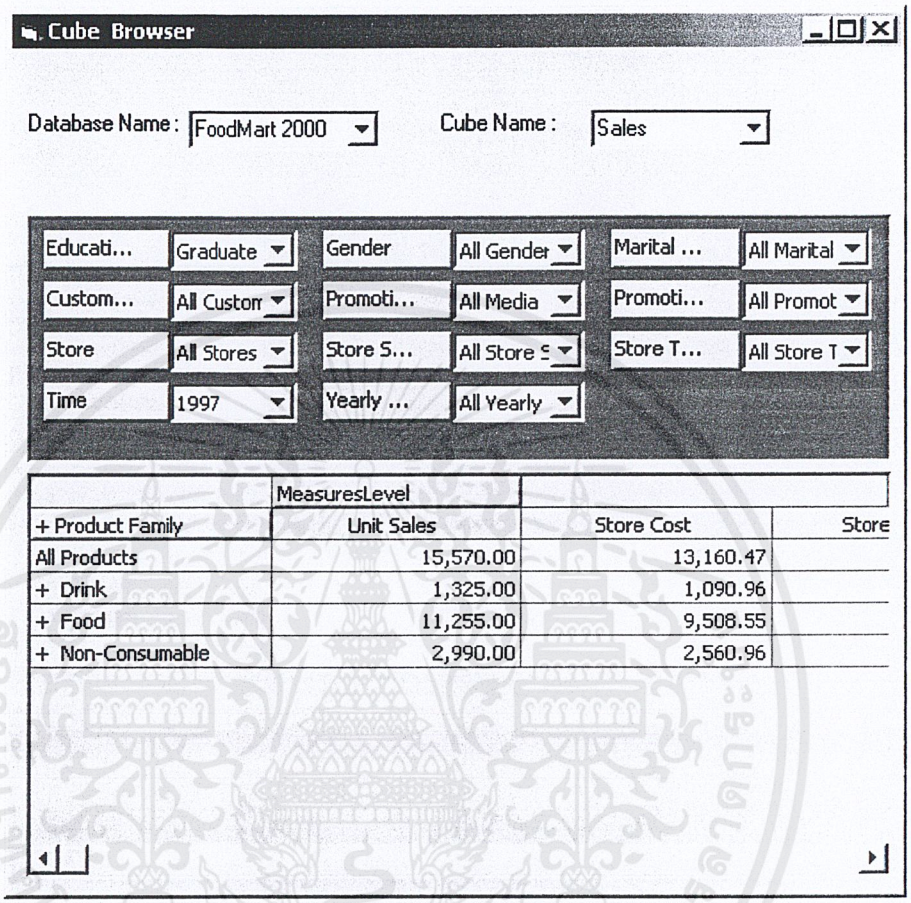
สิ่งที่ผู้ใช้ต้องทำคือ กรอกชื่อ Analysis Server ที่ต้องการติดต่อ และกรอก User name และ Password หลังกรอกรายละเอียดต่างๆ เรียบร้อยแล้ว หากข้อมูลที่ใส่ไปนั้นถูกต้อง โปรแกรมก็จะไปเชื่อมต่อกับเซิร์ฟเวอร์เครื่องนั้นและแสดงฟอร์มดังรูปที่ 7-4



รูปที่ 7-4 แสดงฟอร์มหลักในการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

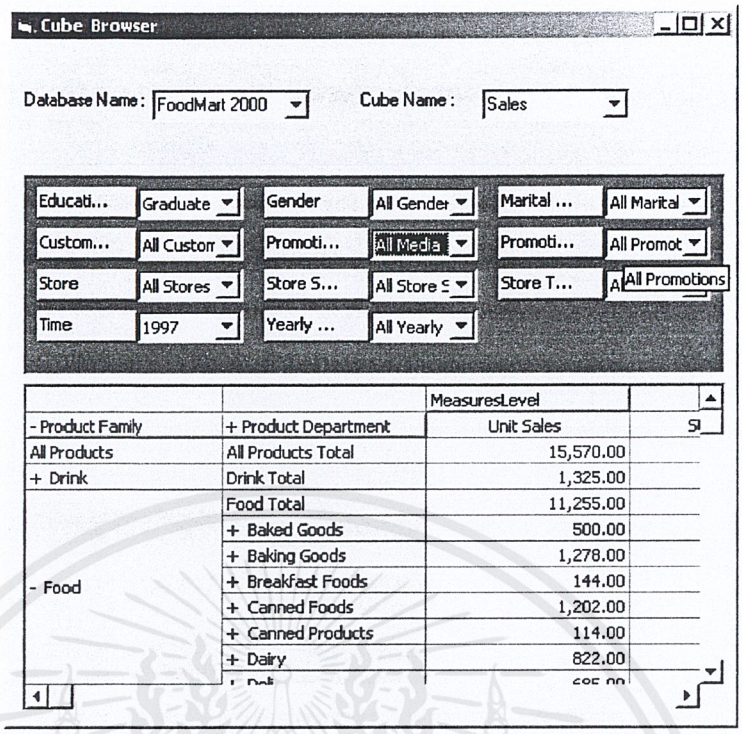
โดยในช่อง Database name เมื่อคลิกที่ปุ่มลูกศร โปรแกรมจะแสดงรายชื่อฐานข้อมูลทั้งหมดที่มีใน Analysis Server นี้ หลังจากเราเลือกฐานข้อมูลเรียบร้อยแล้ว ในช่อง Cube name ก็จะปรากฏชื่อคิวบ์ทั้งหมดที่มีในฐานข้อมูลนี้ และหลังจากเลือกคิวบ์แล้ว ข้อมูลจากคิวบ์ที่เลือกก็จะปรากฏออกมา ดังรูปที่ 7-5



รูปที่ 7-5 การแสดงผลข้อมูลที่อยู่ในคิวบ์ของโปรแกรมประยุกต์

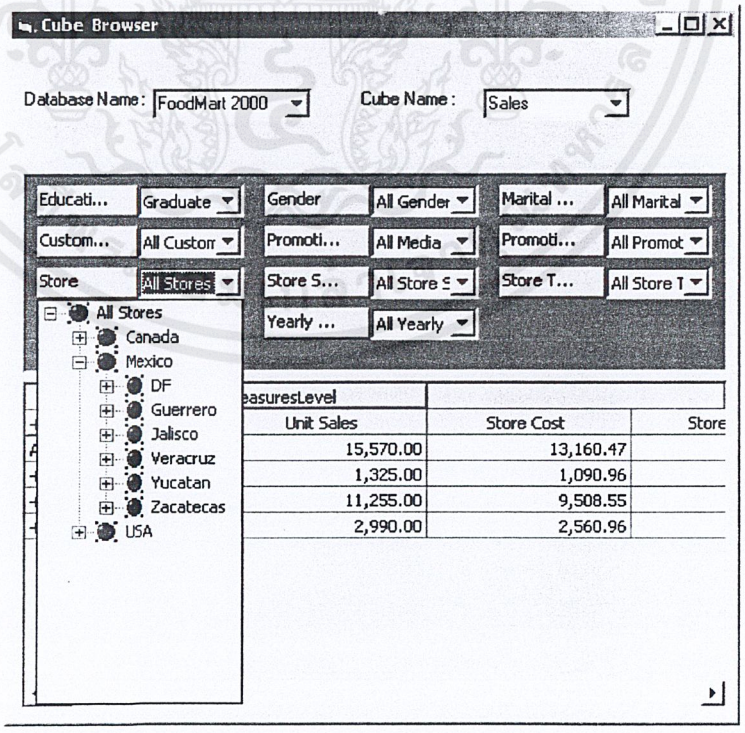
จากนั้นเราก็สามารถใช้งานโปรแกรมนี้ได้เหมือนกับเหมือนกับทำ Browse data ในโปรแกรม Analysis manager ทุกอย่าง เช่น การ drill down จะสังเกตเห็นว่าบางแถวมีเครื่องหมาย + อยู่ข้างหน้านั่นคือ สามารถ drill down ดูข้อมูลในระดับที่ลึกลงไปได้โดยการดับเบิลคลิก ซึ่งจะได้ผลดังรูปที่ 7-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-6 แสดงผลการ drill down

จากรูปที่ 7-6 จะเห็นว่าเราจะเห็นรายละเอียดในระดับที่ลึกลงไปของแถว Food การ slice ดูบางส่วนของข้อมูลก็สามารถทำได้โดยการคลิกที่ปุ่มลูกศรของไดเมนชันใดๆ ที่อยู่ในไดเมนชันเพลน ซึ่งเมื่อคลิกแล้วโปรแกรมจะแสดงรายละเอียดในเลเวลต่างๆ ของไดเมนชันนั้น โดยผู้ใช้สามารถเลือก slice ดูที่เลเวลใดก็ได้ แสดงได้ดังรูปที่ 7-7

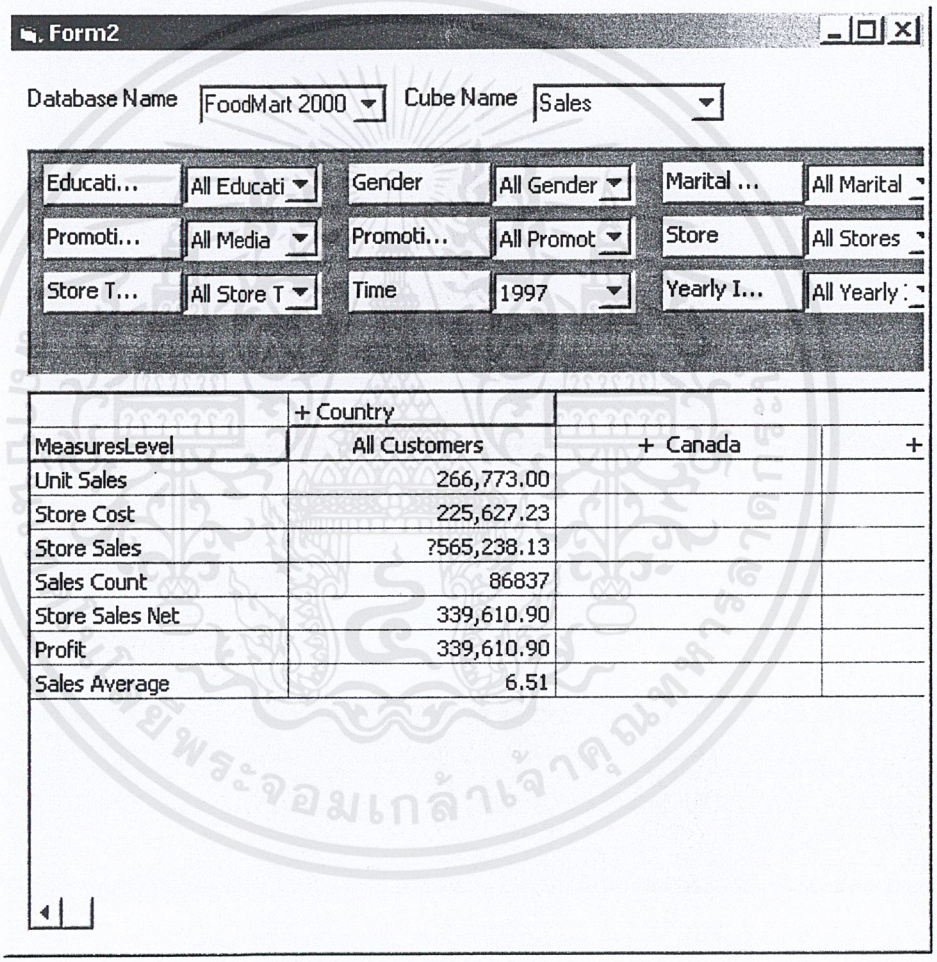


รูปที่ 7-7 แสดงภาพการทำ slice ข้อมูลของโปรแกรมประยุกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำ drill-up ก็จะสามารถทำได้ในลักษณะที่คล้ายกับการทำ drill-down โดยการ drill-up สามารถทำได้โดยการดับเบิลคลิกตรงช่องที่มีเครื่องหมาย – อยู่ข้างหน้า โปรแกรมก็จะรวมช่องแสดงผลย่อยขึ้นมาแสดงรวมในช่องเดียว

การทำ dice สามารถทำได้โดยการคลิกที่ชื่อไดเมนชันในไดเมนชันเพลนแล้วลากมาวางยังตารางแสดงผล ซึ่งจะเป็นการเพิ่มไดเมนชันเข้าไปในตารางหรือเป็นการสลับกับไดเมนชันที่มีอยู่แล้วในตารางก็ขึ้นอยู่กับตำแหน่งที่วางลงไป อีกทั้งยังสามารถสลับการแสดงผลระหว่างแถวกับคอลัมน์ของไดเมนชันในตารางได้ เช่น จากรูปที่ 7-5 เมื่อเราลาก MeasureLevel ซึ่งอยู่ในตาราง มาวางแทนตำแหน่งของ Product Family จะได้ผลลัพธ์ดังรูปที่ 7-8



รูปที่ 7-8 แสดงการสลับการแสดงผลระหว่างแถวกับคอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

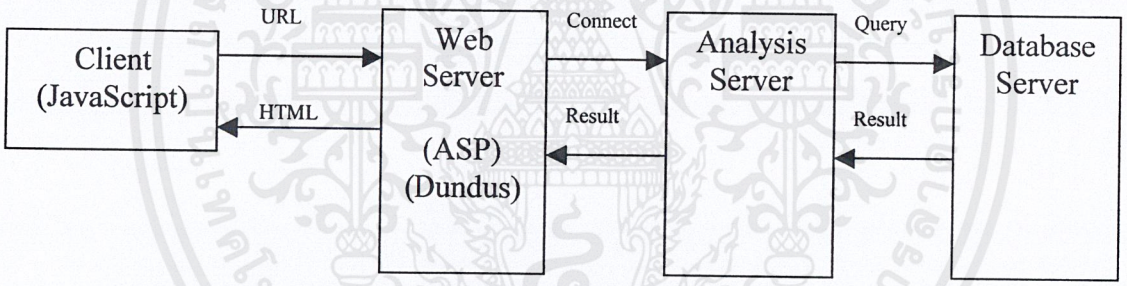
บทที่ 8
การออกแบบและพัฒนาโปรแกรมประยุกต์ที่ทำงานผ่านเว็บบราวเซอร์ (Web Browser)

8.1 จุดประสงค์ในการพัฒนาโปรแกรม

การพัฒนาโปรแกรมประยุกต์ที่สามารถทำงานผ่านเว็บบราวเซอร์นั้น มีจุดประสงค์หลักคือเพื่อให้ผู้ใช้สามารถเรียกใช้งานระบบฐานข้อมูลดาต้าแวร์เฮาส์จากระยะไกลได้ ในบางครั้งผู้ใช้อาจต้องเดินทางไปยังที่ต่างๆ และมีความจำเป็นที่จะต้องเข้าถึงข้อมูลอย่างเร่งด่วนเพื่อใช้ในการตัดสินใจ ดังนั้นโปรแกรมที่พัฒนาขึ้นมานี้สามารถนำไปใช้ในสถานการณ์เช่นนี้ได้ เป็นการเพิ่มความยืดหยุ่นและความสะดวกในการใช้งานระบบฐานข้อมูลดาต้าแวร์เฮาส์อีกทางหนึ่ง

8.2 ทูลส์ที่นำมาใช้ในการพัฒนาโปรแกรม

การพัฒนาโปรแกรมประยุกต์บนเว็บนี้ เราได้นำภาษาสคริปต์(Script) 2 ภาษา คือ Active Server Page (ASP) และ JavaScript มาใช้ในการพัฒนา โดยโครงสร้างการทำงานของโปรแกรมจะเป็นดังรูป ที่ 8-1



รูปที่ 8-1 สถาปัตยกรรมการทำงานของโปรแกรมประยุกต์ที่ทำงานผ่านเว็บบราวเซอร์

ASP เป็นสคริปต์ที่รันอยู่บนฝั่งเซิร์ฟเวอร์ โดยเป็นส่วนหลักในการเชื่อมต่อกับ Analysis Server , ส่งค่าคิวรี และรับผลลัพธ์ที่รีเทิร์นกลับมา จากนั้นก็นำผลลัพธ์ที่ได้มาเขียนในรูปแบบของ HTML เพื่อแสดงบนเว็บให้ผู้ใช้ทราบ

JavaScript เป็นสคริปต์ที่รันอยู่บนฝั่งไคลเอนต์ เราใช้ JavaScript เพื่อเป็นส่วนติดต่อกับผู้ใช้ (User Interface) เช่น การลากเมาส์มาวาง , การดับเบิลคลิกแล้วแสดงผลออกมา, การสร้างหน้าต่างบราวเซอร์ขึ้นมาใหม่ เป็นต้น เพื่อให้ผู้ใช้สามารถใช้งานโปรแกรมประยุกต์ได้อย่างสะดวก

นอกจากนี้โปรแกรมประยุกต์ที่เราได้ทำขึ้นมานั้นสามารถจัดแสดงข้อมูลในรูปแบบของกราฟหรือ Microsoft Excel ได้ด้วย ซึ่งคอมโพเนนท์ที่ใช้ในการวาดกราฟคือ Dundus Chart Evolution เป็นส่วนของกราฟคอมโพเนนท์ซึ่งจะทำงานบนฝั่งเซิร์ฟเวอร์

8.3 การติดตั้งโปรแกรมประยุกต์ที่ทำงานผ่านเว็บ

การติดตั้งมี 2 ส่วนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ก๊อปปี้โฟลเดอร์ **Webwarehouse** ไปไว้ยังโฮมไดเร็คทอรี (Home Directory) ซึ่งภายในโฟลเดอร์ Datawarehouse จะประกอบไปด้วยไฟล์ที่สำคัญ ดังต่อไปนี้

SelectServer.html	ใช้สำหรับให้ผู้ใช้ใส่ชื่อเซิร์ฟเวอร์ที่ต้องการติดต่อ
chooseDB.asp	ใช้สำหรับให้ผู้ใช้เลือกฐานข้อมูลที่จะทำงานด้วย
chooseCube.asp	ใช้สำหรับให้ผู้ใช้เลือกคิวบ์ที่ต้องการจะคิวรี
MDXQuery.asp	เป็นไฟล์หลักเอาไว้ติดต่อ Analysis Server, รับค่า MDX จากเครื่องฝั่งไคลเอนต์ ,ส่งค่า MDX ไปให้ Analysis Server และรับผลลัพธ์กลับมาแสดงผล
DimBrowser.asp	ทำหน้าที่แสดงไดเมนชันสำหรับการทำ Slice
CubeBrowser.js	ทำหน้าที่จัดการส่วนแสดงผล และจัดการส่วนการทำ Dice และ drill up, drill down
CubeAxisBrowser.js	ทำหน้าที่จัดการส่วนแสดงผล และจัดการส่วนการทำ Dice , drill up และ drill down ในส่วนของการกำหนดกราฟด้วยตัวเอง
ExcelReport.asp	ทำหน้าที่จัดการแสดงผลข้อมูลในรูปแบบของ Excel
MakeChart.asp	ทำหน้าที่จัดการแสดงผลข้อมูลในรูปกราฟ
MakeCustomChart.asp	ทำหน้าที่จัดการสร้างกราฟโดยผู้ใช้สามารถระบุแกน xแกน yได้
AxisXBrowser.asp	ส่วนเลือกข้อมูลว่าข้อมูลไหนมาแสดงบนแกน x
AxisYBrowser.asp	ส่วนเลือกข้อมูลว่าข้อมูลไหนมาแสดงบนแกน y
Styles.css, customstyles.css	กำหนดรูปแบบการแสดงผลบนเว็บ
Readme.txt	คำอธิบายการติดตั้งและวิธีการใช้งาน

- ติดตั้ง Dundas Graph 2D Evaulation ซึ่งเป็นคอมโพเนนท์ทำหน้าที่สร้างกราฟโดยแจกให้ดาวน์โหลดฟรี ให้ดับเบิลคลิกที่ตัวติดตั้ง (setup file) จากนั้นโปรแกรมก็จะติดตั้งคอมโพเนนท์นี้ลงยังเครื่อง หลังจากนั้นเราจะสามารถใช้งานกราฟได้

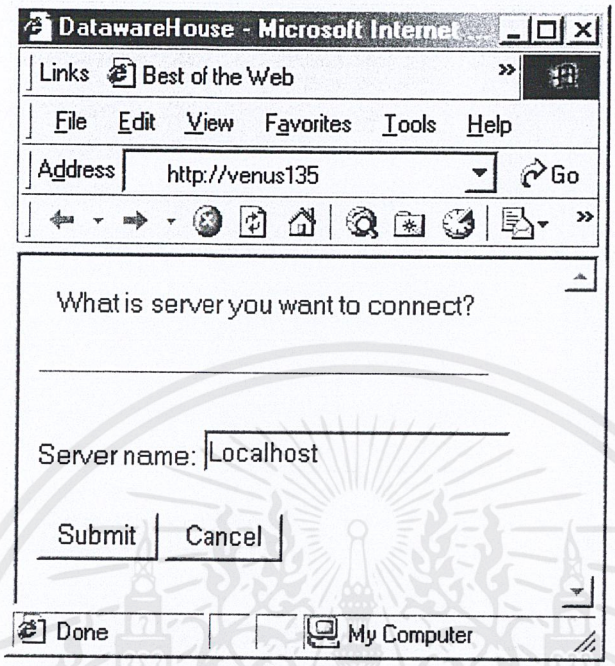
8.4 การใช้งานโปรแกรมประยุกต์ผ่านเว็บ

- เครื่องของผู้ใช้ทางฝั่งไคลเอนต์ เชื่อมต่อไปยังเครื่องเซิร์ฟเวอร์เพื่อรันไฟล์ SelectServer.html ผลลัพธ์จะออกมาเป็นดังรูปที่ 8-2 โดยผู้ใช้สามารถใส่ชื่อเซิร์ฟเวอร์ที่ต้องการได้แต่ถ้าไม่ใส่ค่าดีฟอลต์ จะเป็น Localhost เมื่อใส่ชื่อเซิร์ฟเวอร์แล้วให้กดปุ่ม Submit เพื่อไปยังขั้นตอนต่อไป หรืออาจกดปุ่ม Cancel เพื่อหยุดการทำงานและปิดหน้าต่างนี้

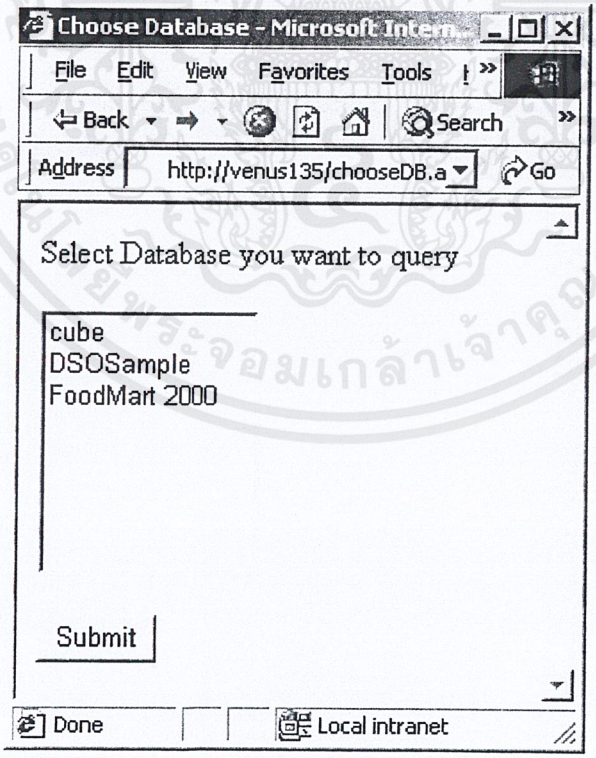
- เลือกฐานข้อมูลที่ต้องการจะทำงานด้วย โดยในช่องเมนูลิสต์ (Menu List) จะปรากฏรายชื่อของฐานข้อมูลทั้งหมดที่มีในเครื่องเซิร์ฟเวอร์ที่ได้เลือกไว้ก่อนหน้า แสดงได้ดังรูปที่ 8-3

- เลือกคิวบ์ที่ต้องการจะดูหรือวิเคราะห์ข้อมูล ในช่องเมนูลิสต์จะปรากฏรายชื่อคิวบ์ทั้งหมดที่อยู่ภายในฐานข้อมูลนั้นและผู้ใช้คนนี้มีสิทธิจะเรียกดูได้ แสดงได้ดังรูปที่ 8-4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

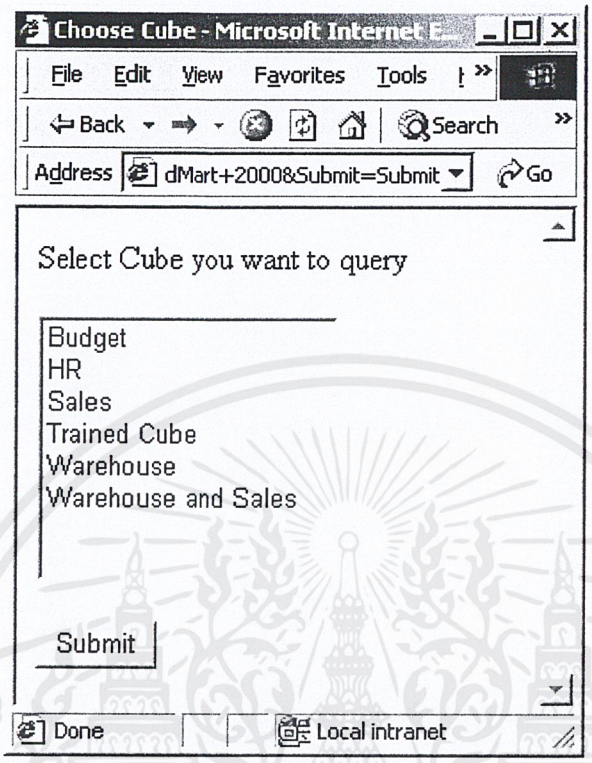


รูปที่ 8-2 แสดงหน้าจอเลือกเซิร์ฟเวอร์ที่ต้องการติดต่อด้วย



รูปที่ 8-3 แสดงหน้าจอ เลือกฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-4 แสดงหน้าจอเลือกคิวบ์

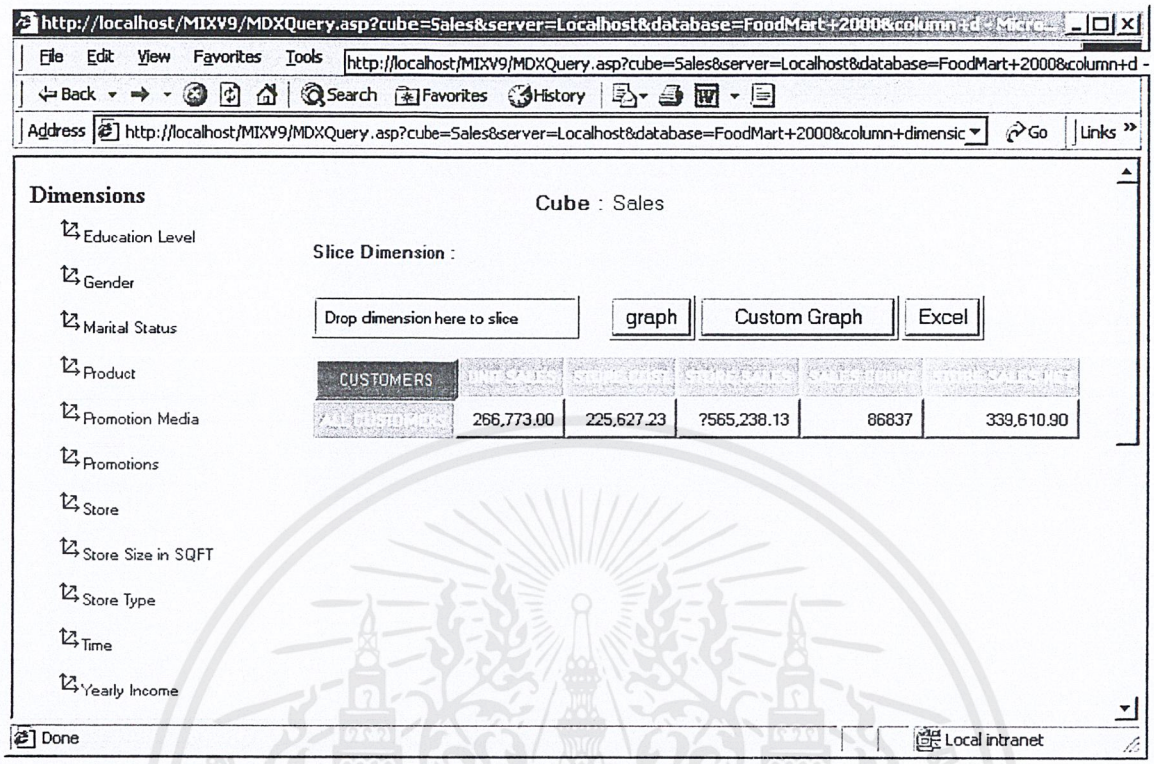
- หลังจากเลือกคิวบ์แล้ว ก็จะปรากฏหน้าจอการทำงานหลักของโปรแกรมดังรูปที่ 8-5 โดยเราสามารถทำโอเปอเรชันของดาต้าแวร์เฮาส์ไม่ว่าจะเป็น Slice ,Dice , Drill up และ Drill down ได้ นอกจากนี้เรายังสามารถดูข้อมูลในรูปแบบกราฟหรือ Excel ได้

- การ Dice

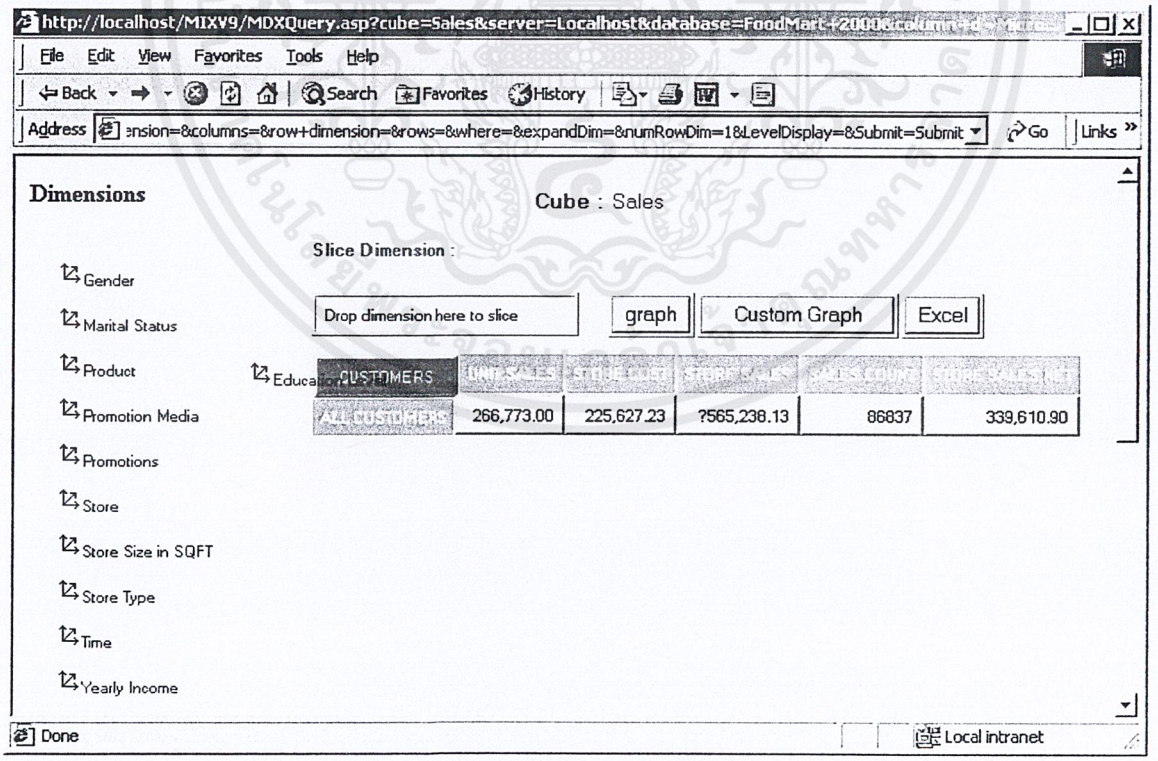
การเปลี่ยนแกนแสดงผลทำได้โดยลากไดเมนชันที่อยู่ทางซ้ายมือมาวางบนแกนแถวหรือแกนคอลัมน์ของตารางแสดงผล แสดงได้ดังรูปที่ 8-6 หลังจากลากไดเมนชันไปวางแล้วเราจะได้ผลลัพธ์ออกมาเป็นดังรูปที่ 8-7

นอกจากนี้หากเราต้องการแสดงผลไดเมนชันมากกว่า 1 ไดเมนชันในแกนเดียวกันก็สามารถทำได้โดยคลิกที่ชื่อไดเมนชันทางด้านซ้าย เราจะเห็นว่าโปรแกรมกางเลเวล (Level) ออกมา จากนั้นก็ให้ลากเลเวลไปไว้บนแกนแถวของตารางแสดงผล แสดงได้ดังรูปที่ 8-8 และ 8-9 ข้อจำกัดของการแสดงผลในลักษณะนี้คือสามารถแสดงผลมากกว่า 1 ไดเมนชันได้บนแกนแถวเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

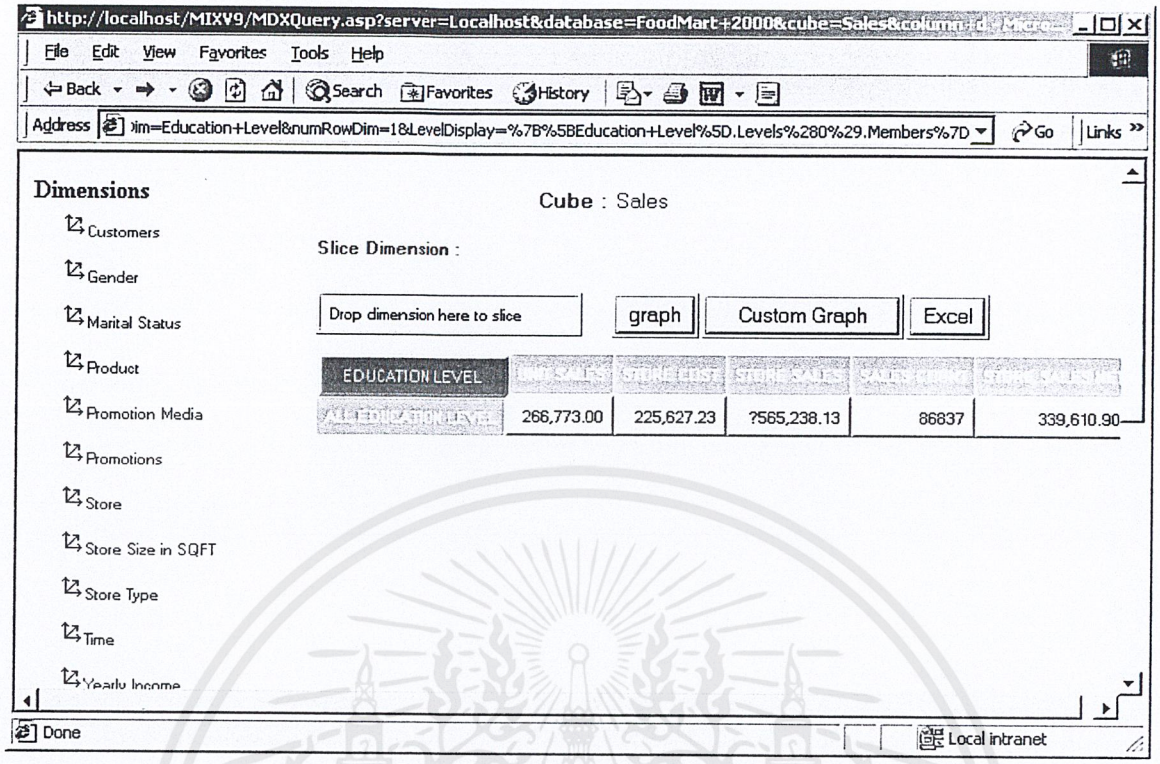


รูปที่ 8-5 แสดงหน้าจอการทำงานหลักของโปรแกรม



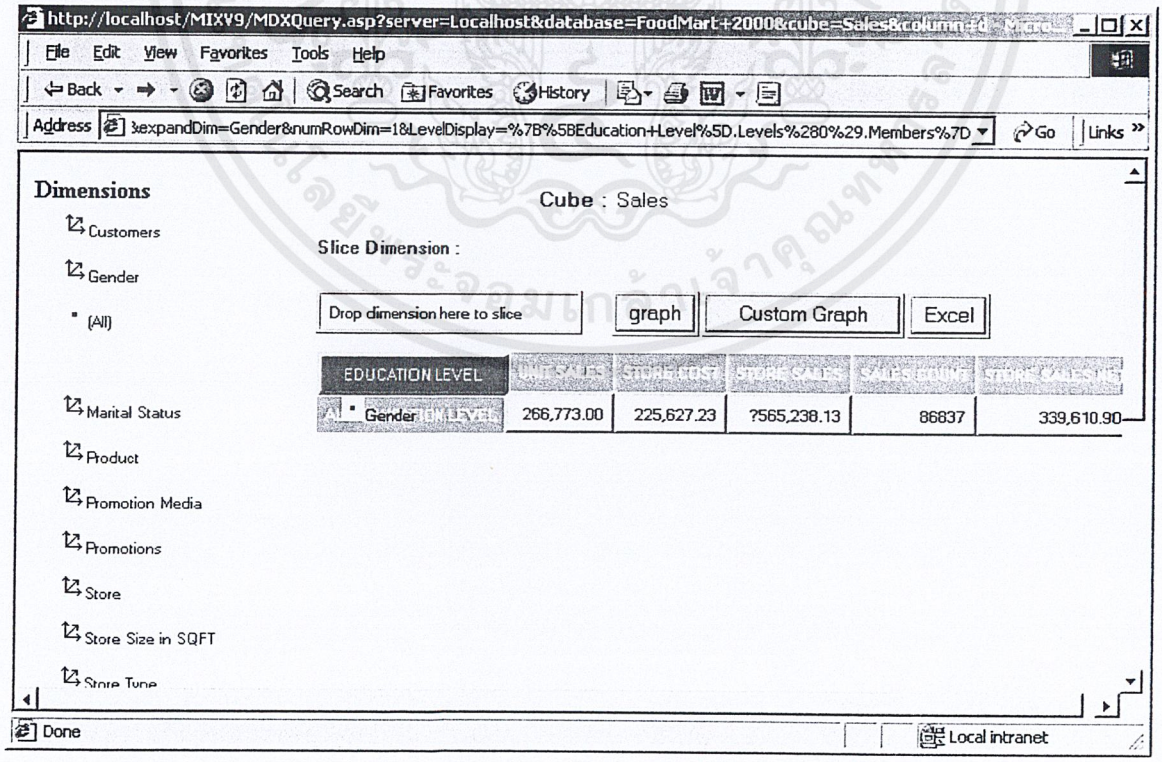
รูปที่ 8-6 แสดงการลากไดเมนชัน Education Level มาวางบนแกนแถว(Row)ของตารางแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-7 แสดงผลลัพธ์หลังจากลากไดเมนชัน Education Level มาวาง

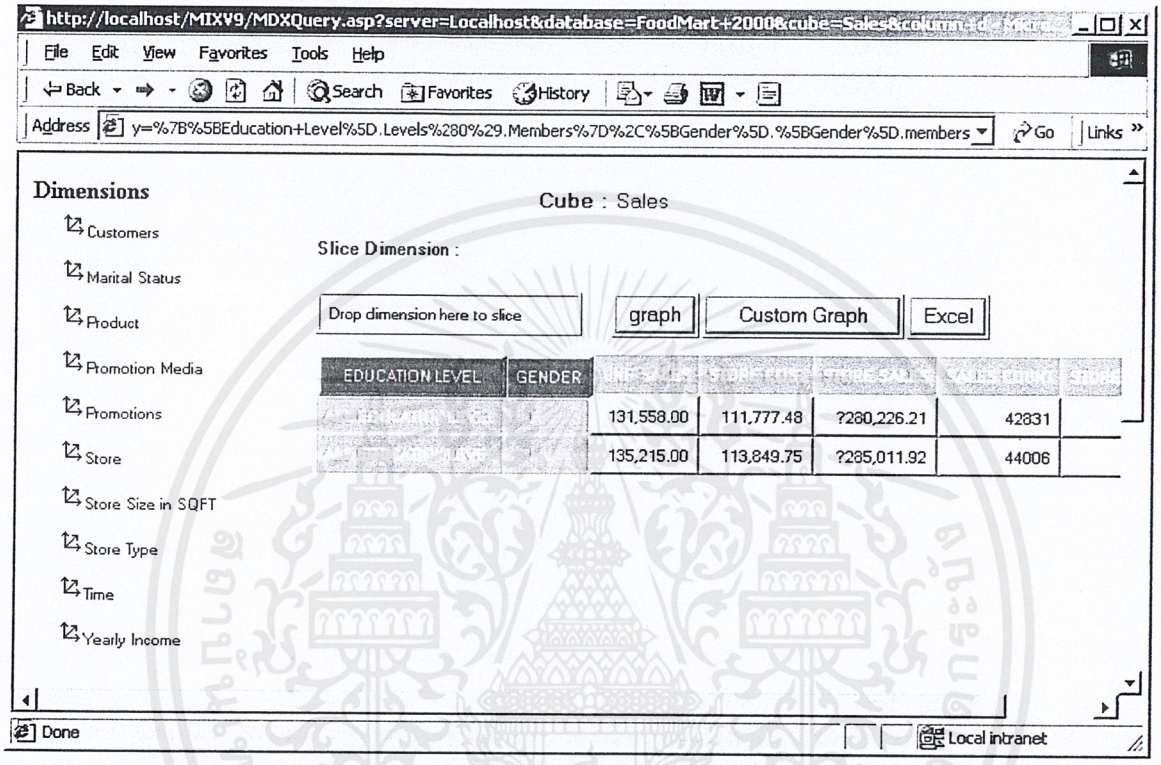
จากรูปที่ 8-7 จะเห็นว่าแกนแถวของตารางแสดงได้เปลี่ยนจาก ALL CUSTOMERS มาเป็น ALL EDUCATION LEVEL แล้ว



รูปที่ 8-8 แสดงการลากเลเวล [Gender].[Gender] มาวางในแกนแถวของตารางแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

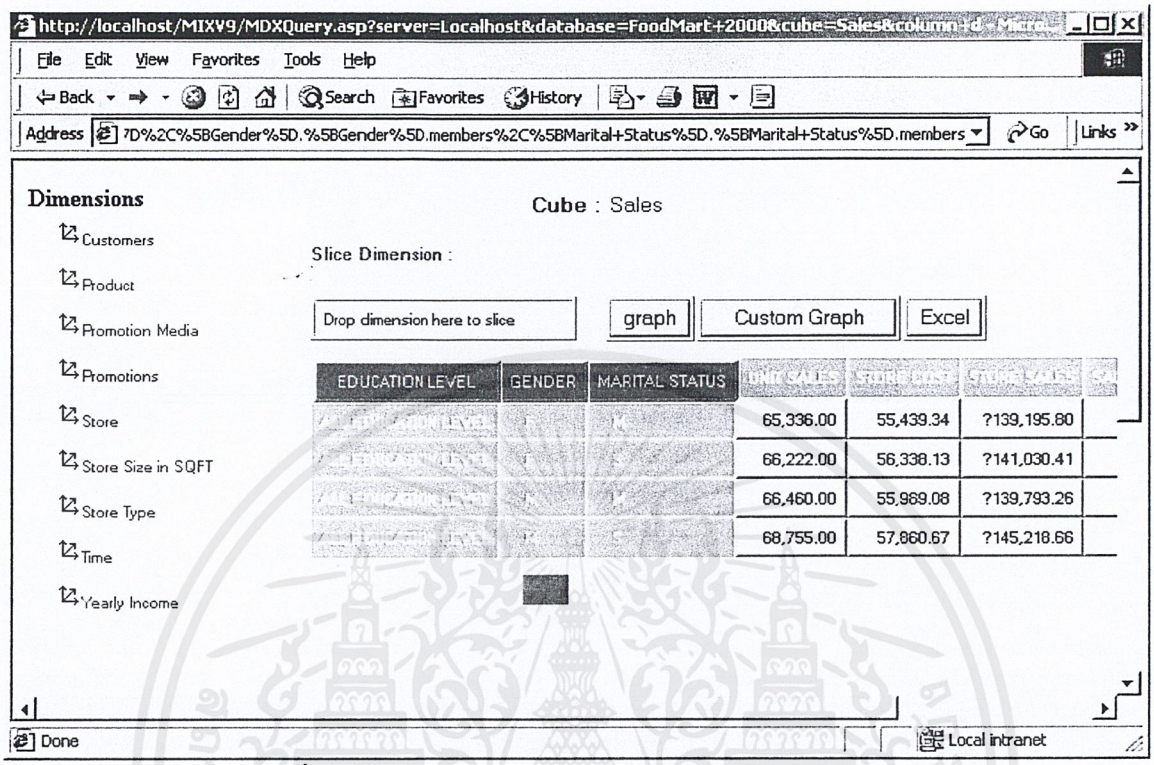
จากรูปที่ 8-7 หลังจากคลิกไดเมนชัน Gender ทางด้านซ้าย โปรแกรมจะกางเลเวลที่ไดเมนชันนี้มีออกมาคือ [Gender].[ALL] และ [Gender].[Gender] จากนั้นเราได้ลากเลเวล [Gender].[Gender] ไปวางบนตารางแสดงผลดังรูปที่ 8-8 ซึ่งผลลัพธ์จะเป็นดังรูปที่ 8-9



รูปที่ 8-9 แสดงผลไดเมนชัน 2 ไดเมนชันบนแกนแถวของตารางแสดงผล

นอกจากนี้ หากมีการแสดงผลบนแกนแถวไว้หลายไดเมนชันแล้ว ผู้ใช้ก็สามารถลบไดเมนชันที่ไม่ต้องการแสดงให้ออกไปจากตารางได้โดยการคลิกที่ชื่อไดเมนชัน (ซึ่งแสดงในช่องสีแดงด้านบน) จากนั้นลากออกมาจากตารางซึ่งโปรแกรมจะแสดงช่องสีแดงตามเมาส์ที่ลากด้วย แสดงได้ดังรูปที่ 8-10 และ 8-11

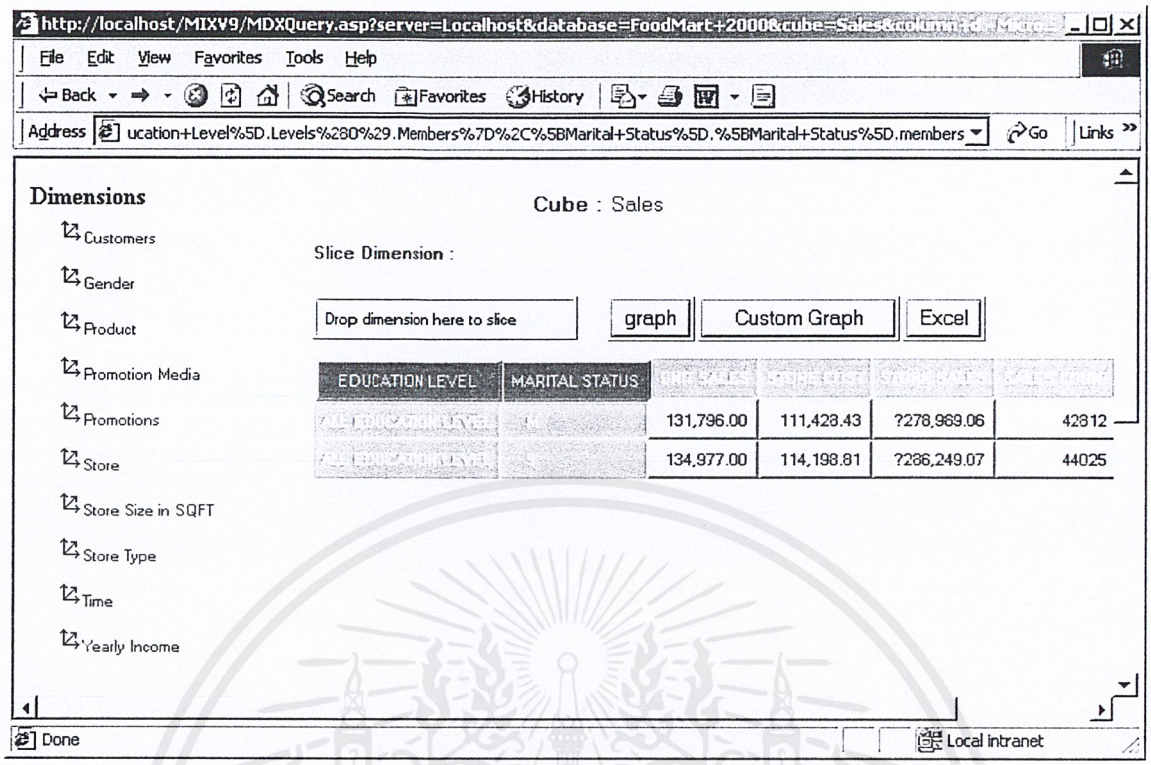
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-10 แสดงการการลากไดเมนชัน Gender ออกจากตารางแสดงผล

เมื่อเราต้องการดูแค่เพียงไดเมนชัน Eduction และ Marital Status เท่านั้น โดยเราไม่ต้องการดูไดเมนชัน Gender อีกต่อไปแล้ว เราก็สามารถคลิกลากไดเมนชัน Gender ออกมานอกตารางซึ่งจะได้ผลลัพธ์ดังรูปที่ 8-11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

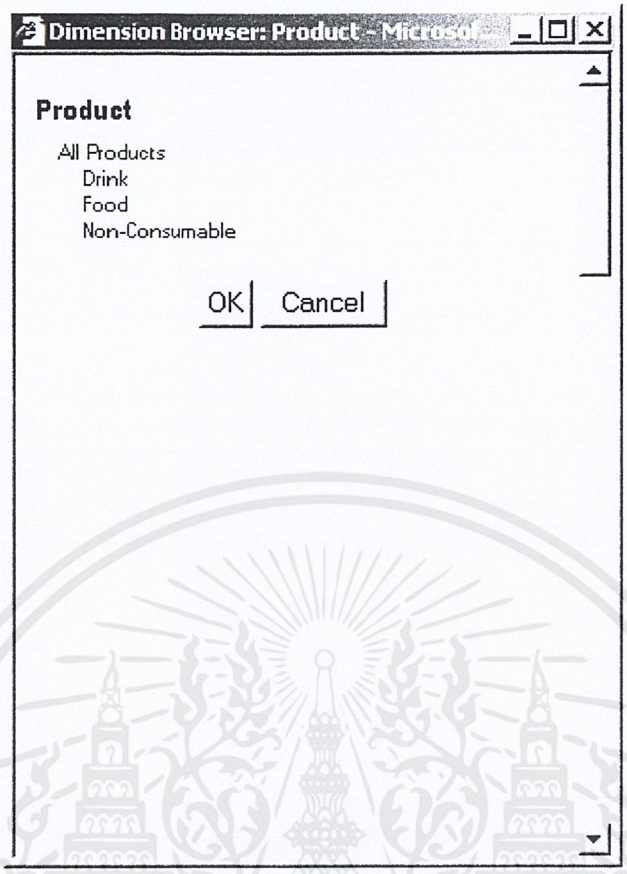


รูปที่ 8-11 แสดงผลลัพธ์หลังจากดึงไดเมนชัน Gender ออกไปแล้ว

- การ Slice

การ Slice สามารถทำได้โดยลากไดเมนชันไปวางไว้ยังส่วนที่เขียนว่า "Drop dimension here to slice dimension" โปรแกรมจะเปิดหน้าต่างบราวเซอร์ใหม่สำหรับทำโอเปอเรชัน Slice ผลลัพธ์จะออกมาเป็นดังรูปที่ 8-12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

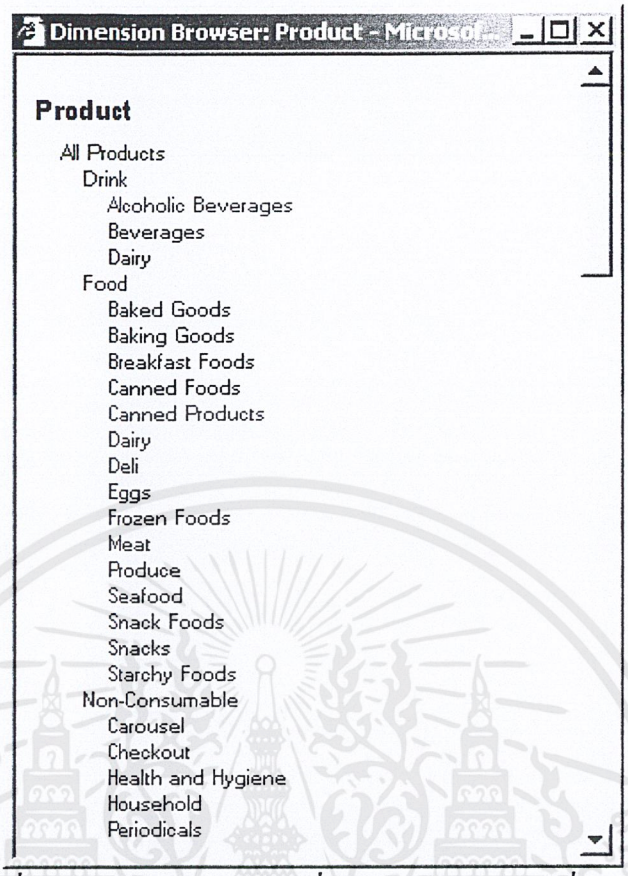


รูปที่ 8-12 แสดงหน้าจอ หลังจากลากไดเมนชัน Product มาวางยังส่วน Slice Dimension

ในหน้า Dimension Browser นี้เราสามารถคลิกที่ชื่อสมาชิกเพื่อกางดูสมาชิกในระดับที่ต่ำกว่าได้ ดังรูปที่ 8-13

หลังจากคลิกเลือกแล้ว ตัวอักษรจะเปลี่ยนเป็นสีแดง เมื่อเลือกแล้วให้กดปุ่ม OK ที่ข้างล่างโปรแกรมจะกลับมายังหน้าต่างแสดงผลตารางอีกครั้งหนึ่ง ซึ่งครั้งนี้จะเห็นว่าข้อมูลในตารางได้มีค่าเปลี่ยนไปแล้ว และจะมีข้อมูลเพิ่มตรงส่วน **Slice Dimension :** โดยข้อมูลที่เพิ่มเข้ามานั้นจะเป็นรายละเอียดของการ Slice เพื่อบอกกับผู้ใช้ว่าขณะนี้ผู้ใช้ได้เลือกตัดดูเฉพาะตรงส่วนไหน

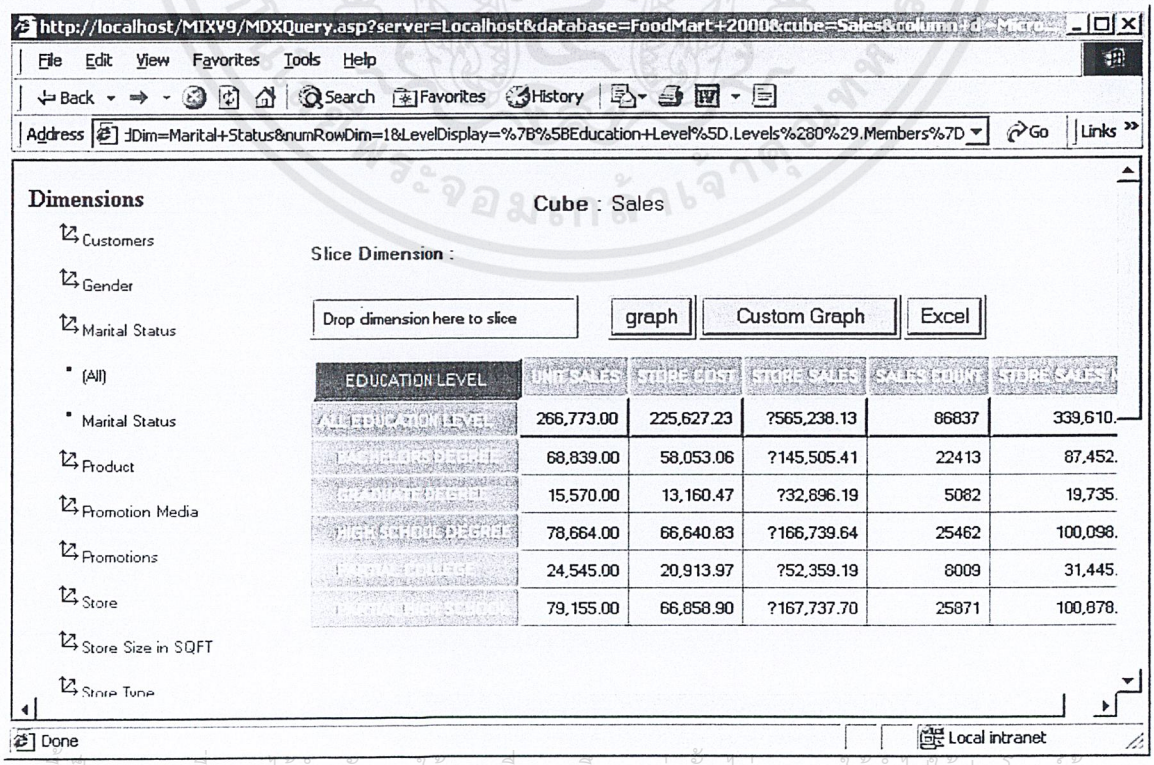
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-13 แสดงการกางสมาชิกเพื่อเลือกสมาชิกในระดับที่ลึกลงไป

- การ Drill up และ การ Drill down

สามารถทำได้โดยดับเบิลคลิกที่เซลล์ในตารางแสดงผล ซึ่งหากเซลล์นั้นสามารถ Drill up หรือ Drill down ได้โปรแกรมก็จะทำงานตามนั้น แต่ถ้าไม่สามารถทำได้ก็จะไม่มีการเปลี่ยนแปลงเกิดขึ้น



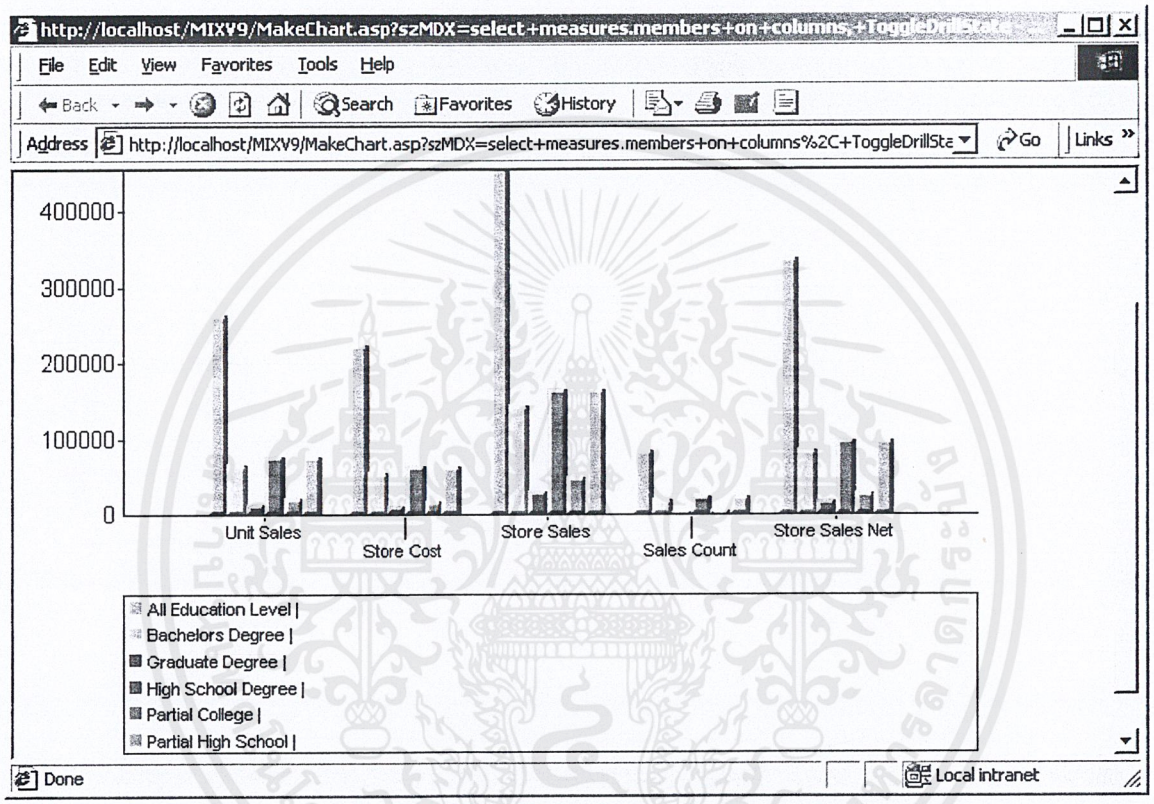
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 8-14 แสดงการทำ Drill down

จากรูปที่ 8-7 หากดับเบิลคลิกตรงเซลล์ ALL EDUCATION LEVEL ในตารางแสดงผล จะเป็นการ Drill down ซึ่งจะได้ผลลัพธ์ออกมาดังรูปที่ 8-14 ส่วนการ Drill up ก็ทำในทางกลับกันคือดับเบิลคลิกเซลล์ ALL EDUCATION LEVEL รูปที่ 8-14 ก็จะได้ผลลัพธ์เป็นรูปที่ 8-7

- การแสดงผลกราฟ

เมื่อต้องการแสดงผลในรูปแบบกราฟ ก็คลิกที่ปุ่ม graph ซึ่งอยู่เหนือตารางก็จะได้ผลลัพธ์ออกมาดังรูปที่ 8-15

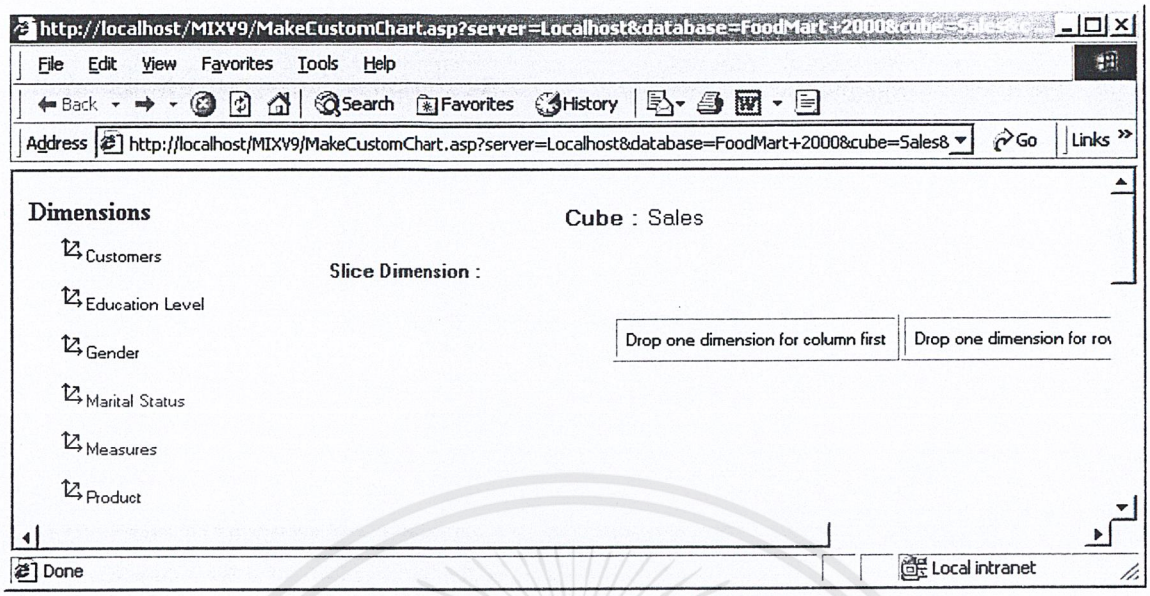


รูปที่ 8-15 แสดงผลข้อมูลในรูปกราฟ โดยการคลิกที่ปุ่มกราฟ

จะเห็นว่าโปรแกรมจะนำข้อมูลทุกอย่างในตารางแสดงผลไปแสดงในรูปแบบกราฟ ดังนั้นจะมีปัญหาเกิดขึ้นในกรณีข้อมูลในตารางแสดงผลมีเป็นจำนวนมาก ทำให้กราฟที่ได้จะออกมาเบียดกันมากและมีขนาดเล็กลงซึ่งทำให้ดูได้ยาก ดังนั้นจึงได้มีการทำให้ส่วนที่ให้ผู้ใช้สามารถเลือกแสดงผลเฉพาะข้อมูลที่ต้องการดูได้ เมื่อนำไปสร้างกราฟก็จะได้กราฟที่มีขนาดเหมาะสม ผู้ใช้สามารถดูหรือวิเคราะห์ได้อย่างสะดวก

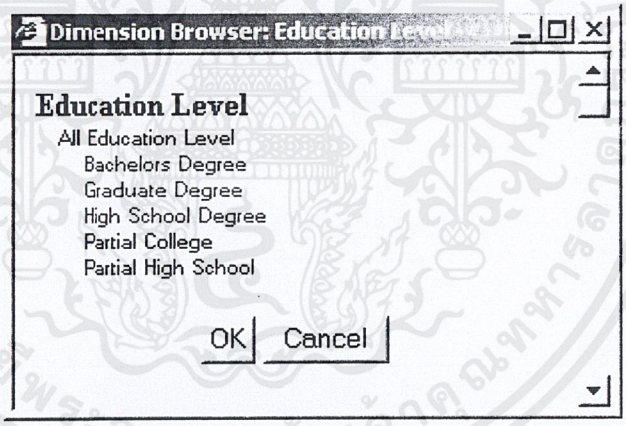
การเลือกข้อมูลที่จะแสดงผลบนแกนแถวและแกนคอลัมน์สามารถทำได้โดยการคลิกที่ปุ่ม Custom Graph ซึ่งโปรแกรมจะเปิดหน้าต่างใหม่ขึ้นมาให้แสดงดังรูปที่ 8-16 โดยมีลักษณะคล้ายกับหน้าจอหลักเพียงแต่ไม่ได้มีการแสดงตารางข้อมูล และมีกรอบสี่เหลี่ยมเพิ่มขึ้นมาคือ Drop one dimension for column first และ Drop one dimension for row first โดยผู้ใช้สามารถลากไดเมนชันมาวางยังกรอบสี่เหลี่ยมนี้ได้ เพื่อเลือกข้อมูลที่จะแสดงบนแกนแถวและแกนคอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

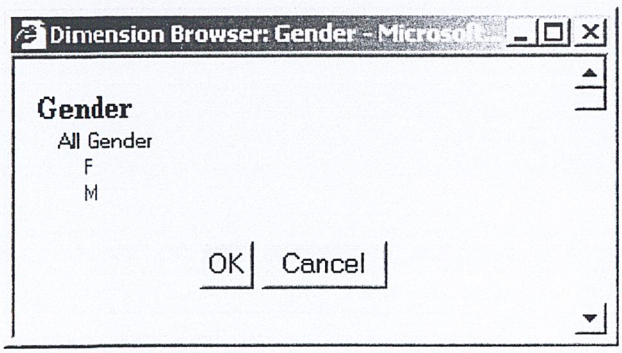


รูปที่ 8-16 แสดงหน้าจอการทำกราฟแบบกำหนดข้อมูลเอง

เมื่อเราลากไดเมนชันมาวางบนกรอบสี่เหลี่ยม โปรแกรมก็จะเปิดหน้าต่างใหม่ให้สำหรับเลือกข้อมูลที่จะแสดงบนแกนแถวและแกนคอลัมน์ แสดงได้ดังรูปที่ 8-17 และ 8-18



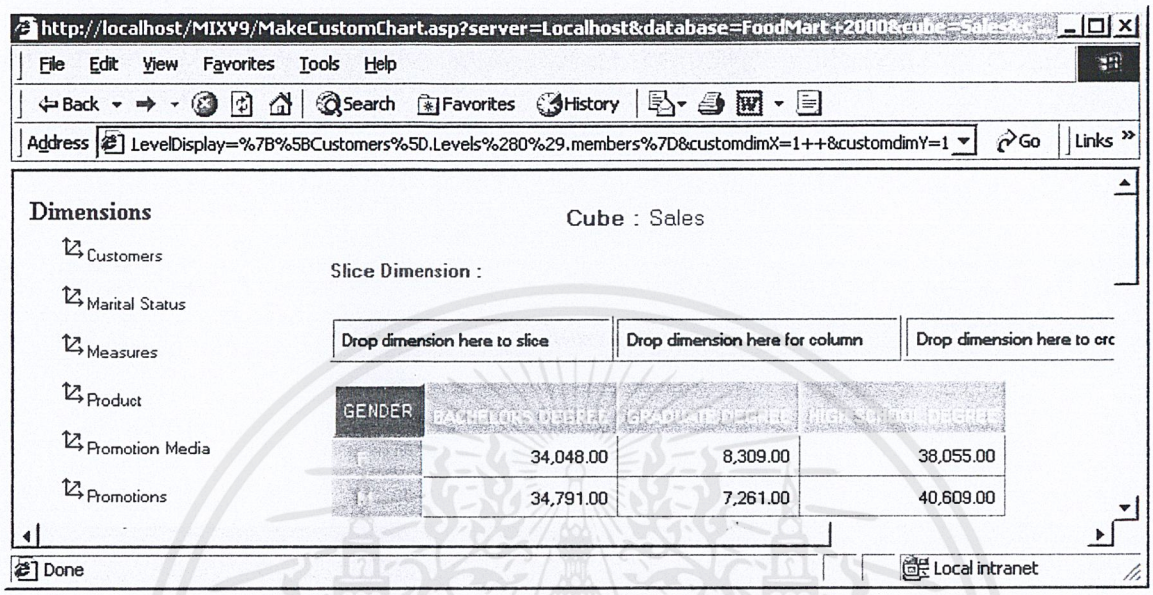
รูปที่ 8-17 แสดงหน้าจอการเลือกข้อมูลที่จะแสดงบนแกนคอลัมน์



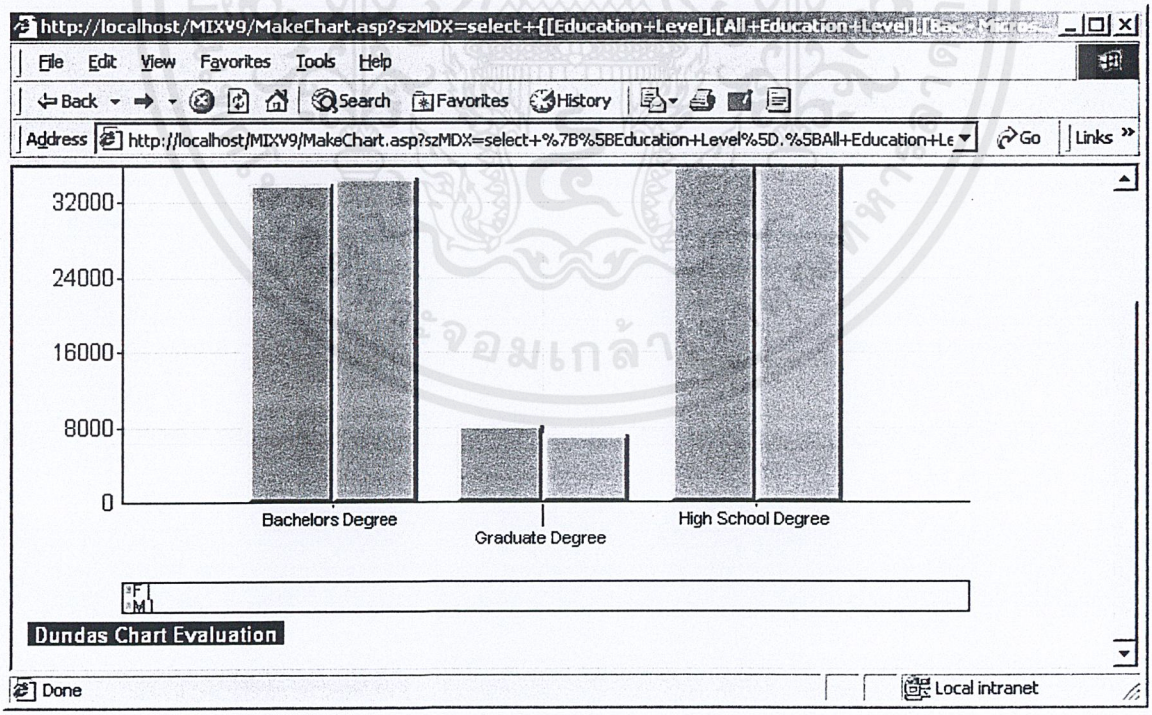
รูปที่ 8-18 แสดงหน้าจอการเลือกข้อมูลที่จะแสดงบนแกนคอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเลือกข้อมูลที่จะแสดงบนแกนแถวและแกนคอลัมน์เรียบร้อยแล้ว โปรแกรมก็จะแสดงตารางข้อมูลออกมาดังรูปที่ 8-19 เมื่อผู้ใช้ต้องการแสดงข้อมูลในรูปกราฟก็เพียงคลิกที่ปุ่ม Graph ทางด้านขวาซึ่งก็จะได้ผลออกมาดังรูปที่ 8-20



รูปที่ 8-19 แสดงผลลัพธ์หลังจากการระบุข้อมูลบนแกนแถวและแกนคอลัมน์

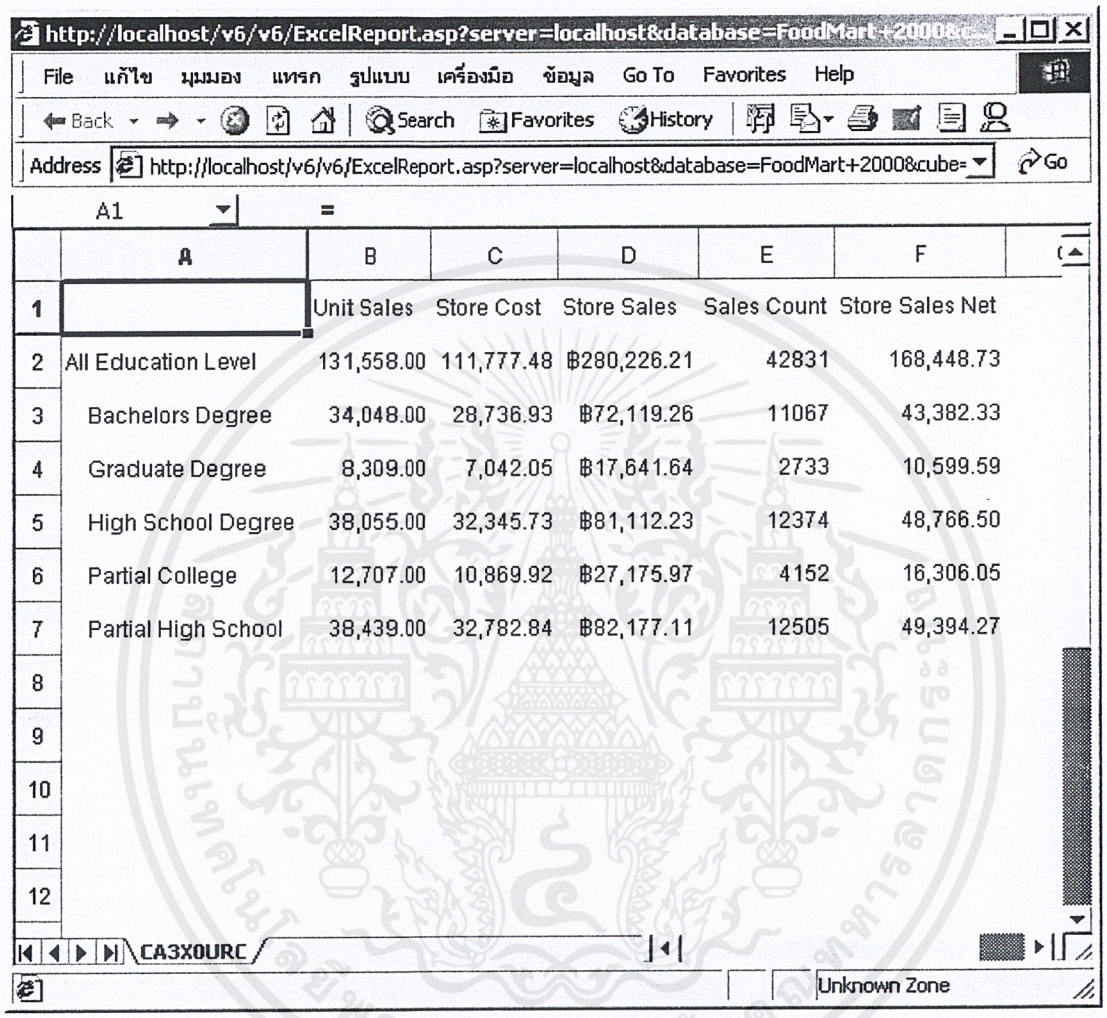


รูปที่ 8-20 แสดงกราฟที่ผู้ใช้ระบุข้อมูลบนแกน x และแกน y เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การแสดงผลในรูปแบบของ Microsoft Excel

เมื่อต้องการแสดงผลข้อมูลรูปแบบของ Microsoft Excel สิ่งที่จำเป็นก็คือเครื่องไคลเอนต์ต้องมี Microsoft Excel 2000 ติดตั้งเอาไว้ จากนั้นก็เพียงคลิกที่ปุ่ม Excel ก็จะได้ผลลัพธ์ออกมาดังรูปที่ 8-21



	A	B	C	D	E	F
1		Unit Sales	Store Cost	Store Sales	Sales Count	Store Sales Net
2	All Education Level	131,558.00	111,777.48	฿280,226.21	42831	168,448.73
3	Bachelors Degree	34,048.00	28,736.93	฿72,119.26	11067	43,382.33
4	Graduate Degree	8,309.00	7,042.05	฿17,641.64	2733	10,599.59
5	High School Degree	38,055.00	32,345.73	฿81,112.23	12374	48,766.50
6	Partial College	12,707.00	10,869.92	฿27,175.97	4152	16,306.05
7	Partial High School	38,439.00	32,782.84	฿82,177.11	12505	49,394.27
8						
9						
10						
11						
12						

รูปที่ 8-21 แสดงผลข้อมูลในรูปแบบของ Microsoft Excel

เราสามารถบันทึกแฟ้มข้อมูลนี้เอาไว้ เพื่อเอาไปใช้ประโยชน์ได้ต่อไป โดยเลือกที่ **File -> Save as** จากนั้นก็ใส่ชื่อของแฟ้มข้อมูลที่ต้องการจะบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9
การทดสอบประสิทธิภาพการทำงานของโปรแกรมประยุกต์และการเก็บข้อมูลของ Analysis Service

ประสิทธิภาพในการคิวรีข้อมูลมีความสำคัญอย่างยิ่งในการศึกษาการเก็บข้อมูลแบบดาต้าแวเฮาส์ เนื่องจากต้องออกรายงานเพื่อการวิเคราะห์ให้ผู้บริหารให้รวดเร็ว การศึกษาทั้งความเร็วในการคิวรีของโปรแกรมประยุกต์ เวลาและเนื้อที่ที่ต้องเสียไปในการสร้างวิธีการเก็บข้อมูลแบบต่างๆ จึงมีความสำคัญเป็นอย่างมากเพื่อใช้ในการตัดสินใจของผู้ดูแลระบบว่าควรเลือกใช้วิธีการแบบใดในการเก็บข้อมูลที่ดีที่สุด

9.1 การทดสอบประสิทธิภาพของโปรแกรมประยุกต์

ทดสอบการทำงานของโปรแกรมประยุกต์ ความสามารถออกรายงานได้มีประสิทธิภาพแค่ไหนในสภาวะแวดล้อมที่ต่างกัน ซึ่งประกอบไปด้วยการทดสอบเพิ่มจำนวนเรคอร์ดในตารางแฟ็ค , การเพิ่มจำนวนไดเมนชันและการเพิ่มจำนวนการรวมค่าว่ามีผลกระทบกับเวลาในการคิวรีข้อมูลอย่างไร โดยที่ทดสอบการเก็บข้อมูลทั้งแบบคือ MOLAP , ROLAP และ HOLAP โดยใช้ฮาร์ดแวร์และซอต์ฟแวร์ดังนี้

โปรแกรมประยุกต์ที่ใช้เป็นโปรแกรมประยุกต์ที่เขียนขึ้นมาจาก Vistual Basic 6.0 โดยใช้คอมโพเนนท์ที่ได้มาเมื่อติดตั้ง Analysis Service ทดสอบทางฝั่งไคลเอ็นต์

เซิร์ฟเวอร์

- Pentium 667
- Ram 256MB
- HD 15 GB ATA/66
- Analysis Server , Microsoft SQL Server 2000

ไคลเอ็นต์

- Pentium 700
- Ram 128MB
- HD 15 GB ATA/66
- PivotTable service
- โปรแกรมประยุกต์ที่ใช้ คอมโพเนนท์ ของ Microsoft SQL Server 2000 นำมาใช้กับ Visual Basic 6.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.1.1 ทดสอบการเพิ่มจำนวนเรคอร์ดบนตารางแฟ็ค

กสีมาที่ใช้ประกอบด้วยตารางแฟ็คที่เพิ่มจำนวนเรคอร์ดจาก 1 ล้านถึง 5 ล้านเรคอร์ด และมีไดเมนชัน จำนวน 3 ไดเมนชัน 2 ไดเมนชันเป็นสกีมาแบบดาวที่เหลืออีกหนึ่งเป็นสโนว์เฟลก ได้ใช้การรวมค่า 0 % เนื่องจากต้องการเวลาในการคิวรีข้อมูลที่เกิดขึ้นจึงติดตั้งค่าการรวมค่าไว้ 0 % เพื่อให้เห็นเวลาในการคิวรีชัดเจนกว่าการรวมค่า 100 % ตารางที่ 9-1 แสดงถึงจำนวน เรคอร์ดของตารางแฟ็คและตารางไดเมนชัน

ตารางแฟ็คและไดเมนชัน	จำนวนเรคอร์ด
Sales_face_1997	1 - 5 ล้าน
Time_by_day	730
Product	1,560
Product_Class	110
Customer	10,281

ตารางที่ 9-1 แสดงจำนวนเรคอร์ดในตารางแฟ็คและไดเมนชันของการเพิ่มเรเคอร์ดในตารางแฟ็ค

ไดเมนชันที่กล่าวในตารางที่ 9-1 นี้ประกอบเลเวลต่างๆ ดังนี้

Time_by_day

Year > Quarter > month

Product และ Product_Class

Product Family > Product department > Product Category > Product Subcategory > Brand name > Product name

Customer

Country > State Province > City > name

การทดสอบโดยติดตั้งโปรแกรมประยุกต์ทางฝั่งไคลเอ็นต์และลองทำโอเปอเรชันต่างๆ เป็นจำนวน 10 โอเปอเรชัน ดังนี้

1. เปิด Cube Browser
2. เลือก drill-down Dimension USA และ drill-up เมื่อเสร็จ
3. เลือก drill-down Dimension Maxico และ drill-up เมื่อเสร็จ
4. นำ Product Dimension มา crossjoin กับ Customer Dimension
5. เลือก Product Dimension ที่เลเวล (Level) ที่ลึกลงไป
6. เลือก Product Dimension ที่เลเวลที่ลึกลงไป
7. เลือก Product Dimension ที่เลเวลที่ลึกลงไป
8. เลือก Product Dimension ที่เลเวลที่ลึกลงไป
9. เลือก Product Dimension ที่เลเวลที่ลึกลงไป
10. เลือก Customer Dimension ที่เลเวลที่ลึกลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลที่ได้จากการทดสอบ

โอเปอเรชันที่	ระยะเวลาในการคิวรี (วินาที)		
	MOLAP	ROLAP	HOLAP
1	0	10	9
2	0	9	10
3	0	10	8
4	0	14	10
5	0	12	9
6	0	70	13
7	0	45	25
8	0	46	21
9	1	28	23
10	1	27	11
เฉลี่ย	**0.2**	**27.1**	**13.9**

ตารางที่ 9-2 แสดงผลการทดสอบระยะเวลาคิวรีของ 1 ล้านเรคอร์ด

โอเปอเรชันที่	ระยะเวลาในการคิวรี (วินาที)		
	MOLAP	ROLAP	HOLAP
1	0	17	11
2	1	14	16
3	0	16	16
4	0	20	12
5	1	14	14
6	1	79	13
7	2	60	25
8	2	61	23
9	0	30	21
10	0	35	12
เฉลี่ย	**0.7**	**34.6**	**16.3**

ตารางที่ 9-3 แสดงผลการทดสอบระยะเวลาคิวรีของ 2 ล้านเรคอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โอเปอเรชันที่	ระยะเวลาในการคิวรี (วินาที)		
	MOLAP	ROLAP	HOLAP
1	0	52	22
2	1	40	21
3	0	92	22
4	0	96	24
5	0	88	91
6	1	76	70
7	1	68	63
8	2	89	93
9	2	85	82
10	5	53	12
เฉลี่ย	**1.2**	**73.9**	**50**

ตารางที่ 9-4 แสดงผลการทดสอบระยะเวลาคิวรีของ 3 ล้านเรคอร์ด

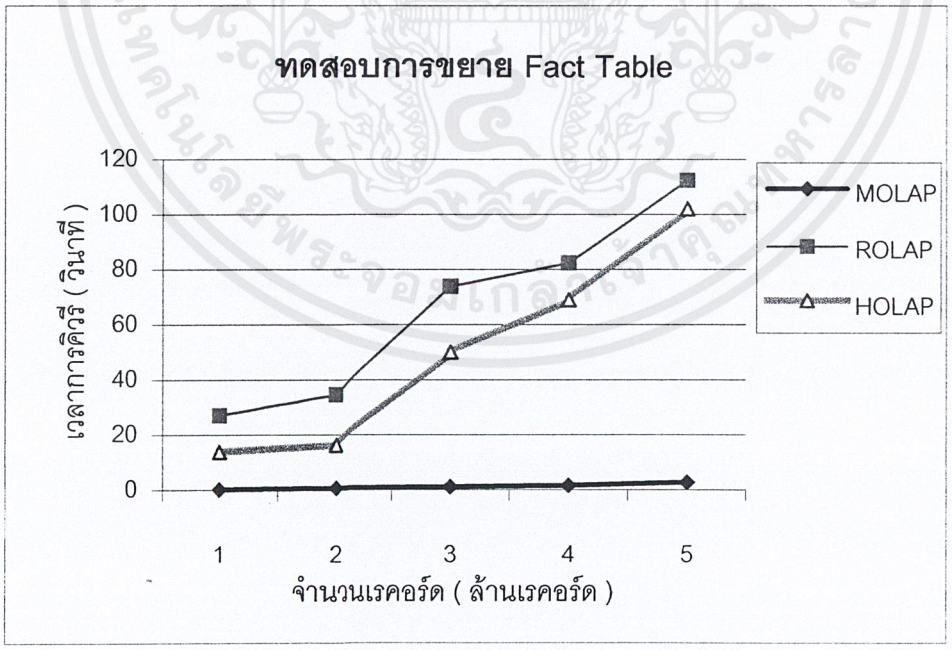
โอเปอเรชันที่	ระยะเวลาในการคิวรี (วินาที)		
	MOLAP	ROLAP	HOLAP
1	0	47	5
2	0	49	42
3	0	23	36
4	1	77	73
5	5	80	72
6	1	160	77
7	6	146	94
8	2	112	83
9	1	52	110
10	0	78	98
เฉลี่ย	**1.6**	**82.4**	**69**

ตารางที่ 9-5 แสดงผลการทดสอบระยะเวลาคิวรีของ 4 ล้านเรคอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โอเปอเรชันที่	ระยะเวลาในการคิวรี (วินาที)		
	MOLAP	ROLAP	HOLAP
1	0	60	10
2	0	53	60
3	0	60	36
4	2	107	100
5	5	150	145
6	5	101	90
7	3	122	60
8	5	180	120
9	3	200	321
10	5	90	78
เฉลี่ย	**2.8**	**112.3**	**102**

ตารางที่ 9-6 แสดงผลการทดสอบระยะเวลาคิวรีของ 5 ล้านเรคอร์ด



รูปที่ 9-1 รูปแสดงเวลาในการคิวรีเมื่อมีจำนวนเรคอร์ดต่างๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดสอบจะเห็นว่าระยะเวลาในการคิวรีของโปรแกรมประยุกต์ผ่านแลนจะเพิ่มมากขึ้นเมื่อมีจำนวนเรคอร์ดในตารางแฟ็คมากขึ้น แต่อย่างไรก็ตามจะเห็นได้ว่า MOLAP ก็ใช้ระยะเวลาในการคิวรีน้อยที่สุด ส่วนของ ROLAP ใช้เวลามากที่สุด

9.1.1 ทดสอบการเพิ่มจำนวนไดเมนชัน

การทดสอบนี้มีตารางเพื่อสร้างสกีมาดังนี้

ตารางแฟ็คและไดเมนชัน	จำนวนเรคอร์ด (เรคอร์ด)
Sales_face_1997	3 ล้าน
Time_by_day	730
Product	1,560
Product_Class	110
Customer	10,281
Store	25

ตารางที่ 9-7 แสดงจำนวนเรคอร์ดในตารางแฟ็คและตารางไดเมนชันของการเพิ่มจำนวนไดเมนชัน

สกีมาแบบ 1 ไดเมนชัน

ใช้ไดเมนชัน Time ประกอบด้วยเลเวล Year > Quarter > month

สกีมาแบบ 2 ไดเมนชัน

ใช้ไดเมนชัน Time และ ไดเมนชัน Product ประกอบด้วยเลเวล Product Family > Product department > Product Category > Product Subcategory > Brand name > Product name

สกีมาแบบ 3 ไดเมนชัน

ใช้ไดเมนชัน Timeไดเมนชัน Product และ ไดเมนชัน Customer ประกอบไปด้วยเลเวล Country > State Province > City > name

การทดสอบโดยติดตั้งโปรแกรมประยุกต์ทางฝั่งไคลเอ็นต์และลองทำโอเปอเรชันต่างๆ โดยกำหนดจำนวนการรวมค่าไว้ 0% การทดสอบใช้วิธีสุ่มโอเปอเรชัน 3 โอเปอเรชัน ซึ่งจะขึ้นอยู่ในแต่ละสกีมา

ผลที่ได้จากการทดสอบ

โอเปอเรชันที่	ระยะเวลาในการคิวรี (วินาที)		
	MOLAP	ROLAP	HOLAP
1	0	70	69
2	0	80	72
3	0	160	159
เฉลี่ย	0	103.3333	100

ตารางที่ 9-8 ผลการทดสอบระยะเวลาการคิวรีของ 1 ไดเมนชัน

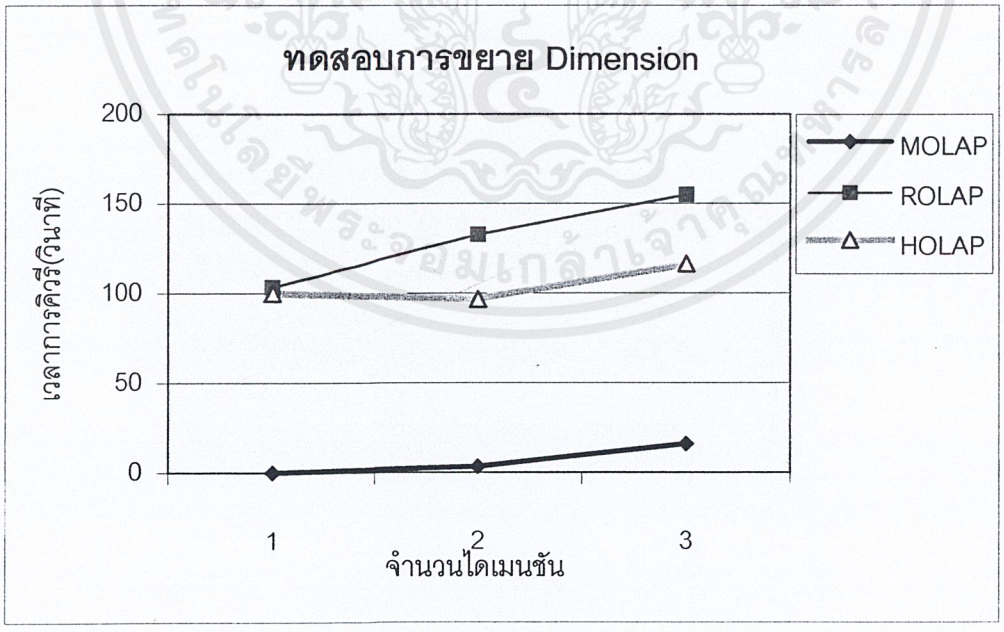
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โอเปอเรชัน ที่	ระยะเวลาในการคิวรี (วินาที)		
	MOLAP	ROLAP	HOLAP
1	0	78	70
2	0	260	60
3	10	60	160
เฉลี่ย	3.333333	132.6667	96.66667

ตารางที่ 9-9 ผลการทดสอบระยะเวลาการคิวรีของ 2 ไดเมนชัน

โอเปอเรชัน ที่	ระยะเวลาในการคิวรี (วินาที)		
	MOLAP	ROLAP	HOLAP
1	13	75	89
2	15	90	80
3	20	300	180
เฉลี่ย	16	155	116.3333

ตารางที่ 9-10 ผลการทดสอบระยะเวลาการคิวรีของ 3 ไดเมนชัน



รูปที่ 9-2 รูปแสดงเวลาในการคิวรีเมื่อมีจำนวนไดเมนชันต่างๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดสอบเห็นว่าการเพิ่มไดเมนชันมีผลทำให้เวลาในการคิวรีข้อมูลของโปรแกรมประยุกต์ไคลเอ็นต์ช้ามากขึ้น อย่างไรก็ตาม การวิธีการใช้การเก็บข้อมูลแบบ MOLAP ก็ยังทำใช้เวลาในการคิวรีน้อยที่สุด ส่วน ROLAP ก็ยังคงใช้เวลาในการคิวรีมากที่สุด

9.1.3 ทดสอบการเพิ่มจำนวนการรวมค่า

สกีมาที่ใช้ประกอบไปด้วยตารางดังต่อไปนี้ โดยเลเวล (Level) ในแต่ละไดเมนชันใช้เหมือนกับการทดสอบแรกๆ

ตารางแฟ็คและไดเมนชัน	จำนวนเรคอร์ด
Sales_face_1997	3 ล้าน
Time_by_day	730
Product	1,560
Product_Class	110
Customer	10,281

ตารางที่ 9-11 แสดงจำนวนเรคอร์ดในตารางแฟ็คและตารางไดเมนชันของการเพิ่มการรวมค่า

การทดสอบโดยติดตั้งโปรแกรมประยุกต์ทางฝั่งไคลเอ็นต์และลองทำโอเปอเรชันต่างๆ โดยกำหนดจำนวนการรวมค่าไว้ 0%, 30%, 60%, 90% การทดสอบนี้ใช้การเลือกทำ 5 โอเปอเรชัน ดังนี้

1. เปิด Cube Browser
2. เลือก Product Dimension มา crossjoin
3. เลือกเลเวลที่ลึกลงไปของ Product
4. เลือกเลเวลที่ลึกลงไปของ Product
5. เลือกเลเวลที่ลึกลงไปของ Product

ผลที่ได้จากการทดสอบ

โอเปอเรชันที่	ระยะเวลาในการคิวรี (วินาที)			
	0%	30%	60%	90%
1	1	1	0	0
2	1	2	0	0
3	2	3	0	0
4	2	4	0	0
5	5	1	0	0
เฉลี่ย	2.2	2.2	0	0

ตารางที่ 9-12 ผลการทดสอบระยะเวลาการคิวรีของ MOLAP ด้วยการรวมค่าค่าต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

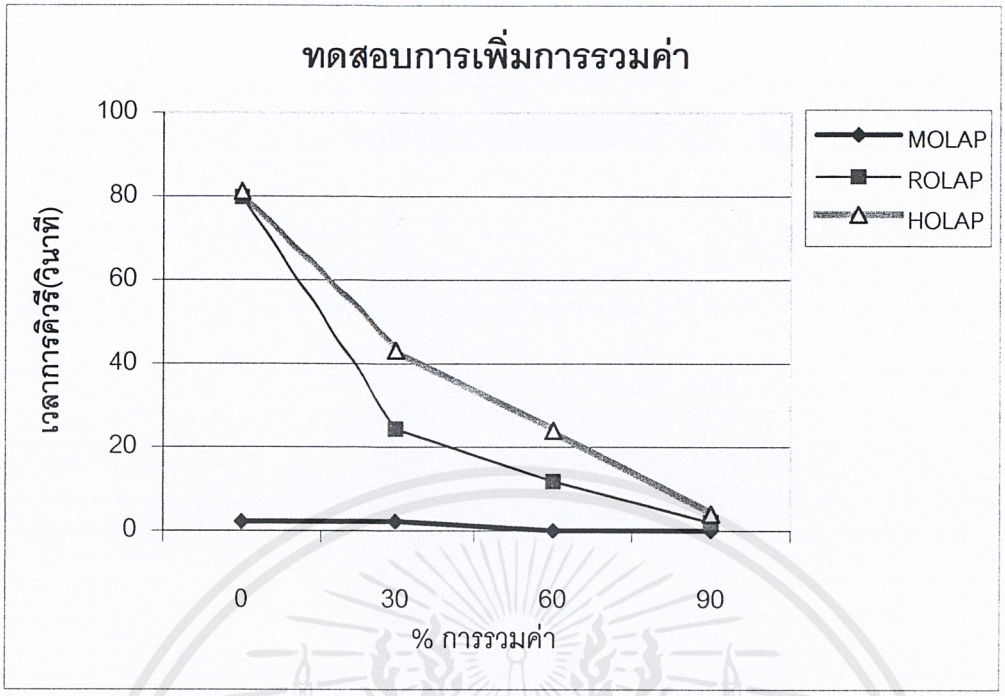
โอเปอเรชัน ที่	ระยะเวลาในการคิวรี (วินาที)			
	0%	30%	60%	90%
1	91	17	2	0
2	70	15	0	3
3	63	5	30	2
4	93	75	15	2
5	82	10	12	3
เฉลี่ย	79.8	24.4	11.8	2

ตารางที่ 9-13 ผลการทดสอบระยะเวลาการคิวรีของ HOLAP ด้วยการรวมค่าค่าต่างๆ

โอเปอเรชัน ที่	ระยะเวลาในการคิวรี (วินาที)			
	0%	30%	60%	90%
1	88	3	0	0
2	76	30	30	10
3	68	20	0	10
4	89	120	0	0
5	85	43	90	0
เฉลี่ย	81.2	43.2	24	4

ตารางที่ 9-14 ผลการทดสอบระยะเวลาการคิวรีของ ROLAP ด้วยการรวมค่าค่าต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9-3 รูปแสดงเวลาในการคิวรีเมื่อมีจำนวนการรวมค่าต่างๆ กัน

การทดสอบการรวมค่าทำให้เห็นว่ายิ่งมีการรวมค่ามาก เวลาในการคิวรียิ่งน้อยลง ทำให้รู้ว่าความเร็วในการคิวรีหรือออกรายงานนั้นขึ้นอยู่กับการเก็บข้อมูลว่าเก็บอย่างไร

การเก็บข้อมูลทั้ง MOLAP, HOLAP และ ROLAP หลังจากสร้างสกีมาเสร็จเราต้องสร้างฐานข้อมูลจากสกีมานั้นๆ ดังนั้นจึงมีระยะเวลามาเกี่ยวข้องกับการเก็บข้อมูลทั้ง 3 แบบ อีกทั้งการเก็บข้อมูลแต่ละแบบยังมีวิธีการเก็บข้อมูลต่างแบบกันดังที่กล่าวในบทที่ 5 เพราะฉะนั้นจึงมีพื้นที่ในการเก็บข้อมูลที่ต่างกัน การทดสอบต่อไปจึงเป็นการทดสอบถึงการเก็บข้อมูลทั้ง 3 แบบให้ชัดเจนมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.2 การทดสอบประสิทธิภาพการเก็บข้อมูลแบบ MOLAP, ROLAP และ HOLAP

ทดสอบประสิทธิภาพ, เวลาในการสร้าง, พื้นที่ที่ใช้ทั้งบนตัว Analsis Server และ RDBMS Server , จำนวนการรวมค่า (aggreagate) ของการเก็บข้อมูลแบบ MOLAP, ROLAP และ HOLAP มีขั้นตอนการทดสอบดังนี้

- สร้างคิวบ์ที่ใช้การเก็บข้อมูลแบบต่างๆ ได้แก่แบบ MOLAP , ROLAP และ HOLAP
- เพิ่มประสิทธิภาพในการคิวรี (%Performance) ไปเรื่อยๆตั้งแต่ 10-100% โดยเว้นช่วงระยะห่างทีละ 10%
- เก็บข้อมูลต่างๆพื้นที่ที่ใช้บน Analysis Server และ RDBMS Server
- เก็บจำนวนการ Aggregation หลังจากทำการสร้างส่วนเก็บข้อมูลเสร็จ
- เก็บระยะเวลาในการทำการเก็บข้อมูลแบบต่างๆไว้

อุปกรณ์

- CPU Pentium 667
- RAM 256MB
- HD 15 GB ATA/66
- Analysis Server , Microsoft SQL Server 2000

รายละเอียดของสกีมา (schema)

มีคิวบ์ 2 แบบที่ใช้ในการทดลอง 2 ไดเมนชัน และ 4 ไดเมนชัน มีรายละเอียดดังนี้

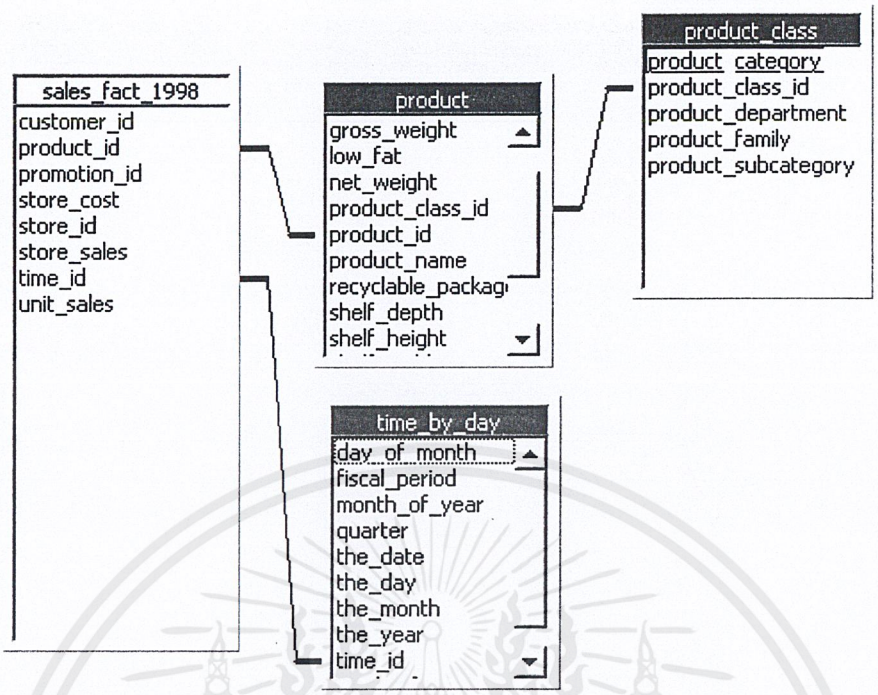
1) สกีมาแบบ 2 ไดเมนชัน

ประกอบด้วย 2 ไดเมนชันมีตารางทั้งหมด 4 ตาราง คือ 1 ตารางแฟ็ค 2 ตารางไดเมนชันมีหนึ่งไดเมนชันเป็นสกีมาสโนว์เฟล็ก ทำให้มีเพิ่มมากอีก 1 ตาราง ตารางที่ 9-4 แสดงจำนวนเรคอร์ดที่มีอยู่ในตารางแฟ็คและตารางไดเมนชัน

ตารางแฟ็คและไดเมนชัน	จำนวนเรคอร์ด (เรคอร์ด)
Sales_face_1997	3 ล้าน
Time_by_day	730
Product	1,560
Product_Class	110

ตารางที่ 9-15 แสดงจำนวนเรคอร์ดของสกีมาแบบ 2 ไดเมนชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9-4 แสดงสกีมาแบบ 2 ไดเมนชัน

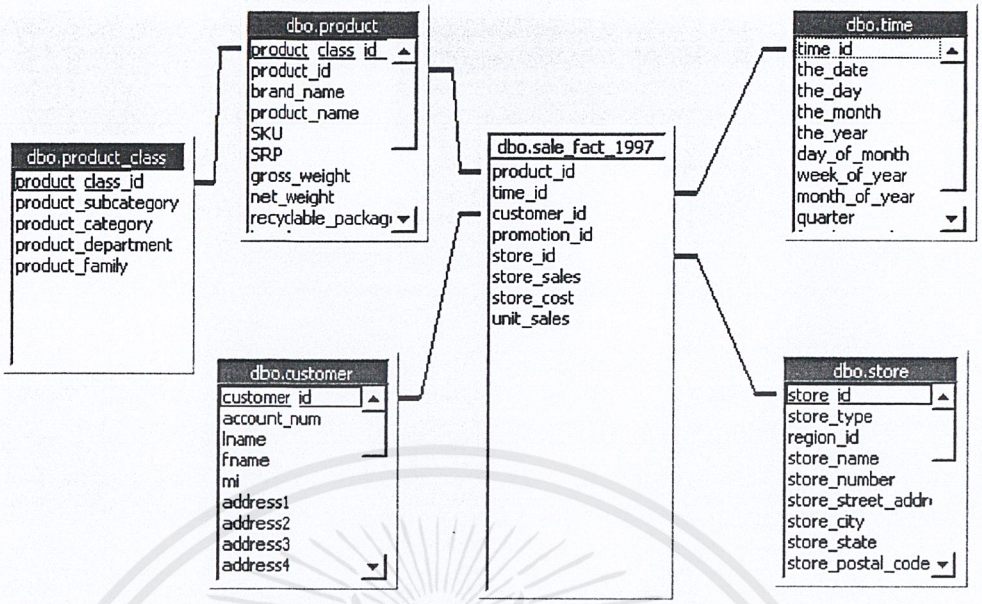
2) สกีมาแบบ 4 ไดเมนชัน

ประกอบด้วย 6 ตารางด้วยกัน ประกอบด้วยตารางที่มีจำนวนเรคอร์ดตามตารางที่ 9-5 และรูปที่ 9-5 แสดงสกีมาที่ออกแบบไว้

ตารางแฟ็คและไดเมนชัน	จำนวนเรคอร์ด (เรคอร์ด)
Sales_face_1997	3 ล้าน
Time_by_day	730
Product	1,560
Product_Class	110
Customer	10,281
Store	25

ตารางที่ 9-16 แสดงจำนวนเรคอร์ดของสกีมาแบบ 4 ไดเมนชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9-5 แสดงสกีมาแบบ 4 ไดเมนชัน

ผลการทดสอบ

ผลการที่ได้จากการทดสอบแสดงเป็น 2 ส่วน คือส่วนที่เป็นตารางและกราฟ ส่วนที่เป็นตารางจะประกอบ

1) Performance (%) หมายถึง เลือกประสิทธิภาพการทำงานเป็นเปอร์เซ็นต์ ยิ่งเพิ่มจำนวน Performance หมายถึงการเพิ่มจำนวนของการรวมค่าด้วย
2) Process Time (hh:mm:ss) หมายถึงเวลาในการสร้างการเก็บข้อมูล (Storage)
3) Size in meta data (MB) เป็น ค่าที่เกิดขึ้นบน Analysis Server
4) Number of Aggregation จำนวนการรวมค่าที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Performances (%)	Process time (hh:mm:ss)	Size in meta data (MB)	Number of aggregation
10	29	0.79	2
20	28.44	0.79	3
30	29.2	0.79	3
40	29.07	0.79	4
50	28.58	0.79	4
60	29.09	0.79	4
70	30.01	0.79	4
80	28.58	0.93	4
90	29.45	1.24	6
100	29.33	2.64	27

ตารางที่ 9-17 ผลทดสอบการเก็บข้อมูลแบบ MOLAP กับคิวบ์ 3 ล้านเรคอร์ด 2 ไดเมนชัน

Performance (%)	Process time (hh:mm:ss)	Size in meta data (MB)	Number of aggregation
10	0.21	0.01	2
20	2.15	0.01	3
30	2.51	0.01	3
40	3.04	0.01	3
50	3.33	0.01	3
60	14.18	0.01	3
70	11.33	0.02	4
80	20.26	0.02	4
90	16.14	0.02	6
100	1.29.46	0.07	26

ตารางที่ 9-18 ผลทดสอบการเก็บข้อมูลกับแบบ ROLAP คิวบ์ 3 ล้านเรคอร์ด 2 ไดเมนชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Performance (%)	Process time (hh:mm:ss)	Size in meta data (MB)	Number of aggregation
10	28.41	0.01	3
20	28.43	0.01	3
30	28.49	0.01	3
40	29.34	0.01	3
50	29.26	0.02	3
60	29.26	0.03	4
70	29.48	0.08	4
80	23.07	0.15	4
90	29.36	0.45	6
100	29.11	1.86	27

ตารางที่ 9-19 ผลทดสอบการเก็บข้อมูลแบบ HOLAP กับคิวบ์ 3 ล้านเรคอร์ด 2 ไดเมนชัน

Performance (%)	Process time (hh:mm:ss)	Size in meta data (MB)	Number of aggregation
10	40.22	3.03	18
20	40.51	7.71	27
30	40.46	7.98	28
40	40.37	8.06	41
50	40.39	8.56	41
60	39.28	9.12	46
70	42.1	11.5	59
80	46	14.9	71
90	48.20	28.7	111
100	48.09	65.6	375

ตารางที่ 9-20 ผลทดสอบการเก็บข้อมูลแบบ MOLAP กับคิวบ์ 3 ล้านเรคอร์ด 4 ไดเมนชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Performance (%)	Process time (hh:mm:ss)	Size in meta data (MB)	Number of aggregation
10	20.50	0	18
20	31.56	0.1	22
30	50.55	0.12	28
40	1.42.35	0.17	38
50	2.20.53	0.19	40
60	3.25.25	0.5	46
70	4.30.25	0.5	59
80	8.27.24	0.8	71
90	10.24.23	0.8	111
100	22.34.30	1.5	375

ตารางที่ 9-21 ผลทดสอบการเก็บข้อมูลแบบ ROLAP กับคิวบ์ 3 ล้านเรคอร์ด 4 ไดเมนชัน

Performance (%)	Process time (hh:mm:ss)	Size in meta data (MB)	number of aggregation
10	30.03	0	18
20	30.02	0.1	22
30	33.09	0.2	28
40	34.35	0.42	38
50	42.20	0.85	40
60	31.05	1.61	46
70	43.54	4.73	59
80	36.53	7.18	71
90	44.21	15.8	111
100	40.49	51.61	376

ตารางที่ 9-22 ผลทดสอบการเก็บข้อมูลแบบ HOLAP กับคิวบ์ 3 ล้านเรคอร์ด 4 ไดเมนชัน

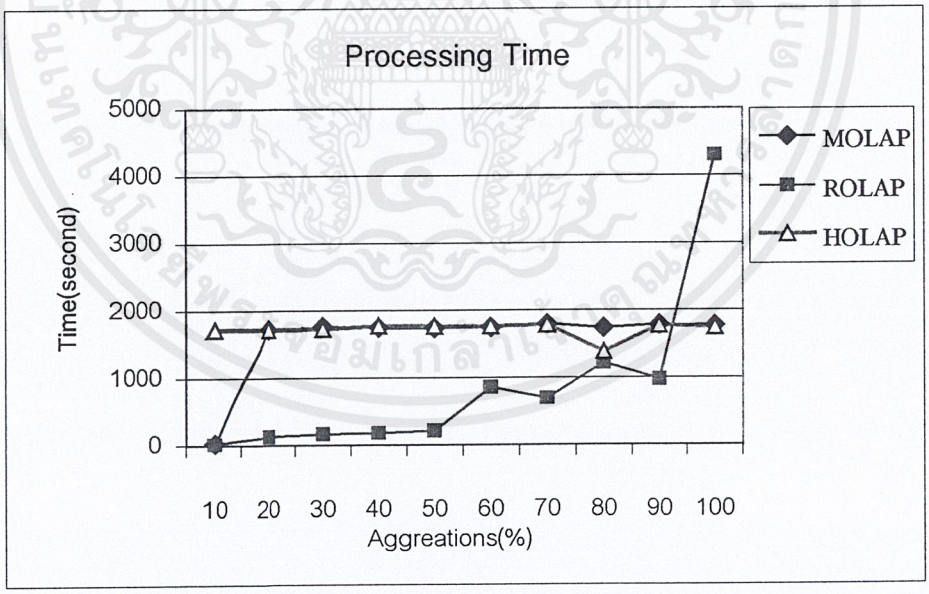
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์จากกราฟ

เปลี่ยนค่าที่ได้ในรูปชั่วโมงให้อยู่ในรูปวินาที

Process Time (second)		
MOLAP	ROLAP	HOLAP
29	21	1721
1724	135	1723
1760	171	1729
1747	184	1774
1738	213	1766
1749	858	1766
1801	693	1788
1738	1223	1387
1785	974	1776
1773	4306	1751

ตารางที่ 9-23 ตาราง Process Time ของสกีมาแบบ 2 ไดเมนชันที่อยู่ในหน่วยวินาที

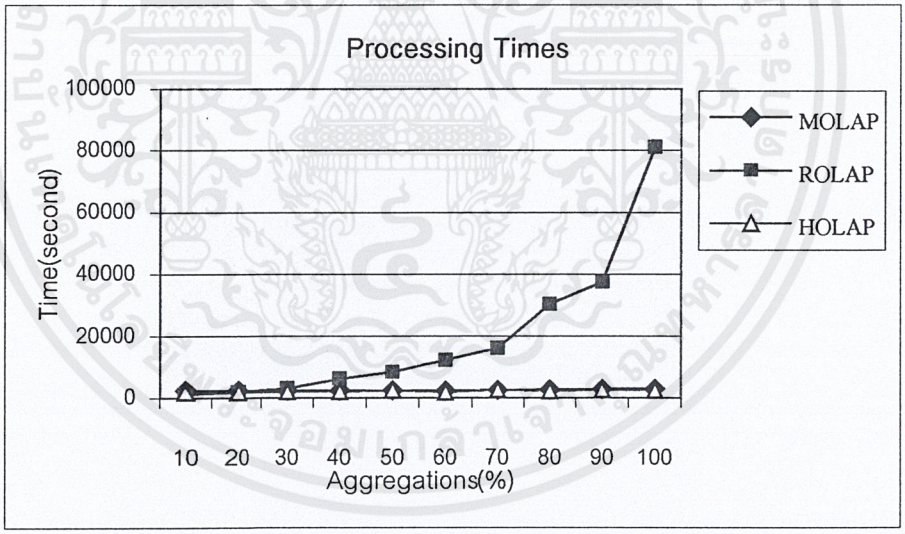


รูปที่ 9-6 กราฟ แสดงเวลาในการสร้างคิวบ์ต่อการเพิ่มของการรวมค่า (2 ไดเมนชัน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Process Time (second)		
MOLAP	ROLAP	HOLAP
2422	1250	1803
2451	1916	1802
2446	3055	1989
2437	6155	2075
2439	8453	2540
2368	12325	1865
2521	16225	2634
2760	30444	2213
2900	37463	2661
2889	81270	2449

ตารางที่ 9-24 ตาราง Process Time ของสกีมาแบบ 4 ไดเมนชันที่อยู่ในหน่วยวินาที



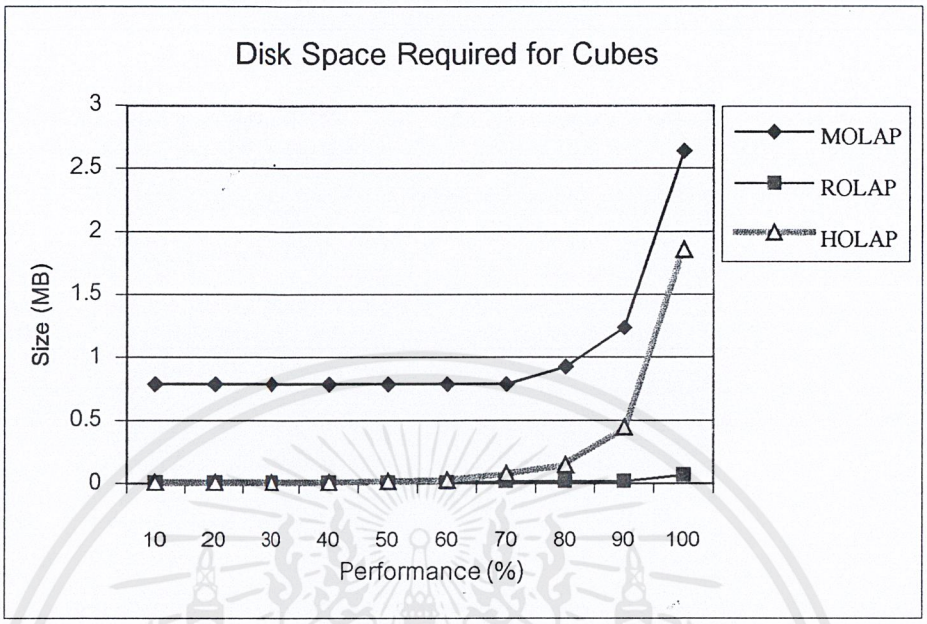
รูปที่ 9-7 กราฟแสดงเวลาในการสร้างคิวบ์ต่อการเพิ่มของการรวมค่า (4 ไดเมนชัน)

สรุปจากกราฟ รูปที่ 9-6 และ 9-7

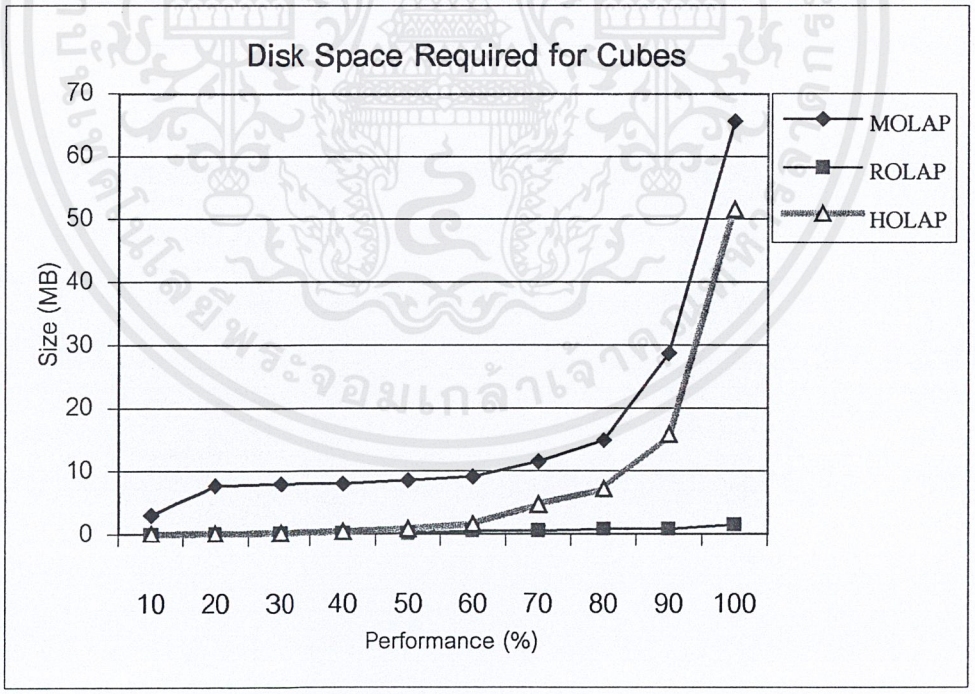
- ในตอนต้นๆ การจัดเก็บข้อมูลแบบ ROLAP ดูเหมือนว่าจะให้เวลาในการสร้างคิวบ์ได้เร็ว แต่ความเป็นจริงเนื่องจากช่วงแรกยังไม่มีการสร้างข้อมูลรวมค่า (Aggregated data)
- เมื่อการรวมค่าเพิ่มขึ้น ROLAP ยิ่งใช้เวลาเพิ่มขึ้นมากกว่าเดิม มากกว่า MOLAP และ HOLAP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แม้ว่าการรวมค่าจะเพิ่มขึ้นแต่ MOLAP และ HOLAP มีการเปลี่ยนแปลงของเวลาในการสร้างคิวบ์คงที่ มีการเพิ่มขึ้นของเวลาน้อยมากหรือแทบไม่มีเลย



รูปที่ 9-8 รูปแสดงพื้นที่ที่ใช้บน Analysis Service (2ไดเมนชัน)



รูปที่ 9-9 รูปแสดงพื้นที่ที่ใช้บน Analysis Service (4 ไดเมนชัน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปจากกราฟรูปที่ 9-8 และ 9-9

- MOLAP ใช้พื้นที่ในการเก็บส่วนนี้มากกว่า ROLAP และ HOLAP เนื่องจากคิวบ์ MOLAP เก็บตารางแฟ็คและ ไดเมนชันไว้ในคิวบ์ด้วย
- HOLAP ใช้พื้นที่ในการเก็บน้อยกว่า MOLAP เนื่องจาก HOLAP ไม่ได้เก็บตารางแฟ็คและไดเมนชันไว้ใน Analysis Service ส่วนค่าการรวมค่าเก็บไว้ใน Analysis Service ด้วยมัลติไดเมนชันแนลดาต้าเบส
- ROLAP ใช้พื้นที่ในการเก็บน้อยสุดเนื่องจากเก็บส่วนตารางแฟ็ค,ไดเมนชันและส่วนค่าการรวมค่าไว้ในรีเลชันแนลดาต้าเบส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

Multidimensional Expressions (MDX)

1. Introduction MDX

MDX หรือ Multidimensional Expression เป็นภาษาที่ใช้ในการกำหนดโครงสร้างและปฏิบัติการกับข้อมูลที่มีลักษณะเป็นมัลติไดเมนชันแนล MDX นี้จะมีหลายส่วนที่คล้ายคลึงกับภาษา SQL แต่ MDX ไม่ใช่ส่วนขยายของ SQL แต่อย่างใด บางฟังก์ชันที่ทำใน MDX ได้ก็อาจจะสามารถทำใน SQL ได้เช่นเดียวกันแต่อาจจะทำได้อย่างไม่มีประสิทธิภาพเท่าใดนัก

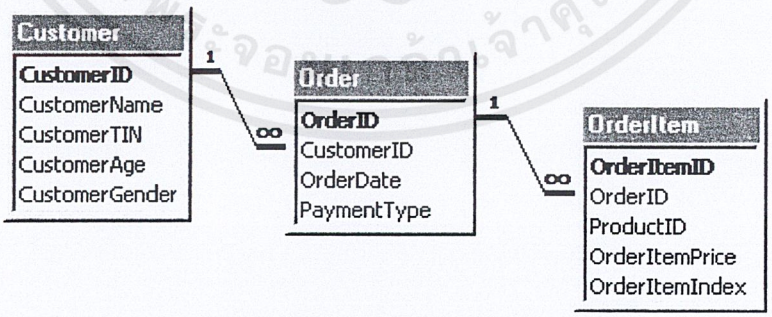
เช่นเดียวกันกับภาษา SQL ,. ภาษา MDX ก็ต้องใช้คีย์เวิร์ด SELECT , FROM ,WHERE ด้วยเช่นกัน โดยคีย์เวิร์ดต่างๆ ใน MDX นี้จะใช้ในการดึงข้อมูลจากคิวบ์เพื่อใช้ทำการวิเคราะห์ MDX ก็จะมีรูปแบบของ Data Definition Language (DDL) สำหรับจัดการโครงสร้างของข้อมูล ซึ่งก็จะมีคำสั่งที่ใช้ในการสร้างหรือลบคิวบ์, ไดเมนชัน, ตัววัด, และออบเจ็กต์ อื่นๆ

2. Key Concept in MDX

วัตถุประสงค์ของ MDX คือ ทำให้สามารถเข้าถึงข้อมูลจากหลายๆ ไดเมนชันได้ง่ายและดีกว่า

2.1 Dimensions, Levels, Members, and Measures

เกือบจะทุกภาษาที่ใช้เกี่ยวกับฐานข้อมูลสำหรับกำหนดโครงสร้างและจัดการข้อมูล เช่น SQL นั้น ได้ถูกออกแบบมาเพื่อดึงข้อมูลออกมาในลักษณะสองไดเมนชัน นั่นคือ เป็น ไดเมนชันของคอลัมน์ และ ไดเมนชันของแถว (row) รูปข้างล่างนี้จะแสดงลักษณะการใช้งานของรีเลชันแนลดาต้าเบสทั่วๆไป

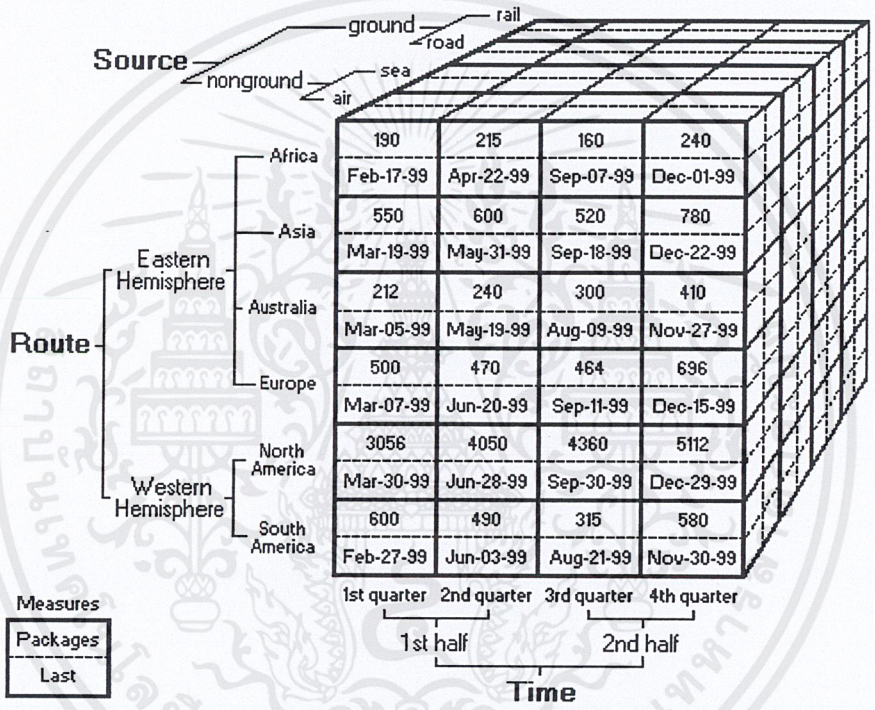


รูปที่ ก-1 แสดงโครงสร้างของรีเลชันแนลดาต้าเบส (relational database)

จากรูปแต่ละตารางจะแสดงข้อมูลในลักษณะสองไดเมนชัน โดยใน SQL นั้นเราสามารถที่จะระบุคอลัมน์ที่ต้องการได้โดยใช้คีย์เวิร์ด SELECT ในขณะที่เราสามารถกำหนดจำนวนแถวที่ต้องการให้ส่งกลับออกมาได้โดยใส่เงื่อนไขไว้ใน WHERE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทางตรงกันข้ามกับข้อมูลที่มีแบบมัลติไดเมนชันแนล (Multidimensional data) ซึ่งข้อมูลสามารถจัดโครงสร้างได้มากกว่าสองไดเมนชัน โครงสร้างในลักษณะนี้จะถูกเรียกว่าคิวบ์ที่จุดอินเตอร์เซค (intersect) กันของไดเมนชันทั้งหลายภายในคิวบ์ อาจจะมีข้อมูลมากกว่า 1 อิลิเมนท์ (element) ซึ่งจะเรียกว่าตัววัด (measure) รูปข้างที่ ก-2 จะแสดงคิวบ์ซึ่งประกอบไปด้วย 3 ไดเมนชัน คือ Route, Service และ Time และประกอบไปด้วย 2 ตัววัด คือ Packages และ Last ซึ่งแต่ละไดเมนชันสามารถแตกย่อยลงไปได้หลายเลเวล โดยที่แต่ละเลเวลก็ยังสามารถแตกย่อยลงไปได้ถึงระดับสมาชิก (members) ได้เช่นกัน ตัวอย่างเช่น ไดเมนชัน Source เราสามารถเลือกสนใจในระดับเลเวลที่เป็น Eastern Hemisphere ซึ่งยังสามารถแตกย่อยได้เป็น 4 สมาชิก คือ Africa, Asia, Australia และ Europe



รูปที่ ก-2 แสดงโครงสร้างของคิวบ์

การจะเข้าใจ MDX ได้ จำเป็นอย่างยิ่งที่จะต้องรู้และเข้าใจคอนเซปต์ของคิวบ์ (Cube), ไดเมนชัน (Dimension), เลเวล (Level), สมาชิก (Members) และตัววัด (Measures)

2.2 Cell, Tuples, and Sets

ใน SQL จะรีเทิร์นซับเซ็ตของข้อมูลจากตารางในลักษณะ 2 ไดเมนชัน ในขณะที่ MDX จะรีเทิร์นซับเซ็ตของข้อมูลจากคิวบ์ในลักษณะมัลติไดเมนชันแนล (multidimensional data)

ในรูปที่ ก-2 ได้แสดงให้เห็นว่า ที่จุดอินเตอร์เซ็คของ multidimensional members จะทำให้เกิด cells ซึ่งจะมีข้อมูลอยู่ภายใน ในการที่จะระบุและดึงข้อมูลนั้น ใน MDX จะใช้สิ่งที่เรียกว่า tuples, Tuples จะแสดง dimensions และ members ในการที่จะระบุ cell ใดเซลล์หนึ่ง และด้วยสาเหตุที่ว่า cell คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดตัดของไดเมนชันต่างๆ ภายในคิวบ์ดังนั้นจึงทำให้ tuple สามารถระบุเซลล์ทุกเซลล์ภายในคิวบ์ได้อย่างเป็นเอกลักษณ์ (unique) ส่วน measures จะถูกจัดการเป็นลักษณะของ private dimension ที่มีชื่อว่า Measures ดังนั้นในการอ้างถึง measures เราก็จะไปอ้างข้อมูลจากไดเมนชัน Measures ตัวอย่างการใช้งาน เช่น

(Source.[Eastern Hemisphere].Africa, Time.[2nd half].[4th quarter], Route.Air, Measures.Packages)

tuple ไม่จำเป็นที่ต้องเกิดจากการระบุรวมทุกไดเมนชันภายใน cube ก็ได้ เช่นตัวอย่างต่อไปนี้
(Source.[Eastern Hemisphere])
(Time.[2nd half], Source.[Western Hemisphere])

การรวม tuple เข้าด้วยกัน เราจะเรียกว่าเซต (set) ในภาษา MDX . ไดเมนชันที่เป็น axis และ slicer จะประกอบไปด้วยเซตของ tuple ตัวอย่างของเซตของ tuple เช่น
{ (Time.[1st half].[1st quarter]), Time.[2nd half].[3rd quarter]) }

นอกจากนี้ เรายังสามารถสร้าง named set ได้ด้วย ซึ่ง named set ก็คือเซตที่ได้มีการตั้งชื่อเอาไว้ เพื่อทำให้ MDX query สามารถเข้าใจได้ง่ายขึ้น

2.3 Axis and Slicer Dimensions

ในภาษา SQL เรามักจะต้องกำหนดจำนวนข้อมูลที่รีเทิร์นกลับมาจากตารางเสมอ โดยการใช้คีย์เวิร์ด SELECT ในการกำหนดจำนวนคอลัมน์ที่รีเทิร์น และใช้คีย์เวิร์ด WHERE ในการระบุจำนวน row ที่รีเทิร์น

ในภาษา MDX ก็จะใช้หลักการในลักษณะเดียวกันนี้ นั่นคือ SELECT statement ใช้สำหรับเลือก dimensions และ members ที่จะรีเทิร์นกลับมา เราจะเรียกนี้ว่า **axis dimensions** ในขณะที่ WHERE statement จะใช้ในการกำหนดจำนวนข้อมูลที่รีเทิร์นกลับมา โดยจะให้รีเทิร์นกลับมาเฉพาะ dimension และ member ที่ตรงตามเงื่อนไขของเรา ซึ่งเราจะเรียกนี้ว่า **slicer dimension**
Axis dimension ถูกคาดการว่าจะรีเทิร์นข้อมูลออกมาในลักษณะของหลาย members ในขณะที่ slicer dimension ถูกคาดการว่าจะรีเทิร์นข้อมูลออกมาในลักษณะของ member เดียว

2.4 Calculated Members

Calculaed members คือ members ที่ไม่ใช่ตัวข้อมูลที่เก็บอยู่ แต่เป็นสมการการคำนวณโดย ใช้ฟังก์ชันต่างๆ ที่มีมาพร้อมกับ MDX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 Comparison of SQL and MDX

ถ้ามองโดยผิวเผินแล้ว MDX จะมีหลายๆ อย่างซึ่งคล้ายกับ SQL หลายๆ ฟังก์ชันที่มีปรากฏใน MDX ก็จะมีลักษณะคล้ายๆ กับฟังก์ชันที่ปรากฏใน SQL เช่นกัน อย่างไรก็ตาม ข้อแตกต่างระหว่าง MDX และ SQL ก็มีอยู่มาก ดังนั้นจำเป็นที่ต้องคำนึงถึงความแตกต่างในระดับคอนเซปต์นี้ด้วย สิ่งหนึ่งที่จะแสดงให้เห็นถึงความไม่เหมือนกันระหว่าง SQL กับ MDX ก็คือ ความสามารถในการอ้างถึงข้อมูล โดย MDX จะอ้างถึงข้อมูลที่เป็นที่มีลักษณะหลายๆ ไดเมนชัน ถึงแม้ว่าเราจะสามารถใช้ SQL ในการคิวรีคิวบ์บน Microsoft SQL Server2000 Analysis Services ได้ก็ตาม แต่เราจะสามารถทำคิวรีได้อย่างมีประสิทธิภาพกว่าเมื่อ เราใช้ MDX เนื่องจากว่า MDX ได้ถูกออกแบบมาเป็นพิเศษสำหรับการดึงข้อมูลที่มีลักษณะเป็นแบบมัลติไดเมนชันแนล ซึ่งจะประกอบไปด้วยหลายไดเมนชัน

เมื่อเราทำการคิวรี ใน SQL จะอ้างถึงข้อมูลเพียงแค่สองไดเมนชัน คือ row และ column ในขณะที่ MDX สามารถทำได้ทั้ง หนึ่ง, สอง หรือมากกว่า โดยคำว่า row และ column จะเปรียบได้กับชื่อของ axis dimension สองอันแรกใน MDX ส่วนไดเมนชันอื่นๆ ก็จะมีชื่อต่างๆ กันไป ในปัจจุบันนี้ทูลหลายๆ ตัวที่ใช้สำหรับ OLAP จะมีความสามารถแสดงผลได้เพียงแค่สองไดเมนชันเท่านั้น

ในภาษา SQL , เราจะใช้ SELECT ในการกำหนด column ที่ต้องการให้รีเทิร์นกลับออกมา และใช้ WHERE เพื่อระบุ row ที่ต้องการ เมื่อมาเปรียบกับ MDX , SELECT จะเอาไว้ใช้กำหนดจำนวน axis dimensions ในขณะที่ WHERE เอาไว้สำหรับกำหนดเงื่อนไขเลือก dimension หรือ member ที่เราต้องการ

ในภาษา SQL , เราจะใช้ WHERE clause ในการกรองข้อมูลที่ต้องการให้รีเทิร์นออกมา ในขณะที่เราใช้ WHERE clause เพื่อให้เกิดการตัดบางส่วนของข้อมูล (slice) ที่จะรีเทิร์นออกมาใน MDX เราจะเห็นได้ว่าทั้งสองอย่างนี้จะมีความคล้ายคลึงกันในระดับคอนเซปต์ แต่จำไว้ว่ามันไม่เหมือนกัน

3. Basic MDX

โครงสร้างพื้นฐานของภาษา MDX ที่เราจะใช้ในการคิวรีเป็นดังนี้

```
SELECT [<axis_specification>
        [, <axis_specification>… ]]
FROM  [<cube_specificaion>]
[WHERE  [<slicer_specification>] ]
```

3.1 โครงสร้างของ SELECT statement ใน MDX

ในภาษา MDX จะใช้ประโยคคำสั่ง SELECT ในการระบุดาต้าเซต (data set) ซึ่งจะเก็บซับเซตของข้อมูลมัลติไดเมนชันแนล ในการระบุดาต้าเซต เราจำเป็นต้องกำหนดข้อมูลต่อไปนี้

- จำนวนของ axis ซึ่ง MDX ได้กำหนดไว้ว่าสามารถใส่ได้สูงสุดถึง 128 axis
- Members จากแต่ละไดเมนชัน ที่เราต้องการจะรวมเข้าไปไว้ใน axis
- ชื่อคิวบ์ที่ต้องการจะทำการคิวรี
- Members ใน slicer dimension ซึ่งเอาไว้สำหรับตัดข้อมูลเฉพาะส่วนที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง เช่น

```
SELECT
  { [Measures].[Unit Sales], [Measures].[Store Sales] } ON COLUMNS,
  { [Time].[1997], [Time].[1998] } ON ROWS
FROM Sales
WHERE ( [Store].[USA].[CA] )
```

SELECT - ใช้เพื่อกำหนด axis dimension ตัวอย่างข้างต้นนี้ได้กำหนดไว้ 2 axis dimension

FROM - ใช้เพื่อเลือกแหล่งข้อมูลที่เราต้องการจะดึงข้อมูลมาใช้ โดยแหล่งข้อมูลนี้จะเก็บข้อมูลในลักษณะของมัลติไดเมนชันแนล

WHERE - อาจจะใสหรือไม่ใส่ก็ได้ โดยเอาไว้สำหรับกำหนด slicer dimension ในตัวอย่างข้างต้นเราได้เลือก member ของไดเมนชัน Store มาทำเป็น slicer dimension

นอกจากนี้ใน SELECT statement เราอาจใช้คีย์เวิร์ด WITH รวมทั้งฟังก์ชันต่างๆ ที่มีมาใน MDX ด้วยก็ได้ Syntax ของ MDX ดูๆ แล้วจะคล้ายกับใน SQL แต่อย่างไรก็ตาม ยังมีส่วนที่ไม่เหมือนกันดังนี้

- MDX จะแยกเซตของ tuples หรือ members ออกจากกันโดยใช้ เครื่องหมาย {}
- MDX สามารถกำหนด axis dimension ได้สูงสุดถึง 128 แกน แต่เฉพาะ 5 axis แรกเท่านั้นที่ได้กำหนดชื่อเรียกไว้แล้ว เราสามารถที่จะอ้างถึง axis ได้โดยอาจจะใช้ชื่อเรียกที่กำหนดเอาไว้แล้ว หรืออีกวิธีหนึ่งคือใช้ลำดับในการระบุ ตัวอย่างเช่น เราสามารถที่จะใช้คิวรีนี้แทนคิวรีข้างบนได้ โดยที่ผลลัพธ์เหมือนกัน

  ```
  SELECT
    { [Measures].[Unit Sales], [Measures].[Store Sales] } ON AXIS(0),
    { [Time].[1997], [Time].[1998] } ON AXIS(1)
  FROM Sales
  WHERE ( [Store].[USA].[CA] )
  ```

- ใน MDX, FROM จะเอาไว้อ้างถึงแหล่งข้อมูลเช่นเดียวกันกับใน SQL แต่อย่างไรก็ตาม มันจะไม่เหมือนกับ SQL ตรงที่ FROM clause ใน MDX ถูกกำหนดให้ใช้กับเพียงคิวบ์เดียว แต่เราก็ยังสามารถที่จะดึงข้อมูลจากคิวบ์อื่นได้โดยใช้ฟังก์ชัน LookupCube
- WHERE clause ใช้ระบุ slicer dimension ถ้าเราไม่ได้มีการระบุไดเมนชันไว้ใน WHERE clause , Microsoft SQL Server2000 Analysis Services จะทำงานโดยนำทุกไดเมนชันที่เราไม่ได้ใช้เป็น axis dimension มาใช้เป็น slicer dimension และ member ที่จะกำหนดให้กับแต่ละไดเมนชันนั้นก็จะกำหนดโดยใช้ค่าดีฟอลต์(default)

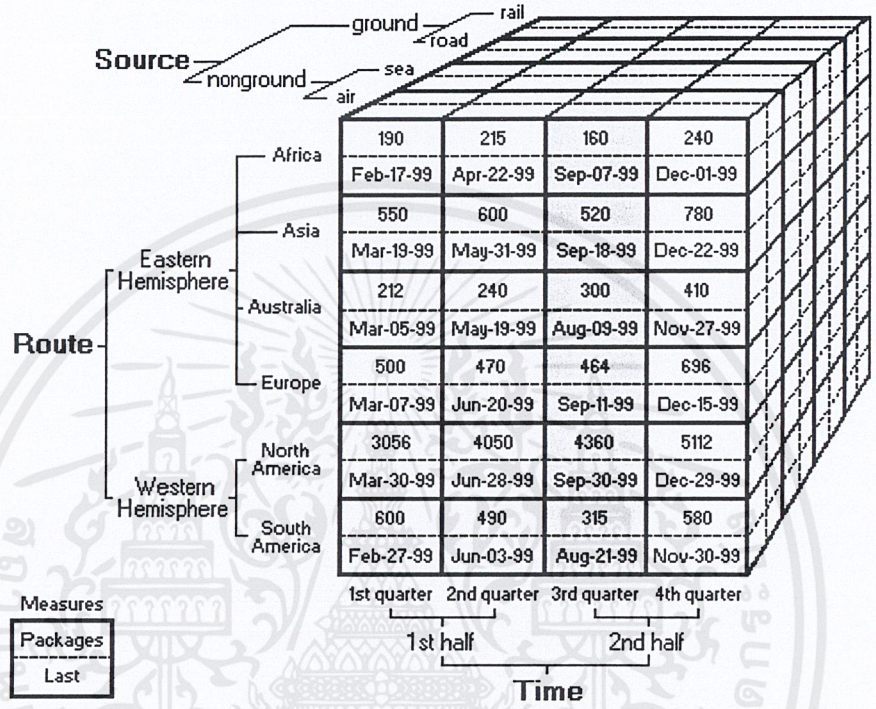
3.2 Members, Tuples, and Sets

ก่อนที่เราจะทำคิวรีโดยใช้ MDX จำเป็นที่จะต้องเข้าใจความหมายของ members, tuples, และเซตเสียก่อน เนื่องจากจะมีการอ้างถึง element เหล่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 Members

Member คือ ส่วนหนึ่งในไดเมนชันที่แสดงถึงข้อมูล โดยอาจจะแสดงข้อมูลเดียวหรือมากกว่า โดยอาจคิดได้ว่า member ในไดเมนชันเปรียบเสมือน เรคอร์ดหนึ่ง (หรือมากกว่า) ที่อยู่ภายในฐานข้อมูล ทั่วๆไป member จะเป็นระดับล่างสุดเมื่อเรากำลังอ้างถึงเซลล์ของข้อมูลภายในคิวบ์
ตัวอย่างข้างล่างนี้ ในส่วนที่แรเงาจะแสดงถึง Time. [2nd half]. [3rd quarter] member



รูปที่ ก-3 แสดงส่วนของ member

เราจะใช้เครื่องหมาย [ ] คร่อมเมื่อชื่อของ member มีช่องว่างหรือตัวเลขรวมอยู่ด้วย ถึงแม้ Time จะเป็นคำธรรมดาที่ไม่ตัวเลขหรือช่องว่างรวมอยู่ แต่เราก็สามารถใส่ [ ] คร่อมได้เช่นกัน ดังเช่น
[Time] . [2nd half]. [4th quarter]

3.2.2 Member Names และ Member Keys

เราสามารถอ้างถึง member ได้โดยใช้ member name หรือ member key ตัวอย่างข้างต้นที่ได้แสดงไปจะอ้างถึง member ในไดเมนชัน Time โดยใช้ member name (4th quarter) อีกวิธีที่จะใช้ในการอ้างถึง member คือการใช้ member key โดยใช้เครื่องหมาย & เพื่อเป็นการบ่งบอกว่าเป็น member key ดังตัวอย่างข้างต่อไปนี้
[Time] . [2nd half] . &[Q4]

โดยการใช้ member key ทำให้เราได้ member ที่ถูกต้องเนื่องจาก member name อาจจะมีซ้ำกัน หรือไดเมนชันอาจจะมีการเปลี่ยนแปลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

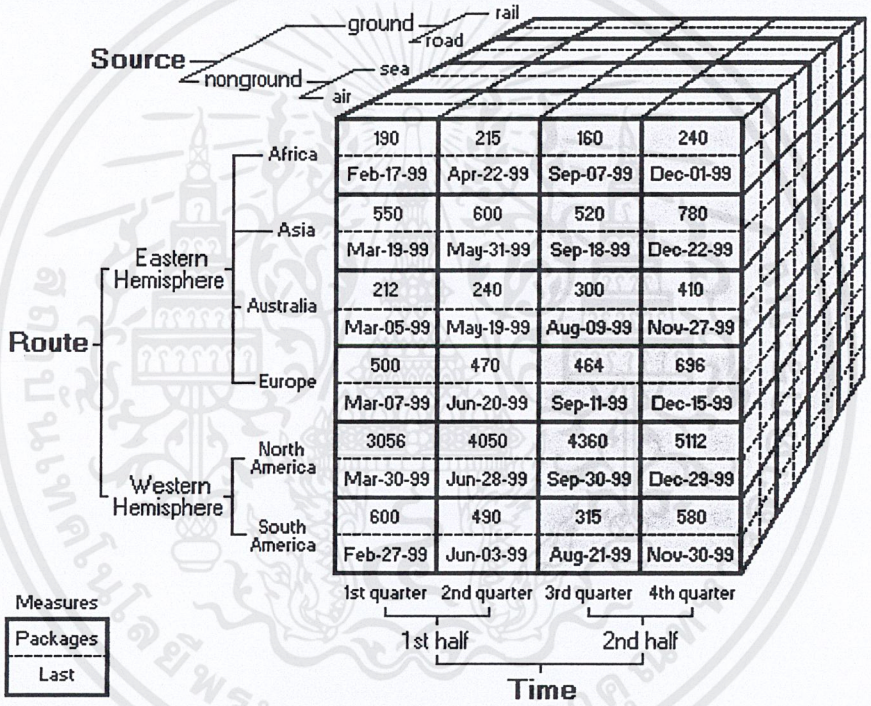
3.2.3 Calculated Members

เราสามารถสร้าง members ใน MDX query โดยให้รีเทิร์นข้อมูลที่เป็นผลมาจากการคำนวณแทนที่จะรีเทิร์นข้อมูลที่เก็บอยู่ในคิวบ์ได้ โดยเราจะเรียก member นี้ว่า calculated members ซึ่งทำให้เกิดความยืดหยุ่นในการใช้ MDX ในการที่จะสร้าง calculated member เราต้องใช้คีย์เวิร์ด WITH เข้ามาร่วมด้วย ตัวอย่างของการสร้าง calculated member แสดงได้ดังข้างล่างนี้

```
WITH MEMBER [Measures] . [PackagesForecast] As
' [Measures] . [Packages] * 1.1 '
```

3.3 Tuples

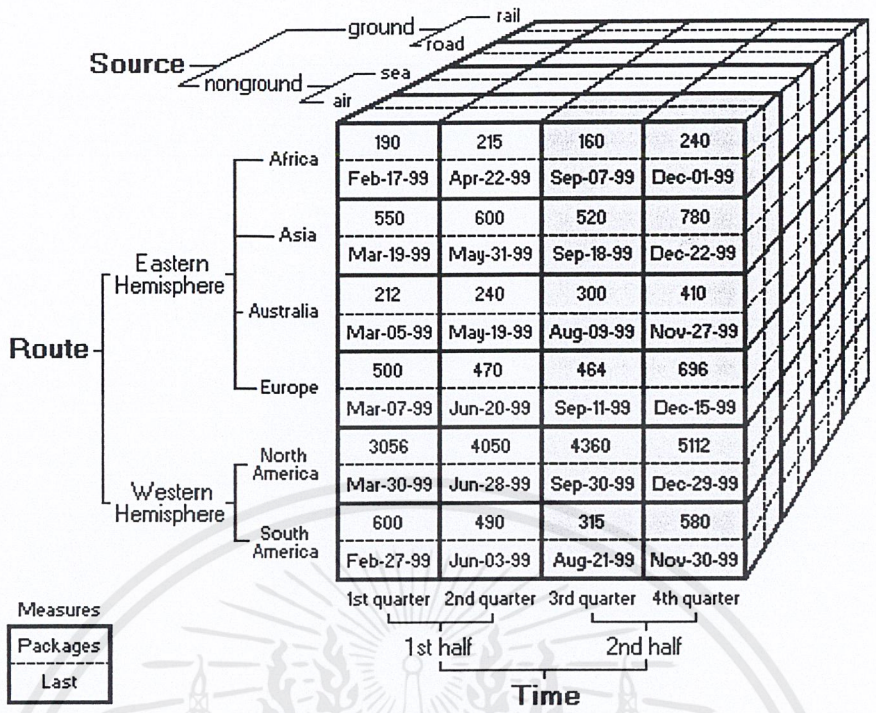
ในการเลือกบางส่วนของข้อมูลจากคิวบ์เราต้องใช้ tuple ซึ่ง tuple ก็คือการนำ member จากหนึ่งไดเมนชันหรือมากกว่ามาประกอบเข้ากัน ตัวอย่างข้างล่างแสดงส่วนของ (Time . [2nd half]) tuple



รูปที่ ก-4 แสดงส่วนของ tuple

รูปข้างล่างนี้จะแสดงส่วนของ (Time . [2nd half] , Route.nonground.air) tuple

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-5 แสดงส่วนของ tuple

ถ้า tuple มีเพียงหนึ่ง member ที่มาจากไดเมนชันเพียงไดเมนชันเดียว เราจะเรียกว่า simple tuple และเราสามารถเขียนในรูปแบบข้างล่างนี้ได้

Time . [2nd half]

แต่ถ้า tuple ประกอบมาด้วยมากกว่าหนึ่ง member และมาจากหลายไดเมนชันแล้ว tuple นี้จะต้องใส่วงเล็บ ( ) คร่อม member แสดงได้ดังนี้

(Time . [2nd half] , Route . nonground . air)

3.4 เซต (Sets)

เซต คือ การรวม tuple เข้าไว้ด้วยกัน อาจจะมีหนึ่ง tuple หรือมากกว่าหรือไม่มีเลยก็ได้เซตจะถูกใช้เพื่อกำหนด axis และ slicer dimension ในคิวรี MDX ตัวอย่างข้างล่างนี้แสดงเซตของ 2 tuple

{ (Time . [1st half] , Route . nonground . air) , (Time . [2nd half] , Route . nonground . sea) }

นอกจากนี้เซตยังสามารถประกอบไปด้วย tuple เดียวกันมากกว่าหนึ่ง tuple ก็ได้ ดังเช่น

{ Time . [2nd half] , Time . [2nd half] }

3.4.1 Set Functions

MDX ได้มีฟังก์ชันมาให้หลายฟังก์ชันในการรีเทิร์นเซต เช่น ใช้ colon (:) ในการแสดงแทนลำดับของ member ในการสร้างเซต จะได้

{ [1st quarter] : [4th quarter] }

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะให้ผลเหมือนกับ

{ [1st quarter] , [2nd quarter], [3rd quarter], [4th quarter] }

colon operator นั้นเป็นฟังก์ชันที่ทำงานในลักษณะนับรวมทั้งต้นและท้าย โดยที่ member ที่เป็นหัวและท้ายของฟังก์ชันก็จะถูกนับรวมเข้ามาเป็นเซตด้วย

3.4.2 Named Sets

Named set คือ เซตที่มีการตั้งชื่อเรียกไว้เรียบร้อยแล้ว named set จะใช้มากใน MDX เพื่อที่จะทำให้ง่ายต่อความเข้าใจและบำรุงรักษาง่าย นั่นคือ กลับมาดูอีกทีก็เข้าใจได้อย่างง่ายและถูกต้อง

3.5 Axis และ Slicer Dimensions

เมื่อเราทำการคิวรีโดยใช้ MDX , ลักษณะการทำงานของโปรแกรมคือ แบ่งเซตของไดเมนชันออกเป็นสองซับเซต คือ Axis dimension และ Slicer dimension

ด้วยการใช้ axis dimension และ slicer dimension ทำให้เราสามารถเลือกส่วนของข้อมูลที่ต้องการให้รีเทิร์นออกมาได้ ตัวอย่างเช่น สมมุติมีคิวบ์ชื่อ TestCube ซึ่งประกอบไปด้วยสองไดเมนชันคือ Route และ Time และ measures ของคิวบ์จะอยู่ในไดเมนชันที่ชื่อ Measures ดังนั้น cube นี้จึงมีทั้งหมดสามไดเมนชัน เมื่อเราต้องการจะคิวรีเพื่อเปรียบเทียบ Packages measures ระหว่าง routes และ times ก็จะสามารถทำได้ดังนี้

```
SELECT
        { Route . nonground . Members }  ON COLUMNS,
        { Time . [1st  half] . Members } ON ROWS
FROM TestCube
WHERE ( [Measures] . [Packages] )
```

ในที่นี้เราใช้ฟังก์ชัน Members เข้ามาช่วย ทำให้เราไม่จำเป็นต้องเขียนทุก member ที่จะใช้ในการสร้าง set ผลลัพธ์ของคิวรีข้างต้นจะออกมาเป็นดังนี้

	air	sea
1st quarter	60	50
2nd quarter	45	45

ตารางที่ ก-1 แสดงผลลัพธ์การคิวรีจากคิวบ์ TestCube

การทำงานของโปรแกรมคือ MDX จะเริ่มจากคำนวณหา axis และ slicer dimension ก่อนเป็นลำดับแรก แล้วจึงมาทำโครงสร้างของผลลัพธ์ที่ต้องรีเทิร์นไป จากนั้นค่อยดึงข้อมูลออกมาจากคิวบ์ที่เราทำการคิวรี

3.5.1 การกำหนด Axis dimension

Axis dimension จะใช้เพื่อระบุเซ็ตของคำตอบ โดย MDX ใช้ SELECT clause ในการกำหนด axis dimension มีรูปแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<axis_specification> :: = <set> ON <axis_name>
<axis_name> :: = COLUNMS | ROWS | PAGES | SECTIONS | CHAPTERS | AXIS(<index>)

โดย MDX อนุญาตให้เราสามารถกำหนด axis dimension ได้มากที่สุดถึง 128 axis แต่ในความเป็นจริงแล้วจะมีน้อยมากที่คิวรีโดยใช้มากกว่า 5 axis การระบุ axis ก็สามารถทำได้โดยใช้ axis name หรือใช้ axis index นั่นคือ AXIS(0), AXIS(1), AXIS(2), AXIS(3) และ AXIS(4) ก็เปรียบได้กับ COLUMNS, ROWS, PAGES, SECTIONS และ CHAPTERS ตามลำดับ ใน MDX ได้กำหนดไว้ว่าเราไม่สามารถข้าม axis ได้ นั่นคือ เราจะกำหนด COLUMNS และ PAGES โดยที่เราไม่กำหนด ROWS ไม่ได้

3.5.2 การกำหนด Slicer Dimension

Slicer dimension ใช้สำหรับกรองเอาเฉพาะข้อมูลที่เราต้องการให้รีเทิร์นกลับมาเป็นผลลัพธ์ โดยเราจะระบุ slicer dimension ใน WHERE clause ของ MDX

โดยไดเมนชันที่ไม่ได้กำหนดเป็น axis dimension จะถูกนำมาใช้เป็น slicer dimension โดยจะใช้ค่าดีฟอล์ตในการระบุ member ซึ่งสามารถกำหนดได้ผ่านทางพร็อพเพอร์ตี้ของ Default Member ใน Analysis Manager (ปกติแล้ว ค่าดีฟอล์ตจะถูกตั้งไว้เป็น All member) รูปแบบของการกำหนด slicer dimension เป็นดังนี้

[WHERE [<slicer_specificaion>]]

slicer dimension จะรับเฉพาะรูปแบบของ expression ที่สามารถหาค่าเป็น tuple เดียวได้ ไม่ได้หมายความว่าเฉพาะ single tuple เท่านั้นที่จะสามารถใช้เป็น slicer dimension ได้ โดย single tuple แสดงได้ดังนี้

WHERE ([Time]. [1st half], [Route]. [nongroud])

เมื่อเรากำหนด slicer dimension เป็นเซ็ตของ tuple , MDX ก็จะพยายามหาค่าของเซตโดยการรวมค่าเซลล์ผลลัพธ์ในแต่ละ tuple เข้าด้วยกันซึ่งทำได้โดยใช้ฟังก์ชัน Aggregate ดังนั้น where clause ข้างล่างนี้ก็จะสามารถให้ผลลัพธ์ได้อย่างถูกต้อง

WHERE { ([Time].[1st half], [Route].[nonground]), ([Time].[1st half], [Route].[ground]) }

3.6 การระบุคิวบ์

เราสามารถระบุคิวบ์ที่เราต้องการคิวรีได้โดยใช้ FROM clause ตัวอย่างจะเป็นดังนี้

FROM SalesCube

โดยที่การระบุชื่อคิวบ์เราสามารถใส่ได้เพียงชื่อเดียวเท่านั้น แต่ก็ไม่ได้หมายความว่าจะเป็นข้อจำกัดในการใช้ข้อมูล เนื่องจากว่าเราสามารถใช้ฟังก์ชัน LookupCube จัดการได้เมื่อเราต้องการข้อมูลจากคิวบ์อื่นด้วย ดังนั้นตัวอย่างการระบุคิวบ์ต่อไปนี้จึงเป็นตัวอย่างที่ผิดเนื่องจาก MDX ไม่ได้อนุญาตให้มีการ join กันดังเช่นใน SQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FROM SalesCube, OtherCube

4. CrossJoin

Crossjoin เป็นฟังก์ชันที่มีประโยชน์อย่างมาก ช่วยให้สามารถจัดการแสดงผลออกมาตามที่เราต้องการได้ ซึ่งรูปแบบของการใช้ฟังก์ชันนี้จะเป็นดังนี้

Crossjoin(«Set1», «Set2»)

หรือเราสามารถเขียนอีกแบบได้ดังนี้

«Set1» * «Set2»

ลำดับของ tuple ในเซตของผลลัพธ์จะขึ้นกับลำดับของ <<Set1>> และ <<Set2>> รวมทั้งลำดับของ member ภายในเซตนั้น

ดังนั้น ถ้า «Set1» = {*x1, x2,...,xn*} and «Set2» = {*y1, y2, ..., yn*} แล้ว ผลลัพธ์ของ **Crossjoin**(*Set1, Set2*) จะเป็นดังนี้

{(*x1, y1*), (*x1, y2*),...,(*x1, yn*), (*x2, y1*), (*x2, y2*),..., (*x2, yn*),..., (*xn, y1*), (*xn, y2*),..., (*xn, yn*)}

ตัวอย่างการใช้ Crossjoin เป็นดังนี้

```
SELECT
      Crossjoin(  [Product].[Product Family].members ,[Store].[Store City].members)
on rows,
       {[Measures].[MeasuresLevel].members}      on columns
FROM sales
```

โดยจะได้ผลลัพธ์ออกมาเป็นดังรูปที่ ก-6

		Unit Sales	Store Cost	Store Sales	Sales Count	Store Sales Net
Drink	Alameda					
	Beverly Hills	1,945.00	1,547.71	฿3,940.54	618	2,392.83
	Los Angeles	2,422.00	1,953.55	฿4,823.88	767	2,870.33
	San Diego	2,560.00	2,014.67	฿5,065.10	797	3,050.43
	San Francisco	175.00	146.34	฿373.72	109	227.38
	Portland	2,371.00	1,880.59	฿4,743.04	751	2,862.45
	Salem	3,735.00	2,955.76	฿7,394.25	1202	4,438.49
	Bellingham	208.00	167.10	฿413.08	130	245.98
	Bremerton	2,288.00	1,860.05	฿4,621.14	730	2,761.09
	Seattle	2,213.00	1,746.74	฿4,370.65	709	2,623.91
	Spokane	2,238.00	1,709.79	฿4,293.30	694	2,583.51
	Tacoma	3,092.00	2,412.35	฿6,092.91	984	3,680.56
	Walla Walla	191.00	143.28	฿355.81	121	212.53
	Yakima	1,159.00	939.29	฿2,348.79	366	1,409.50
Food	Vancouver					
	Victoria					
	Mexico City					
	San Andres					

รูปที่ ก-6 แสดงผลลัพธ์ของคำสั่ง Crossjoin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. การ Drill-Up และ Drill-Down

การ Drill-Up และ Drill-Down สามารถแบ่งออกได้เป็น 2 ประเภทด้วยกัน คือ

5.1 Drill by Member มีอยู่ 2 อย่าง ได้แก่

5.1.1 DrillDownMember () เป็นการดูรายละเอียดข้อมูลในระดับที่ลึกลงไป 1 ระดับ เฉพาะสมาชิกที่ระบุชื่อไว้เท่านั้น

Syntax : DrillDownMember (Set1, Set2 [, Recursive])

ตัวอย่าง : DrillDownMember ({ Ford , Honda } , { Ford})

ผลลัพธ์ : { Ford , Taurus , Explorer , Honda }

5.1.2 DrillUpMember () เป็นการดูรายละเอียดข้อมูลที่สูงขึ้น 1 ระดับ

Syntax : DrillUpMember (Set1, Set2)

ตัวอย่าง : DrillUpMember ({ Ford, Taurus, [Taurus-2000], Explorer, [Explorer-1999], [Explorer – 2000], Honda } , { Taurus, Explorer})

ผลลัพธ์ : { Ford, Honda, Taurus, Explorer }

5.2 Drill by Level มีอยู่ 2 อย่างเช่นเดียวกันกับ Drill by Member ได้แก่

5.2.1 DrillDownLevel () : เป็นการดูรายละเอียดข้อมูลในระดับที่ลึกลงไปตามชื่อของระดับที่ระบุไว้

Syntax : DrillDownLevel (Set[, Level])

ตัวอย่าง : DrillDownLevel ({ Ford, Honda } , [Car Make])

ผลลัพธ์ : { Ford, Taurus, Explorer, Honda, Accord, City }

5.2.2 DrillUpLevel () : เป็นการดูรายละเอียดข้อมูลในระดับที่สูงขึ้น ตามชื่อของระดับที่ระบุไว้

Syntax : DrillUpLevel (Set [, Level])

ตัวอย่าง : DrillUpLevel ({ Ford, Taurus, [Taurus- 2000], Explorer, [Explorer-1999], [Explorer-2000], Honda, Accord, [Accord-1999], [Accord-2000], City, [City-2000] } , [Car Model])

ผลลัพธ์ : { Ford, Honda, Taurus, Explorer, Accord, City}

นอกจากนี้เรายังสามารถทำการ Drill up หรือ Drill Down ได้อีกวิธีหนึ่งคือ การใช้ ToggleDrillState

5.3 ToggleDrillState

เป็นฟังก์ชันที่รวมการทำงานของ DrillupMember และ DrilldownMember เข้าด้วยกัน รูปแบบการใช้งานจะเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ToggleDrillState(«Set1», «Set2»[, RECURSIVE])

ถ้า member m ของ <<Set2>> ปรากฏอยู่ใน <<Set1>> แล้วการทำงานจะเป็น DrillupMember(«Set1», {*m*})

ถ้า member m ของ <<Set2>> ไม่ได้ปรากฏอยู่ใน <<Set1>> แล้ว การทำงานก็จะเป็น DrilldownMember(«Set1», {*m*}[, RECURSIVE])

ตัวอย่างการใช้ ToggleDrillState

เมื่อเราทำการคิวรีด้วย MDX นี้

```
Select Measures.members on columns,
        [Store].[(All)].members on rows
From Sales
```

จะได้ผลลัพธ์ดังรูปที่ ก-7

	Unit Sales	Store Cost	Store Sales	Sales Count	Store Sales I
All Stores	266,773.00	225,627.23	฿565,238.1:	86837	339,610.90

รูปที่ ก-7 แสดงผลลัพธ์การคิวรี

และเมื่อคิวรีโดยการใช้ ToggleDrillState ด้วย MDX ข้างล่างนี้

```
Select Measures.members on columns,
        ToggleDrillState({[Store].[(All)].members},{[Store].[All Store]})
        on rows
From Sales
```

จะได้ผลลัพธ์ดังรูปที่ ก-8 ซึ่งจะเป็นการ Drill down

	Unit Sales	Store Cost	Store Sales	Sales Count	Store Sales I
All Stores	266,773.00	225,627.23	฿565,238.1:	86837	339,610.90
Canada					
Mexico					
USA	266,773.00	225,627.23	฿565,238.1:	86837	339,610.90

รูปที่ ก-8 แสดงผลลัพธ์การคิวรีเมื่อใช้ ToggleDrillState

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
คู่มือสำหรับโปรแกรมเมอร์

1. การพัฒนาโปรแกรมประยุกต์ทำงานผ่านเน็ตเวิร์คแลน

โปรแกรมประยุกต์ได้พัฒนาขึ้นมาโดยใช้ภาษา Visual Basic 6.0 โดยสิ่งที่จำเป็นต้องทำก่อนก็คือ ติดตั้ง PivotTable Service ซึ่งสามารถติดตั้งในได้จากแผ่น CD Microsoft SQL Server 2000 และต้องมีการลงทะเบียนไฟล์ msmdcb.ocx บนเครื่องคอมพิวเตอร์เสียก่อน เพราะในโปรแกรมได้เรียกใช้คอมโพเนนท์ในไฟล์นี้

หลังจากนั้นเวลาเขียนโปรแกรมในภาษา Visual Basic ก็ต้องไปเพิ่มคอมโพเนนท์ OLAP Manager Cube Browser เข้ามา โดยคลิกที่ Project -> Component แล้วที่แท็ป Controls ให้คลิกเลือก OLAP Manager Cube Browser

1.1 ฟอร์มทั้งหมดที่ใช้ในมีในโปรแกรม

- LoginForm : เป็นฟอร์มแรกของการทำงานทำหน้าที่รับชื่อเซิร์ฟเวอร์ ชื่อล็อกอิน (Login) และรหัสผ่าน จากผู้ใช้
- CubeBrowserForm : เป็นฟอร์มหลักของโปรแกรมทำหน้าที่ติดต่อกับ Analysis Server และเป็นตัวทำ Slice , Dice , Drill up และ Drill down

1.2 LoginForm

ตัวแปรที่ใช้

txtserv As String	ใช้เก็บชื่อเซิร์ฟเวอร์ที่ผู้ใช้ใส่เข้ามาทางช่อง Server
txtlog As String	ใช้เก็บชื่อล็อกอินของผู้ใช้
txtpwd As String	ใช้เก็บรหัสผ่านของผู้ใช้

Procedure และ Function

- **Private Sub cmdOk_Click()**
  - เก็บชื่อเซิร์ฟเวอร์ ชื่อล็อกอินและรหัสผ่านของผู้ใช้ ใส่ยังตัวแปร txtserv, txtlog, txtpwd
  - เรียกฟอร์ม CubeBrowserForm ขึ้นมาทำงาน และปิดฟอร์มตัวเองลง

1.3 CubeBrowserForm

Procedure และ Function

- **Public Sub initDatabase()**

ตัวแปรที่ใช้ในฟังก์ชัน

cnn As ADODB.Connection	ใช้สร้างการเชื่อมต่อไปยังเซิร์ฟเวอร์
rstSchema As ADODB.Recordset	ใช้สำหรับดูโครงสร้าง (schema)
strCnn As String	ใช้เป็น Connection String

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงาน

- สร้าง Connection String โดยใช้ตัวแปรที่ได้จาก LoginForm
- สร้างการเชื่อมต่อไปยัง Analysis Server โดยใช้ Connectioin String
- ดูชื่อฐานข้อมูลที่มีอยู่ในเซิร์ฟเวอร์ที่ติดต่อด้วย โดยใช้เมธอด OpenSchema
- ใส่ชื่อฐานข้อมูลลงในคอมโบบ๊อกซ์ (Combo Box)

- **Public Sub initCube()**

ตัวแปรที่ใช้ในฟังก์ชัน

cnn As ADODB.Connection	ใช้สร้างการเชื่อมต่อไปยังเซิร์ฟเวอร์
rstSchema As ADODB.Recordset	ใช้สำหรับดูโครงสร้าง (schema) ของฐานข้อมูล
strCnn As String	ใช้เป็น Connection String

การทำงาน

- สร้าง Connection String โดยใช้ค่าตัวแปรที่ได้จากคอมโบบ๊อกซ์ของฐานข้อมูล
- เชื่อมต่อไปยัง Analysis Server โดยใช้ Connection String
- แสดงชื่อคิวบ์ที่มีในฐานข้อมูล โดยใช้เมธอด OpenSchema(adSchemaCubes)
- ใส่ชื่อคิวบ์ลงในคอมโบบ๊อกซ์ Cube Name

- **Private Sub cboCube_Click()**
  - สร้าง Connection String โดยนำชื่อฐานข้อมูล และชื่อคิวบ์มาจากคอมโบบ๊อกซ์
  - เชื่อมต่อออบเจ็กต์ Connection เข้ากับคอมโพเนนท์ CubeBrowser
- **Private Sub cboDatabase_Click()**
  - เมื่อผู้ใช้เลือกฐานข้อมูลแล้ว โปรแกรมจะไปเรียกโพรซีเยอร์ initCube
- **Private Sub Form_Load()**
  - เมื่อฟอร์มนี้ถูกเรียก โปรแกรมจะไปเรียกโพรซีเยอร์ initDatabase
- **Private Sub Form_Unload(Cancel As Integer)**
  - เมื่อฟอร์มถูกปิดก็จะปิดการเชื่อมต่อกับเซิร์ฟเวอร์

2. การพัฒนาโปรแกรมประยุกต์ทำงานผ่านเว็บบราวเซอร์

โปรแกรมประยุกต์ที่ทำงานผ่านเว็บบราวเซอร์ ได้ถูกพัฒนาขึ้นโดยใช้ Active Server Page และ Java Script โดยใช้ Active Server Page สำหรับติดต่อกับ Analysis server จัดการกับการคิวรีด้วย MDX ส่วน Java Script เอาไว้สำหรับจัดการในส่วนแสดงผลและโต้ตอบกับผู้ใช้ เช่น การลากไปวาง (drag drop) การดับเบิลคลิก

ไฟล์ที่ใช้สำหรับโปรแกรม

- **SelectServer.asp**

การทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รับชื่อเซิร์ฟเวอร์จากผู้ใช้ ผ่านทางเท็กท์บ๊อกซ์ (Text Box) โดยมีค่าดีฟอลต์เป็น Localhost
- ส่งชื่อเซิร์ฟเวอร์ไปให้ chooseDB.asp โดยผ่าน Query String

- **chooseDB.asp**

การทำงาน

- รับชื่อเซิร์ฟเวอร์จาก Query string ที่ส่งมาจาก SelectServer.asp
- สร้างออบเจ็กต์ **Connection** และ **Recordset** ของ ADODB
- เรียกเมธอด Open ของออบเจ็กต์ **Connection** เพื่อสร้างการเชื่อมต่อไปยัง Analysis Server โดยใน Connection string ใส่ค่า Data Source เป็นชื่อเซิร์ฟเวอร์ที่ได้จาก query string
- เรียกเมธอด OpenSchema ของออบเจ็กต์ **Connection** โดยส่งค่าพารามิเตอร์เป็น adSchemaCatalogs(กำหนดค่าคงที่เป็น 1) เพื่อที่จะแสดงชื่อฐานข้อมูลทั้งหมดที่มี และนำผลลัพธ์ใส่ไว้ในออบเจ็กต์ **Recordset**
- วนลูปนำชื่อฐานข้อมูลใส่ลงในเมนูลิสต์

- **chooosCube.asp**

การทำงาน

- รับชื่อเซิร์ฟเวอร์และชื่อฐานข้อมูลจาก Query string ที่ส่งมาจาก chooseDB.asp
- สร้างออปเจ็กต์ **Connection** และ **Recordset** ของ ADODB
- เรียกเมธอด Open ของออบเจ็กต์ **Connection** เพื่อสร้างการเชื่อมต่อไปยัง Analysis Server โดยใน Connection string ใส่ค่า Data Source เป็นชื่อเซิร์ฟเวอร์ที่ได้จาก query string และใส่ค่า Initial Catalog เป็นชื่อฐานข้อมูลที่ได้จาก query string เช่นกัน
- เรียกเมธอด OpenSchema ของออบเจ็กต์ **Connection** โดยส่งค่าพารามิเตอร์เป็น adSchemaCubes(มีค่าคงที่เป็น 32) เพื่อที่จะแสดงรายชื่อคิวบ์ทั้งหมดที่มี และนำผลลัพธ์ใส่ไว้ในออบเจ็กต์ **Recordset** เราสามารถเรียกดูชื่อคิวบ์ที่เก็บไว้ในออบเจ็กต์นี้ได้โดยใช้ Recordset.Field(2).Value
- วนลูปนำชื่อคิวบ์ใส่ลงในเมนูลิสต์
- กำหนดค่า input ใน HTML โดยให้กำหนดชนิดเป็นแบบ hidden เนื่องจากต้องการให้เป็นเพียงตัวแปรและสามารถส่งไปมาระหว่างไฟล์ได้ โดยไม่ได้มีความจำเป็นที่ต้องให้ผู้ใช้เห็นค่า

  Input ที่กำหนดมีความหมายดังต่อไปนี้

  - server เก็บชื่อเซิร์ฟเวอร์ ที่ผู้ใช้เลือกตั้งแต่เริ่มทำงาน
  - database เก็บชื่อฐานข้อมูลที่ผู้ใช้เลือก
  - column dimension เก็บชื่อไดเมนชันที่แสดงผลอยู่บนแกนคอลัมน์ของตารางแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- columns เก็บค่า MDX ที่ใช้สำหรับแกนคอลัมน์
- row dimension เก็บชื่อไดเมนชันที่แสดงผลบนแกนแถว (row)
- rows เก็บค่า MDX ที่ใช้สำหรับแกนแถว
- where เก็บค่า MDX เมื่อผู้ใช้ทำ Slice
- expanDim ระบุว่าแกนไดเมนชันทางซ้ายมือ แกนใดถูกคลิกกางออก
- numRowDim ระบุจำนวนไดเมนชันที่แสดงบนแกนแถวของตารางแสดงผล
- LevelDisplay ใช้สำหรับเก็บค่า MDX ที่ใช้สำหรับการ cross join เมื่อมีการแสดงผลไดเมนชันบนแกนแถวมากกว่า 1 ไดเมนชัน

- เมื่อผู้ใช้กดปุ่ม Submit โปรแกรมจะไปรันไฟล์ MDXQuery.asp

- **MDXQuery.asp**

การทำงาน

- สร้างการเชื่อมต่อไปยัง Analysis Server โดยใช้ชื่อเซิร์ฟเวอร์และชื่อฐานข้อมูลจาก query string ของ server และ database
- ตรวจสอบว่าถ้าหากเป็นการเข้ามารันไฟล์นี้เป็นครั้งแรกก็จะเลือกไดเมนชัน Measures แสดงบนแกนคอลัมน์และเลือกไดเมนชันแรก(ที่ไม่ใช่ไดเมนชัน Measures)ที่อ่านเจอแสดงบนแกนแถว (row)
- สร้างออบเจ็กต์ **Catalog** ของ ADOMD เพื่อที่จะใช้ในการดูรายละเอียดของคิวบ์ (Cube Definition) เชื่อม Catalog.CubeDefs เข้ากับชื่อ Cube ที่ส่งมาจาก query string
- วาดไดเมนชันทุกไดเมนชันออกมาบนหน้าจอเว็บบราวเซอร์ โดยมีการตรวจสอบก่อนว่าไดเมนชันที่จะวาดนี้จะต้องไม่ใช่ไดเมนชันที่แสดงอยู่บนแกนของตารางแสดงผล
- วาดเลเวล (Level) ของไดเมนชันที่มีชื่อตรงกับค่า query string ของExpanDim
- สร้างตารางสำหรับทำ slice dimension
- สร้างประโยค MDX ที่สมบูรณ์ โดยการใช้ค่าจาก query string ของ rows,columns และ where
- สร้างออบเจ็กต์ **Cellset** ของ ADOMD เพื่อใช้ในการส่งค่า MDX ไปให้กับ Analysis Server และรับผลลัพธ์กลับมา
- แสดงผลลัพธ์จากการคิวรีที่อยู่ในออบเจ็กต์ **Cellset** โดยแสดงในรูปแบบของตาราง โดยใช้วิธีการวนลูปแสดงผลตั้งแต่แถวแรกไปจนถึงแถวสุดท้าย
- สร้าง queryForm ซึ่งภายในฟอร์มจะมี input แบบเดียวกันกับที่ได้อธิบายไว้ใน chooseCube.asp เพื่อที่จะใช้เป็นตัวแปรเก็บค่า และส่งค่าไปมาระหว่างไฟล์
- สร้างฟอร์ม สำหรับไปรันกราฟ
- สร้างฟอร์มสำหรับไปรัน Excel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **CubeBrowser.js**

การทำงาน

- **Function init()**

ใช้วิธีการตรวจจับอีเวนต์ (event) ควบคุมการทำงานของเมาส์ (mouse) มีดังนี้

onmousemove หากมีการเคลื่อนเมาส์ให้ไปรันฟังก์ชัน mouseMove

onmousedown เมื่อมีการกดเมาส์ ไปรันฟังก์ชัน mouseDown

onmouseup เมื่อปล่อยเมาส์หลังจากกด ไปรันฟังก์ชัน mouseUp

onmouseover เมื่อเคลื่อนเมาส์มาอยู่ด้านบน ไปรันฟังก์ชัน moueOver

onmouseout เมื่อเคลื่อนเมาส์ออก ไปรันฟังก์ชัน mouseout

onclick หากมีการคลิกเมาส์ ไปรันฟังก์ชัน expand

- **Function insideResultTable()**

เป็นฟังก์ชันที่ทำหน้าที่ตรวจสอบตำแหน่งของเมาส์ว่าอยู่ในตารางแสดงผลหรือไม่

- **Function insideSliceTable()**

เป็นฟังก์ชันที่ทำหน้าที่ตรวจสอบตำแหน่งของเมาส์ว่าอยู่ในตารางสำหรับ slice หรือไม่

- **Function insideXAxis()**

เป็นฟังก์ชันที่ทำหน้าที่ตรวจสอบตำแหน่งของเมาส์อยู่ในแกนคอลัมน์ของตารางแสดงผลหรือไม่

- **Fuction insideYAxis()**

เป็นฟังก์ชันที่ทำหน้าที่ตรวจสอบตำแหน่งของเมาส์อยู่ในแกนแถวของตารางแสดงผลหรือไม่ ในฟังก์ชันมีการเรียกใช้ค่า query string ของ numRowDim เพื่อดูว่ามีกี่ไดเมนชันที่แสดงอยู่บนแกนแถวของตารางแสดงผล ที่ทำแบบนี้เพื่อที่จะได้ตรวจสอบตำแหน่งได้อย่างถูกต้องยิ่งขึ้น

- **Function expand()**

ทำหน้าที่ตรวจสอบว่าตอนนี้เมาส์อยู่ที่ออบเจ็กต์ไดเมนชันหรือไม่ ถ้าใช่ก็จะนำค่าของออบเจ็กต์นั้นมาใส่ใน query string ของ expanDim

- **Function mouseDown()**

เมื่อมีการกดเมาส์ลงจะตรวจสอบว่า เป็นการลากออบเจ็กต์ (drag) หรือไม่ ถ้าใช่ก็จะเก็บค่าตำแหน่งที่กดเมาส์ และตำแหน่งแรกเริ่มของออบเจ็กต์เอาไว้ จากนั้นก็เปลี่ยนค่าของตัวแปร move ให้มีค่าเป็น 1

- **Function mouseMove()**

ถ้าค่าของตัวแปร move มีค่าเป็นหนึ่ง นั่นหมายความว่ากำลังจะเคลื่อนย้ายออบเจ็กต์ ดังนั้นจึงย้ายตำแหน่งของออบเจ็กต์ตามตำแหน่งของเมาส์

- **Function mouseUp()**

มีการตรวจสอบ 3 อย่างคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเป็นการลากไดเมนชัน จะตรวจสอบว่าตำแหน่งที่วางนั้นอยู่ในตารางแสดงผลหรือตารางสำหรับทำ Slice หรือไม่ ลำดับต่อไปคือตรวจสอบว่าตำแหน่งที่วางอยู่บนแกนคอลัมน์หรือแกนแถว ถ้าอยู่บนแกนใดก็จะเปลี่ยนค่า MDX ใน query string ของแกนนั้นๆ แล้วสั่งรันไฟล์ MDXQuery.asp เพื่อแสดงผลลัพธ์

ถ้าเป็นการลากเลเวล นั่นหมายถึงผู้ใช้ต้องการจะแสดงผลในแกนแถวมากกว่า 1 ไดเมนชัน ดังนั้นจำเป็นต้องตรวจสอบว่าตำแหน่งที่วางอยู่บนแกนแถวหรือไม่ ถ้าอยู่ก็จะเปลี่ยนค่า MDX ของ query string ของ row โดยใช้คำสั่ง crossjoin เพื่อนำข้อมูลจากหลายไดเมนชันมาแสดงร่วมกันได้ ตรงส่วนนี้จะมีการนำค่าจาก LevelDisplay เข้ามาใช้ จากนั้นก็สั่งรันไฟล์ MDXQuery.asp เพื่อแสดงผลลัพธ์

ถ้าไม่ใช่ทั้งการลากไดเมนชัน และการลากเลเวล สิ่งที่อาจเป็นไปได้คือการลากไดเมนชันออกจากตารางแสดงผลเพื่อลดจำนวนไดเมนชันที่แสดงบนแกนแถว ดังนั้นต้องตรวจสอบค่า dimRemove ซึ่งจะเป็นตัวบอกว่าลาก ไดเมนชันใดออกจากตารางจากนั้นก็เปลี่ยนค่า MDX แล้วสั่งรันไฟล์ MDXQuery.asp ใหม่

- **Function writeMDX(Level,NumRow)**

ทำหน้าที่เขียนค่า MDX ใหม่หลังจากที่เราได้ลากไดเมนชันออกจากตารางแสดงผล โดยตัดค่าที่เกี่ยวกับไดเมนชันที่ได้ลากออก ใน LevelDisplay ออกไป แล้วนำที่เหลือมา crossjoin กัน

- **Function mouseOver()**

เมื่อเมาส์มาอยู่เหนือ Dimension , Level และ axisYCell นั่นหมายความว่าอาจเป็นไปได้ที่ผู้ใช้กำลังจะลากออบเจ็กต์ใดออบเจ็กต์หนึ่ง ดังนั้นจึงเปลี่ยนค่าตัวแปร drag เป็น 1

- **Function SliceShow()**

ทำหน้าที่สร้างหน้าต่างใหม่ขึ้นมาเพื่อที่จะทำหน้าที่เกี่ยวกับการ Slice โดยจะมีการสร้างฟอร์มเพื่อไปรันไฟล์ SliceBrowser.asp โดยจะมีการส่งค่า query string ไปให้

- **Function drill(evt)**

จะมีการเรียกใช้ฟังก์ชันนี้ เมื่อมีการคลิกในตารางแสดงผล ซึ่งฟังก์ชันนี้จะทำหน้าที่ Drill up และ Drill down โดยใช้คำสั่ง ToggleDrillState

- **DimBrowser.asp**

ในไฟล์นี้จะมีทั้งส่วนการทำงานที่เป็น ASP เพื่อเอาไว้ติดต่อเซิร์ฟเวอร์ ติดต่อฐานข้อมูล คิวบ์ เพื่อดูรายละเอียดของไดเมนชัน และมีการทำงานในส่วนของ Java Script เพื่อใช้เป็นส่วนควบคุมการทำงานของผู้ใช้

การทำงานในส่วนของ ASP

- **Function ConnectToServer()**

ทำหน้าที่เชื่อมต่อกับ Analysis Server โดยใช้ชื่อเซิร์ฟเวอร์และฐานข้อมูลที่ได้จาก query string

- **Function GetCubeDefinition()**

ทำหน้าสร้างออบเจ็กต์ **Catalog** เพื่อที่จะเอาไว้ติดต่อกับคิวบ์ที่ผู้ใช้ได้เลือกเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Function DisplayDimensionMembers()**

ทำหน้าที่วาดไดเมนชันที่ส่งมาจาก query string ออกทางหน้าจอ และหากมีการกางไดเมนชัน ก็จะมีการวาดเลเวลที่กางออกมาด้วย

- **Function DisplayHierarchyMembers(hierarchy, lowestLevel)**

ทำหน้าที่สร้างผลลัพธ์สำหรับสมาชิกที่เป็นลำดับขั้น (Hierarchy members)

- **Function DisplayMember(member, lowestLevel)**

ทำหน้าที่วนแสดงผลแบบรีเคอร์ซีพ (recursive) จนกว่าจะถึงเลเวลที่อยู่ใน query string "level"

การทำงานในส่วนของ Java Script

- **Function init()**

ทำหน้าที่ควบคุมการใช้งานเมาส์

onmousedown ถ้ามีการกดเมาส์ลง ให้ไปรันฟังก์ชัน mouseDown

ondbclick หากมีการดับเบิลคลิก ให้ไปรันฟังก์ชัน mouseDblClick

- **Function mouseDown()**

ทำหน้าที่เปลี่ยนสีตัวที่กด จากสีน้ำเงินให้เป็นสีแดง

- **Function mouseDblClick()**

เปลี่ยนค่า query string ของ level ให้มีค่าเพิ่มขึ้นจากตัวที่คลิก 1 จากนั้นสั่งแสดงผลใหม่

- **Function cmdOK()**

หากผู้ใช้กดปุ่ม OK โปรแกรมก็จะเปลี่ยนค่า query string ของ where จากนั้นก็สั่งรัน MDXQuery.asp เพื่อแสดงผลข้อมูลใหม่

- ***AxisXBrowser.asp , AxisYBrowser.asp***

ไฟล์นี้มีฟังชันการทำงานคล้ายฟังชันใน DimBrowser.asp เพียงแต่แก้ให้เวลาคลิกเลือกสมาชิในไดเมนชันให้เลือกให้หลายตัวในเวลาเดียวซึ่งทำให้สามารถเลือกแกนได้มีความยืดหยุ่นมากขึ้น

- ***MakeCustomChart.asp***

ไฟล์นี้มีฟังชันการทำงานเดียวกันกับ MDXQuery.asp แต่ว่าเขียนเพิ่มในส่วนอินเทอร์เฟสง่ายต่อผู้ใช้ในการสร้างกราฟมากกว่า ใช้ร่วมกับไฟล์ AxisXBrowser.asp , AxisYBrowser.asp และ CubeAxisBrowser.js

- ***Styles.css, customstyles.css***

ไฟล์นี้ไว้เก็บรูปแบบในการแสดงผลของแต่ละส่วน styles.css ใช้ในส่วนการแสดงผลในหน้าจอหลัก ส่วน cusomstyles.css ใช้ในส่วนการสร้างกราฟแบบกำหนดเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **CubeAxisBrowser.js**

ไฟล์ช่วยทำส่วนแสดงผลสวยงาม ไฟล์นี้ใช้ในส่วนการแสดงกราฟแบบกำหนดเอง ใช้ร่วมกับไฟล์ MakeCustomChart.asp

- **MakeChart.asp**

ไฟล์นี้สร้างกราฟจากตารางแสดงผลซึ่งจะให้ แกนคอลัมน์เป็นแกน X และให้แกนแถวเป็นแกน Y เซิร์ฟเวอร์จะต้องติดตั้งโปรแกรมที่สามารถสร้างกราฟผ่านเว็บได้ โปรแกรมที่เอามาใช้ในโปรเจคนี้เป็นโปรแกรมของบริษัท Dundas Software ซึ่งได้ให้ดาวโหลดโปรแกรมฟรีแต่ต้องแลกกับการมีชื่อติดไว้ในกราฟ

- **ExcelReport.asp**

ไฟล์สร้างรายงานลง Excel เพื่อนำไปใช้สร้างกราฟต่อๆไป โดยเครื่องที่จะรันส่วนนี้ต้องโปรแกรม Microsoft Excel อยู่แล้วบนเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

หนังสืออ้างอิง

[1] Harry Singh : " *Data Warehousing : Concepts, Technologies, Implementations and Management* ", Prentice-Hall , 1996.

[2] Olap Train and Reed Jacobson : " *Microsoft SQL Server 2000 Analysis Services Step by Step* ", Microsoft Press , 2001.

[3] อำไพ สินลิขิตกุล : " *อินไซท์ SQL Server 7 Step by Step ครอบคลุมเวอร์ชัน 2000* " , โปรวิชั่น , 2544.

เว็บไซต์อ้างอิง

[1] http://www.msdn.microsoft.com

[2] http://www.dynamicdrive.com

[3] http://www.sqlmag.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้