ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

โปรแกรมวิเคราะห์การขายสินค้าโดยใช้ OLAP บน SQL Server

A Sales Analysis System for OLAP on SQL Server

โดย

นายธนเดช จันทฑีโร

รหัส 44067479

H002125

อาจารย์ที่ปรึกษา

ผศ.ดร.วรพจน์ กรีสุระเดช

วัน เดือน ปี 05 ก.พ. 2550
เลขทะเบียน 02125
เลขเรียกหนังสือ วพ ธ141 ป 2546
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2546
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

รายงานนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	โปรแกรมวิเคราะห์การขายสินค้าโดยใช้ OLAP บน SQL Server
นักศึกษา	นายธนเดช จันทตีโร
อาจารย์ที่ปรึกษา	ผศ.ดร.วรพจน์ กรีสุระเดช
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2546

บทคัดย่อ

กระบวนการในการรวบรวมข้อมูลต่างๆ มาไว้ในที่เดียวกัน ที่เรียกว่า Data Warehouse นั้น ทำให้สามารถเรียกใช้ข้อมูลที่ต้องการได้อย่างสะดวก รวดเร็ว แต่หากต้องการนำข้อมูลมาใช้ให้มีประสิทธิภาพยิ่งขึ้น จะต้องมีกระบวนการนำข้อมูลมาวิเคราะห์ หรือที่เรียกว่า OLAP มาใช้ใน Data Warehouse ด้วย

โครงการพัฒนาระบบงานนี้ เป็นการพัฒนาในส่วน User Interface เพื่อนำ OLAP มาใช้ในงานวิเคราะห์ข้อมูลการขายขององค์กรที่ทำการผลิต และขายสินค้า เพื่อให้ผู้ใช้สามารถนำข้อมูลที่ได้ไปใช้ประโยชน์ในระดับการตัดสินใจ หรือระดับการบริหาร และพัฒนาความสามารถในการแข่งขันในทางธุรกิจขององค์กร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	A Sales Analysis System for OLAP on SQL Server
Student	Mr. Thanadetch Chantateero
Advisor	Assist. Prof. Dr. Worapoj Kreesuradej
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2003

ABSTRACT

The data integration from multiple sources into the Data Warehouse makes the retrieval process of data to easier and faster. To get more efficiency for the use of data, we must have the On-line Analytical Processing or OLAP system.

This development project is use to improve user interface when using OLAP system in the Sales Analysis of the trading organization to enhance the user in the decision making or the management and improve competitive positive of the company.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

สารบัญ III

สารบัญตาราง V

สารบัญภาพ VI

บทที่

1. บทนำ 1

1.1 ความเป็นมาของปัญหา 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ขอบเขตของโครงงาน 3

1.4 ขั้นตอนและแผนงานในการพัฒนา 4

1.5 เครื่องมือที่ใช้ในการพัฒนาระบบงาน 6

1.6 ประโยชน์ที่จะได้รับ 7

1.7 รายละเอียดในบทต่าง ๆ 8

2. ทฤษฎีที่เกี่ยวข้องกับการพัฒนาระบบงาน 10

2.1 Data Warehouse และ OLAP Cube 10

2.2 สถาปัตยกรรมระบบวิเคราะห์ข้อมูลของไมโครซอฟท์ 12

2.3 MDX – Multidimensional Expression Language 24

2.4 ADO-MD Object Library 26

3. การออกแบบการจัดเก็บข้อมูลการขายสินค้าสำหรับทำ OLAP Cube 28

3.1 การคัดเลือกข้อมูลจากระบบการขายที่เป็น OLTP 28

3.2 การออกแบบตารางข้อมูลใน Data Warehouse 28

3.3 กระบวนการนำข้อมูลจาก OLTP เข้าสู่ Data Warehouse 31

3.4 การสร้าง OLAP Cube สำหรับงานวิเคราะห์การขายสินค้า 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4. สถาปัตยกรรม และการพัฒนาโปรแกรม OLAP 40
   4.1 สถาปัตยกรรมด้านฮาร์ดแวร์ 40
   4.2 สถาปัตยกรรมด้านซอฟท์แวร์ 41
   4.3 การออกแบบโปรแกรมสำหรับวิเคราะห์ข้อมูลใน OLAP Cube 43
5. การนำโปรแกรมไปใช้งานและบำรุงรักษาระบบ..... 48
   5.1 การติดตั้งโปรแกรมให้กับผู้ใช้งาน 47
   5.2 การบำรุงรักษาข้อมูลในระบบ 48
   5.3 การจัดการสิทธิ์ของผู้ใช้ข้อมูล 49
6. สรุปผลการศึกษา และข้อเสนอแนะ 51
   6.1 สรุปผลการดำเนินงาน 51
   6.2 ข้อเสนอแนะ 51

บรรณานุกรม 53

ภาคผนวก 54

ประวัติผู้เขียน 66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

3.1 Product Dimension Table 29

3.2 Customer Dimension Table 30

3.3 Time Dimension Table 31

3.4 Sales Fact Table 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

ภาพที่

2.1 กระบวนการของ Data Warehouse และ OLAP 11

3.1 DTS Package สำหรับ Product Dimension Transformation 33

3.2 DTS Package สำหรับ Customer Dimension Transformation 34

3.3 DTS Package สำหรับ Sales Fact Transformation 35

3.4 Product Dimension ในมุมมองด้าน Brand (Brand Hierarchy) 36

3.5 Product Dimension ในมุมมองด้านขนาดบรรจุ (Size Hierarchy) 36

3.6 Product Dimension ในมุมมองด้านรสชาติ (Flavor Hierarchy) 37

3.7 Customer Dimension ในมุมมองด้านปริมาณการขาย (Account Hierarchy) 37

3.8 Customer Dimension ในมุมมองด้านการทำงานของฝ่ายขาย (SalesOrg Hierarchy) 38

3.9 Time Dimension 38

3.10 รูปสมบูรณ์ของ Sales Cube 39

4.1 ฮาร์ดแวร์ทั้งหมดที่เกี่ยวข้องกับระบบ 41

4.2 หน้าจอหลักของโปรแกรม Sales Analysis 44

4.3 หน้าจอ Cube Browser พร้อมทั้ง MDX Builder 44

4.4 แสดงการ Drill Down 45

5.1 การส่ง Package ของโปรแกรมไปติดตั้งยังเครื่องผู้ใช้ 48

5.2 Role สำหรับการเข้าถึงข้อมูลของผู้ใช้กลุ่มต่าง ๆ 50

5.3 การกำหนดสิทธิ์ของผู้ใช้ในกลุ่มต่าง ๆ 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

การนำข้อมูลการขายสินค้าในองค์กรมาใช้งาน หรือนำมาวิเคราะห์ โดยทั่วไปนั้น ยังคงนิยมใช้วิธีการแบบดั้งเดิม คือรวบรวมข้อมูลเท่าที่มีอยู่ นำมาจัดเรียงในเชิงสถิติแล้วจึงนำข้อมูลเหล่านั้นมาให้กับผู้บริหารเพื่อนำไปใช้งานในเชิงวิเคราะห์ ซึ่งวิธีการในการรวบรวมข้อมูลดังกล่าวเป็นวิธีการที่ใช้บุคลากรในองค์กรเป็นผู้ที่ทำหน้าที่รวบรวม ทำให้สิ้นเปลืองทรัพยากรบุคคล และยังอาจเกิดข้อผิดพลาดได้ง่าย นอกจากนี้ หากแหล่งข้อมูลมีปริมาณมาก หรือมีรูปแบบที่หลากหลาย ผู้ที่ทำหน้าที่รวบรวมข้อมูล จะต้องใช้เวลามากขึ้นในการรวบรวมข้อมูลเหล่านั้นมาทำเป็นรายงานสรุปให้ผู้บริหารนำไปใช้ ซึ่งอาจส่งผลให้ได้ข้อมูลไม่ทันต่อความต้องการ ซึ่งสิ่งที่สำคัญที่สุดในการนำข้อมูลมาใช้ให้มีประสิทธิภาพ คือ ต้องได้ข้อมูลที่ถูกต้อง และรวดเร็ว และข้อมูลเหล่านั้นจะเป็นตัวกำหนด หรือทำนายการขายที่จะเกิดขึ้นในอนาคต เพื่อวางแผนเกี่ยวกับการทำตลาด หรือการผลิตสินค้าให้เพียงพอกับความต้องการของตลาดมากที่สุด ซึ่งการเก็บรวบรวมข้อมูลด้วยวิธีการเดิมโดยที่มีข้อมูลที่ไม่สมบูรณ์ ทำให้ขาดประสิทธิภาพในการนำข้อมูลไปใช้ประโยชน์

อีกประเด็นหนึ่งคือ การนำเสนอข้อมูลแบบเดิม มักจะใช้วิธีนำเสนอในรูปแบบที่เป็นรายงานที่มีรูปแบบที่แน่นอน ที่เรียกว่า Fixed Format ซึ่งบางครั้งผู้บริหารที่ได้รับข้อมูลนั้น อาจต้องการปรับเปลี่ยนรูปแบบ หรือมุมมองอย่างอื่น ในกรณีนี้จะต้องให้ผู้ทำข้อมูลเป็นผู้ทำการปรับเปลี่ยนรูปแบบให้ โดยผู้ที่ใช้ข้อมูลไม่สามารถเปลี่ยนรูปแบบเองได้ ทำให้ไม่สะดวก และเป็นสาเหตุหนึ่งของการขาดประสิทธิภาพในการนำข้อมูลไปใช้งาน

จากปัญหาข้างต้น ทำให้เกิดกระบวนการในการรวบรวมข้อมูลที่ต้องการใช้งาน มาไว้ในที่เดียวกัน เพื่อความสะดวกในการเรียกใช้ และเรียกที่ในการเก็บรวบรวมข้อมูลทั้งหมดว่าคลังข้อมูล หรือ Data Warehouse โดยมีกระบวนการในการรวบรวมข้อมูลจากหลายแหล่ง หลายรูปแบบ ให้มาอยู่ในรูปแบบเดียวกัน ซึ่งกระบวนการในการนำข้อมูลเข้าสู่ Data Warehouse ประกอบด้วยสามกระบวนการหลัก ๆ ได้แก่ Extraction Transformation และ Loading ซึ่งเมื่อได้ข้อมูลที่ครบถ้วนใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Warehouse แล้ว ขั้นตอนที่สำคัญถัดมา ก็คือการนำข้อมูลที่อยู่ใน Data Warehouse ออกมาใช้งาน หรือนำเสนอในรูปแบบที่มีประสิทธิภาพ กล่าวคือ ผู้ใช้สามารถปรับเปลี่ยนมุมมอง หรือรูปแบบของรายงานได้อย่างง่ายดาย และหลังจากปรับเปลี่ยนรูปแบบแล้ว จะต้องเห็นผลการเปลี่ยนแปลงนั้นทันที

การนำเสนอข้อมูลใน Data Warehouse ที่มีประสิทธิภาพ จะต้องพึ่งกระบวนการที่เรียกว่า On-line Analytical Processing หรือ OLAP โดย OLAP จะทำการแบ่งข้อมูลออกเป็นหลายมิติ และยังทำการจัดกลุ่มข้อมูลต่าง ๆ รวมทั้งคำนวณผลรวมในแต่ละกลุ่มไว้ล่วงหน้า ทำให้สามารถเรียกดูข้อมูลได้รวดเร็ว ซึ่งในโครงงานนี้ เป็นการจัดทำโปรแกรมวิเคราะห์การขายสินค้าโดยใช้ OLAP บน SQL Server ที่ชื่อว่า Analysis Service เพื่อช่วยนำข้อมูลที่อยู่ใน Data Warehouse มาวิเคราะห์การขายสินค้าในองค์กรให้ถูกต้อง และสามารถทำให้องค์กรสามารถใช้ประโยชน์จากผลการวิเคราะห์นั้นให้เกิดประโยชน์สูงที่สุด

1.2 วัตถุประสงค์ของโครงงาน

การจัดทำโปรแกรมวิเคราะห์การขายสินค้าโดยใช้ OLAP บน SQL Server นี้ เกิดขึ้นเนื่องจากความประสงค์ที่จะแก้ไขปัญหาของความผิดพลาดของการวิเคราะห์การขายสินค้า โดยที่มีข้อมูลที่ไม่ครบถ้วน หรือไม่สมบูรณ์ รวมถึงการที่ต้องเสียเวลาในการรวบรวมข้อมูลที่อาจมาจากหลายแหล่ง ซึ่งอาจจะไม่รองรับต่อสภาพเศรษฐกิจในปัจจุบัน ทำให้ผลการวิเคราะห์ผิดพลาด หรือคลาดเคลื่อนไปจากความเป็นจริง ทั้งยังจะช่วยแก้ไขปัญหาด้านความสามารถของบุคลากรในองค์กร และเพื่อให้สามารถรองรับระบบงานเกี่ยวกับการนำเสนอข้อมูลการขายสินค้าที่มีประสิทธิภาพมากขึ้น โดยโปรแกรมวิเคราะห์การขายสินค้าโดยใช้ OLAP บน SQL Server นี้มีจุดประสงค์ดังต่อไปนี้

1. เพิ่มประสิทธิภาพในการรวบรวมข้อมูลการขายสินค้าที่ต้องรับผิดชอบโดยบุคลากรในองค์กร โดยการนำ Data Warehouse เข้ามาช่วยในการเก็บรวบรวมข้อมูลการขายซึ่งอาจมาจากหลายแหล่ง ทำให้เกิดความสะดวก ถูกต้อง และรวดเร็วมากกว่าการทำงานแบบเดิม
2. เพิ่มความสามารถในด้านการบริหารเวลา โดยได้ข้อมูลที่รวดเร็ว ถูกต้อง และสามารถนำไปใช้ได้อย่างทันต่อเหตุการณ์
3. เพิ่มขีดความสามารถในการเรียกใช้ข้อมูลสำหรับการตัดสินใจของผู้บริหาร เพื่อการวิเคราะห์การขายได้อย่างถูกต้อง และมีประสิทธิภาพสูงที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำให้การวิเคราะห์ข้อมูลการขายเป็นไปอย่างเป็นมาตรฐาน ทั้งนี้ เนื่องจากมีการนำเอา Data Warehouse และ OLAP เข้ามาใช้ในการจัดการด้านต่างๆ ดังต่อไปนี้
   - ด้านการจัดเก็บข้อมูล มีการจัดเก็บข้อมูลเกี่ยวกับข้อมูลทางด้านการขายสินค้าทั้งหมด ให้เป็นไปในรูปแบบเดียวกัน ทำให้สามารถจัดการกับข้อมูลได้ง่าย สะดวก รวดเร็ว และเป็นมาตรฐานเดียวกัน
   - ด้านการสืบค้นข้อมูล มีการแบ่งกลุ่มข้อมูล และคำนวณผลไว้ล่วงหน้า ทำให้สามารถเรียกใช้ข้อมูลได้รวดเร็ว

1.3 ขอบเขตของโครงงาน

โปรแกรมที่ทำการพัฒนานี้ จะเป็นเครื่องมือที่ช่วยให้การทำงานของบุคลากรที่มีหน้าที่รับผิดชอบรวบรวมข้อมูลเกี่ยวกับการขายสินค้าขององค์กร และนำข้อมูลเหล่านั้นมาวิเคราะห์หาผลลัพธ์ที่จะนำมาซึ่งการวางแผนการผลิตและการวางแผนการขายสินค้าในอนาคตนั้น ทำงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น โดยโปรแกรมที่ทำการพัฒนาขึ้นมานี้ จะมุ่งเน้นในการนำข้อมูลที่มีอยู่ใน Data Warehouse ออกมาวิเคราะห์ หรือมาแสดงผลได้หลากหลายรูปแบบ ซึ่งต้องง่ายต่อการใช้งาน ทั้งนี้เนื่องจากได้เล็งเห็นว่าผู้ใช้งานข้อมูลเพื่อการวิเคราะห์นี้ มักเป็นผู้บริหารระดับสูง ต้องการความเรียบง่ายในการใช้งาน และให้ผลการทำงานที่มีประสิทธิภาพ

นอกจากนี้ ขอบเขตการทำงานของโครงการนี้จะกล่าวถึงกระบวนการในการเก็บรวบรวมข้อมูล มาไว้ใน Data Warehouse และการออกแบบในกระบวนการ OLAP ด้วยบางส่วน เพื่อความสอดคล้องกันกับการนำข้อมูลมาวิเคราะห์และนำเสนอในขั้นสุดท้าย ด้วยโปรแกรมที่จะพัฒนาขึ้นมา ทั้งนี้รูปแบบการนำเสนอข้อมูล จะเป็นการติดต่อกับผู้ใช้ผ่านทางหน้าจอของโปรแกรมที่พัฒนาขึ้นบนระบบปฏิบัติการ Windows และอาจมีการนำเสนอรายงานในรูปแบบของแผนภูมิ โดยทั้งหมดนี้จะสามารถปรับเปลี่ยนรูปแบบของรายงานได้อย่างง่ายดายตามความต้องการของผู้ใช้งาน โดยที่กระบวนการทั้งหมด จะครอบคลุมการทำงานดังต่อไปนี้

- การเก็บรวบรวมข้อมูลที่เกี่ยวข้องกับการขายสินค้า เพื่อให้เป็นรูปแบบฐานข้อมูลอิเล็กทรอนิกส์ และเป็นรูปแบบเดียวกันสำหรับทั้งองค์กร
- การจำแนกประเภท และรูปแบบของข้อมูลที่รวมรวมได้ ให้เป็นประเภทเดียวกัน เพื่อง่ายต่อการใช้งาน
- การสร้างรูปแบบสำหรับการวิเคราะห์การขายสินค้า เช่น การแบ่งกลุ่มข้อมูล หรือการสร้างสูตรสำหรับการวิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แสดงผลข้อมูลให้ออกมาในรูปแบบของตัวเลขสรุป หรือ ในรูปแบบของแผนภูมิเพื่อความเข้าใจและนำไปใช้งานได้ง่าย และสามารถปรับเปลี่ยนรูปแบบรายงานได้
- มีการประมวลผลสรุปในเชิงสถิติ เพื่อนำไปใช้ในการตัดสินใจ

1.4 ขั้นตอนและแผนงานในการพัฒนา

1.4.1 การกำหนดระยะเวลาในการดำเนินงาน

- กำหนดระยะเวลาสำหรับการพัฒนาโปรแกรม โดยคำนึงถึงการดำเนินงานจริง

1.4.2 การวิเคราะห์การพัฒนา (Analysis Phase)

- ศึกษาการทำงานของระบบการขายสินค้า โดยศึกษาลักษณะการทำงานเดิมของผู้ที่มีหน้าที่รับผิดชอบ และศึกษาขั้นตอนในการทำงาน ข้อมูลที่ได้จากระบบงานดังกล่าว รวมถึงเอกสารที่จำเป็นในการดำเนินงานนั้น
- สอบถามปัญหา และเก็บรวบรวมความต้องการของบุคลากรที่เกี่ยวข้องกับการใช้ข้อมูลการขายเพื่อการวิเคราะห์
- ศึกษาโครงสร้างของทรัพยากรทางด้านคอมพิวเตอร์ที่ต้องใช้ในระบบงาน ทั้งทางด้านซอฟต์แวร์และฮาร์ดแวร์
- ศึกษาข้อมูลของเครื่องมือที่จะนำมาใช้ในระบบงาน ทั้งทางอินเตอร์เน็ต หนังสือ รวมถึงระบบที่ถูกพัฒนาขึ้นมาในรูปแบบเดียวกันที่องค์กรอื่นได้นำมาใช้แล้ว
- สรุปผลที่ได้ทำการศึกษามาทั้งหมด รวมถึงกำหนดทางเลือกเพื่อการตัดสินใจที่ดีที่สุด
- กำหนดขอบเขตของโปรแกรมที่จะพัฒนาโดยใช้ข้อมูลจากที่สรุป

1.4.3 การออกแบบ (Design Phase)

- วิเคราะห์ข้อมูลที่มีทั้งหมด และออกแบบระบบในการจัดเก็บข้อมูลดังกล่าว
- ออกแบบโครงสร้างข้อมูล และจัดหมวดหมู่ข้อมูล
- กำหนดเครื่องมือ และทรัพยากรที่ใช้ในการพัฒนาโปรแกรม
- ออกแบบโครงสร้างของโปรแกรม ในส่วนนำเข้าข้อมูล การแสดงผลของข้อมูล และส่วนที่ติดต่อกับผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ออกแบบโครงสร้างของระบบเครือข่าย
- การวิเคราะห์และออกแบบทั้งหมด จะต้องอยู่ภายในขอบเขตที่กำหนดไว้ โดยจะต้องออกแบบให้เหมาะสมและสอดคล้องกับการทำงานภายในองค์กรจริง

1.4.4 การติดตั้งโปรแกรม (Implementation Phase)

- เขียนโปรแกรมตามที่ออกแบบไว้
- ติดตั้งโปรแกรมที่ทำการพัฒนาขึ้น
- กำหนดสิทธิการเข้าถึงข้อมูลของผู้ใช้ในระดับต่าง ๆ ที่แตกต่างกัน

1.4.5 การทดสอบโปรแกรม (Testing Phase)

- ทำการทดสอบโปรแกรมที่ได้ทำการพัฒนาขึ้น โดยแบ่งการทดสอบออกเป็น 3 ภาคการทำงานคือ
  - ทดสอบการทำงานได้ของโปรแกรม โดยทดสอบเป็นฟังก์ชันการทำงานย่อยที่มีอยู่ในโปรแกรม ว่าสามารถทำงานได้ตามที่ออกแบบไว้หรือไม่
  - ทดสอบการทำงานได้ของโปรแกรม โดยทดสอบรวมทั้งโปรแกรม ว่าสามารถทำงานได้ตามที่ออกแบบไว้ และทำงานถูกต้องครบถ้วนหรือไม่
  - ทดสอบโดยการให้ผู้ใช้ใช้งานจริง ว่าระบบสามารถรอบรับความต้องการและช่วยสนับสนุนผู้ใช้ในการทำงานให้มีประสิทธิภาพและประหยัดเวลามากขึ้นหรือไม่
  - ทดสอบโดยการให้ผู้ใช้ใช้งานจริง เพื่อตรวจสอบระดับการเข้าถึงข้อมูลว่ามีความแตกต่างกันตามที่ได้กำหนดไว้หรือไม่

- เมื่อทำการทดสอบโปรแกรม หากมีสิ่งที่จะต้องแก้ไข จะกลับไปดำเนินการในขั้นตอนการออกแบบ และการติดตั้งโปรแกรม อีกครั้ง จนกว่าโปรแกรมจะใช้งานได้ครบถ้วนสมบูรณ์

1.4.6 การทำเอกสารประกอบการทำงานของโปรแกรม

- ทำเอกสารประกอบการออกแบบโปรแกรมทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำเอกสารประกอบการใช้งานของโปรแกรม

1.4.7 การบำรุงรักษาโปรแกรม (Maintenance Phase)

- บำรุงรักษาระบบการควบคุมสิทธิการเข้าถึงข้อมูลของผู้ใช้
- ติดตามและประเมินผลการใช้งานของผู้ใช้โปรแกรม

1.5 เครื่องมือที่ใช้ในการพัฒนาระบบงาน

1.5.1 รายละเอียดทรัพยากรทางด้านฮาร์ดแวร์ที่จะต้องใช้ในโปรแกรม

- เครื่องคอมพิวเตอร์ที่ใช้เป็น Server จะต้องมีคุณสมบัติดังต่อไปนี้
  - หน่วยประมวลผลกลาง (CPU) มีประสิทธิภาพอย่างน้อยเทียบเท่า Intel Pentium 4 1.7 GHz
  - หน่วยความจำหลัก (RAM) ความจุอย่างน้อย 512 MB
  - หน่วยความจำสำรอง (Harddisk) ความจุอย่างน้อย 40 GB
- เครื่องคอมพิวเตอร์ที่จะเป็น Client จะต้องมีคุณสมบัติดังต่อไปนี้
  - หน่วยประมวลผลกลาง (CPU) มีประสิทธิภาพอย่างน้อยเทียบเท่า Intel Celeron 1.3 GHz
  - หน่วยความจำหลัก (RAM) ความจุอย่างน้อย 128 MB
  - หน่วยความจำสำรอง (Harddisk) ความจุอย่างน้อย 20 GB
- เครื่องคอมพิวเตอร์ที่จะใช้พัฒนาโปรแกรม จะต้องมีคุณสมบัติดังต่อไปนี้
  - หน่วยประมวลผลกลาง (CPU) มีประสิทธิภาพอย่างน้อยเทียบเท่า Intel Pentium 4 1.7 GHz
  - หน่วยความจำหลัก (RAM) ความจุอย่างน้อย 256 MB
  - หน่วยความจำสำรอง (Harddisk) ความจุอย่างน้อย 20 GB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

1.5.2 รายละเอียดทรัพยากรทางด้านซอฟต์แวร์ที่จะต้องใช้ในระบบ

- เครื่องคอมพิวเตอร์ที่จะเป็น Server จะต้องมีคุณสมบัติดังต่อไปนี้
  - Operation System อย่างน้อยควรใช้ Windows 2000 Server
  - Relational Database ใช้ MS SQL Server 2000
  - OLAP Server เป็น MS SQL 2000 Analysis Service
- เครื่องคอมพิวเตอร์ที่จะเป็น Client จะต้องมีคุณสมบัติดังต่อไปนี้
  - Operation System อย่างน้อยควรใช้ Windows 2000 Professional
  - ติดตั้งโปรแกรม Microsoft Data Access Component 2.7 เพื่อใช้เชื่อมต่อด้านข้อมูลกับ Server
  - อาจมี Microsoft Excel XP เพื่อวิเคราะห์ข้อมูลเพิ่มเติม
- เครื่องคอมพิวเตอร์ที่จะใช้พัฒนาโปรแกรมจะต้องมีคุณสมบัติดังต่อไปนี้
  - Operation System อย่างน้อยควรใช้ Windows 2000 Professional
  - ติดตั้งโปรแกรม Microsoft Data Access Component 2.7 เพื่อใช้เชื่อมต่อด้านข้อมูลกับ Server
  - ติดตั้งโปรแกรม Visual Basic 6.0 ในการพัฒนาโปรแกรม

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 ประโยชน์ต่อผู้ทำการพัฒนาโปรแกรม

- ได้นำความรู้ความสามารถที่ได้จากการเรียนรู้และประสบการณ์มาใช้ในทุกขั้นตอนของการพัฒนาระบบงานทางด้านเทคโนโลยีสารสนเทศ
- รู้จักการแก้ไขปัญหาที่เกิดขึ้นระหว่างที่ทำการพัฒนาโปรแกรม
- สนับสนุนการทำงานร่วมกับผู้อื่น

1.6.2 ประโยชน์ต่อองค์กร

- ได้ระบบการรวบรวมข้อมูลการขายสินค้า ภายในองค์กรและสามารถนำข้อมูลมาใช้ได้อย่างมีประสิทธิภาพ
- ประหยัดทรัพยากรประเภทบุคคล แรงงาน และเวลา
- เป็นแนวทางเริ่มต้นในการที่จะนำโปรแกรมที่พัฒนาขึ้นเองมาใช้งาน ทั้งนี้ทำให้สามารถปรับเปลี่ยนตามความต้องการของผู้ใช้งานได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6.3 ประโยชน์โดยทั่วไป

- เป็นกรณีศึกษาเพื่อเป็นแนวทางในการเก็บรวบรวมข้อมูลด้านอื่น ๆ นอกเหนือจากข้อมูลการขายสินค้า มาไว้ใน Data Warehouse
- เป็นกรณีศึกษาเพื่อเป็นแนวทางในการพัฒนาโปรแกรมอื่น ๆ ต่อไป

1.7 รายละเอียดในบทต่าง ๆ

- **บทที่ 2** กล่าวถึงทฤษฎีที่เกี่ยวข้องในการพัฒนาระบบทั้งหมด ไม่ว่าจะเป็น Data Warehouse เครื่องมือที่ช่วยในการเก็บรวบรวมข้อมูล (ETL — Extraction Transformation and Loading) เครื่องมือที่ช่วยในการทำ OLAP และจะกล่าวถึงรายละเอียดของเครื่องมือที่ใช้ในการพัฒนาโปรแกรมเพื่อนำเสนอข้อมูลด้วย

- **บทที่ 3** กล่าวถึงการวิเคราะห์ การดำเนินงานขององค์กรที่เกี่ยวกับการขายสินค้า การออกแบบการจัดเก็บข้อมูลการขายสินค้าลงใน Data Warehouse รวมถึงการออกแบบ Multidimensional Database หรือ OLAP Cube

- **บทที่ 4** กล่าวถึงการออกแบบสถาปัตยกรรมของระบบ ออกแบบโปรแกรมสำหรับนำเสนอข้อมูล ให้ตรงกับความต้องการของผู้ใช้มากที่สุด โดยจะมีการศึกษาความต้องการของผู้ใช้เป็นหลัก

- **บทที่ 5** กล่าวถึงการติดตั้งระบบเพื่อการใช้งานจริง การบำรุงรักษาข้อมูลที่อยู่ในระบบ และการดูแลสิทธิ์การเข้าถึงข้อมูลของผู้ใช้แต่ละคนให้ถูกต้อง

- **บทที่ 6** กล่าวถึงการสรุปผลการดำเนินการพัฒนาระบบการจัดเก็บ และการนำเสนอข้อมูลการขายสินค้าเพื่อการวิเคราะห์ รวมไปจนถึงข้อดี ข้อเสีย และข้อจำกัดของระบบที่ได้รับการพัฒนาขึ้น นอกจากนี้ยังอาจมีการเปรียบเทียบในแง่การนำเสนอข้อมูลของระบบที่พัฒนาขึ้น กับการนำเสนอข้อมูลด้วยโปรแกรมสำเร็จรูปอื่น ๆ

ความเป็นมาของโครงการทั้งหมดซึ่งประกอบไปด้วย ความเป็นมาของโครงการ วัตถุประสงค์ ขอบเขตและการวางแผน รวมถึงประโยชน์ที่ได้รับจากการพัฒนาโครงงาน จะแสดงให้เห็นถึงการพัฒนาระบบที่เป็นไปอย่างมีแบบแผน และเป็นขั้นตอน ซึ่งจะส่งผลให้การดำเนินการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนาโปรแกรมวิเคราะห์การขายสินค้าโดยใช้ OLAP บน SQL Server บรรลุวัตถุประสงค์ตามที่ต้องการ

แม้จะมีการวางแผนและออกแบบอย่างครอบคลุมทั้งหมดแล้ว แต่ในการดำเนินการพัฒนาระบบจริง ๆ อาจจะต้องมีการปรับเปลี่ยนในบางส่วนเพื่อความเหมาะสมกับสภาวะปัจจุบัน รวมถึงเพื่อใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้องกับการพัฒนาระบบงาน

เอกสารบทนี้จะกล่าวถึง ทฤษฎีและเครื่องมือที่ใช้ในการพัฒนาโปรแกรมวิเคราะห์ข้อมูลการขายสินค้าด้วย OLAP บน SQL Server ดังต่อไปนี้

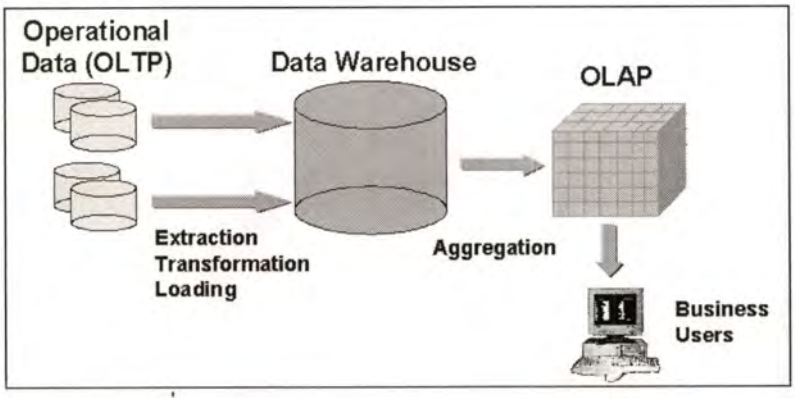
2.1 Data Warehouse และ OLAP Cube

ในองค์กรที่มีขนาดใหญ่ ข้อมูลที่ได้จากการประมวลผลงานประจำวันต่าง ๆ หรือที่เรียกว่า OLTP - Online Transactions Processing จะมีปริมาณมาก หลากหลายรูปแบบ ทำให้นำไปใช้ประโยชน์เพื่อการตัดสินใจได้ยาก ดังนั้นจึงได้มีการออกแบบระบบที่เรียกว่า Data Warehouse ซึ่งเป็นการนำข้อมูลที่มีปริมาณมากและหลายรูปแบบเหล่านั้น มารวมอยู่ด้วยกันและทำให้มีรูปแบบเดียวกัน เพื่อง่ายต่อการนำข้อมูลไปใช้ในการตัดสินใจ ซึ่งการที่จะนำ Data Warehouse มาใช้งานให้มีประสิทธิภาพนั้น ประกอบด้วยสามส่วนหลัก ๆ คือ หนึ่งต้องมีการออกแบบ Data Warehouse ที่ดี สองมีกระบวนการนำข้อมูลเข้าสู่ Data Warehouse ที่มีประสิทธิภาพ และสาม คือส่วนที่ผู้ใช้สามารถนำข้อมูลจาก Data Warehouse ไปใช้ได้ง่าย ซึ่งในส่วนที่สาม คือการนำข้อมูลจาก Data Warehouse ไปใช้ในเพื่อการวิเคราะห์และตัดสินใจ โดยมีเครื่องมือหรือกระบวนการที่เรียกว่า OLAP – Online Analytical Processing แต่เนื่องจากข้อมูลใน Data Warehouse มีปริมาณมาก หากผู้ใช้มีการ Query ข้อมูลโดยตรงจาก Data Warehouse ต้องใช้เวลามากจึงจะได้ข้อมูลที่ต้องการ ดังนั้น เครื่องมือในการทำ OLAP จึงได้ทำการจัดกลุ่มของข้อมูลเอาไว้ก่อน แล้วจึงรวม (Aggregation) ข้อมูลเหล่านั้นเข้าด้วยกัน จะทำให้ผู้ใช้ได้ข้อมูลที่เร็วขึ้น

2.1.1 กระบวนการของ Data Warehouse และ OLAP

กระบวนการในการนำข้อมูลจาก OLTP เข้ามาสู่ Data Warehouse นั้น มีขั้นตอนในการทำงาน 3 ขั้นตอนหลัก ๆ ได้แก่ Data Extraction, Data Transformation และ Data Loading ซึ่งทำให้ข้อมูลที่อยู่ใน Data Warehouse มีความครบถ้วนสมบูรณ์ และระบบ Data Warehouse ส่วนใหญ่มักจะมีเครื่องมือที่เรียกว่า OLAP เข้ามาช่วยเพิ่มประสิทธิภาพในการทำงานกับข้อมูลที่อยู่ภายใน Data Warehouse ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.1 กระบวนการของ Data Warehouse และ OLAP

2.1.2 ส่วนประกอบของ Data Warehouse

ปกติข้อมูลจาก OLTP จะประกอบด้วย Table หลาย ๆ Table มี Index ที่เป็นทั้ง Primary Key และ Foreign Key แต่ใน Data Mart นั้น Table จะถูกออกแบบใหม่ให้เหลือเพียงสองประเภทหลัก ๆ ได้แก่ Fact Table และ Dimension Table โดยส่วนใหญ่ในหนึ่ง Data Mart มักจะมี Fact Table เพียงหนึ่ง Table เท่านั้น แต่มี Dimension Table หลาย Table

2.1.2.1 Fact Table

Fact Table จะเก็บข้อมูลคล้าย ๆ ข้อมูลประเภท Transaction ที่ได้จาก OLTP แต่จะมีการผ่านกระบวนการ Summarize มาบ้างแล้วจากกระบวนการ ETL – Extraction Transformation and Loading ยกตัวอย่างเช่น เป็นข้อมูลยอดขาย ของสินค้า ต่อลูกค้า ในหนึ่งวัน เป็นต้น โดยจะไม่มีรายละเอียดของสินค้า หรือ ลูกค้า อยู่ด้วย โดยปกติ Fact Table นี้จะมีขนาดใหญ่มาก เนื่องจากต้องเก็บข้อมูลที่ผ่านมาแล้ว (historical data) เพื่อใช้วิเคราะห์และเปรียบเทียบด้วย และคุณสมบัติอีกอย่างของ Fact Table คือมี Foreign Key เป็นแบบ Multiple Foreign Key ซึ่งจะสัมพันธ์กับ แต่ละ Dimension Table ใน Data Mart เดียวกัน

2.1.2.2 Dimension Table

Dimension Table จะมีข้อมูลที่เป็นรายละเอียดเพิ่มเติมของข้อมูลใน Fact Table เช่น ที่อยู่ของลูกค้า หรือ รายละเอียดอื่น ๆ ของสินค้า เป็นต้น จำนวน Dimension Table จะมากหรือน้อยขึ้นอยู่กับการออกออกแบบ Data Mart และใน Dimension Table หนึ่ง ๆ จะมีจำนวนข้อมูลไม่มากนัก คุณสมบัติอย่างหนึ่งของ Dimension Table คือ มี Primary Key ที่สัมพันธ์กับ Fact Table ด้วย ซึ่งความสัมพันธ์ของ Key ระหว่าง Fact Table และ Dimension Table จะเรียกว่า Star Schema

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.3 OLAP Cube

จากความสัมพันธ์ของ Fact Table และ Dimension Table แบบ Star Schema นั้น สามารถนำไปสร้างเป็น OLAP Cube ได้ ซึ่งการสร้าง Cube ก็คือการรวมข้อมูลกลุ่มเดียวกันเข้าด้วยกัน แล้วเก็บผลบวกที่ได้เอาไว้ ทำให้ลดเวลาในการ Query ข้อมูล เนื่องจากไม่ต้องเสียเวลาคำนวณใหม่ ซึ่งภายใน OLAP Cube ประกอบไปด้วย

1. Dimension

คือโครงสร้างของ Cube ถ้ามองอย่างง่าย ก็คือด้านต่าง ๆ ของ Cube นั่นเอง โดยจำนวน Dimension ของ Cube อาจจะเท่ากับจำนวน Dimension Table ของ Star Schema หรือไม่ก็ได้ ทั้งนี้ขึ้นอยู่กับการออกแบบ Cube

2. Member

คือสมาชิกของ Dimension ซึ่งชื่อของ Member จะเป็นข้อมูลที่อยู่ภายใน Dimension Table ใน Star Schema และ Member เป็นส่วนสำคัญในการเข้าถึงข้อมูลในภาย Cube

3. Level

เป็นการแบ่ง Member ออกเป็นลำดับขั้น ภายในแต่ละ Dimension นั้น ๆ

4. Hierarchy

คือความสัมพันธ์แบบ Parent – Child ภายในแต่ละ Dimension

5. Measurement

เป็นข้อมูลประเภทตัวเลขที่จะนำมาวิเคราะห์ ซึ่ง Measurement นี้จะอยู่ใน Fact Table ใน Star Schema ถ้ามองในมุมมองของ Cube แล้ว Measurement ก็คือ ค่าต่าง ๆ ที่ถูกบรรจุอยู่ภายในช่องต่าง ๆ ภายใน Cube นั่นเอง

2.2 สถาปัตยกรรมระบบวิเคราะห์ข้อมูลของไมโครซอฟท์

แนวคิดของไมโครซอฟท์ที่มีต่อ Business Intelligence คือการออกแบบให้ระบบการจัดการคลังข้อมูลเป็นเรื่องง่าย ซึ่ง SQL Server 2000 ได้มีสิ่งที่จำเป็นทั้งหมดต่อการสร้างแอปพลิเคชั่นและ Business Intelligence ซึ่งประกอบด้วย

- ฐานข้อมูลเชิงสัมพันธ์สำหรับจัดเก็บข้อมูลเชิงสัมพันธ์หรือคลังข้อมูล (ทำงานโดย SQL Server เอง)
- กลไก OLAP (Analysis Server)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Data Transformation Services (DTS) ซึ่งเป็นซอฟต์แวร์สำหรับคัดแยก แปลงรูปแบบ และโหลดข้อมูล (ETL) จากระบบปฏิบัติการสู่หน่วยจัดเก็บข้อมูลเชิงสัมพันธ์
- ซอฟต์แวร์บริหารระบบ ใช้ในการจัดการทั้งฐานข้อมูลเชิงสัมพันธ์และกลไก OLAP
- ความสามารถในการจัดทำ Data Mining
- บริการ Meta Data
- การบริหารระบบแบบกราฟิก และสนับสนุนการเชื่อมต่ออย่างกว้างขวาง

2.2.1 มาตรฐานและการเชื่อมโยงข้อมูล

องค์ประกอบที่สองของระบบข้อมูลอัจฉริยะของไมโครซอฟต์คือ การประกาศใช้อินเทอร์เฟสและมาตรฐานแบบเปิดสำหรับการทำงานกับส่วนประกอบเหล่านี้ และระหว่างส่วนประกอบเหล่านี้กับผลิตภัณฑ์ของผู้ผลิตรายอื่น ซึ่งไมโครซอฟต์ได้ชี้ถึงวัตถุประสงค์ในหลายทางด้วยกัน

ทั้ง SQL Server และ Analysis Server เผยให้เห็นถึงออปเจ็คโมเดล และเปิดโอกาสให้เชื่อมต่อผ่านทางโปรแกรมโดยตรง โดยข้ามขั้นตอนของอินเทอร์เฟสในการจัดการของไมโครซอฟต์

- SQL Distributed Management Object (SQL-DMO) เป็นออปเจ็คโมเดลสำหรับ SQL Server
- DSO (Decision Support Objects) เป็นออปเจ็คโมเดลของ Analysis Manager
- ActiveX Data Objects Multidimensional (ADO MD) เป็นออปเจ็คโมเดลสำหรับ PivotTable Services ซึ่งเป็นส่วนประกอบด้านไคลเอนท์ของ Analysis Services

2.2.2 การเชื่อมต่อแบบ OLE DB

การเชื่อมต่อแบบ OLE DB เป็นการเชื่อมต่อในระดับต่ำไปยังข้อมูลของทั้งองค์กร ในขณะที่มาตรฐานเฉพาะฐานข้อมูล เช่น ODBC ได้รับการออกแบบให้ครอบคลุมฐานข้อมูลเชิงสัมพันธ์ทั้งหมด ส่วนมาตรฐาน OLE DB ได้รับการออกแบบให้เรียกใช้ได้ทั้งแหล่งข้อมูลแบบสัมพันธ์และไม่ใช่แบบสัมพันธ์ จึงให้ความสามารถในการทำงานร่วมกันได้ในขอบเขตที่กว้างกว่า

มาตรฐาน OLE DB เป็นข้อกำหนดแบบเปิด ที่ได้รับการพัฒนาร่วมกันจากหลายองค์กรในอุตสาหกรรม และเปิดให้นักพัฒนาทั่วโลกได้ใช้งานอย่างทั่วถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Analysis Services ได้ขยายมาตรฐาน OLE DB ด้วยข้อกำหนดเพิ่มเติม 3 ประการได้แก่

- OLE DB for OLAP

เป็นซอฟต์แวร์สำหรับการเรียกค้นและจัดทำรายงานนั้นมีให้เลือกใช้เป็นจำนวนมาก และไม่มีซอฟต์แวร์ใดที่สามารถตอบสนองความต้องการของธุรกิจได้ทุกประเภท ดังนั้นธุรกิจหนึ่งๆ จึงอาจมีมาตรฐานของซอฟต์แวร์เรียกค้นและทำรายงานได้ 2 หรือ 3 แบบ

และจากการมีซอฟต์แวร์ที่มีศักยภาพในการเป็นฟรอนท์เอนด์จำนวนมาก จึงจำเป็นอย่างยิ่งที่จะต้องมีมาตรฐานในการเชื่อมต่อหรือเรียกใช้ข้อมูล OLAP ซึ่งเซิร์ฟเวอร์แบบหลายมิติ หรือ OLAP เช่น Analysis Server ได้จัดเก็บข้อมูลในรูปแบบที่ต่างไปจากแหล่งข้อมูลเชิงสัมพันธ์แบบเดิมที่มีอยู่ ดังนั้น ข้อกำหนด OLE DB for OLAP จึงทำหน้าที่ในการเชื่อมต่อกับแหล่งข้อมูลแบบหลายมิติ

มาตรฐานเปิดที่เป็นที่ยอมรับอย่างรวดเร็วนี้ จะทำให้แอปพลิเคชั่นไคลเอนท์เรียกใช้แหล่งข้อมูลใดๆ จากผู้ให้บริการ OLE DB for OLAP และยอมให้แหล่งข้อมูล OLAP เชื่อมกับแหล่งข้อมูลหลากหลายประเภท เป็นผลให้มีการปฏิบัติงานร่วมกันได้ในระดับหนึ่งในตลาดของ OLAP ซึ่งไม่เคยปรากฏมาก่อน

ตัวอย่างเช่น องค์กรสามารถซื้อแอปพลิเคชั่นเฉพาะทางจาผู้ผลิตรายหนึ่ง ซึ่งถ้าหากผลิตภัณฑ์นี้รองรับมาตรฐาน OLE DB for OLAP ก็สามารถเรียกใช้แหล่งข้อมูลของ Analysis Server ได้เช่นเดียวกับแหล่งข้อมูลของผู้ให้บริการรายอื่น สิ่งนี้ทำให้ธุรกิจมีทางเลือกมาขึ้น ทั้งในเรื่องของซอฟต์แวร์และแหล่งข้อมูล และทำให้การรวมกันของผลิตภัณฑ์เหล่านี้ง่ายขึ้นกว่าเดิม

- OLE DB for Data Mining

ไมโครซอฟต์ได้ขยายขีดความสามารถของ OLE DB โดยรวมข้อกำหนดสำหรับ data mining ลงไปด้วย โดยการเชื่อมต่อ OLE DB for data mining ได้รับการพัฒนาจากความร่วมมืออย่างกว้างขวางในวงการ data mining ทำให้ผลิตภัณฑ์ data mining ที่มีอยู่อย่างกระจัดกระจายสามารถแลกเปลี่ยนทั้งข้อมูลและผลลัพธ์ระหว่างกันได้

การเสนอมาตรฐานเปิดสำหรับ data mining ก่อให้เกิดการปฏิบัติงานร่วมกันในอีกระดับหนึ่ง ซึ่งในอดีตจะขึ้นกับข้อกำหนดเฉพาะของแต่ละผู้ผลิต การใช้การเชื่อมต่อที่กำหนดใหม่นี้ ทำให้นักพัฒนาและผู้ผลิตซอฟต์แวร์อิสระสามารถรวมความสามารถของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

data mining ไว้ในแอปพลิเคชั่นที่ต้องการ ผู้ใช้จำนวนมากจึงเรียกใช้ความสามารถของ data mining ได้ในแอปพลิเคชั่น เช่น การทำตลาดแบบ one-to-one การตรวจจับการทุจริต การวิเคราะห์ความเสี่ยงของเครดิต และแอปพลิเคชั่นที่เป็นไปได้อื่นๆ สำหรับการทำ data mining ในวงการธุรกิจ

ด้วยข้อกำหนดของ OLE DB for data mining บริษัทต่างๆ สามารถใช้ความสามารถ data mining ที่มีอยู่กับ Analysis Services ทำการรวมอัลกอริธึมพิเศษหรือแอปพลิเคชั่นเข้าไปเป็นส่วนหนึ่งของแอปพลิเคชั่น Analysis Services หรือใช้ Analysis Services cube ได้ในฐานะของแหล่งข้อมูลสำหรับแอปพลิเคชั่น data mining พิเศษอื่นๆ

และที่สำคัญที่สุด ขณะนี้บริษัทต่าง ๆ สามารถรวมประโยชน์ที่ได้รับจาก data mining เข้าไปเป็นส่วนหนึ่งของการทำงานและการตัดสินใจทางธุรกิจทั่วไป สิ่งซึ่งครั้งหนึ่งเคยเป็นกฎเกณฑ์ค่อนข้างเป็นอุปสรรค แต่ปัจจุบันนี้สามารถแนะแนวการตัดสินใจทางธุรกิจเพื่อผลตอบแทนและความสำเร็จที่มากขึ้นได้

- XML for Analysis

ในข้อกำหนดของทั้ง OLE DB for OLAP และ OLE DB for data mining ต่างต้องใช้องค์ประกอบที่ไคลเอนท์ซึ่งขึ้นอยู่กับเซิร์ฟเวอร์อย่างมาก รูปแบบเช่นนี้ใช้งานได้ดีในสภาพแวดล้อมหลายแบบ แต่ไม่เหมาะกับไคลเอนท์ความสามารถต่ำที่ทำงานบนเว็บ เช่น ไคลเอนท์ประเภทบราวเซอร์

ข้อกำหนดของ XML for Analysis สนับสนุนการทำงานแบบไม่ผูกติดและเชื่อมโยงแบบไม่มีสถานะในสภาพแวดล้อมของเว็บ จึงเอื้อต่อการเรียกใช้แหล่งข้อมูลวิเคราะห์ทุกรูปแบบ (OLAP และ data mining) ผ่านเว็บ ข้อกำหนดนี้ถูกออกแบบให้ทำงานได้ดีที่สุดผ่านเว็บ โดยลดการส่งข้อมูลไปมา และลดปริมาณการใช้เครือข่ายลง ช่วยให้เกิดสภาพแวดล้อมแบบผสมผสานจริงๆ สำหรับการเรียกใช้ข้อมูลจากแพล็ตฟอร์มใดก็ได้ เนื่องจากว่า ในข้อกำหนดนั้นใช้ XML สำหรับการแลกเปลี่ยนข้อมูล แอปพลิเคชั่นที่นำข้อกำหนดนี้ไปใช้จึงสามารถเขียนขึ้นด้วยภาษาใด บนแพล็ตฟอร์มใดก็ได้

แม้ว่าข้อกำหนดนี้เพิ่งได้รับการเสนอใช้งานเป็นครั้งแรกในปี 2001 แต่ก็กำหนดขึ้นบน OLE DB for OLAP และ OLE DB for data mining ซึ่งมีความสมบูรณ์แล้ว XML for Analysis จึงได้รับการสนับสนุนอย่างกว้างขวางและเป็นที่ยอมรับจากผู้ผลิตซอฟต์แวร์เรียกค้นจำนวนมาก แม้กระทั่งจากแพล็ตฟอร์ม OLAP อื่นก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การส่งต่อข้อมูลในคลังข้อมูลที่รวมกัน

การทำงานร่วมกันเพื่อรายงานการวิเคราะห์ให้ผู้ใช้

1. ข้อมูลถูกส่งจากระบบข้อมูลหลายแหล่งเข้าสู่คลังข้อมูล ซึ่งอาจจะเป็นเว็บเซิร์ฟเวอร์ เซิร์ฟเวอร์อีคอมเมิร์ช ระบบ ERP แหล่งข้อมูลภายนอก และอื่นๆ โดยซอฟต์แวร์ ETLM เช่น DTS จะช่วยดำเนินการอัพเดทและปรับแต่งข้อมูลโดยอัตโนมัติ
2. ข้อมูลได้รับการจัดเก็บในฐานข้อมูลเชิงสัมพันธ์บน SQL Server สำหรับแอปพลิเคชั่นด้านการวิเคราะห์
3. เพื่อให้การสนับสนุนแอปพลิเคชั่นวิเคราะห์เฉพาะทางหรือการวิเคราะห์แบบโต้ตอบ การสร้าง OLAP cube โดยใช้ Analysis Services จะทำให้แอปพลิเคชั่นของผู้ใช้ติดต่อกับ cube แทนฐานข้อมูลเชิงสัมพันธ์ จึงง่ายต่อการจัดเก็บข้อมูลเชิงสัมพันธ์อย่างเห็นได้ชัด และมุ่งความสนใจไปยังที่การค้นหาใน Analysis Services ได้อย่างเต็มที่
4. ไคลเอนท์สามารถเรียกใช้ข้อมูลบน cube เหล่านี้ได้หลายวิธี ซึ่งเป็นผลจากการสนับสนุนอินเทอร์เฟสจำนวนมาก ไคลเอนท์บนเว็บ ไคลเอนท์บนเครือข่าย Windows หรือแม้แต่ไคลเอนท์ที่ออฟไลน์และไคลเอนท์บนอุปกรณ์เคลื่อนที่ ก็สามารถเรียกใช้ข้อมูลวิเคราะห์เพื่อช่วยในการตัดสินใจทางธุรกิจได้

นอกจากนี้ ยังสามารถเพิ่มความสามารถด้าน data mining ให้กับ cube เพื่อช่วยให้ผู้ใช้ค้นพบข้อมูล แนวโน้ม หรือรูปแบบใหม่ ที่อาจจะซ่อนอยู่ในข้อมูลจำนวนมากได้

2.2.4 Data Transformation Services: จัดการการส่งต่อข้อมูล

ขั้นแรกของการส่งต่อข้อมูลใน Business Intelligence คือ การสร้างหน่วยเก็บข้อมูลเชิงสัมพันธ์ด้วยข้อมูลที่เหมาะสมสำหรับการวิเคราะห์ ซึ่งอาจเป็นขั้นตอนง่าย ๆ หรือซับซ้อนก็ได้ เพราะผู้ใช้ไม่เพียงต้องการจะคัดเลือกข้อมูลจากแหล่งข้อมูลเท่านั้น แต่อาจต้องการเชื่อมโยงและแปลงข้อมูลด้วย เพื่อให้มั่นใจถึงความถูกต้องและแน่นอนของข้อมูลที่มาจากหลายระบบ จึงเห็นได้ชัดว่าบริการเหล่านี้ ต้องยืดหยุ่นและสามารถปรับเปลี่ยนได้อย่างมาก และควรรับมือกับข้อมูลปริมาณมากได้อย่างมีประสิทธิภาพด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคโนโลยีสารสนเทศ ธจล.

มีซอฟต์แวร์จำนวนมากที่ให้บริการเกี่ยวกับการส่งต่อข้อมูลเช่นนี้ Data Transformation Services (DTS) ซึ่งรวมมากับ SQL Server 2000 และ SQL Server 7.0 เป็นซอฟต์แวร์สำหรับการแปลงข้อมูลที่ยืดหยุ่นและมีประสิทธิภาพมาก ในความเป็นจริงแล้ว DTS เป็นรูปแบบออปเจ็คที่ไมโครซอฟต์และผู้ผลิตรายอื่นพัฒนาขึ้น เพื่อการเชื่อมต่อกับผู้ใช้ในรูปแบบต่างๆ ซึ่งมีหลายวิธีในการใช้ DTS เพื่อจัดการกับการส่งต่อข้อมูล

DTS นั้นมีประสิทธิภาพเพียงพอที่จะรับมือกับสภาพแวดล้อมของการแปลงข้อมูลตามต้องการ โดยสามารถสร้างแอปพลิเคชั่น DTS เป็นโมดูล โดยสร้างคอมโพเนนท์ที่นำกลับมาใช้ใหม่ในขั้นตอนอื่นหรือการแปลงอื่นได้ ซึ่งโดยโครงสร้างพื้นฐานของ DTS แล้ว จะบันทึกการทำงานผลที่ได้ และ จำนวนข้อมูลที่นำเข้าในแต่ละขั้นตอนให้โดยอัตโนมัติ และสามารถแจ้งให้ผู้บริหารฐานข้อมูล (DBA) ทราบผ่านอีเมล์ หรือเพจเจอร์ได้ในกรณีที่เกิดปัญหาขัดข้อง

ในขณะเดียวกัน DTS ทำให้การรวมข้อมูลจากแหล่งข้อมูลต่างๆ ง่ายขึ้นเป็นอย่างมาก แหล่งข้อมูลใดที่มีไดรเวอร์ OLE DB หรือ ODBC ก็สามารถแปลงและรวมเข้าด้วยกันได้โดยใช้ DTS ซึ่งรวมถึงแหล่งข้อมูลที่เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS) และยังสามารถใช้ DTS ในการกำหนดเวลาและทำงานอัตโนมัติ เพื่อจัดการการส่งต่อข้อมูลในกระบวนการของ Business Intelligence ทั้งหมด ซึ่งงานพิเศษของ DTS ที่เกี่ยวข้องกับ Business Intelligence ได้แก่

- การประมวลผล cube จะสามารถประมวลผลแบบเพิ่มเติมหรือแบบสมบูรณ์ของมิติและ cube ได้
- การคาดการณ์โดยใช้ data mining เมื่อพัฒนาและทดสอบโมเดลของ data mining เรียบร้อยแล้ว ผู้ใช้สามารถใช้ DTS Data Mining Prediction เพื่อใช้โมเดลดังกล่าวกับข้อมูลชุดใหม่ได้

2.2.5 การรวม OLAP เข้ากับคลังข้อมูล

Analysis Services ให้ประสิทธิภาพการทำงานมาก และทำให้ง่ายต่อการสร้างแอปพลิเคชั่นสำหรับงานวิเคราะห์ที่มีความเปลี่ยนแปลงและซับซ้อนอย่างมาก

- ประสิทธิภาพการเรียกค้นที่เหนือชั้น OLAP Cube ให้ประสิทธิภาพที่เหนือกว่าสำหรับการเรียกค้นที่ซับซ้อนเมื่อเทียบกับฐานข้อมูลเชิงสัมพันธ์มาตรฐานทั่วไป วิธีหนึ่ง OLAP ประสบผลสำเร็จในเรื่องของประสิทธิภาพการทำงานคือ ใช้การสรุปผลที่คำนวณผลไว้ล่วงหน้า ด้วยการเตรียมผลสรุป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของการเชื่อมโยงหลายมิติ และการวัดผลไว้ล่วงหน้า OLP cube จึงสามารถให้ผลลัพธ์ได้อย่างรวดเร็ว โดยไม่ต้องอ่านและคำนวณข้อมูลในฐานข้อมูลใหม่ตั้งแต่ต้น

- การสร้างการเรียกค้น OLAP cube ทำให้ผู้ใช้ทางธุรกิจสามารถเรียกใช้ข้อมูลได้มากขึ้น โดยเตรียมโครงสร้างที่เป็นธรรมชาติและเข้าใจง่าย ควบคู่ไปกับการเรียกค้นแบบโต้ตอบ ในขณะที่ฐานข้อมูลเชิงสัมพันธ์เป็นการจัดเก็บข้อมูลทั่วไป แต่ OLAP cube มีมิติ ซึ่งเข้ากันกับวิธีคิดทางธุรกิจ คือพิจารณาถึงลูกค้า กรอบเวลา ผลิตภัณฑ์ เป็นต้น มิติเหล่านี้อาจจะมีลำดับชั้นมากมาย เช่น ผลิตภัณฑ์ที่อยู่นายผลิตภัณฑ์ หรือไตรมาสซึ่งอยู่ภายใต้ปีของงบการเงิน สิ่งเหล่านี้ทำให้ง่ายต่อการพัฒนาแอปพลิเคชั่นซึ่งตรงตามความต้องการของแต่ละธุรกิจ

Structure Query Language (SQL) ซึ่งเป็นภาษาที่ใช้ในการเรียกค้นฐานข้อมูลเชิงสัมพันธ์นั้น เหมาะสำหรับโครงสร้างแบบตารางที่ผ่านการ normalize แล้ว คำถามที่พบโดยทั่วไปในการวิเคราะห์เชิงธุรกิจ เช่น "ผลกำไรของช่วงระยะเวลานี้เป็นอย่างไรเมื่อเทียบกับเวลาช่วงเดียวกันของปีที่แล้ว" ค่อนข้างยากมาก หากจะเรียกค้นด้วยคำสั่ง SQL

Analysis Services ใช้การเรียกค้นแบบหลายมิติ และ MDX ซึ่งเป็นภาษาที่ดีที่สุดสำหรับทำงานกับมิติที่เป็นลำดับขั้น จึงง่ายมากต่อการเรียกค้นเพื่อตอบคำถามทั่วไปทางธุรกิจ ดังนั้น เวลาที่เสียไปในการพัฒนา OLAP ในช่วงเริ่มต้นจะได้รับผลตอบแทนอย่างเต็มที่ โดยทำให้สร้างแอปพลิเคชั่นได้อย่างรวดเร็ว และสร้างและปรับปรุงการเรียกค้นสำหรับการวิเคราะห์แบบโต้ตอบได้ง่ายขึ้น

Analysis Manager จะทำให้กระบวนการพัฒนา cube ง่ายและเร็วขึ้นด้วยวิซาร์ด ซึ่งแนะนำการสร้างอย่างเป็นขั้นตอน และทำงานในรูปของกราฟฟิก

2.2.6 การใช้ OLAP เป็นฟรอนท์เอนด์ของคลังข้อมูล

การใช้ Analysis Services ในฐานะของฟรอนท์เอนด์สำหรับผู้ใช้ทุกคน ในการเรียกใช้ข้อมูลของหน่วยเก็บข้อมูลเชิงสัมพันธ์ หรือกล่าวอีกนัยหนึ่ง คือ จะมีเฉพาะแอปพลิเคชั่นและนักพัฒนาเท่านั้นที่ได้รับอนุญาตให้เรียกค้นไปยังหน่วยเก็บข้อมูลเชิงสัมพันธ์ได้โดยตรง

ผลดีหลายประการสำหรับวิธีการนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ช่วยให้จัดการกับหน่วยเก็บข้อมูลเชิงสัมพันธ์ได้ง่าย ในกรณีที่ไม่ต้องการให้ผู้ใช้เรียกค้นข้อมูล ตัวอย่างเช่น หากตัวหน่วยเก็บข้อมูลเชิงสัมพันธ์ต้องการใช้ดัชนี 2-3 ดัชนี ก็จะช่วยลดขนาดของฐานข้อมูลลงได้ โดยอาจเลือกใช้การพาร์ทิชันโครงสร้างที่ต่างกันให้กับฐานข้อมูลเชิงสัมพันธ์ ซึ่งผู้ใช้จะไม่รู้ถึงการเปลี่ยนแปลง
- กระบวนการจัดการข้อมูลจะไม่ส่งผลต่อประสิทธิภาพของผู้ใช้ ในขณะที่ผู้ใช้เรียกค้น cube ที่ไม่ใช้แปลงข้อมูลเชิงสัมพันธ์ Analysis Services จะป้องกันไม่ให้ผู้ใช้ถูกขัดจังหวะจากการประมวลผล cube ใหม่
- Analysis Manager ทำให้กระบวนการสร้างแอปพลิเคชั่นง่ายและรวดเร็วขึ้นด้วยอินเทอร์เฟสและวิซาร์ดจำนวนมาก

ในขณะที่ cube ได้รับการแยกออกจากหน่วยเก็บข้อมูลเชิงสัมพันธ์ และตอบสนองต่อการอัพเดทและเรียกค้นจากผู้ใช้ การรวมกันระหว่าง Analysis Server และ SQL Server ได้ก่อให้เกิดมุมมองของข้อมูลที่เข้ากันได้อย่างไร้ที่ติ ตัวอย่างเช่น

- ความสามารถในการลงสู่รายละเอียดใน Analysis Server ทำให้ผู้ใช้เข้าถึงรายละเอียดที่ย่อยที่สุดในหน่วยเก็บข้อมูลเชิงสัมพันธ์ได้ (ตราบเท่าที่ผู้บริหารฐานข้อมูลอนุญาต) เพื่อดูรายละเอียดที่ของตัวเลขดังกล่าว การเชื่อมโยงอย่างใกล้ชิดระหว่างฟรอนท์เอนด์และแบ็คเอนด์จึงหมายถึงการที่ผู้ใช้สามารถเข้าถึงข้อมูลทุกระดับในคลังข้อมูลได้อย่างเต็มที่
- ในบางกรณี ผู้บริหารระบบอาจจะเลือกจัดเก็บ OLAP cube ไว้ที่ฐานข้อมูลเชิงสัมพันธ์เอง (เรียกว่าการจัดเก็บแบบ Relational OLAP หรือ ROLAP) หรือจัดเก็บเฉพาะลำดับชั้นของข้อมูลไว้ใน Analysis Server และเก็บรายละเอียดไว้ที่ SQL Server (คือการจัดเก็บแบบ Hybrid OLAP หรือ HOLAP) Analysis Services สามารถทำงานกับรูปแบบการจัดเก็บข้อมูลแบบใดก็ได้ เปิดโอกาสผู้ใช้มองเห็นข้อมูลได้อย่างสมบูรณ์ โดยไม่ต้องสนใจว่าจริง ๆ แล้ว ข้อมูลได้รับการจัดเก็บไว้ที่ใด

2.2.7 การให้ข้อมูลเชิงลึก

Data mining สามารถให้ข้อมูลเชิงลึกจากข้อมูลดิบปริมาณมากๆ ช่วยให้ค้นพบองค์ความรู้ซึ่งซ่อนอยู่ในข้อมูล เช่น ความสัมพันธ์ระหว่างผลิตภัณฑ์ใดที่ลูกค้านิยมซื้อและลูกค้าซื้อเมื่อไร หรือลูกค้าอาศัยอยู่ที่ไหน และสัมพันธ์กับสินค้าที่ซื้ออย่างไร การเข้าใจถึงพฤติกรรมของผู้บริโภค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ดีขึ้น มักสามารถเปลี่ยนให้กลายเป็นข้อได้เปรียบในการแข่งขันผ่านทางการปรับแคมเปญทางการตลาด การบรรจุหรือการโปรโมชั่นผลิตภัณฑ์ใหม่ หรือการพัฒนาผลิตภัณฑ์ใหม่ นอกจากนี้ data mining ยังเป็นประโยชน์ในด้านการเงินสำหรับดำเนินงาน เช่น ประเมินความเสี่ยงของเครดิต และการตรวจจับการปลอมแปลงต่างๆ

ด้วยการรวม data mining เข้ากับผลิตภัณฑ์ Analysis Services ไมโครซอฟต์ทำให้ความสามารถนี้ใช้ได้กับธุรกิจทุกประเภท โดย Analysis Services มีอัลกอริธึมของ data mining ที่มีประสิทธิภาพและง่ายต่อการใช้งาน 2 อัลกอริธึมที่สามารถนำมาใช้กับข้อมูลที่มีอยู่ใน cube ซึ่งได้แก่ Decision Trees และ Clustering และยังมีอินเทอร์เฟสเปิด ทำให้สะดวกต่อการเพิ่มความสามารถให้กับ data mining ภายใต้สภาพแวดล้อมของแอปพลิเคชั่นสนับสนุนการตัดสินใจ

2.2.8 การส่งผลวิเคราะห์สู่ผู้ใช้

สถาปัตยกรรมของ Analysis Server ได้รับการออกแบบมาเพื่อให้ทำงานแบบไคลเอนท์/เซิร์ฟเวอร์ ที่ซึ่งส่วนประกอบของไคลเอนท์ (PivotTable Services) อยู่บนระบบเดสก์ท็อปและเชื่อมโยงผ่านระบบเครือข่ายไปยัง Analysis Server ภายใต้สภาพแวดล้อมของไคลเอนท์บนเว็บ อุปกรณ์เคลื่อนที่ หรือไคลเอนท์ที่ไม่ได้เชื่อมต่อกับระบบเครือข่าย Analysis Services 2000 สนับสนุนทางเลือกในการเชื่อมต่อของไคลเอนท์หลายประเภทด้วยกันคือ

- Rich Client แบบออนไลน์ ไคลเอนท์จะมีความสามารถต่างๆ มากมายจากแอปพลิเคชั่นบนไคลเอนท์เอง ในเวอร์ชั่นแรกๆ ไคลเอนท์เหล่านี้จำเป็นต้องอยู่บนโดเมนเดียวกันกับเซิร์ฟเวอร์ แต่ด้วยความสามารถในการเชื่อมโยงแบบ HTTP ของ Analysis Services ไคลเอนท์เหล่านี้จึงสามารถอยู่ที่ใดก็ได้บนอินเตอร์เน็ต และทำการเชื่อมโยงผ่านไฟร์วอลล์มายังเซิร์ฟเวอร์ที่ให้การสนับสนุนวิธีการเชื่อมต่อดังกล่าว
- Rich Client แบบออฟไลน์ สำหรับผู้ใช้ทางธุรกิจที่ต้องการเรียกใช้ความสามารถในการวิเคราะห์ โดยไม่ต้องเชื่อมต่อกับเครือข่ายตลอดเวลา Analysis Services ยังสนับสนุนการวิเคราะห์แบบออฟไลน์ผ่านการใช้งานของ cube บนเครื่องที่ใช้งาน โดย cube นี้จะเป็นส่วนหนึ่งของ cube บนเซิร์ฟเวอร์ที่ถูกกำหนดไว้แล้ว ซึ่งสามารถดาวน์โหลด ไปยังผู้ใช้เพื่อทำงานวิเคราะห์แบบออฟไลน์
- Thin client การกำหนดค่าการทำงานต่างๆ ที่อธิลายไว้ตั้งแต่ต้น อยู่ภายใต้เงื่อนไขว่ามี PivotTable Services บนเครื่องไคลเอนท์แล้ว แต่บ่อยครั้งที่องค์กรธุรกิจต้องการใช้ไคลเอนท์บนเว็บเพื่อให้ใช้งานได้จากระบบทุกชนิดซึ่งมีสิ่งเดียวเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่มีอยู่ในระบบเหล่านี้ นั่นคือบราวเซอร์ ดังนั้นเพื่อที่จะใช้ Analysis Services นั้น สามารถทำให้เว็บเซิร์ฟเวอร์ Microsoft Internet Information Services (IIS) ทำหน้าที่เสมือนกับไคลเอนท์ของ Analysis Services (ที่ติดตั้ง PivotTable Services) ได้ และสร้างเว็บเพจแบบไดนามิค (Active Server Pages หรือ ASP) เพื่อการโต้ตอบกับ Analysis Server

2.2.9 การสร้างและจัดการ Cube

วิธีหนึ่งที่ทำให้ระบบ OLAP ประสบความสำเร็จในเรื่องประสิทธิภาพของการเรียกค้นที่ซับซ้อนก็คือ การออกแบบกลุ่มข้อมูลไว้ล่วงหน้าจำนวนหนึ่ง ซึ่งในการปรับแต่งประสิทธิภาพในการเรียกค้น จำเป็นต้องจัดการกับกลุ่มข้อมูลเหล่านี้ก่อน

การจัดกลุ่มข้อมูลนั้น แท้ที่จริงแล้วคือการเพิ่มข้อมูลลงไปใน cube ซึ่งอาจจะไปถึงจุดที่กลุ่มข้อมูลมีปริมาณมากกว่าข้อมูลเริ่มต้น ปัญหานี้เรียกว่า การระเบิดของข้อมูล (Data Explosion) ซึ่งการจัดการกลุ่มเหล่านี้จะต้องมีการคิดอย่างรอบคอบ

Analysis Manager ให้การจัดการกลุ่มข้อมูลที่ชาญฉลาด โดยทำให้กระบวนการจัดการระบบ OLAP ง่ายขึ้น ลดการจัดเก็บข้อมูลและกลุ่ม โดย

- ให้การจัดกลุ่มข้อมูลล่วงหน้าที่ชาญฉลาด ในการเลือกทำเฉพาะกลุ่มที่มีความสำคัญสูงสุด เช่น กลุ่มซึ่งใช้เป็นฐานในการคำนวณให้กับกลุ่มข้อมูลอื่น
- ติดตามการเรียกค้นซึ่งทำให้ผู้บริหารระบบสามารถปรับแต่งการจัดกลุ่มข้อมูลได้ตรงกับการเรียกค้นที่เกิดขึ้นกับ cube อย่างแท้จริง
- ช่วยบ่งชี้ให้เห็นถึงผลดีผลเสียระหว่างปริมาณเนื้อที่จัดเก็บกลุ่มข้อมูลกับประสิทธิภาพที่ได้รับ และยอมให้ผู้บริหารระบบเลือกและปรับแต่งให้สมดุลได้ ซึ่งส่วนใหญ่จะมีจุดที่ทำให้เกิดประสิทธิภาพสูงสุดโดยไม่เพิ่มเนื้อที่ในการจัดเก็บ ซึ่งหากอยู่ในระดับที่ต่ำกว่านี้ จะแสดงให้เก็บว่ากำลังเพิกเฉยต่อการปรับปรุงประสิทธิภาพ และ ณ ระดับที่สูงกว่า ก็หมายถึงกำลังใช้เนื้อที่จัดเก็บข้อมูลมากเกินไป เพื่อให้ได้มาซึ่งประสิทธิภาพที่เพิ่มขึ้นเพียงเล็กน้อย

2.2.10 ระบบงานที่รองรับการเติมโตของ Business Intelligence

ผลอย่างหนึ่งที่พบเหมือนๆ กันในการจัดทำคลังข้อมูลคือ ผู้ใช้จะไม่รู้ว่าตัวเองต้องการอะไรจากแอปพลิเคชั่น ด้านการวิเคราะห์ จนกระทั่งได้รับผลการวิเคราะห์แล้ว ดังนั้นแพล็ตฟอร์มสำหรับการวิเคราะห์ที่ดีต้องยืดหยุ่นและปรับตัวได้ และสามารถขยายระบบเพื่อรองรับการเติบโต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของทั้งจำนวนผู้ใช้และข้อมูลได้ในปริมาณมาก รวมทั้งรองรับความซับซ้อนของการวิเคราะห์ได้ด้วย

นอกเหนือจากความง่ายในการใช้งานและการรวมเข้าด้วยกันดังที่ได้กล่าวไปแล้ว Analysis Services ยังมีความสามารถด้านการวิเคราะห์ที่สลับซับซ้อน และสามารถทำงานในสภาพแวดล้อมส่วนใหญ่ ในส่วนต่อไปนี้ จะกล่าวถึงวิธีที่ทำให้ Analysis Services สามารถตอบสนองความต้องการใหม่ๆ ทางธุรกิจได้อย่างต่อเนื่อง

2.2.10.1 การวิเคราะห์ขั้นสูง

ผลิตภัณฑ์ Analysis Services มีทั้งประสิทธิภาพสูง และสามารถขยายขีดความสามารถออกไปได้อีก นักพัฒนาแอปพลิเคชั่นสามารถเรียกใช้คุณสมบัติต่างๆ ซึ่งช่วยให้แอปพลิเคชั่นสนองความต้องการเฉพาะทางได้เต็มที่ ซึ่งได้แก่

- การคำนวณ สามารถเรียกใช้ฟังก์ชันที่มีอยู่มากกว่า 130 ฟังก์ชันใน Analysis Services รวมทั้งสามารถเรียกใช้ฟังก์ชันของ Excel และ Visual Basic ได้ หรือเขียนสูตรคำนวณขึ้นเองในรูปของภาษา MDX ที่ถูกออกแบบมาสำหรับแหล่งข้อมูลแบบหลายมิติ ซึ่งฟังก์ชันเหล่านี้จะทำให้เรียกใช้ขอ้มูลในลำดับชั้นหรือมิติต่างๆ และมีข้อมูลเกี่ยวกับมิติที่ถูกกำหนดไว้ให้แล้ว เช่น ข้อมูลด้านเวลา
- เซลล์ที่มีการคำนวณ สามารถกำหนดสูตรคำนวณให้กับเซลล์หรือชุดของเซลล์
- Virtual cube คล้ายกับ view ในโลกของฐานข้อมูลเชิงสัมพันธ์ โดย cube แบบเสมือนจะทำหน้าที่เชื่อมโยงข้อมูลจากหลายๆ cube ที่มีการใช้มิติร่วมกัน
- ผู้ใช้สามารถบันทึกข้อมูลกลับได้โดยตรงบน cube ที่อนุญาตให้ทำการแก้ไขได้ และยังมีมิติที่อนุญาตให้แก้ไขได้ ซึ่งถูกใช้สำหรับการจัดทำโมเดลโครงสร้างของมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.10.2 ติดตั้งการวิเคราะห์ให้กับธุรกิจ

ในขณะที่แอปพลิเคชั่นสำหรับการวิเคราะห์ที่สร้างบน Analysis Services เริ่มมีความจำเป็นต่อธุรกิจเพิ่มขึ้นเรื่อยๆ นั้น อาจพบว่า ต้องการรวมแอปพลิเคชั่นเหล่านี้เข้ากับกระบวนทางธุรกิจให้แนบแน่นมากขึ้นกว่าเดิม

Analysis Services มีคุณสมบัติที่เรียกว่า Actions ซึ่งสามารถเชื่อมโยงการวิเคราะห์เข้ากับการทำงานในธุรกิจ จึงสามารถช่วยให้ธุรกิจตอบสนองต่อเงื่อนไขที่เปลี่ยนแปลงอยู่ตลอดเวลาได้อย่างรวดเร็ว

2.2.10.3 การรักษาความปลอดภัย

การรักษาความปลอดภัยที่ยืดหยุ่น เป็นความต้องการที่สำคัญในสภาพแวดล้อมของการทำธุรกิจ Analysis Services มีการรักษาความปลอดภัยตามลักษณะหน้าที่ พร้อมกับมีระดับความปลอดภัยให้เลือกใช้

ที่ระดับของเซิร์ฟเวอร์ ฐานข้อมูล cube และการทำโมเดลการวิเคราะห์ข้อมูล ผู้บริหารระบบสามารถกำหนดสิทธิการเข้าใช้ให้กับผู้ใช้ได้ตั้งแต่เรียกใช้ได้ทั้งหมดไปจนถึงเรียกใช้ไม่ได้เลย

การรักษาความปลอดภัยในระดับของมิตินั้น ผู้บริหารสามารถเปิดให้เรียกใช้มิติที่กำหนด หรือบางส่วนของมิติก็ได้

การรักษาความปลอดภัยในระดับเซลล์ ให้การควบคุมการเรียกใช้ข้อมูลที่ละเอียดกว่าเดิม ผู้บริหารระบบสามารถพัฒนากฎเกณฑ์ที่ซับซ้อนเพื่อจำกัดผู้ใช้เฉพาะชุดของเซลล์ที่ระบุ ที่ระดับต่างๆ ใน cube เท่านั้น

2.2.10.4 รับมือกับข้อมูลจำนวนมาก

SQL Server และ Analysis Services สามารถรับมือกับข้อมูลปริมาณมาก ซึ่งเป็นเรื่องปกติของคลังข้อมูลขนาดใหญ่ Analysis Services จึงเตรียมซอฟต์แวร์บริหารให้แก่ผู้บริหารระบบ สำหรับจัดการกับข้อมูลจำนวนมาก ดังนี้

- การบีบอัดหน่วยเก็บข้อมูลที่ใช้เก็บ cube ของ Analysis Services ลงเหลือเพียง 25-40 เปอร์เซ็นต์ของพื้นที่ที่ฐานข้อมูลเชิงสัมพันธ์ใช้ (ไม่รวมอินเด็กซ์) โดย cube นั้นได้รวมอินเด็กซ์และข้อมูลสรุปไว้ด้วย
- การปรับต่างตามการใช้งาน อนุญาตให้ผู้บริหารระบบทำการสร้างกลุ่มข้อมูลชุดเล็ก ซึ่งตรงตามความต้องการเรียกค้นของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การทำพาร์ทิชันช่วยให้ผู้บริหารระบบสร้าง logical cube ที่อาศัยข้อมูลจาก cube ที่มีอยู่จริงหลายๆ cube สิ่งนี้เป็นทางเลือกสำหรับการทำข้อมูลสรุป การประมวลผล หรือการจัดเก็บส่วนต่างๆ ของ cube แยกออกจากกัน เช่น พาร์ทิชันของ cube ที่ต่างกัน อาจมีแผนจัดทำข้อมูลสรุปต่างกัน โดยพาร์ทิชันสามารถใช้วิธีการเก็บที่ต่างกัน และถูกกระจายไปยังเซิร์ฟเวอร์หลายตัว ซึ่งผู้บริหารระบบสามารถประมวลผลพาร์ทิชันเดียว เมื่อต้องการอัพเดท cube หรือประมวลผลหลายพาร์ทิชันพร้อมกันเพื่อประสิทธิภาพการปรมวลผลที่รวดเร็ว
- หน่วยจัดเก็บแบบ ROLAP มีประโยชน์สำหรับมิติขนาดใหญ่มาก เนื่องจากทั้ง แหล่งข้อมูล กลุ่มข้อมูล และ มิติ จะถูกเก็บอยู่ในฐานข้อมูลเชิงสัมพันธ์
- หน่วยเก็บแบบ HOLAP มีประโยชน์สำหรับข้อมูลจำนวนมาก ซึ่งข้อมูลรายละเอียดถูกเรียกค้นไม่บ่อยนัก

2.2.11 บทสรุปของ SQL Server 2000 และ Analysis Services

ทั้ง SQL Server 2000 และ Analysis Services เป็นเครื่องมือที่มีประสิทธิภาพและเหมาะสมที่จะนำมาใช้ในการจัดการคลังข้อมูล และสามารถใช้ประโยชน์จากการวิเคราะห์ข้อมูลได้อย่างลึกซึ้ง ในขณะที่แพลตฟอร์มมีความยืดหยุ่น เปิดกว้าง เพื่อให้ผู้ใช้ข้อมูลสามารถเลือกแอปพลิเคชั่นที่เหมาะสมในการนำข้อมูลจาก Analysis Server มาวิเคราะห์ได้

2.3 MDX - Multidimensional Expressions Language

MDX เป็นคำสั่ง Query Language ประเภท DML – Data Manipulation Language ซึ่งเป็นส่วนหนึ่งใน Microsoft SQL 2000 Analysis Services มีโครงสร้างของคำสั่งและหน้าที่การทำงานคล้ายกับ SQL แต่ส่วนที่ต่างกันคือ MDX สามารถดึงข้อมูลมาจาก Multidimensional Database หรือ OLAP Cube ได้ทันที เพราะมีโครงสร้างของภาษาที่ออกแบบมาให้ทำหน้าที่แบบนี้อยู่แล้ว และข้อมูลที่ได้มาอาจยังคงอยู่ในสภาพที่เป็นลักษณะหลายมิติก็ได้ ในขณะที่ข้อมูลที่ได้จาก SQL จะมีเพียงสองมิติคือ Column และ Row เท่านั้น ดังนั้นในคำสั่ง MDX จึงต้องมีการระบุ Dimension ให้กับข้อมูลที่ได้รับมาด้วย เช่น Row หรือ Column หรือ Page

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบคำสั่ง MDX มีโครงสร้างคล้ายกับ SQL โดยมีลักษณะดังนี้

SELECT [<axis_specification> [,<axis_specification> …]]

FROM [<cube_specification>]

[WHERE [<slicer_specification>]]

โดยที่

-axis_specification เป็นการเลือก member หรือ เซตของ member ซึ่งตามที่ได้กล่าวไปแล้วว่าจะต้องระบุ dimension ให้กับ member ที่เลือกด้วย ดังนั้น รูปแบบของ axis_specification จะต้องกำหนดด้วยว่า on row หรือ on column ไว้ด้วยเสมอ เช่น [Product].[Food] on column เป็นต้น ดังนั้น axis_specification จะมีรูปแบบคือ <set> on <axis_name> ซึ่ง set ก็คือเซตของ member และ axis_name สามารถระบุโดยชื่อ ได้แก่ ได้แก่ columns, rows, pages, sections, chapters หรือระบุเป็น index เช่น axis(0), axis(1), axis(2) ไปเรื่อย ๆ ก็ได้ ซึ่ง columns คือ axis(0), rows คือ axis(1) และ pages คือ axis(2) นั่นเอง ใน MDX นี้ยอมให้มี axis ได้ทั้งหมด 64 axis แต่สามารถระบุแบบชื่อได้ถึง chapters เท่านั้นหลังจากนั้นต้องระบุเป็นแบบ index และในการเขียนควรเรียงจาก index น้อยไปหามากเพื่อป้องกันการสับสน

-cube_specification คือชื่อ cube ที่ต้องการ เหมือนกับการระบุชื่อ table ในคำสั่ง SQL

-slicer_specification เป็นการระบุเงื่อนไขในการเลือกข้อมูล เหตุที่ใช้คำว่า slicer เนื่องจากโครงสร้าง cube เป็นลักษณะ multidimensional เมื่อระบุเงื่อนไขใน dimension หนึ่งแล้ว ก็เหมือนกับว่าข้อมูลที่ได้มานี้จะอยู่ภายใต้เงื่อนไขของ dimension นี้เท่านั้น ทั้งนี้ dimension ที่ถูกระบุใน axis_specification แล้ว จะไม่สามารถนำมาใช้ใน slicer_specification ได้อีก หรือกล่าวอีกนัยหนึ่งคือ axis_specification และ slicer_specification จะต้องเป็น dimension ที่ต่างกันเสมอ

คำสั่ง MDX มีความสำคัญสำหรับการพัฒนาโปรแกรมสำหรับวิเคราะห์ข้อมูลการขายสินค้าสำหรับโครงงานนี้ เนื่องจากโปรแกรมดังกล่าวจะมีการนำสิ่งต่าง ๆ ที่ผู้ใช้ต้องการมาสร้างเป็นคำสั่ง MDX เพื่อส่งไปขอข้อมูลจาก OLAP Server แล้วนำข้อมูลมานำเสนอในโปรแกรมต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ADO-MD Object Library

ADO-MD (ActiveX Data Objects Multidimensional) เป็น Object Library สำหรับใช้ติดต่อกับ OLAP Server ของ Microsoft ® SQL Server™ 2000 Analysis Services ซึ่งมีประโยชน์มากสำหรับการพัฒนา Application ในเชิงวิเคราะห์ เนื่องจากสามารถดึงข้อมูลจาก OLAP Cube ได้โดยตรงไม่ว่าจะเป็นโครงสร้างของ Cube (Cube Schema) หรือข้อมูลภายใน Cube (Cube Data)

โดยปกติฐานข้อมูลที่เป็น Multidimensional หรือ OLAP Cube นี้จะประกอบด้วยส่วนหลักสองส่วน คือ โครงสร้าง (Multidimensional Schema) และส่วนที่เป็นข้อมูล (Data) ซึ่ง Schema จะเป็นส่วนที่กำหนดว่า Cube นั้นมี โครงสร้างอย่างไร ส่วน Data จะเป็นข้อมูลที่เก็บอยู่ใน Cube ซึ่งบางกรณีอาจเรียกว่า Measurement ก็ได้ และ ADO-MD ก็จะมีกระบวนการในการเข้าถึง Schema และ Data ได้ทั้งสองส่วน

2.4.1 Cube Schema

Cube Schema คือโครงสร้างของ Cube โดยก่อนที่จะทำการสร้าง Cube จะต้องมีการกำหนดโครงสร้างเหล่านี้ขึ้นมาก่อน แล้วจึงทำการสร้าง (Process Cube) ขึ้นมา ซึ่งโครงสร้างเหล่านั้นประกอบด้วย

-Dimension คือโครงสร้างหลักของ Cube ถ้ามองอย่างง่าย ถ้า Cube เป็นสี่เหลี่ยมลูกบาศก์ Dimension ก็คือด้านทั้งสามของ Cube นั่นเอง แต่ในความเป็นจริง Dimension อาจจะมีได้มากกว่า 3 Dimension ก็ได้

-Hierarchy คือความสัมพันธ์แบบ Parent – Child ภายใน Dimension เดียวกัน เนื่องจากในแต่ละ Dimension นั้น อาจมีการจัดหมวดหมู่ของข้อมูล ตัวอย่างเช่น ใน Geographic Dimension มีการแบ่งเป็นภาคและจังหวัด ซึ่งจังหวัดหนึ่ง ๆ จะต้องขึ้นอยู่กับภาคใดภาคหนึ่งเสมอ ในกรณีนี้ภาคจะเป็น Parent และ จังหวัดที่อยู่ภายในภาคก็จะเป็น Child ของภาคนั้น เป็นต้น

-Level เป็นการแบ่งลำดับขั้น ภายใน Dimension จากตัวอย่างเดิมคือภาคและจังหวัด จะมี Level ที่เป็นภาค และ Level ที่เป็นจังหวัด เป็นต้น

-Member คือข้อมูลที่อยู่ภายในแต่ละ Dimension ซึ่งก็คือสมาชิกทั้งหมดภายใน Dimension นั้น ๆ นั่นเอง

ใน ADO-MD จะมี Object กลุ่มที่ทำงานกับ Cube Schema โดยเฉพาะ ซึ่งก็คือ Catalog, CubeDef, Dimension, Hierarchy และ Level โดยทำงานสอดคล้องกับ Cube Schema โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 Cube Data

Cube Data หรือข้อมูลภายใน Cube บางทีเรียกว่า Measurement ซึ่งใน ADO-MD จะสามารถส่งคำสั่งไปขอข้อมูลใน Cube โดยใช้คำสั่ง MDX และผลของคำสั่ง MDX นี้จะกลับมาอยู่ในรูปแบบ Table ธรรมดาที่มีสองมิติ หรือให้อยู่ในรูปแบบที่มากกว่าสองมิติก็ได้ โดยเรียกกลุ่มข้อมูลที่ได้กลับมาจากกระบวนการนี้ว่า CellSet ซึ่งใน ADO-MD จะมี Object ที่ทำหน้าที่ในการจัดการผลของ CellSet ประกอบด้วย

-Axis ก็คือ แกนของข้อมูลที่ได้กลับมา ถ้าเปรียบเทียบกับฐานข้อมูลแบบธรรมดา ผลของการ Query จะได้เป็นสองแกน คือ แนวตั้งและแนวนอน แต่ใน CellSet นี้สามารถมีได้มากกว่าสองแกน ซึ่ง axis นี้มักจะสอดคล้องกับ Dimension ที่อยู่ใน Cube ซึ่งส่วนมากมักจะ Query โดยระบุให้ axis เป็น Dimension หนึ่งใน Cube แต่ก็มีบางกรณี ที่สามารถระบุให้ axis ประกอบด้วย Dimension มากกว่าหนึ่ง Dimension ก็ได้ ซึ่งจะเรียก axis นั้นว่าเป็น Nested Dimension

-Position คือตำแหน่งต่าง ๆ หรือสมาชิกใน axis เช่นถ้าระบุให้ Query ดึงข้อมูลหนึ่ง Dimension ลงมาใน axis ผลที่ได้ก็คือ Position ก็คือ Member ของ Dimension นั้นนั่นเอง แต่หาก axis นั้นเป็น Nested Dimension จะได้ว่า Position ก็คือคู่อันดับของ Member ของทั้งสอง Dimension นั้น

-Cell คือที่เก็บข้อมูลจริง ๆ หรือมองอีกนัยหนึ่งก็คือข้อมูลที่เก็บอยู่ที่เกิดจากการระบุ Position จากทุก ๆ axis โดยที่ Cell หนึ่ง Cell จะเก็บข้อมูลตัวเลขได้เพียงหนึ่งชุดเท่านั้น

ในบางกรณีคำสั่ง MDX ที่ส่งไปขอข้อมูลนั้นอาจระบุให้ Measurement ใน Cube กลับมาเป็น axis ก็ได้ โดยลักษณะนี้จะมอง Measurement เป็นเหมือนกับ Dimension และ Position ของ axis ก็คือค่าต่าง ๆ ที่เป็น Measure นั่นเอง

ADO-MD Object Library มีความสำคัญต่อการพัฒนาโปรแกรมสำหรับวิเคราะห์ข้อมูลการขายสินค้า สำหรับโครงงานนี้ เนื่องจากโปรแกรมดังกล่าวจะต้องมีการเรียกใช้ ADO-MD Object เพื่อติดต่อกับ OLAP Cube ในการทำงานกับโครงสร้างของ Cube หรือขอข้อมูลที่ถูกบรรจุอยู่ภายใน Cube ดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบการจัดเก็บข้อมูลการขายสินค้าสำหรับทำ OLAP Cube

เพื่อให้บรรลุถึงวัตถุประสงค์ของการวิเคราะห์ข้อมูลการขายอย่างมีประสิทธิภาพ จึงต้องมีการศึกษากระบวนการที่เกี่ยวข้องกับข้อมูลการขายสินค้าที่มีอยู่ในระบบเดิม และออกแบบการจัดเตรียมข้อมูลให้อยู่ในรูปแบบที่เหมาะสมในการจัดเก็บลง Data Warehouse และยังต้องคำนึงถึงความสะดวกในการใช้งานข้อมูลใน Data Warehouse ผ่านทางโปรแกรม OLAP ที่จะพัฒนาขึ้นมาอีกด้วย

3.1 การคัดเลือกข้อมูลจากระบบการขายที่เป็น OLTP

ระบบการขายสินค้าขององค์กรที่เป็น OLTP อยู่บนเครื่อง AS/400 และข้อมูลการขายสินค้าต่าง ๆ ถูกจัดเก็บอยู่ในฐานข้อมูล DB2 โดยมีการเก็บข้อมูลการขายสินค้าทุกวัน และภายในฐานข้อมูลได้มีการจัดเก็บข้อมูลที่เป็นรายละเอียดสินค้า และข้อมูลของลูกค้าเป็น Master File ด้วย ดังนั้นจึงนำข้อมูลจากทั้งสามส่วนดังกล่าว ได้แก่ ข้อมูลการขาย (Sales Transaction) ข้อมูลสินค้า (Product Master) และข้อมูลลูกค้า (Customer Master) เข้ามาเก็บไว้ใน Data Warehouse เพื่อใช้งานในการวิเคราะห์ผ่านโปรแกรม OLAP ต่อไป

3.2 การออกแบบตารางข้อมูลใน Data Warehouse

การออกแบบตารางสำหรับฐานข้อมูลใน Data Warehouse จะต้องคำนึงถึงการนำไปใช้โดย OLAP เป็นสำคัญ นั่นคือหลักการออกแบบ ต้องมาจากความต้องการของผู้ใช้งาน และจะต้องสอดคล้องกับข้อมูลที่มีอยู่ในระบบการขายที่เป็น OLTP ด้วย

ความต้องการในการวิเคราะห์การขายสินค้าในองค์กรแบ่งเป็นสามส่วนหลัก ๆ ได้แก่

1. การวิเคราะห์ด้านสินค้า
2. การวิเคราะห์ด้านลูกค้า
3. การวิเคราะห์ด้านเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวิเคราะห์ด้านสินค้านั้น เนื่องจากเป็นองค์กรที่ขายสินค้าที่ประเภทเครื่องดื่ม และมีสินค้าทั้งหมดหลายรายการ แต่ก็ได้มีการจัดหมวดหมู่สินค้าแต่ละชนิดเป็นสามแบบ ได้แก่ Brand ของสินค้า ขนาดบรรจุของสินค้า และรสชาติของสินค้าแต่ละชนิด ดังนั้นในสินค้าแต่ละชนิดจะต้องถูกระบุไว้เสมอว่าเป็น Brand อะไร ขนาดบรรจุเท่าไหร่ และเป็นรสชาติใด นอกจากนี้ยังมี Attribute ที่จะนำมาใช้อีกหนึ่ง Attribute ได้แก่ จำนวนหน่วยย่อยในหนึ่งหน่วยใหญ่ ซึ่งเรียกว่า Number of SubUnit เนื่องจากการขายสินค้านั้น จำนวนที่ขายไปจะนับเป็นหน่วยย่อย เช่นขวด แต่การขนส่งจะสนใจหน่วยที่ใหญ่กว่า เช่นลัง เป็นต้น จึงต้องมี Attribute นี้สำหรับการวิเคราะห์ด้วย ซึ่ง Attribute ทั้งสี่ที่กล่าวมานี้ถูกบรรจุอยู่ใน Product Master ในฐานข้อมูลแบบ OLTP อยู่แล้ว ซึ่งสามารถใช้กระบวนการ ETL นำข้อมูลเหล่านี้เข้ามาบรรจุอยู่ใน Data Warehouse ได้ทันที ดังนั้นจึงออกแบบตาราง Product Dimension Table ที่เก็บในฐานข้อมูล Data Warehouse ได้ดังตารางที่ 3.1

ตารางที่ 3.1 Product Dimension Table

Column Name	Description	Data Type	Length
ProductCode	รหัสสินค้า	int	4
ProductDesc	ชื่อสินค้า	varchar	30
NumSubUnit	จำนวนหน่วยย่อยใน 1 หน่วยใหญ่	int	4
PSize	ขนาดบรรจุ	varchar	30
PFlavour	รสชาติ	varchar	30
PBrand	Brand ของสินค้า	varchar	30

การวิเคราะห์ด้านลูกค้านั้น เนื่องจากองค์กรทำการขายสินค้าแบบขายส่ง ดังนั้นลูกค้าขององค์กรได้แก่ร้านค้าปลีก หรือซุปเปอร์มาร์เก็ตต่าง ๆ รวมถึงร้านอาหาร และโรงภาพยนตร์ต่าง ๆ ด้วย และจากการที่มีลูกค้าจำนวนมากและหลากหลายกลุ่ม องค์กรจึงได้มีแนวทางในการแบ่งกลุ่มลูกค้าเป็นสองแนวทาง คือ

1. แบ่งตามปริมาณการขาย โดยกลุ่มลูกค้าที่มีปริมาณการขายต่อเดือนสูง เช่นพวกซุปเปอร์มาร์เก็ต หรือโรงภาพยนตร์ กลุ่มลูกค้าเหล่านี้จะมี พนักงานขายที่เรียกว่า Key Account มาดูแลโดยเฉพาะ และการขนส่งสินค้าก็จากแตกต่างออกไป ส่วนกลุ่มลูกค้าปกติ ส่วนมากเป็นร้านค้าปลีก ร้านค้าเหล่านี้มีปริมาณมาก แต่ยอดขายแต่ละร้านจะไม่สูงมากนัก ร้านค้าเหล่านี้ก็จะมีพนักงานขายดูแลเช่นเดียวกัน แต่พนักงานขายหนึ่งคน จะต้องดูแลร้านค้าหลายร้าน

2. แบ่งตามการทำงานของฝ่ายขาย โดยรวมทั้งประเทศเป็นหน่วยใหญ่ที่สุด จากนั้นแบ่งเป็น Sales Director (SD) ซึ่งเปรียบเหมือนกับภาคต่าง ๆ ของประเทศนั่นเอง รองลงมาอีกได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาขา ซึ่งเปรียบเทียบได้กับจังหวัดนั่นเอง เนื่องจากองค์กรมีสาขาอยู่ทุกจังหวัด แต่มีข้อยกเว้นในจังหวัดใหญ่ ๆ เช่นกรุงเทพมหานครจะมีหลายสาขาเนื่องจากลูกค้ามีปริมาณมาก แบ่งย่อยจากสาขาลงไป จะเป็น Route Manager ซึ่งเป็นหัวหน้าพนักงานขาย แบ่งย่อยลงไปอีกก็จะเป็นพนักงานขาย หรือเรียกว่า Sales Route

จากการวิเคราะห์ด้านลูกค้าดังกล่าว ทำให้ออกแบบตารางสำหรับเก็บข้อมูลลูกค้าได้ดังตารางที่ 3.2

ตารางที่ 3.2 Customer Dimension Table

Column Name	Description	Data Type	Length
CustomerCode	รหัสลูกค้า	int	8
CustomerName	ชื่อลูกค้า	varchar	30
AccountType	ประเภทของร้านค้าตาม Account	varchar	30
AccParent	Parent Account ของร้านค้า	varchar	30
Channel	ช่องทางการจัดจำหน่าย	varchar	30
SD	Sales Director	varchar	30
Location	สาขาที่ดูแลร้านค้า	varchar	30
RteMgr	Route Manager	varchar	30
SalesRte	Salesman	varchar	30

การวิเคราะห์ข้อมูลทางด้านเวลานั้น ออกแบบให้สามารถดูยอดขายรวมทั้งปีได้ และเมื่อต้องการดูรายละเอียดที่ย่อยลงมา ก็จะเป็นรายไตรมาส จากนั้นก็เป็นเดือน และเป็นสัปดาห์ ส่วนหน่อยย่อยที่เล็กที่สุดสามารถวิเคราะห์ได้ในระดับวัน แต่การดูข้อมูลระดับสัปดาห์นั้น ไม่ได้ดูตามปฏิทิน เนื่องจากจะมีการเหลื่อมกันกับข้อมูลในระดับเดือน จึงได้ทำการแบ่งวันในแต่ละเดือนออกเป็นสัปดาห์ โดยดูจากวันที่แทน ซึ่งยึดหลักคือ วันที่ 1-7 จะนับเป็นสัปดาห์ที่ 1 ของแต่ละเดือน และนับวันที่ 8-15 เป็นสัปดาห์ที่สอง นับวันที่ 16-22 เป็นสัปดาห์ที่สาม และวันที่ 23 เป็นต้นไปเป็นสัปดาห์ที่สี่ ดังนั้นเมื่อการวิเคราะห์ข้อมูลในด้านเวลาเป็นแบบนี้จึงไม่สามารถใช้ Time Dimension ที่เป็นมาตรฐานของระบบได้ จึงได้ทำการสร้างตาราง Time Dimension ไว้สำหรับเก็บข้อมูลด้านเวลาโดยตารางนี้สามารถถูกสร้างจาก Store Procedure ใน SQL ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 Time Dimension Table

Column Name	Description	Data Type	Length
TranDate	วันที่	smalldatetime	4
TranDateShow	ใช้สำหรับแสดงข้อมูลวันที่	varchar	10
TranYear	ปี	varchar	10
TranQuarter	ไตรมาสที่	varchar	10
TranMonth	เดือน	varchar	10
TranWeekOfMonth	สัปดาห์ที่ ในแต่ละเดือน	varchar	10

จากความต้องการในการวิเคราะห์ทั้งสามด้านที่กล่าวมาแล้วนั้น เป็นเพียงแนวทางในการวิเคราะห์เท่านั้น ซึ่งในการทำงานจริง ๆ จะต้องมีตัวชี้วัด หรือ Measurement และสิ่งที่ต้องการสำหรับวิเคราะห์ข้อมูลการขายสินค้าที่สำคัญที่สุดคือ ปริมาณสินค้าที่ขายได้ หรือ Quantity Sold นั่นเอง ส่วนตัวชี้วัดอีกอย่างที่ผู้ใช้ต้องการคือ จำนวนเงินของสินค้าที่ขายไป หรือ Revenue ด้วย แต่เนื่องจากองค์กรทำการขายสินค้าแบบขายส่ง ดังนั้นข้อมูลการขายสินค้าที่เกิดขึ้น จากระบบการขายแบบ OLTP จึงสามารถยอมให้ลูกค้าคืนสินค้าได้ และนอกจากนี้ ผู้ที่จะทำการวิเคราะห์ยังมีความต้องการตัวเลขของสินค้าที่คืนเหล่านี้ด้วย ทำให้การออกแบบตารางสำหรับเก็บข้อมูลที่เข้ามาแต่ละวัน ลงใน Data Warehouse ต้องแยกเก็บตัวเลขกัน ระหว่างปริมาณสินค้าที่ขายไปได้ และปริมาณสินค้าที่คืนกลับมา จึงทำให้ตารางที่เก็บข้อมูลรายละเอียดการขายสินค้า เป็นดังตารางที่ 3.4

ตารางที่ 3.4 Sales Fact Table

Column Name	Description	Data Type	Length
TranDate	วันที่	smalldatetime	4
ProductCode	รหัสสินค้า	int	4
CustomerCode	รหัสลูกค้า	int	8
DSQSLD	Quantity Sold	decimal	13.0
DSASLD	Amount Sold / Revenue	decimal	15.2
DSQRTN	Quantity Return	decimal	13.0
DSARTN	Amount Return	decimal	15.2

3.3 กระบวนการนำข้อมูลจาก OLTP เข้าสู่ Data Warehouse

ข้อมูลการขายสินค้าในแต่ละวัน จากระบบการขายสินค้าที่เป็น OLTP ทั้งที่เป็นข้อมูลการขายสินค้ารายวัน หรือแม้กระทั่งข้อมูลลูกค้า และสินค้าที่เป็น Master File สามารถนำเข้าสู่ Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Warehouse ที่เป็นฐานข้อมูล SQL Server ได้โดยผ่านกระบวนการที่เรียกว่า ETL ซึ่งระบบฐานข้อมูล SQL Server 2000 มีเครื่องมือในการทำ ETL ที่ชื่อว่า Data Transformation Services รวมอยู่ด้วย โดยสามารถสร้างการ Transform ข้อมูลในแต่ละส่วนแยกกันเรียกว่าเป็น Package เช่น การ Transform ข้อมูลจาก Customer Master เป็น Package หนึ่ง และการ Transform ข้อมูลรายละเอียดการขายสินค้าในแต่ละวัน ก็แยกเป็นอีก Package หนึ่งเป็นต้น และสิ่งสำคัญที่สุดที่ทำให้เครื่องมือนี้สามารถนำมาใช้งานในระดับ Implementation ได้นั่นคือ สามารถกำหนดให้แต่ละ Package ที่สร้างไว้ทำงานตามเวลาที่กำหนดได้ เนื่องจากการทำงานในกระบวนการเหล่านี้ต้องทำหลังจากระบบ OLTP ทำการปิดการขายในแต่ละวันเรียบร้อยแล้ว เพื่อให้ข้อมูลไม่เกิดการเปลี่ยนแปลงอีก ซึ่งมักจะเป็นเวลาหลังเวลาทำงานในแต่ละวัน

3.3.1 การ Transform ข้อมูลสินค้า

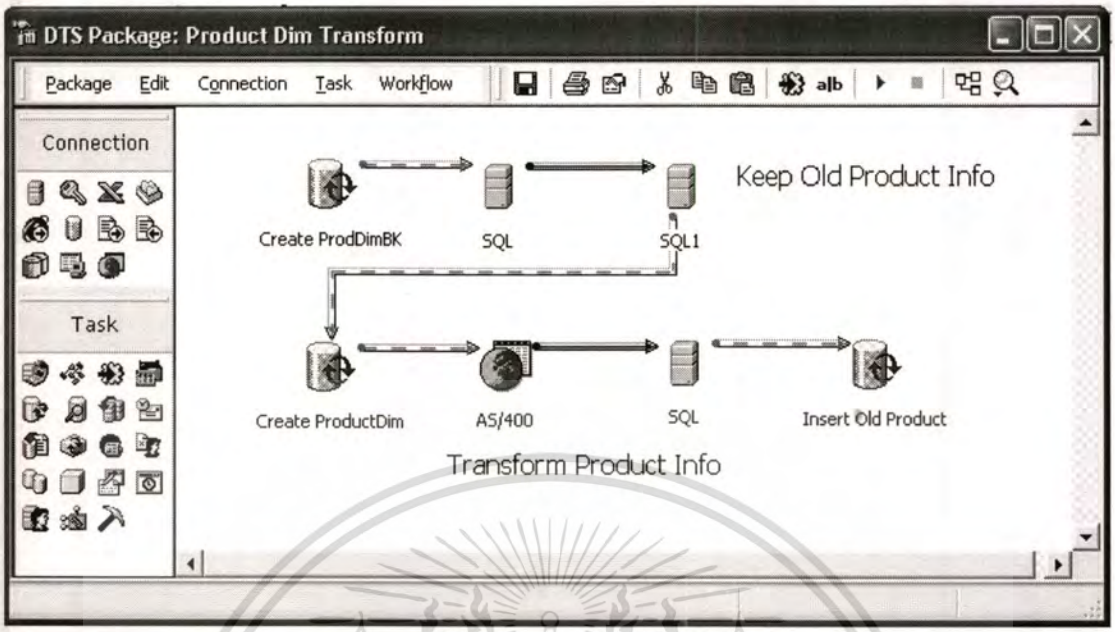
ในที่นี้จะไม่ขอกล่าวถึงรายละเอียดการเขียน DTS Package มากนัก จะกล่าวถึงเพียงคร่าว ๆ เท่านั้น ทั้งนี้เนื่องจากโครงงานนี้จะเน้นในด้านการใช้งาน OLAP Cube มาวิเคราะห์ข้อมูลมากกว่า ซึ่งในการเก็บข้อมูลลงใน Data Warehouse นั้น ข้อมูลรายละเอียดการขายสินค้า (Sales Transaction) แต่ละวันจะถูกนำเข้ามาเก็บไว้ทุกวันและเพิ่มปริมาณขึ้นเรื่อย ๆ ทำให้มีข้อมูลที่เป็นข้อมูลในอดีต (Historical Data) ดังนั้นในกระบวนการดึงข้อมูลอื่น ๆ ที่เกี่ยวข้อง จึงต้องสนใจข้อมูลที่เป็นอดีตเช่นกัน ไม่เช่นนั้นอาจทำให้ข้อมูลในอดีตเกิดการสูญหายไปได้ จึงออกแบบกระบวนการ Transform ข้อมูลที่เกี่ยวข้อง เช่น สินค้า หรือลูกค้า ให้สนใจข้อมูลในอดีตเช่นกัน

ขั้นตอนในการ Transform ข้อมูลสินค้า เพื่อสร้าง Product Dimension มีดังนี้

1. นำข้อมูลใน Product Dimension เก็บไว้เป็นชุด Backup
2. ลบข้อมูลสินค้าทั้งหมดใน Product Dimension
3. Transform ข้อมูลรายละเอียดสินค้าจาก OLTP มาใส่ Product Dimension
4. นำข้อมูลรายละเอียดสินค้าที่มีอยู่ในชุด Backup แต่ไม่มีในชุดปัจจุบันมาเพิ่มเข้าไปใน Production Dimension

ซึ่งวิธีการแบบนี้ทำให้สามารถเก็บข้อมูลสินค้าเอาไว้ได้ ถึงแม้ว่าหากมีการลบข้อมูลสินค้าชนิดนั้นออกไปจากฐานข้อมูล OLTP แล้วก็ตาม รูปแบบ DTS Package สำหรับการ Transform ข้อมูลสินค้าเพื่อทำ Product Dimension แสดงไว้ในภาพที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.1 DTS Package สำหรับ Product Dimension Transformation

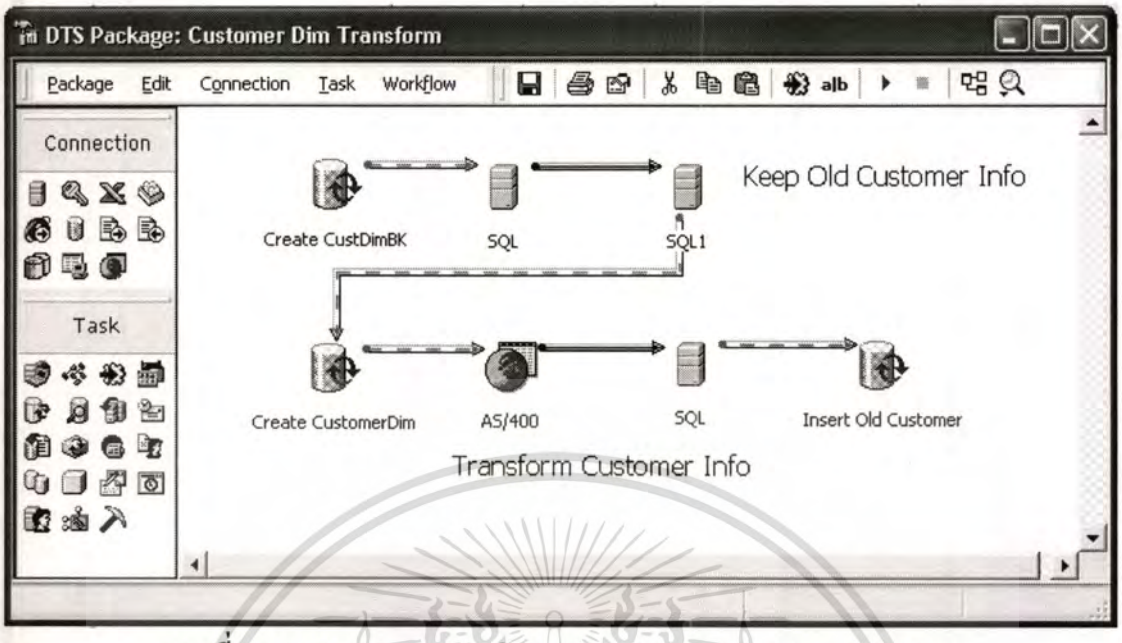
3.3.2 การ Transform ข้อมูลลูกค้า

ในทำนองเดียวกับกระบวนการ Transform ข้อมูลสินค้า การ Transform ข้อมูลลูกค้าก็มีขั้นตอนการ Transform เพื่อสร้าง Customer Dimension มีดังนี้

1. นำข้อมูลใน Customer Dimension เก็บไว้เป็นชุด Backup
2. ลบข้อมูลสินค้าทั้งหมดใน Customer Dimension
3. Transform ข้อมูลรายละเอียดสินค้าจาก OLTP มาใส่ Customer Dimension
4. นำข้อมูลรายละเอียดสินค้าที่มีอยู่ในชุด Backup แต่ไม่มีในชุดปัจจุบันมาเพิ่มเข้าไปใน Customer Dimension

รูปแบบ DTS Package สำหรับการ Transform ข้อมูลลูกค้าเพื่อทำ Customer Dimension แสดงไว้ในภาพที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.2 DTS Package สำหรับ Customer Dimension Transformation

3.3.3 การ Transform ข้อมูลรายละเอียดการขายสินค้า (Sales Transaction)

ข้อมูลรายละเอียดการขายสินค้า หรือ Sales Transaction ที่ได้จาก OLTP นั้น มีข้อมูลเข้ามาสู่ระบบทุกวัน แต่จะมีช่วงเวลาที่ข้อมูลเหล่านี้ไม่เคลื่อนไหว ได้แก่สิ้นวัน และในระบบ OLTP จะมีการทำการปิดสิ้นวันเพื่อนำข้อมูลไปทำงานในส่วนอื่น ๆ ต่อไป กระบวนการ Transform ข้อมูล Sales Transaction นี้ได้อาศัยช่วงเวลาที่ข้อมูลหยุดการเคลื่อนไหว ดึงข้อมูลรายละเอียดการขายสินค้าเข้าสู่ระบบ Data Warehouse โดยจะต้องเลือกข้อมูลที่เป็นวันที่ปัจจุบันเท่านั้น และทำการดึงข้อมูลเข้ามาจาก OLTP เพียงวันเดียว แล้วอาศัยการเก็บ Historical Data เอาไว้ใน Data Warehouse เอง

ขั้นตอนการ Transform ข้อมูลจาก Sales Transaction เพื่อทำเป็น Sales Fact มีดังนี้

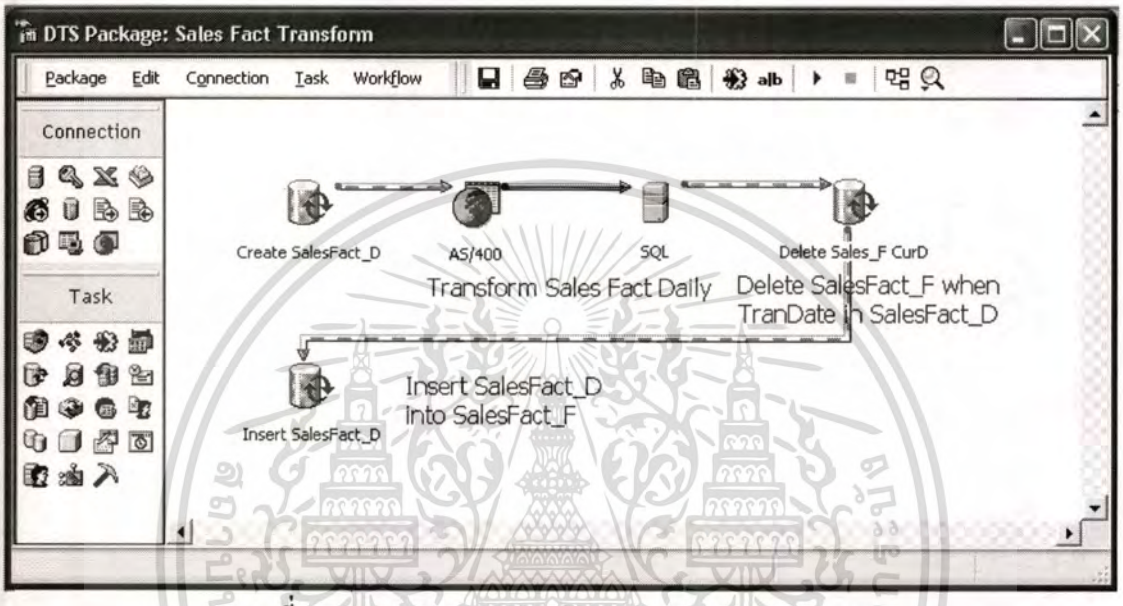
1. ดึงข้อมูลจาก OLTP เฉพาะวันปัจจุบันมาใส่ที่ SalesFact_D Table
2. หากพบข้อมูลใน SalesFact_F ที่มีวันที่ตรงกับ SalesFact_D ก็ทำการลบข้อมูลใน SalesFact_F
3. เพิ่มข้อมูลจาก SalesFact_D ลงใน SalesFact_F ทั้งหมด

โดยที่ SalesFact_F เป็นที่เก็บข้อมูลรายละเอียดการขายทั้งหมด รวมถึง Historical Data ด้วย ส่วน SalesFact_D จะมีข้อมูลรายละเอียดการขายเพียงวันปัจจุบันเท่านั้น การออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการ Transform แบบนี้ ทำให้สามารถ Rerun Package นี้ได้ใหม่ โดยไม่ต้องกังวลว่าข้อมูลใน SalesFact_F จะเกิดการซ้ำซ้อน

รูปแบบ DTS Package สำหรับการ Transform ข้อมูลรายละเอียดการขายสินค้าเพื่อทำ Sales Fact แสดงไว้ในภาพที่ 3.3



ภาพที่ 3.3 DTS Package สำหรับ Sales Fact Transformation

3.4 การสร้าง OLAP Cube สำหรับงานวิเคราะห์การขายสินค้า

เมื่อได้ข้อมูลที่ต้องการทั้งหมดมาแล้ว ขั้นตอนต่อไปก็คือการนำข้อมูลเหล่านั้นมาสร้าง OLAP Cube สำหรับงานวิเคราะห์ด้านการขายสินค้า ซึ่งจะเรียกสั้น ๆ ว่า Sales Cube เพื่อที่ผู้ใช้งานจะสามารถติดต่อกับ Cube และนำข้อมูลไปวิเคราะห์ได้ตามต้องการ

การสร้าง OLAP Cube โดยทั่ว ๆ ไปนั่นต้องประกอบด้วยการสร้าง Dimension ซึ่งสร้างได้จาก Dimension Table ที่ได้ออกแบบไว้ จากนั้นจึงทำการสร้าง Cube ซึ่งเปรียบได้กับการนำ Dimension หลาย ๆ Dimension มาประกอบเข้าด้วยกัน และเพิ่มตัวชี้วัด (Measurement) เข้ามาใน Cube

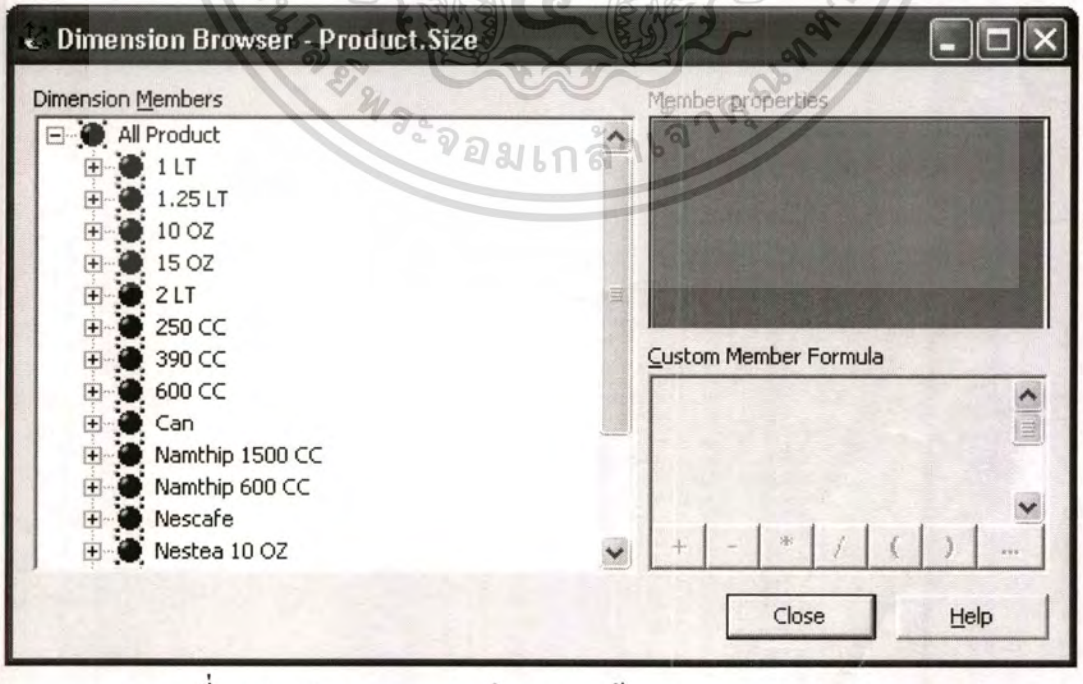
จาก Dimension Table ที่ได้ออกแบบไว้ และหลังจากผ่านการ Transform ข้อมูลทั้งหมดมาแล้ว สามารถนำมาสร้าง Dimension ได้ทันที โดยใช้เครื่องมือที่มีอยู่ใน SQL Server 2000 Analysis Service ที่ชื่อว่า Dimension Editor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 Product Dimension

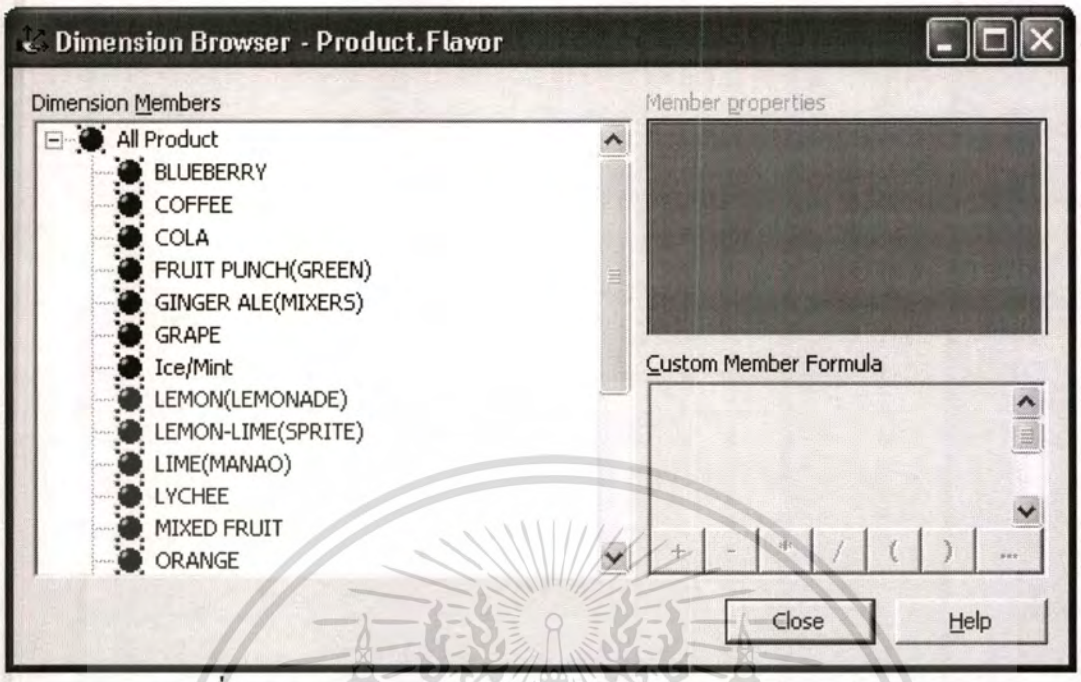
จากความต้องการของผู้ใช้งาน คือต้องการวิเคราะห์ข้อมูลด้านสินค้าที่สามารถแบ่งเป็น 3 รูปแบบ ได้แก่ แบ่งตาม Brand แบ่งตามขนาดบรรจุ และแบ่งตามรสชาติของสินค้า ดังนั้นจึงทำการสร้าง Product Dimension ให้มี 3 Hierarchy ตามมุมมองของผู้ใช้งาน ได้แก่ Brand Size และ Flovor ซึ่งรูปแสดง Product Dimension ที่สร้างเสร็จสมบูรณ์ แสดงไว้ดังภาพที่ 3.4 3.5 และ 3.6

ภาพที่ 3.4 Product Dimension ในมุมมองด้าน Brand (Brand Hierarchy)



ภาพที่ 3.5 Product Dimension ในมุมมองด้านขนาดบรรจุ (Size Hierarchy)

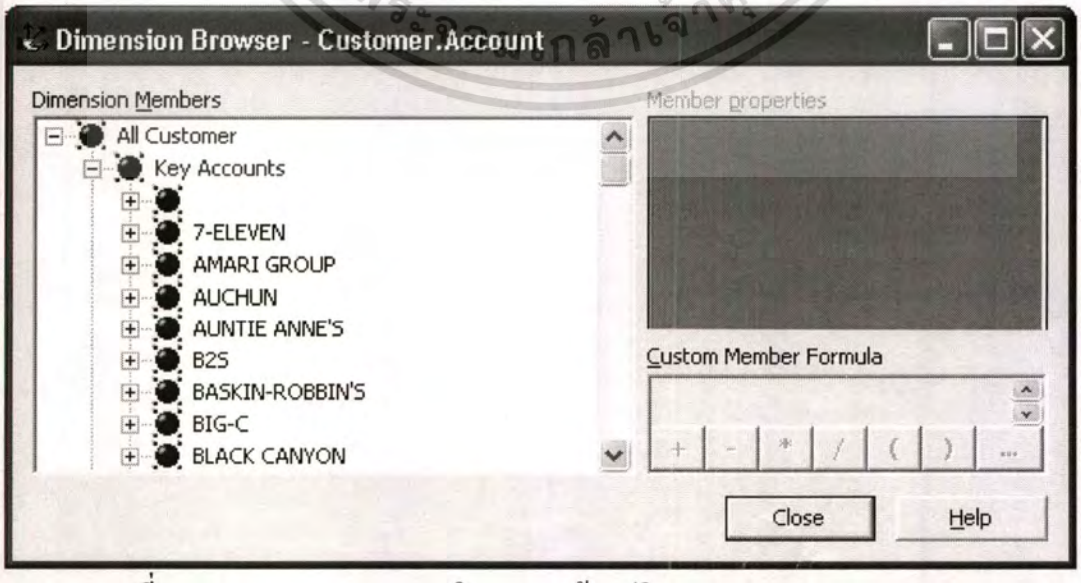
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.6 Product Dimension ในมุมมองด้านรสชาติ (Flavor Hierarchy)

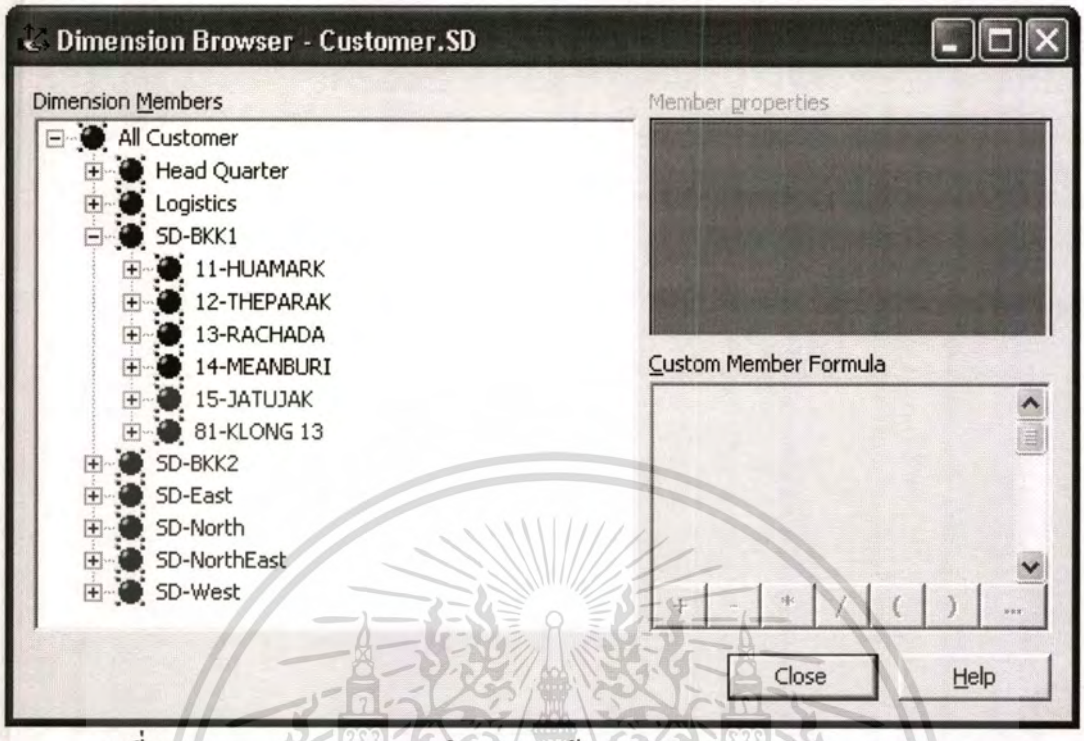
3.4.2 Customer Dimension

ทางด้าน Customer Dimension นั่นมีการแบ่งกลุ่มลูกค้าไว้เป็นสองแนวทาง คือ แบ่งตามปริมาณการขาย และแบ่งตามการทำงานของฝ่ายขาย ดังนั้น Customer Dimension จึงกำหนดให้มีสอง Hierarchy ได้แก่ Account และ Sales Director (SD) ซึ่งรูปแสดง Customer Dimension ที่สร้างเสร็จสมบูรณ์ แสดงไว้ดังภาพที่ 3.7 และ 3.8



ภาพที่ 3.7 Customer Dimension ในมุมมองด้านปริมาณการขาย (Account Hierarchy)

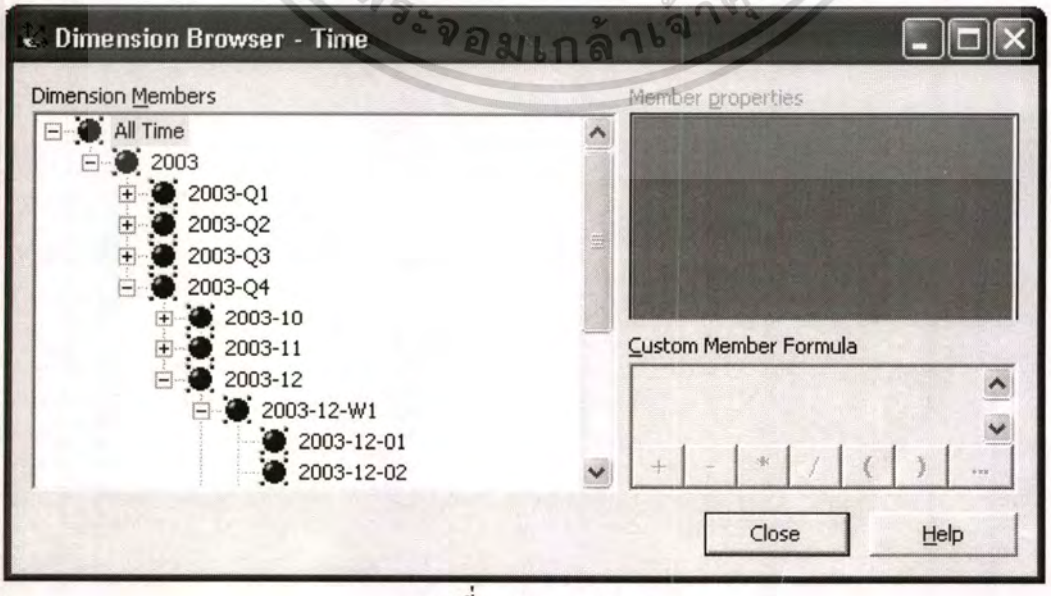
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.8 Customer Dimension ในมุมมองด้านการทำงานของฝ่ายขาย (SD Hierarchy)

3.4.3 Time Dimension

ทางด้าน Time Dimension สามารถดูข้อมูลรายปี รายไตรมาส รายเดือน รายสัปดาห์ และเล็กที่สุดสามารถดูข้อมูลเป็นรายวันได้ รูปแสดง Time Dimension ที่สร้างเสร็จสมบูรณ์ แสดงไว้ดังภาพที่ 3.9



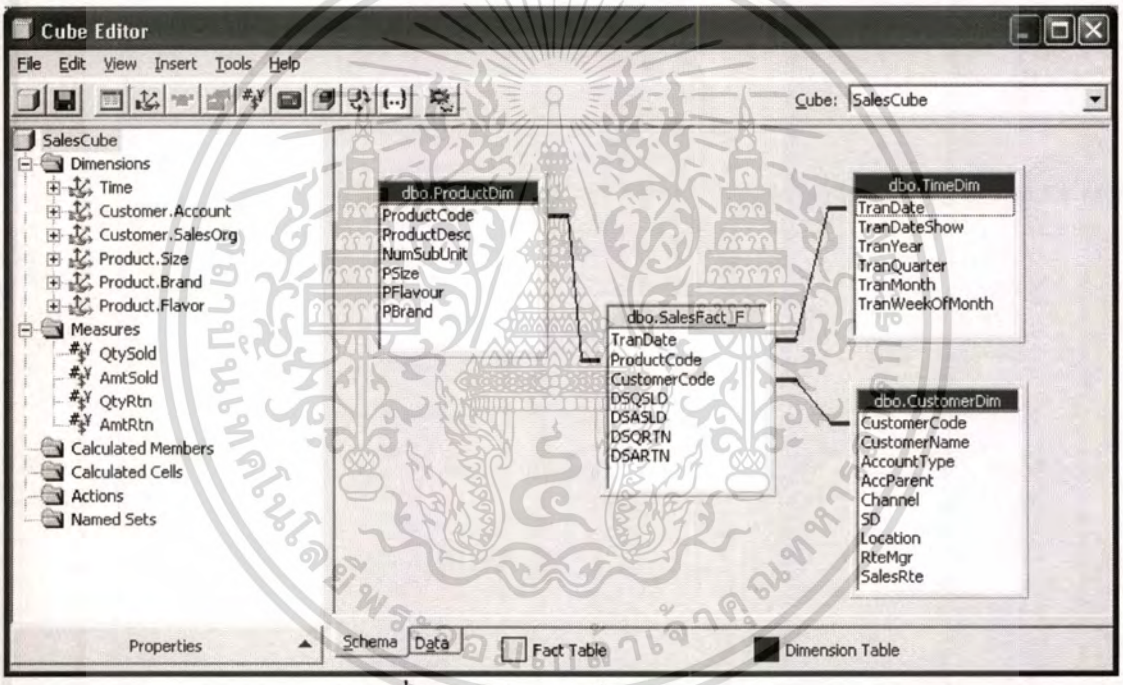
ภาพที่ 3.9 Time Dimension

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.4 Sales Cube ตัวชี้วัด หรือ Measurement

Sales Cube ที่สร้างขึ้นมานี้ กำหนดให้มี Measurement ที่มากับ Sales Fact Table สี่ตัว ได้แก่ ปริมาณสินค้าที่ขาย (Quantity Sold), จำนวนเงินของสินค้าที่ขาย (Amount Sold หรือ Revenue), ปริมาณสินค้าที่มีการคืนกลับเข้ามา (Quantity Return) และ จำนวนเงินของสินค้าที่คืนกลับมา (Amount Return)

จาก Dimension ทั้งสาม และ Measurement ทั้งสี่ทำให้ทำให้ Sales Cube ที่ถูกสร้างขึ้นมาเป็นดังภาพที่ 3.10 ซึ่งการสร้าง Sales Cube นี้ทำได้โดยใช้เครื่องมือที่ชื่อว่า Cube Editor ที่มากับ SQL Server 2000 Analysis Service ด้วยเช่นเดียวกับการสร้าง Dimension



ภาพที่ 3.10 รูปสมบูรณ์ของ Sales Cube

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

สถาปัตยกรรมและการพัฒนาโปรแกรม OLAP

ในบทนี้จะกล่าวถึงสถาปัตยกรรมที่ใช้สำหรับระบบโดยรวมทั้งหมด ไม่ว่าจะเป็นทางด้านฮาร์ดแวร์ ซอฟท์แวร์ โดยคำนึงถึงการใช้งานระบบร่วมกันจากผู้ใช้หลาย ๆ คน และผู้ใช้แต่ละคนมีสิทธิ์ในการเข้าถึงข้อมูลแตกต่างกัน นอกจากนี้ยังจะกล่าวถึงการพัฒนาโปรแกรมสำหรับวิเคราะห์ข้อมูลใน OLAP Cube การออกแบบส่วนติดต่อกับผู้ใช้ โดยเน้นความง่ายในการใช้งาน และสามารถทำการวิเคราะห์การขายได้อย่างมีประสิทธิภาพ

4.1 สถาปัตยกรรมด้านฮาร์ดแวร์

การใช้งานจริงของระบบวิเคราะห์การขายโดยใช้ OLAP นี้ จะมีเซิร์ฟเวอร์อยู่ที่ส่วนกลาง และผู้ใช้งานอาจจะอยู่ที่สาขาต่าง ๆ ขององค์กร โดยอาศัยเน็ตเวิร์คแบบ WAN ที่ได้มีการติดตั้งไว้แล้ว และอาจมีผู้ใช้บางส่วนอยู่ในวง LAN เดียวกันกับเซิร์ฟเวอร์ด้วย ส่วนสิทธิ์ในการเข้าถึงข้อมูลใน OLAP Cube ได้อาศัยการตรวจสอบสิทธิ์จาก Active Directory ของ Windows 2000 เครื่องคอมพิวเตอร์ในองค์กรเป็นแพลตฟอร์มของ Microsoft ทั้งหมด โดยผู้ที่จะใช้คอมพิวเตอร์ได้นั้นต้องทำการ Login เข้าสู่ Active Directory อยู่แล้ว

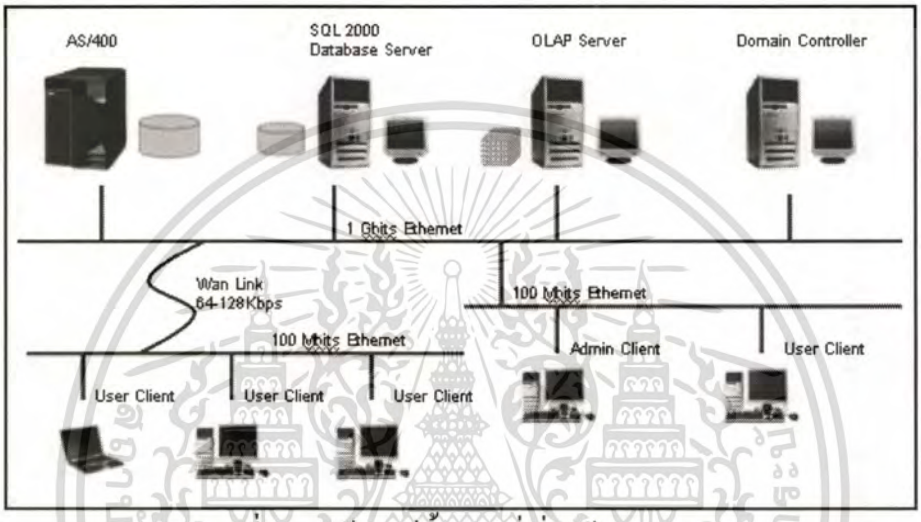
การกำหนดสถาปัตยกรรมทางด้านฮาร์ดแวร์นั้น ต้องคำนึงถึงส่วนที่เกี่ยวข้องกับระบบทั้งหมด ได้แก่ การได้มาของข้อมูล การนำข้อมูลมาเก็บไว้ใน Data Warehouse การสร้าง OLAP Cube จนกระทั่งการที่ผู้ใช้งานสามารถเข้ามาใช้งานข้อมูลที่อยู่ใน OLAP Cube ได้ ดังภาพที่ 4.1

ฮาร์ดแวร์ที่เกี่ยวข้องกับระบบทั้งหมด ประกอบด้วย

1. เครื่อง AS/400 ซึ่งมี DB2 เป็นฐานข้อมูล ที่ทำงานในระบบ OLTP ซึ่งเป็นต้นทางของข้อมูลที่นำการวิเคราะห์ หรือนำมาสร้างเป็น OLAP Cube
2. เครื่องเซิร์ฟเวอร์ที่ทำหน้าที่เป็น Data Warehouse Database Server สำหรับเก็บข้อมูลที่ผ่านกระบวนการทำ ETL มาแล้วจากเครื่อง AS/400
3. เครื่องเซิร์ฟเวอร์ที่ทำหน้าที่เป็น OLAP Server สำหรับเก็บ OLAP Cube

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เครื่องไคลเอนท์ สำหรับใช้กับโปรแกรม OLAP Client ที่สร้างขึ้น เพื่อทำหน้าที่เป็นฟรอนท์เอนด์ สำหรับผู้ใช้ทำการวิเคราะห์ข้อมูลจาก OLAP Cube
5. เครื่องเซิร์ฟเวอร์ที่ทำหน้าที่เป็น Domain Controller ซึ่งจะทำการตรวจสอบชื่อผู้ใช้งานและกำหนดสิทธิ์ต่าง ๆ ตามผู้ที่ใช้คนนั้นสามารถใช้ได้
6. อุปกรณ์เน็ตเวิร์คต่าง ๆ สำหรับการเชื่อมต่อทุกระบบเข้าด้วยกัน



ภาพที่ 4.1 ฮาร์ดแวร์ทั้งหมดที่เกี่ยวข้องกับระบบ

แต่อุปกรณ์ฮาร์ดแวร์ส่วนใหญ่เป็นสิ่งที่มีอยู่ภายในองค์กรอยู่แล้ว เช่นอุปกรณ์เน็ตเวิร์ค เครื่อง AS/400 หรือแม้กระทั่งเครื่องเซิร์ฟเวอร์ที่ทำหน้าที่เป็น Domain Controller แต่สิ่งที่ต้องการเพิ่มเติมสำหรับพัฒนาระบบวิเคราะห์ข้อมูลด้วย OLAP นี้ ได้แก่ เครื่องเซิร์ฟเวอร์ที่เป็น Data Warehouse Database Server และเครื่อง OLAP Server เท่านั้น

4.2 สถาปัตยกรรมด้านซอฟท์แวร์

การพัฒนาระบบวิเคราะห์ข้อมูลด้วย OLAP นี้ มีซอฟท์แวร์ที่เกี่ยวข้องซึ่งสามารถจำแนกเป็นหมวดหมู่ได้แก่ Operating System, Database Server, Middle Ware ต่าง ๆ รวมทั้งเครื่องมือในการพัฒนาโปรแกรม โดยแจกแจงรายละเอียดได้ดังนี้

1. Operating System

เครื่องคอมพิวเตอร์ที่เกี่ยวข้องทั้งหมดใช้ OS เป็น Windows 2000 ยกเว้น OLTP Database Server เครื่องเดียวเท่านั้นที่เป็น AS/400 ซึ่งเป็น OS ของ IBM โดยเครื่องที่ทำหน้าที่เป็นเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ Windows 2000 Server ส่วนเครื่องที่เป็นไคลเอนท์ใช้ Windows 2000 Professional หรืออาจเป็น Windows XP Professional ก็ได้

2. Database Server

ซอฟท์แวร์ที่ทำหน้าที่จัดการฐานข้อมูลสำหรับ Data Warehouse ใช้เป็น SQL Server 2000 ส่วน OLAP Server ที่ทำหน้าที่ให้บริการข้อมูลจาก OLAP Cube ใช้เป็น SQL 2000 Server Analysis Service ซึ่งทั้ง Database Server และ OLAP Server นั้นเป็นซอฟท์แวร์ที่อยู่ในชุดเดียวกัน

3. Middleware

เนื่องจากมีการส่งต่อข้อมูลระหว่างเซิร์ฟเวอร์หลายเครื่อง จึงต้องมีซอฟท์แวร์ที่เป็น Middleware เพื่อทำหน้าที่ให้ข้อมูลในแต่ละเซิร์ฟเวอร์ รวมถึงการไคลเอนท์สามารถติดต่อกันได้ Middleware ที่เกี่ยวข้องได้แก่

- ODBC for IBM DB2 สำหรับติดต่อกับฐานข้อมูล DB2 บนเครื่อง AS/400
- OLE DB for SQL Server สำหรับติดต่อกับฐานข้อมูล SQL ที่เป็น Data Warehouse
- OLE DB for OLAP Service สำหรับติดต่อกับ OLAP Server

4. Client Tools

ซอฟท์แวร์ที่ต้องใช้ในฝั่งไคลเอนท์ มีสองระบบ คือ ซอฟท์แวร์สำหรับผู้ดูแลระบบ และซอฟท์แวร์สำหรับผู้ใช้งานข้อมูลใน OLAP Server

- ซอฟท์แวร์สำหรับผู้ดูแลระบบ คือ SQL Server Enterprise Manager สำหรับจัดการฐานข้อมูล SQL Server และอีกตัวหนึ่งคือ Analysis Manager สำหรับจัดการ OLAP Server ซึ่งการสร้างหรือแก้ไข Dimension และ Cube จะต้องทำผ่านซอฟท์แวร์ตัวนี้เสมอ ซึ่งซอฟท์แวร์ทั้งสองนี้รวมอยู่ในชุด SQL 2000 Server ทั้งหมด

- ซอฟท์แวร์สำหรับผู้ใช้งานข้อมูลใน OLAP Server ส่วนนี้เป็นซอฟท์แวร์ที่จะนำข้อมูลที่อยู่ใน OLAP Cube ไปวิเคราะห์ข้อมูลการขายสินค้า ซึ่งจะเป็นโปรแกรมที่ผู้จัดทำโครงงานนี้พัฒนาขึ้นมาเพื่อให้สามารถปรับเปลี่ยนความสามารถให้ตรงกับความต้องการของผู้ใช้งานที่สุด

5. เครื่องมือที่ใช้ในการพัฒนาโปรแกรม OLAP

ซอฟท์แวร์ที่ใช้เป็นเครื่องมือในการพัฒนาโปรแกรมเพื่อดึงข้อมูลจาก OLAP Cube เพื่อให้ผู้ใช้สามารถวิเคราะห์ข้อมูลได้ตรงตามความต้องการได้แก่ Microsoft Visual Basic 6 เนื่องจากมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความคล่องตัวในการใช้งาน นอกจากนี้ยังมี Component ต่าง ๆ สำหรับการเชื่อมต่อกับฐานข้อมูลทั้ง SQL Server หรือแม้กระทั่ง OLAP Server ซึ่งสามารถนำมาใช้งานได้อย่างมีประสิทธิภาพ

4.3 การออกแบบโปรแกรมสำหรับวิเคราะห์ข้อมูลใน OLAP Cube

โปรแกรมที่ทำการพัฒนาขึ้นมานี้สามารถนำไปใช้กับ OLAP Cube ที่อยู่ใน SQL Server Analysis Service ได้ทุกระบบ แต่ในโครงงานนี้จะเน้นการใช้งานกับ Sales Cube ที่ได้สร้างขึ้นมาจากบทที่ผ่านมา เพื่อใช้กับงานวิเคราะห์ด้านการขายสินค้าขององค์กรเท่านั้น

4.3.1 หน้าจอหลักสำหรับดูข้อมูล

ในการออกแบบหน้าจอหลักสำหรับดูข้อมูลนั้น ได้ออกแบบให้เป็นตาราง เพราะผู้ใช้งานส่วนใหญ่จะมีความเคยชินกับการใช้งาน Spread Sheet โดยมีการแสดงเป็นแถวและคอลัมน์ ซึ่งข้อมูลด้านแถว และคอลัมน์ดังกล่าว คือ Dimension ที่อยู่ใน Cube ซึ่งสามารถเลือก Dimension ใดมาแสดงก็ได้ นอกจากนี้ Dimension อื่น ๆ ที่อยู่นอกเหนือจากการเลือกมาแสดงด้านแถว และคอลัมน์ จะสามารถนำมาใส่ไว้ในระดับ Page และสามารถนำมาทำเป็น Filter ข้อมูลได้ด้วย

นอกจากนี้ข้อมูล Dimension ที่แสดงอยู่ในด้านแถว สามารถทำการ Drill Down เพื่อดูข้อมูลที่ Level ถัดลงไปได้ด้วย การ Drill Down นี้จัดเป็นมาตรฐานของฟรอนท์เอนด์ สำหรับ OLAP Cube เกือบทุกระบบ ดังนั้นการพัฒนาโปรแกรมนี้จึงต้องมีความสามารถในการ Drill Down อยู่ด้วย

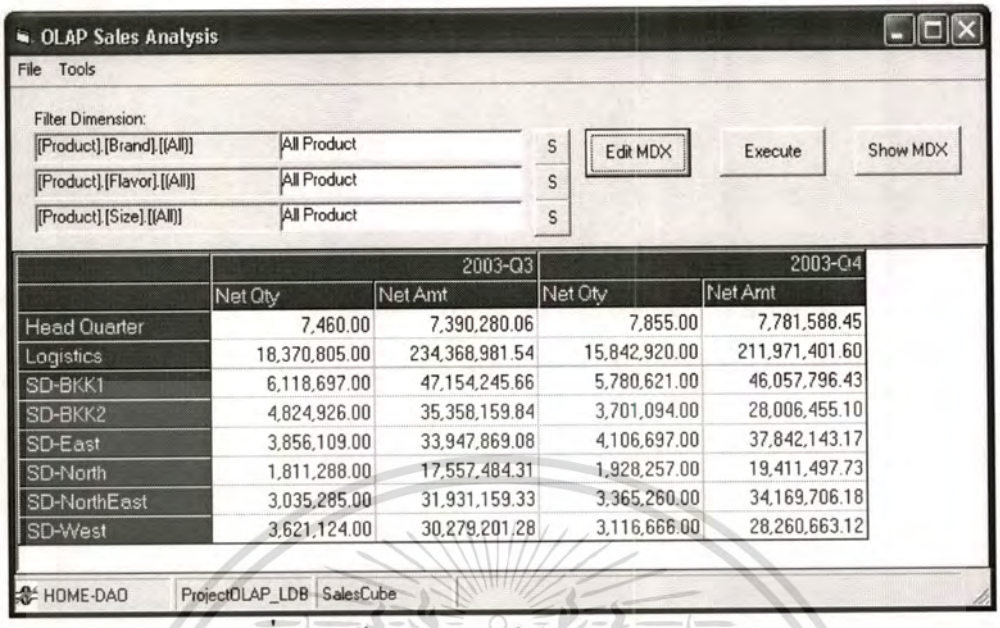
ส่วนข้อมูลที่แสดงอยู่ระหว่างแถว ตัดกับคอลัมน์นั้น คือตัวชีวัด หรือ Measurement ที่ได้สร้างไว้ใน Cube นั่นเอง โดยโปรแกรมสามารถเลือกได้ว่าจะนำ Measurement ใดมาแสดง หรือไม่แสดงก็ได้

การออกแบบหน้าจอหลัก จะเน้นพื้นที่แสดงข้อมูลที่เป็นตารางให้เป็นพื้นที่ส่วนใหญ่ เพื่อความสะดวกในการดูตัวเลขที่ปรากฏ ซึ่งพื้นที่ตารางดังกล่าวสามารถขยายขนาดตามความกว้างของหน้าต่างของโปรแกรมได้ด้วย

ส่วนบนของหน้าจอหลักของโปรแกรม นอกจากจะแสดง Dimension ที่ต้องการ Filter ข้อมูลแล้ว ยังมีปุ่มอีกสองปุ่ม โดยปุ่มแรกทำหน้าที่ Execute คำสั่ง MDX เพื่อนำข้อมูลมาแสดงในตาราง ส่วนปุ่มที่สองเป็นปุ่มที่ใช้สำหรับแสดงคำสั่ง MDX ที่โปรแกรมได้สร้างขึ้นมา

เมนูของโปรแกรมก็ออกแบบให้เป็นมาตรฐาน คล้ายกับโปรแกรมทั่วไป ซึ่งโปรแกรมนี้สามารถ Save คำสั่ง MDX เก็บไว้ใช้งานภายหลังได้ ซึ่งรูปหน้าจอหลักของโปรแกรม แสดงไว้ในภาพที่ 4.2

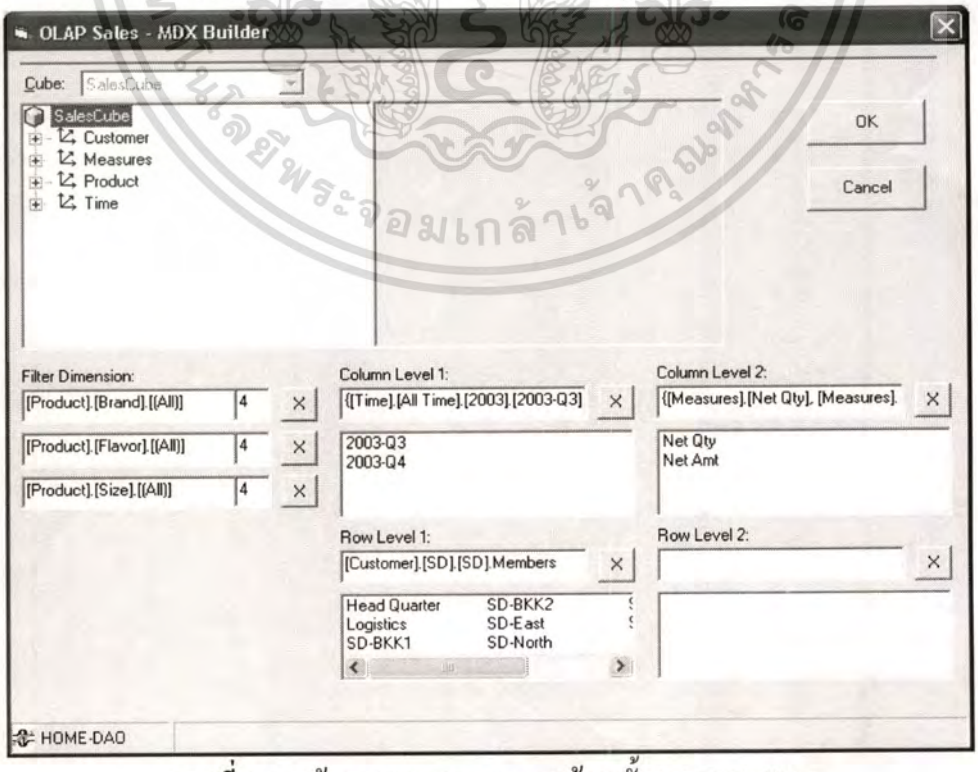
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.2 หน้าจอหลักของโปรแกรม Sales Analysis

4.3.2 หน้าจอ Cube Browser สำหรับดูโครงสร้างของ Cube และเลือกข้อมูล

หน้าจอสำหรับแสดงโครงสร้างของ Cube และทำการเลือกข้อมูลจาก Dimension ต่าง ๆ มาทำการสร้างรายงาน เพื่อแสดงผลข้อมูลตามที่ผู้ใช้ต้องการ ดังภาพที่ 4.3



ภาพที่ 4.3 หน้าจอ Cube Browser พร้อมทั้ง MDX Builder

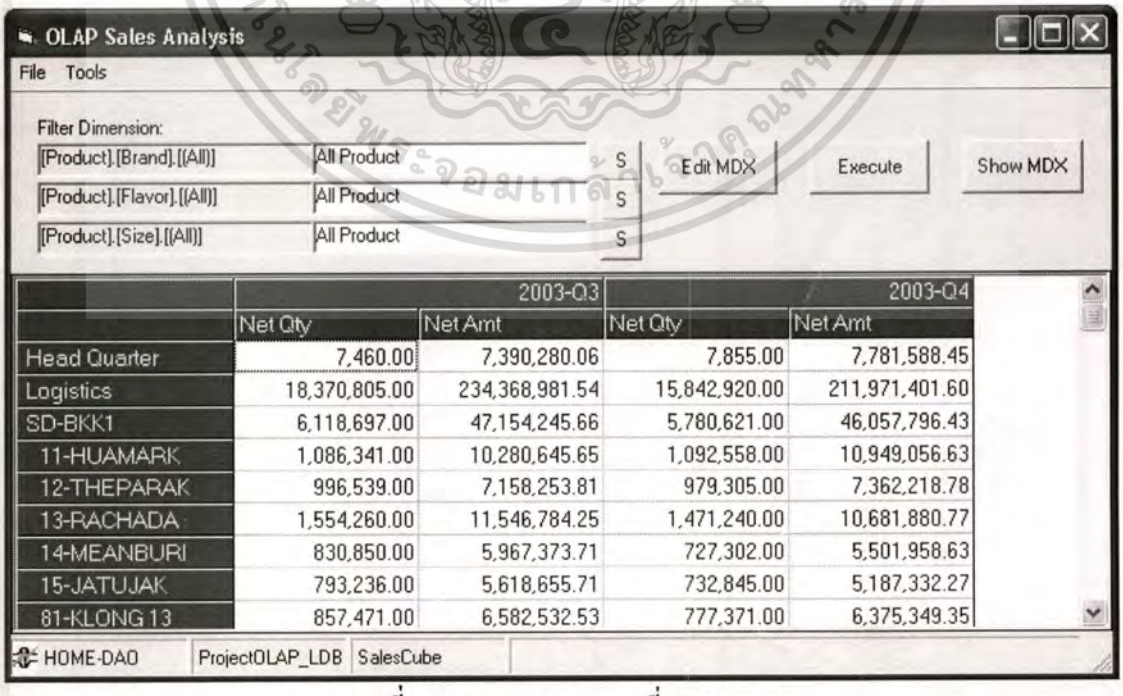
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 การทำงาน และความสามารถต่าง ๆ ของโปรแกรม

การออกแบบการทำงานของโปรแกรมสำหรับวิเคราะห์ข้อมูลการขายสินค้า หรือที่เรียกว่า OLAP Sales Analysis นี้ จะมีความสามารถในการทำงานหลัก ๆ คือการ Generate คำสั่ง MDX ขึ้นมา แล้วส่งคำสั่งไปขอข้อมูลจาก OLAP Server ซึ่งคำสั่ง MDX นี้จะถูกสร้างโดยอัตโนมัติโดยที่ผู้ใช้งานไม่จำเป็นต้องทราบโครงสร้างของคำสั่งแต่อย่างใด ผู้ใช้เพียงแต่กำหนดว่าต้องการดูข้อมูลอะไรในแกนใดของตาราง และต้องการ Filter ที่ Dimension ใดบ้างเท่านั้น

ความสามารถต่าง ๆ ที่โปรแกรมทำได้ ได้แก่

1. สามารถเชื่อมต่อกับ OLAP Server ที่เป็น SQL 2000 Server Analysis Service เพื่อทำงานกับ OLAP Cube เช่นการดึงรายชื่อ Dimension, Hierarchy รวมถึง Measurement ที่อยู่ภายใน Cube ได้
2. สามารถ Generate คำสั่ง MDX เพื่อดึงข้อมูลที่อยู่ภายใน OLAP Cube มาแสดงในตาราง ตามความต้องการของผู้ใช้งานได้ และคำสั่ง MDX ที่ถูกสร้างขึ้นมานี้ สามารถ Save เก็บไว้เพื่อนำมาใช้งานภายหลังได้อีก
3. มีความสามารถในการ Drill Down ข้อมูล Dimension ที่อยู่ในแถวของตาราง เพื่อการวิเคราะห์ หรือดูข้อมูลแบบเจาะจง ซึ่งรูปการ Drill Down แสดงไว้ในภาพที่ 4.3



ภาพที่ 4.4 การ Drill Down ที่ SD-BKK1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในภาพที่ 4.3 นี้แสดงการ Drill Down ที่ SD-BKK1 ซึ่งเป็น Member ตัวหนึ่งใน Level ชื่อ SD ที่อยู่ภายใน Dimension ชื่อ Customer.SD ซึ่งการ Drill Down นี้จะแสดง Level ที่อยู่ถัดลงไปจาก SD นั่นคือ Level ชื่อ Location ตามที่ได้ออกแบบ Cube เอาไว้ในบทก่อน ซึ่ง Level ชื่อ Location นี้มี Member 8 ตัว ที่อยู่ภายใต้ SD-BKK1

4. สามารถเลือกการจัดเรียงลำดับข้อมูลในตารางได้ โดยอาจเลือกให้เรียงตาม Measurement หรือ ให้เรียงลำดับตามรายชื่อ Member ที่ปรากฏด้านซ้ายมือสุดก็ได้ ในภาพที่ 4.3 เป็นการเรียงลำดับตามรายชื่อ Member ด้านซ้ายมือ
5. มีความสามารถนำข้อมูลในตารางทั้งหมด ส่งออกไปยังโปรแกรม Excel เพื่อให้ผู้ใช้สามารถนำตัวเลขไปทำการวิเคราะห์ต่อ หรือสร้างเป็นกราฟเพิ่มเติมได้

นอกจากความสามารถหลัก ๆ ของโปรแกรมดังกล่าวมาข้างต้นนี้ ยังมีความสามารถอีกหลาย ๆ ด้าน เช่นการเลือกให้แสดงเจาะจงเฉพาะข้อมูลที่ต้องการ การรวมข้อมูลเข้าด้วยกัน การสร้าง Member ใหม่จาก Member เก่า หรือที่เรียกว่า Calculated Member เป็นต้น และหากผู้ใช้งานต้องการให้โปรแกรมมีความสามารถเพิ่มเติม ก็สามารถพัฒนาเพิ่มได้อย่างไม่ยากนัก หากมีคำสั่ง MDX รองรับการทำงานดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การนำโปรแกรมไปใช้งาน และบำรุงรักษาระบบ

ในบทนี้จะกล่าวถึงการนำโปรแกรมวิเคราะห์ข้อมูลการขายสินค้าที่พัฒนาขึ้นมา ไปให้กับผู้ใช้งานได้ใช้เพื่อวิเคราะห์การขายสินค้าขององค์กร ซึ่งผู้ใช้งานส่วนใหญ่จะอยู่ตามสาขา แต่มีการเชื่อมต่อแบบเครือข่ายกับสำนักงานใหญ่ ซึ่งผู้ใช้งานสามารถใช้งานโปรแกรมเพื่อดูข้อมูลได้โดยผ่านทางเครือข่ายดังกล่าว เนื้อหาส่วนที่สองจะกล่าวถึงการบำรุงรักษาระบบข้อมูลให้ถูกต้อง การจัดตารางเวลาดึงข้อมูลเข้าสู่ Cube เพื่อให้ผู้ใช้สามารถเปิดใช้งานข้อมูลได้ในชั่วโมงทำงาน และส่วนสุดท้ายจะกล่าวถึงการจัดการในเรื่องสิทธิ์ต่าง ๆ ที่ผู้ใช้แต่ละคนได้รับในการเข้าถึงข้อมูลที่แตกต่างกัน

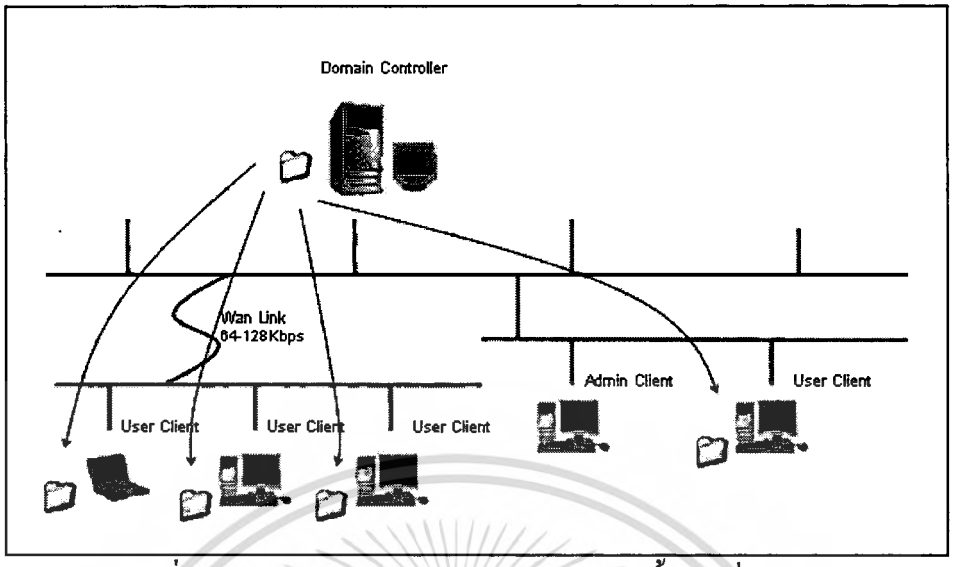
5.1 การติดตั้งโปรแกรมให้กับผู้ใช้งาน

โปรแกรมวิเคราะห์ข้อมูลการขายสินค้านี้ถูกพัฒนาจาก Visual Basic ซึ่งนอกจากเป็นเครื่องมือที่จะมีความคล่องตัวในการพัฒนาแล้ว ยังมีความคล่องตัวในการ Deploy โปรแกรมด้วย โดย Visual Basic นี้จะมีเครื่องมือในการรวบรวมโปรแกรมที่เขียนขึ้น และคอมโพเน้นท์ต่าง ๆ ที่ต้องใช้ในโปรแกรม ทั้งหมดเข้าด้วยกัน แล้วสร้างเป็น Package หากต้องการติดตั้งโปรแกรมที่พัฒนาขึ้นมานี้ที่เครื่องใดก็สามารถนำ Package นี้มาติดตั้งได้ทันที

แต่เนื่องจากซอฟท์แวร์ที่เป็น Operating System ของเครื่องผู้ใช้ทั้งหมดเป็น Windows 2000 ทำให้มีความสะดวกในการส่ง Package ของโปรแกรมที่พัฒนาขึ้นมานี้ไปติดตั้งยังเครื่องของผู้ใช้ได้ โดยอาศัยความสามารถของ Active Directory ที่อยู่ใน Windows 2000 Server โดยให้เครื่องที่ทำให้หน้าที่เป็น Domain Controller เป็นผู้ส่ง Package ไปติดตั้งยังเครื่องที่ต้องการใช้โปรแกรมโดยอัตโนมัติ ซึ่งผู้ใช้ไม่จำเป็นต้องติดตั้งโปรแกรมด้วยตนเอง ดังภาพที่ 5.1

ข้อดีของการส่ง Package ไปติดตั้งโดยอัตโนมัตินี้คือการที่ผู้ดูแลระบบคอมพิวเตอร์ไม่ต้องเดินทางไปติดตั้งโปรแกรมให้กับผู้ใช้เอง และหากกรณีที่โปรแกรมมีการแก้ไข หรือมีการ Upgrade ก็สามารถส่งไปติดตั้งให้กับผู้ใช้ได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.1 การส่ง Package ของโปรแกรมไปติดตั้งยังเครื่องผู้ใช้

เมื่อทำการติดตั้งโปรแกรมลงที่เครื่องคอมพิวเตอร์ของผู้ใช้งานแล้ว ผู้ใช้ก็สามารถเรียกโปรแกรมเพื่อเข้าดูข้อมูลได้ทันที ซึ่งในส่วนนี้จะขึ้นอยู่กับสิทธิ์ของผู้ใช้คนนั้น ๆ ว่ามีสิทธิ์เข้าดูข้อมูลที่อยู่ใน OLAP Cube ได้เพียงใด

โปรแกรมสำหรับการวิเคราะห์ข้อมูลด้วย OLAP ถึงแม้ว่าจะมีความสะดวกในการเรียกดูข้อมูลต่าง ๆ ก็ตาม แต่มักเกิดปัญหากับผู้ใช้งานเนื่องจากผู้ใช้ส่วนใหญ่ไม่ชินกับการดูข้อมูลในลักษณะหลายมิติ ดังนั้นเพื่อให้การใช้งานโปรแกรมให้เกิดประโยชน์สูงสุดสำหรับองค์กร จึงต้องมีการเทรนนิ่งให้กับผู้ที่จะใช้ข้อมูลด้วย

5.2 การบำรุงรักษาระบบข้อมูล

ข้อมูลที่ถูกเก็บอยู่ใน Data Warehouse สำหรับเตรียมนำไปสร้างเป็น OLAP Cube นั้นเป็นข้อมูลที่ได้ผ่านกระบวนการ Transform และ Cleaning มาแล้ว ดังนั้นข้อมูลส่วนใหญ่จะเชื่อถือได้และมักไม่ค่อยเกิดความผิดพลาดมากนัก แต่ทั้งนี้ต้องขึ้นอยู่กับกระบวนการในการ Transform ข้อมูลด้วย โดยต้องมีการวางแผนในการทำงานเป็นอย่างดี โดยต้องทราบว่าระบบ OLTP ที่ทำงานกับข้อมูลการขายประจำวันมีการจบการทำงานเมื่อใด ซึ่งในการทำงานจริง เวลาดังกล่าวมักจะเปลี่ยนแปลงได้เสมอ ไม่แน่นอน ดังนั้น การที่จะเริ่มต้องกระบวนการนำข้อมูลจาก OLTP เข้าสู่ระบบ Data Warehouse จะต้องแน่ใจเสมอว่าระบบการขายสินค้าประจำวันได้หยุดการทำงานแล้ว ข้อมูลการขายไม่มีการเปลี่ยนแปลงอีกแล้วนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ต้องคำนึงถึงเวลาที่ใช้ในการทำงานของ OLAP Cube ด้วย เนื่องจากในขั้นตอนการนำข้อมูลที่อยู่ใน Data Warehouse มาสร้างเป็น OLAP Cube นั้นใช้เวลาค่อนข้างมากเนื่องจากมีการคำนวณสูง เพราะเป็นการคำนวณผลบวกหลาย ๆ ตัวเก็บเอาไว้ ทำให้ความจำเป็นในการเริ่มต้นกระบวนการนำข้อมูลเข้าสู่ Data Warehouse จนกระทั่งได้ OLAP Cube ที่สมบูรณ์ ถูกจำกัดในกรอบเวลา ซึ่ง Cube ต้องเสร็จพร้อมให้ผู้ใช้งานสามารถใช้งานได้ทันทีที่เข้ามาถึงออฟฟิศในตอนเช้า

แต่ถึงแม้ว่าจะมีการวางแผนด้านเวลาในการนำข้อมูลเข้าระบบเป็นอย่างดีก็ตาม บางครั้งก็อาจเกิดปัญหาที่ไม่คาดคิดได้ ทำให้อาจต้องมีการนำข้อมูลเข้าระบบใหม่ และข้อมูลที่เก็บใน Data Warehouse มีข้อมูลเก่าอยู่ด้วยเสมอ ดังนั้นหากเกิดปัญหาในการดึงข้อมูล และมีการดึงข้อมูลใหม่อีกครั้ง ถ้าหากออกแบบ Package ที่ใช้ในการ Transform ไม่ดีจะทำให้ข้อมูลเกิดความซ้ำซ้อนได้ ซึ่งในโครงงานนี้ได้มีการออกแบบ Package สำหรับ Transform ให้สามารถทำงานซ้ำได้ โดยข้อมูลจะไม่เกิดความซ้ำซ้อน เพราะมีการตรวจสอบข้อมูลที่เข้ามาใหม่กับข้อมูลที่มีอยู่ ซึ่งในกรณีข้อมูลการขาย ก็จะตรวจสอบจากวันที่ เป็นต้น

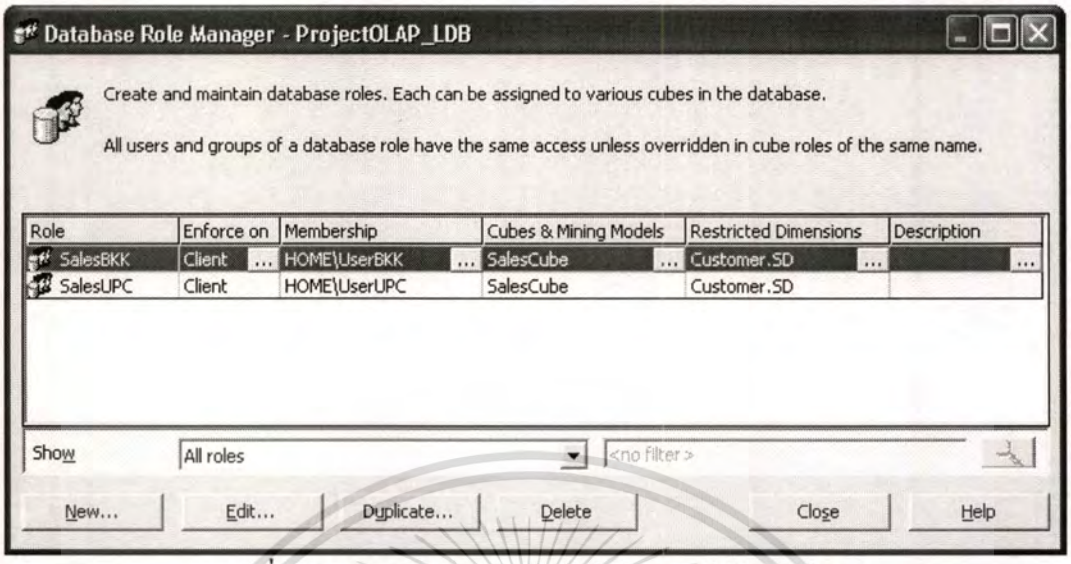
5.3 การบำรุงรักษาสิทธิ์ของผู้ใช้ข้อมูล

ข้อมูลที่ถูกนำเข้าไปเก็บไว้ใน OLAP Cube นี้ จะเป็นข้อมูลรวมทั้งหมดขององค์กร ซึ่งแน่นอนว่าผู้ใช้ทุกคนไม่มีสิทธ์ในการเข้าดูข้อมูลทั้งหมดได้ จะมีเพียงบางคนเท่านั้นที่สามารถเข้าดูข้อมูลของทั้งองค์กรได้ และผู้ใช้ส่วนใหญ่จะดูข้อมูลได้เพียงบางส่วนเท่านั้น ซึ่งก็จะเป็นในส่วนที่ตนเองรับผิดชอบอยู่นั่นเอง

ในการกำหนดสิทธิ์การเข้าถึงข้อมูลใน Cube นี้ มีเครื่องมือสำหรับ Admin ก็คือ SQL Analysis Manager ซึ่งเป็นเครื่องมือเดียวกับที่ใช้ในการสร้าง Dimension และสร้าง Cube นั่นเอง

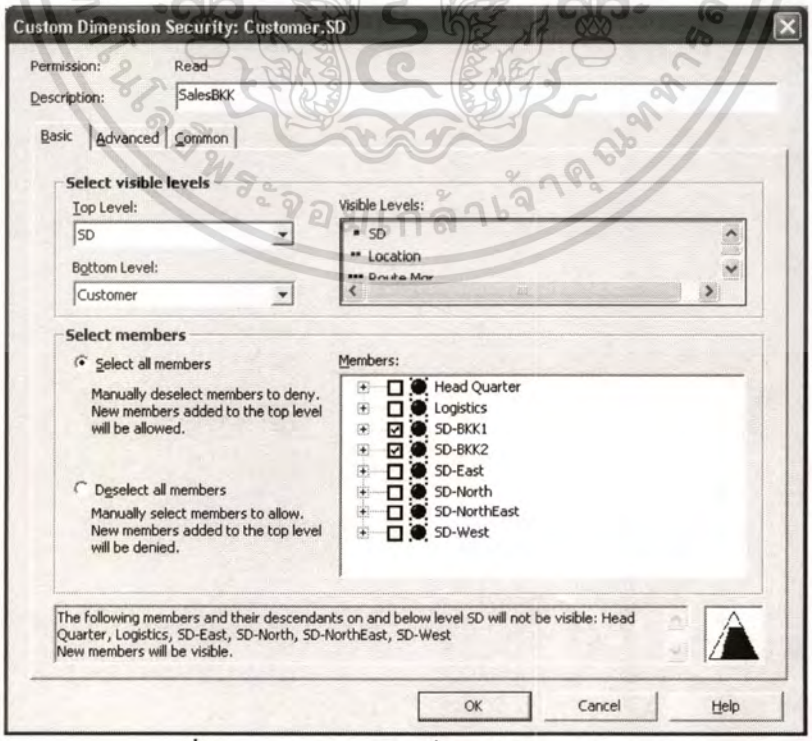
และข้อดีอีกอย่างของ SQL Server Analysis Service (OLAP Server) ก็คือ ความสามารถในการกำหนดสิทธิ์ผู้ใช้ โดยอาศัยรายชื่อผู้ใช้จาก Active Directory ของ Windows 2000 ได้ นั่นคือระบบ OLAP Server ไม่ต้องสร้างรายชื่อผู้ใช้ขึ้นใหม่แต่อย่างใด เพราะตามปกติแล้วผู้ใช้งานคอมพิวเตอร์จะต้องมีรายชื่ออยู่ใน Active Directory อยู่แล้ว ใน OLAP Server เพียงแต่สร้างกลุ่มขึ้นมา จากนั้นกำหนดสิทธิ์การเข้าถึงข้อมูลเป็นกลุ่ม เพื่อที่จะง่ายในการเปลี่ยนแปลงสิทธิ์ และก็นำรายชื่อผู้ใช้จาก Domain เข้ามาใส่ในกลุ่มตามสิทธิ์ที่ผู้นั้นควรจะได้รับเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.2 Role สำหรับการเข้าถึงข้อมูลของผู้ใช้กลุ่มต่าง ๆ

การกำหนดสิทธิ์ในการเข้าถึงข้อมูลของกลุ่มต่าง ๆ นั้นสามารถระบุเป็น Dimension ได้ โดยบอกว่าผู้ใช้กลุ่มนี้ มองเห็น Member ใดได้บ้างใน Dimension หนึ่ง ๆ ดังเช่นภาพ 5.3 เป็นการกำหนดให้ผู้ใช้ที่อยู่ในกลุ่ม BKK มองเห็นข้อมูล SD-BKK1 และ SD-BKK2 ใน Dimension ชื่อ Customer.SD เท่านั้น ดังนั้นผู้ใช้ในกลุ่มนี้จะไม่สามารถมองเห็นข้อมูลที่อยู่ใน SD อื่น ๆ ได้



ภาพที่ 5.3 การกำหนดสิทธิ์ของผู้ใช้ในกลุ่มต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปผลการศึกษา และข้อเสนอแนะ

โครงการพัฒนาระบบงานสำหรับวิเคราะห์ข้อมูลการขายสินค้าด้วย OLAP นี้ จัดทำขึ้นเพื่อให้สามารถนำข้อมูลที่เป็นข้อมูลเชิงสัมพันธ์มาสร้างเป็นข้อมูลหลายมิติ สำหรับการใช้งานด้านการวิเคราะห์ข้อมูล เพื่อนำข้อมูลที่ได้ไปเพิ่มประสิทธิภาพการทำงานได้

6.1 สรุปผลการดำเนินงาน

ฐานข้อมูลแบบหลายมิติ หรือ OLAP Cube มีประโยชน์อย่างมากในการนำมาใช้วิเคราะห์ข้อมูลต่าง ๆ ได้หลากหลายรูปแบบ ขึ้นอยู่กับการออกแบบ Dimension และ Measurement และรูปแบบการ Drill Down ในแต่ละ Dimension แต่ในโครงงานนี้ได้นำมาใช้กับข้อมูลด้านการขายสินค้า เนื่องจากมีรูปแบบที่เข้าใจได้ง่าย และเป็นพื้นฐานในการนำไปใช้กับระบบที่มีความซับซ้อนเพิ่มมากขึ้นต่อไป

โครงงานนี้ได้นำฐานข้อมูลแบบหลายมิติที่ชื่อว่า SQL Server 2000 Analysis Service มาใช้ในการพัฒนาระบบทั้งหมด เนื่องจากสามารถทำความเข้าใจได้ง่าย มีเครื่องมือที่ใช้ในการนำข้อมูลเข้าระบบที่มีประสิทธิภาพ และยังมีคำสั่ง MDX ที่ทำให้สามารถพัฒนาโปรแกรมสำหรับวิเคราะห์ข้อมูลจาก OLAP Cube ขึ้นมาใช้เองได้ด้วย

ในส่วนการพัฒนาโปรแกรมนั้น ได้เลือกใช้เครื่องมือคือ Visual Basic เนื่องจากผู้พัฒนามีความคล่องตัวในการใช้งานอยู่แล้ว ทำให้ไม่เสียเวลาในการเรียนรู้การใช้เครื่องมือ และสามารถมุ่งประเด็นไปที่การทำงานกับ OLAP Cube ได้ทันที

6.2 ข้อเสนอแนะ

โปรแกรมที่พัฒนาขึ้นมานี้ นอกจากนำไปใช้กับ OLAP Cube ของข้อมูลการขายสินค้าที่ได้ออกแบบไว้แล้ว ยังสามารถนำไปใช้กับ Cube อื่น ๆ ที่อยู่ใน SQL Server Analysis Service ได้ด้วย เนื่องจากพัฒนาให้เป็นเครื่องมือในการดูข้อมูลใน Cube แบบทั่ว ๆ ไป ไม่ได้เฉพาะเจาะจง ทั้งนี้โปรแกรมจะทำงานตาม Cube ที่ได้ออกแบบไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาโปรแกรมวิเคราะห์ข้อมูลนี้ เป็นการพัฒนาแบบ Windows Form กล่าวคือ สามารถทำงานได้บน Windows เท่านั้น ซึ่งต้องมีการติดตั้งโปรแกรมด้วย อาจทำให้ไม่สะดวกบ้างในการใช้งาน ทั้งนี้สามารถพัฒนาเพิ่มเติมเพื่อให้ทำงานบน Web Form ได้ โดยใช้ ASP และ VBScript ซึ่งลักษณะการเขียนโปรแกรมก็ไม่แตกต่างกันมากนัก

นอกจากนี้โปรแกรมที่พัฒนาขึ้นมาได้มีการเรียกใช้คอมโพเน้นท์สำหรับติดต่อกับ OLAP Server ได้แก่ OLE DB for OLAP ซึ่งเป็นเพียงส่วนหนึ่งของคอมโพเน้นท์ที่มีอยู่เท่านั้น ซึ่งหากมีการเรียนรู้คอมโพเน้นท์ที่มีมากับ OLAP Server เหล่านี้เพิ่มเติมสามารถนำไปพัฒนาโปรแกรมที่มีประสิทธิภาพสูงขึ้นต่อไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

George Hernandez. 2003. **ADO-MD Object.** [Online]. Available : http://www.georgehernandez.com/xDatabases/ADO/ADOMDObjects/Index.htm

Microsoft Corporation. 2003. **MSDN Library (ADO-MD).** [Online]. Available : http://msdn.microsoft.com/library /en-us/ado270/htm/ammscadomdobjects.asp

Microsoft Corporation. 2003. **MSDN Library (ADO-MD 2.7).** [Online]. Available : http://msdn.microsoft.com/library/en-us/ado270/htm/amobjaspadocomplexexamplevb scriptx.asp

Microsoft Corporation. 2003. **MSDN Library (MDX).** [Online]. Available : http://msdn.microsoft.com/library/en-us/ olapdmad/agmdxbasics_04qg.asp

Microsoft Corporation. 2003. **MSDN Library (VB Example for ADO-MD).** [Online]. Available : http://msdn.microsoft.com/library/en-us/ado270/htm/amobjaspadocomplex examplevbscriptx.asp

Sakhr Youness. 2003. **Accessing Microsoft OLAP Data with ADO-MD and MDX.** [Online]. Available : http://www.topxml.com/conference/wrox/2000_vegas/Powerpoints/sakhr_ olap.ppt.

Sneath Tim. 2003. **Building OLAP Applications with SQL Server and Visual Basic.** [Online]. Available : http://www.vbug.co.uk/vbug2000/Slides/OLAP.ppt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

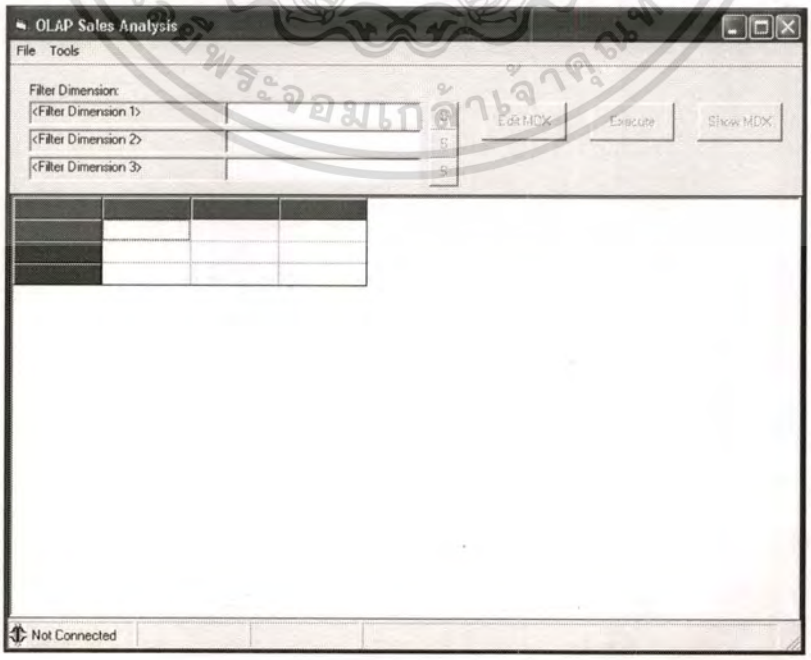
ภาคผนวก

คู่มือการใช้งานโปรแกรม OLAP Sales Analysis

โปรแกรม OLAP Sales Analysis ที่ได้พัฒนาขึ้นมานี้ มีหน้าจอหลักสำหรับการทำงานสองหน้าจอ คือ หน้าจอแสดงข้อมูล และหน้าจอ Cube Browser ซึ่งในส่วนของหน้าจอ Cube Browser นี้นอกจากจะสามารถดูข้อมูลโครงสร้างของ OLAP Cube ที่ได้สร้างไว้แล้ว ยังสามารถเลือก Dimension ต่าง ๆ เพื่อทำการสร้างรายงานได้ด้วย ซึ่งเรียกการทำงานในส่วนนี้ว่า MDX Builder เนื่องจากเมื่อเลือก Dimension สำหรับแสดงข้อมูลในแต่ละส่วนเรียบร้อยแล้ว โปรแกรมจะทำการสร้างคำสั่ง MDX ให้โดยอัตโนมัติ เพื่อเรียกข้อมูลจาก OLAP Cube มาแสดงในหน้าจอแสดงข้อมูลต่อไป

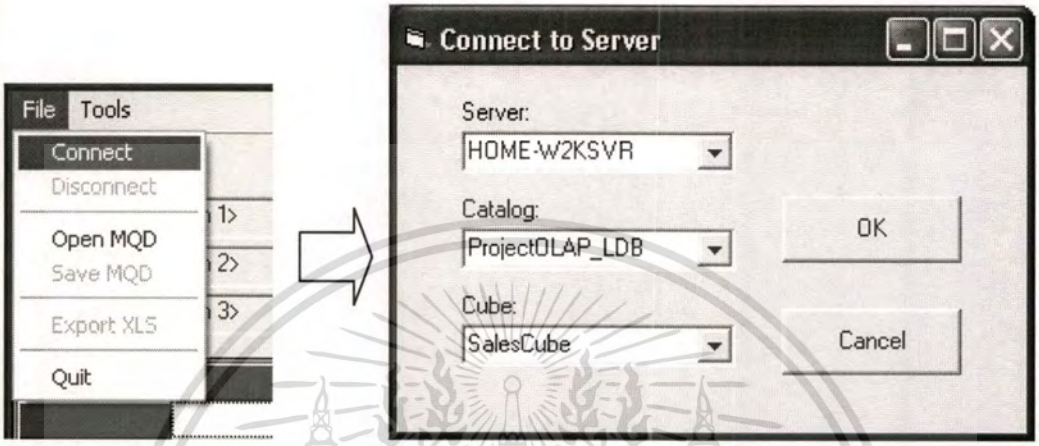
ขั้นตอนการใช้งาน – สร้างรายงานใหม่

1. เมื่อเปิดโปรแกรมขึ้นมาครั้งแรก โปรแกรมจะเปิดหน้าจอสำหรับแสดงข้อมูล แต่ยังไม่มีข้อมูลใด ๆ แสดงอยู่ พร้อมทั้งบอกสถานะว่ายังไม่มีการ Connect ไปยัง Cube ใด ๆ ดังภาพ

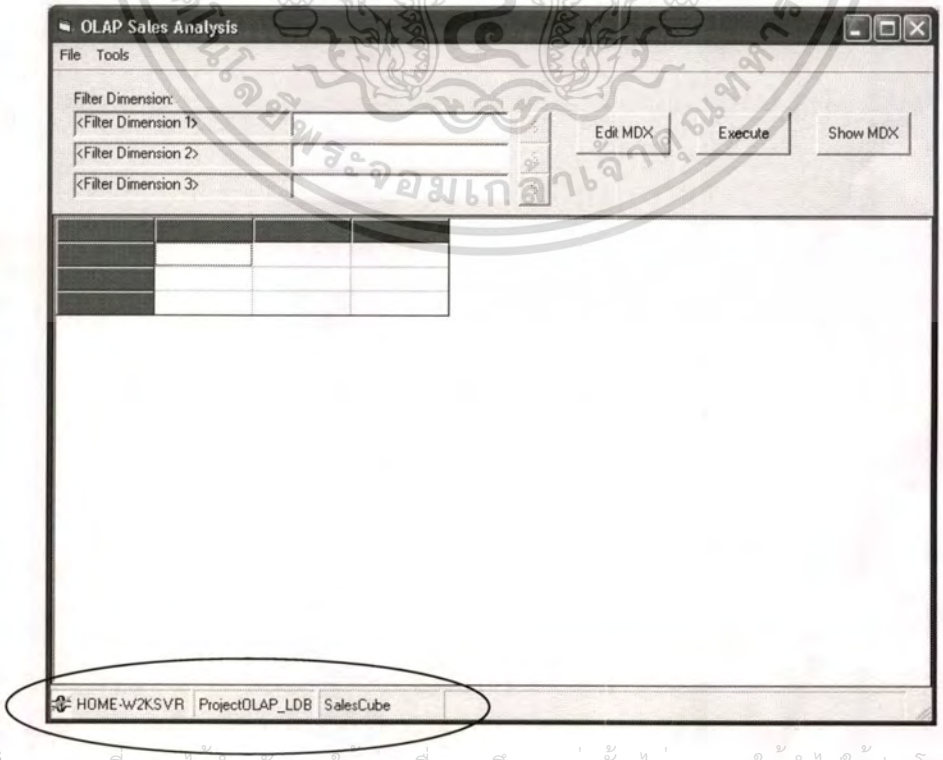


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ต้องทำการติดต่อไปยัง OLAP Server เพื่อทำงานกับข้อมูลที่อยู่ใน Server โดยเลือกเมนู File และเลือกที่ Connect เพื่อติดต่อไปยัง OLAP Server ซึ่งโปรแกรมจะเปิด Dialog Box สำหรับติดต่อกับ Server ขึ้นมาให้ ดังภาพ

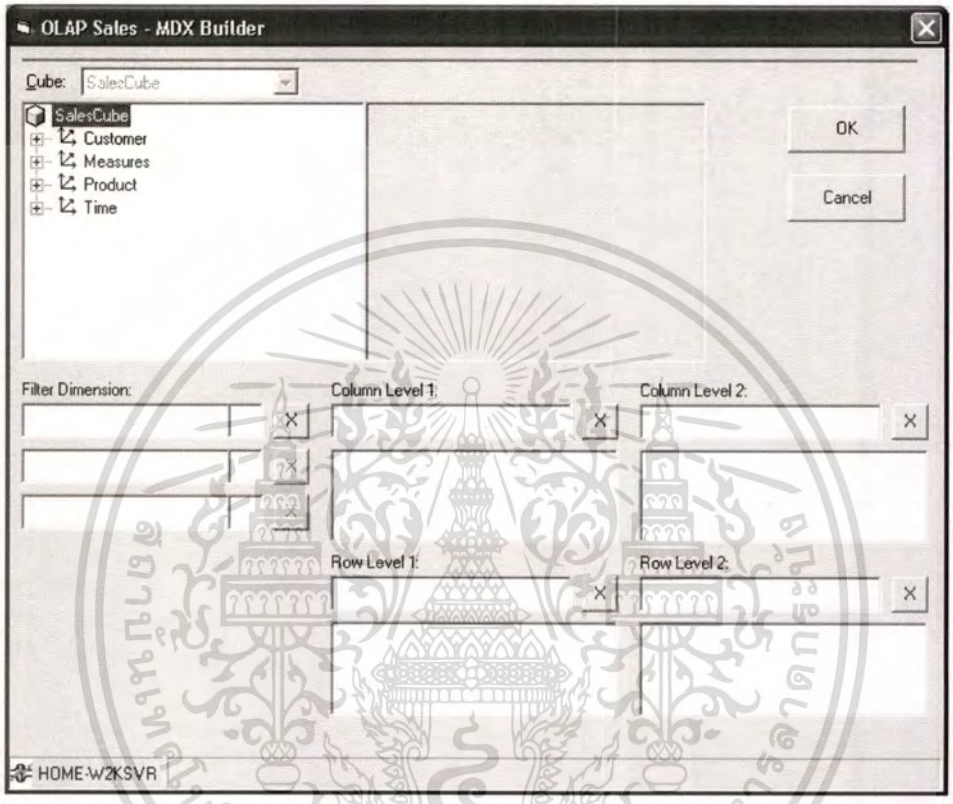


3. ทำการเลือก Server, Catalog และ Cube เรียบร้อยแล้ว เมื่อกด OK โปรแกรมจะกลับมาสู่หน้าจอแสดงข้อมูลอีกครั้ง แต่บอกสถานะว่า สามารถติดต่อกับ OLAP Server ได้แล้ว แต่ยังไม่มีการแสดงข้อมูลใด ๆ เนื่องจากต้องทำการสร้างคำสั่ง MDX ก่อน ดังภาพ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. กดปุ่ม Edit MDX เพื่อเปิดหน้าจอหลักอีกหน้าจอหนึ่ง คือ Cube Browser และ MDX Builder ขึ้นมา เพื่อจะทำการเลือกข้อมูลที่จะมาแสดงในหน้าจอนี้ ซึ่งหน้าจอ MDX Builder จะมีลักษณะดังภาพ



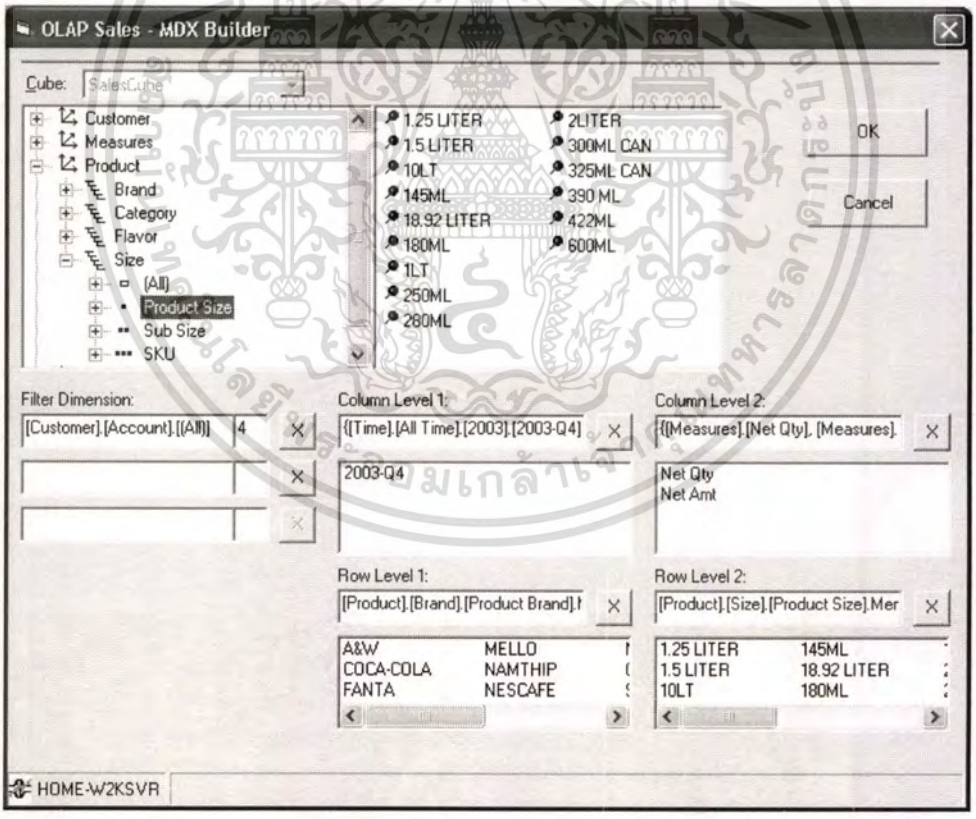
ซึ่งอธิบายรายละเอียดในแต่ละส่วนได้ดังนี้

1. Cube Browser จะทำหน้าที่ในการดูข้อมูลใน Cube ว่ามีโครงสร้างเป็นอย่างไร ซึ่งสามารถดูได้ว่า มี Dimension ทั้งหมดเท่าไร และมี Dimension อะไรบ้าง และในแต่ละ Dimension มี กี่ Hierarchy มี Member เป็นอะไร เป็นต้น โดยแสดงข้อมูลในรูปแบบที่เป็น Tree View ทำให้ง่ายต่อการใช้งาน
2. Filter Dimension เป็นส่วนที่ทำการเลือกว่าจะนำ Dimension หรือ Hierarchy ใดมาทำหน้าที่เป็น Filter บ้าง ยกตัวอย่างเช่น หากเลือก Customer.Account มาเป็น Filter Dimension เมื่อกลับออกไปที่หน้าจอแสดงข้อมูล จะสามารถกด Drop Down เพื่อให้ข้อมูลที่แสดงอยู่ แสดงเฉพาะ Key Account หรือ แสดงเฉพาะ Normal Account ก็ได้ หรืออาจจะให้แสดงข้อมูลของทั้งสอง Account รวมกันเลยก็ได้
3. Column และ Row เป็นการเลือก Dimension หรือ Hierarchy ใดมาแสดงข้อมูลด้าน Column และ Row โดยที่ทั้ง Column และ Row มีได้อย่างละสอง Dimension

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

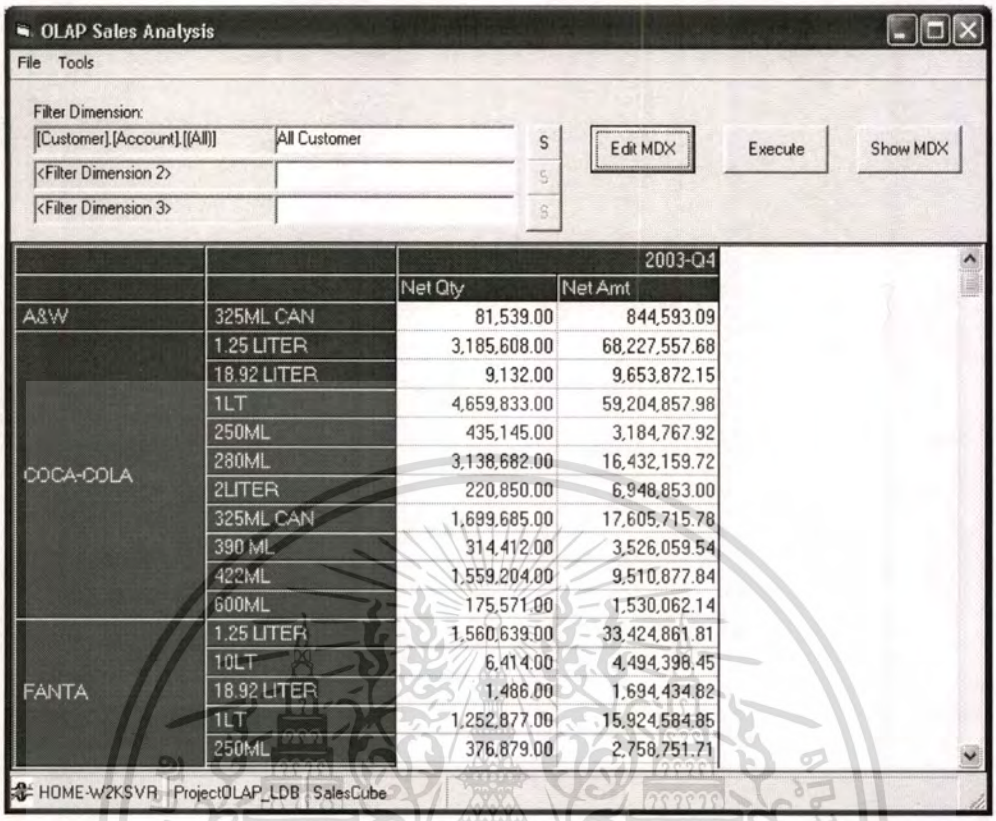
หมายความว่า สามารถเลือก Product.Brand และ Product.Size มาอยู่ทางด้าน Row ได้พร้อมกัน โดยเลือก Product.Brand ใส่ลงใน Row Level 1 ส่วน Product.Size ใส่ลงใน Row Level 2 เป็นต้น

5. ทำการเลือกข้อมูลเพื่อสร้างคำสั่ง MDX และให้แสดงข้อมูลในหน้าจอแสดงข้อมูล ในกรณีนี้จะยกตัวอย่างการดูข้อมูลด้าน Product ที่นำ Brand และ Size มา Cross กันในด้าน Row และต้องการดูยอดขายที่เป็น Net Quantity และ Net Amount ณ. Quarter 4 ปี 2003 โดยในระหว่างการแสดงข้อมูล สามารถเปลี่ยนดูได้ทั้งที่เป็นลูกค้าประเภท Key Account และ Normal Account ดังนั้นในตัวอย่างนี้จะต้องทำการเลือก Dimension ต่าง ๆ ดังภาพ ซึ่งในขณะที่ทำการเลือกนั้น ใช้การ Drag Drop และถ้าเลือก Level โปรแกรมจะทำการเลือก Member ใน Level นั้นมาให้ทั้งหมด แต่ก็สามารถเลือก Member เป็นรายตัวได้ด้วย

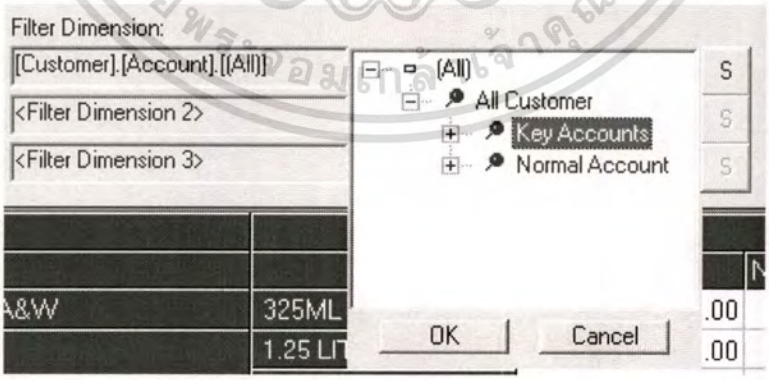


6. กดปุ่ม OK เพื่อสร้างคำสั่ง MDX และแสดงข้อมูลที่ได้ ในหน้าจอแสดงข้อมูล ดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

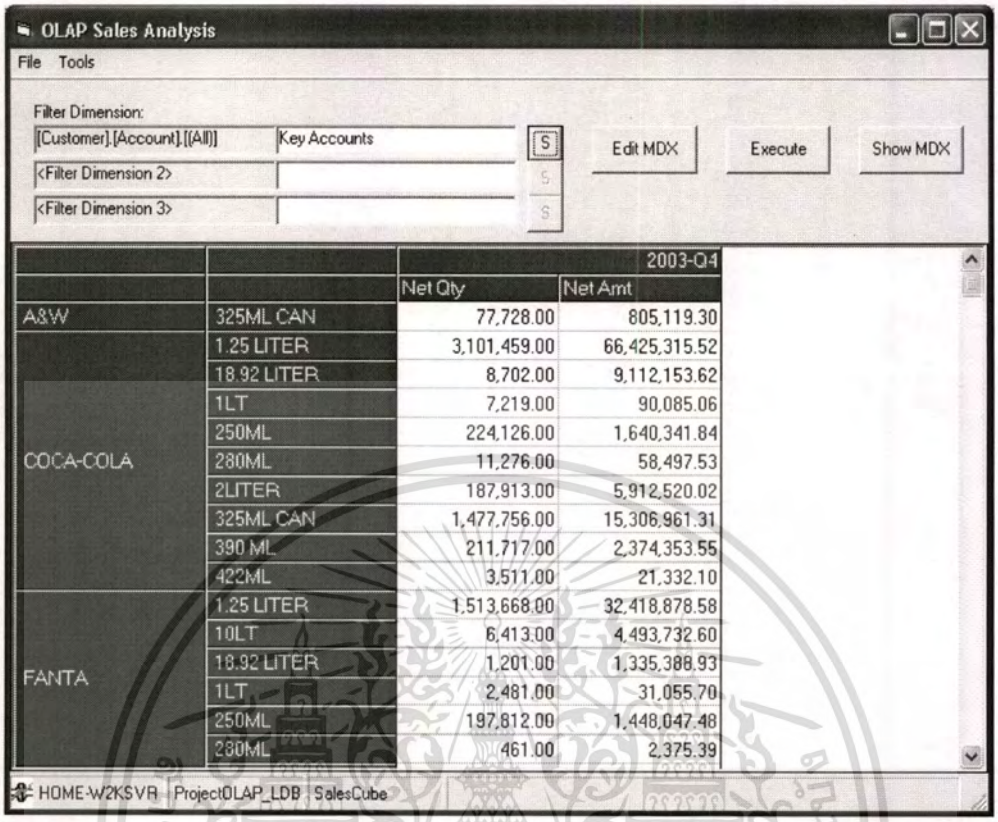


จะเห็นว่า Filter Dimension ที่ได้กำหนด Customer.Account มาใส่ไว้นั้น บอกว่าขณะนี้แสดงข้อมูลของลูกค้าทั้งหมด (All Customer) ซึ่งสามารถเลือกให้แสดงเฉพาะลูกค้า Key Account ได้โดยการกดปุ่ม S เล็ก ๆ ที่อยู่ด้านข้าง แล้วทำการเลือก Key Account ได้ ดังภาพ



7. ซึ่งทำให้ข้อมูลที่แสดงในตารางเป็นตัวเลขเฉพาะของ Key Account เท่านั้น ดังภาพ

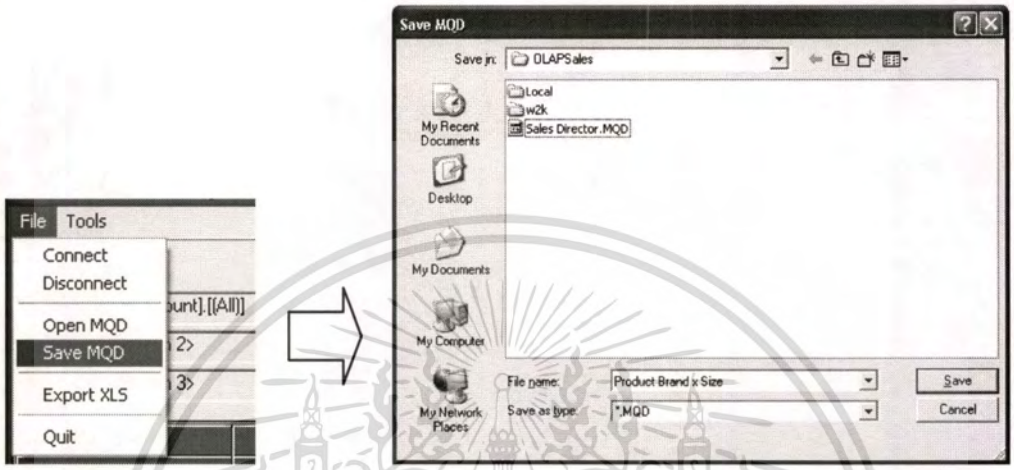
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการใช้งาน – Save รายงานเพื่อนำกลับมาใช้ใหม่ในภายหลัง

1. เมื่อได้รายงานที่สมบูรณ์แล้วต้องการ Save เพื่อเก็บไว้ใช้งานในภายหลัง ทำได้โดย เลือกเมนู File แล้วเลือก Save MQD (MDX Query Definition) ดังภาพ

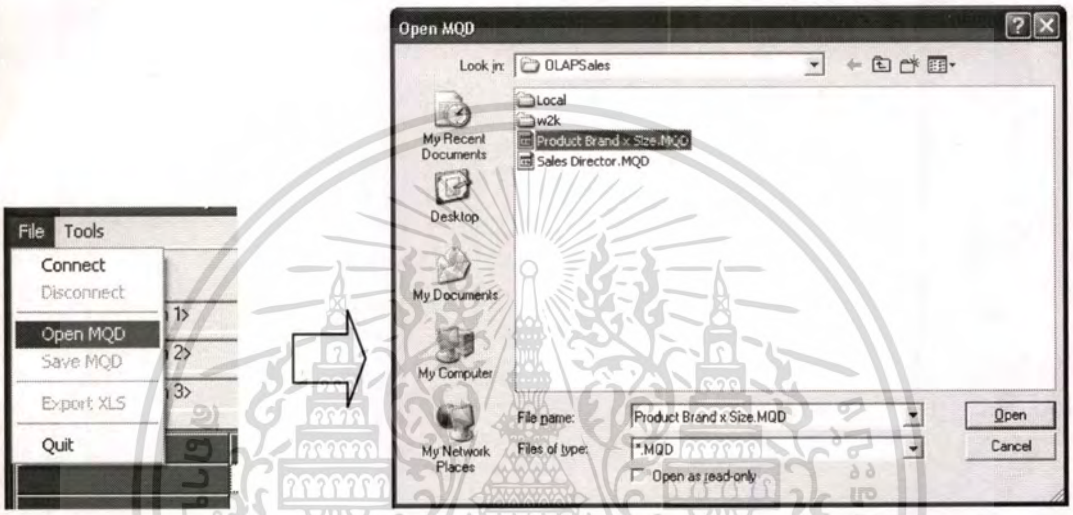


2. เลือก Folder ที่ต้องการเก็บไฟล์ จากนั้นตั้งชื่อไฟล์ แล้วกด Save โปรแกรมจะทำการสร้างไฟล์ MQD เก็บไว้เพื่อนำมาเปิดใช้งานในภายหลังได้

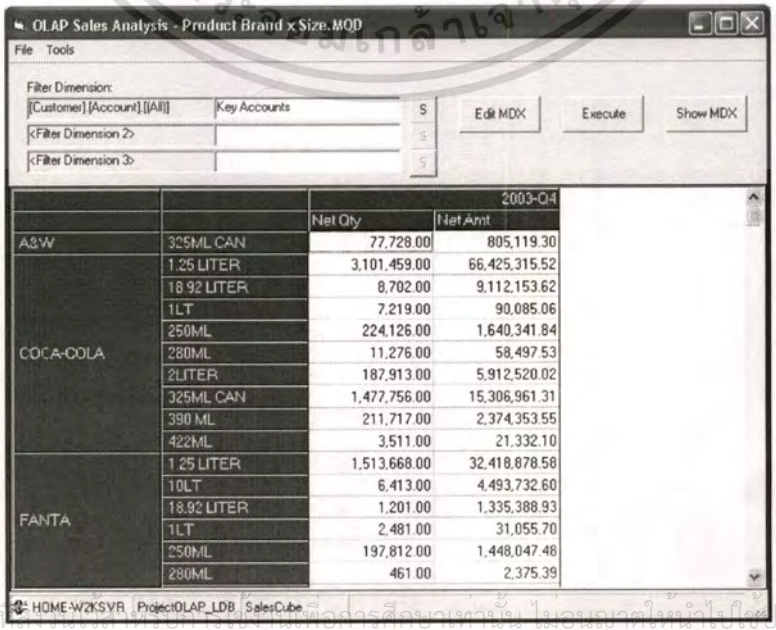
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการใช้งาน – Open รายงานที่เคย Save ไว้มาใช้งาน

1. หากเคยทำการ Save ไฟล์ MDQ ไว้แล้ว ในขั้นตอนการเปิดโปรแกรมขึ้นมาทำงานในครั้งแรก สามารถ สั่ง Open ไฟล์ MDQ ขึ้นมาทำงานได้เลย โดยไม่ต้องทำการ Connect ไปยัง Server ด้วย เนื่องจากไฟล์ MDQ จะมีการเก็บชื่อ OLAP Server และรายละเอียดต่าง ๆ ไว้ทั้งหมด การ Open ไฟล์ MDQ ทำได้โดยเลือกเมนู File แล้วเลือก Open MDQ ดังภาพ



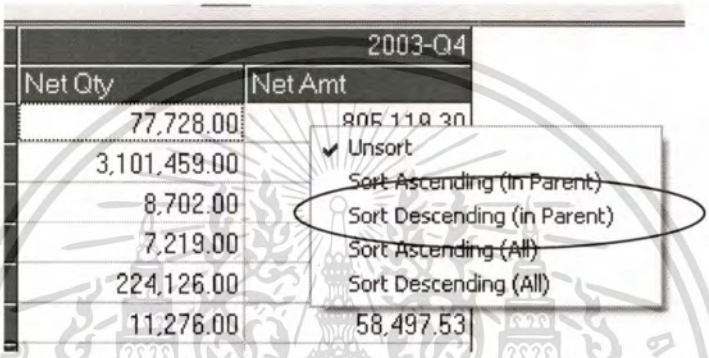
2. เลือก Folder ที่เก็บไฟล์ MDQ จากนั้นเลือกไฟล์ที่ต้องการ แล้วกด Open ซึ่งโปรแกรมจะทำการเปิดไฟล์ดังกล่าวขึ้นมา พร้อมทั้งทำการติดต่อไปขอข้อมูลจาก OLAP Server มาแสดงในหน้าจอแสดงข้อมูลทันที ดังภาพ



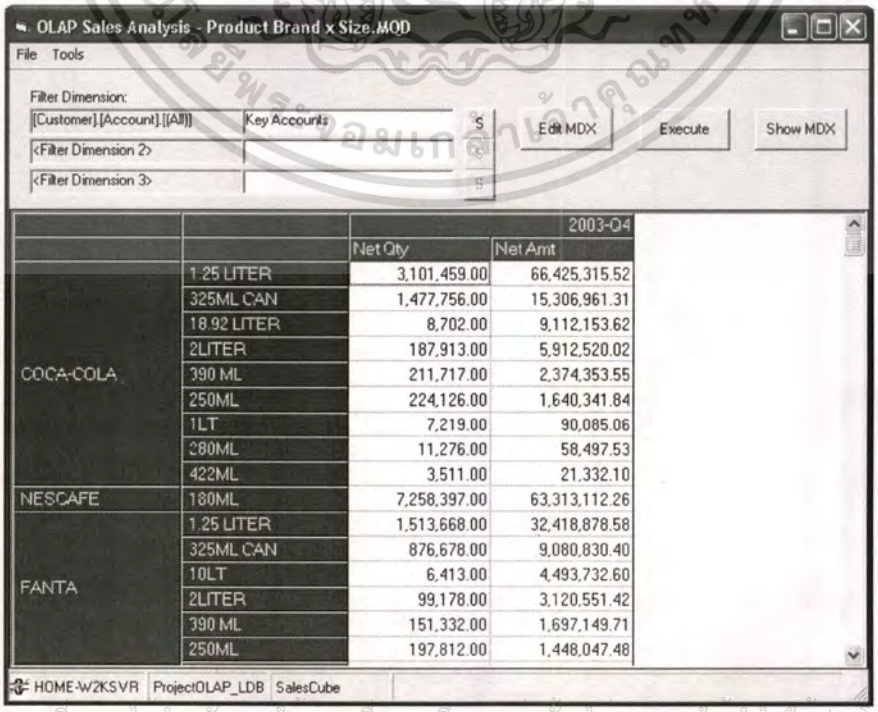
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานอื่น ๆ – การจัดเรียงข้อมูลตามต้องการ

การจัดเรียงข้อมูล หรือการ Sort นั้นมีรูปแบบการ Sort สี่แบบด้วยกัน ซึ่งแบ่งเป็นสองรูปแบบคือเรียงจากน้อยไปมาก หรือเรียงจากมากไปน้อย แต่ใน OLAP จะมีการเรียงแบบสนใจ Parent หรือไม่สนใจ Parent เข้ามาร่วมด้วย ทำให้เกิดการจัดเรียงทั้งหมดสี่แบบ การใช้งานการ Sort นั้นสามารถทำได้โดยคลิ้กเม้าส์ปุ่มขวา บน Column ที่ต้องการ Sort ในหน้าแสดงข้อมูลได้เลย ซึ่งจะเกิด Menu ขึ้นมาดังภาพ

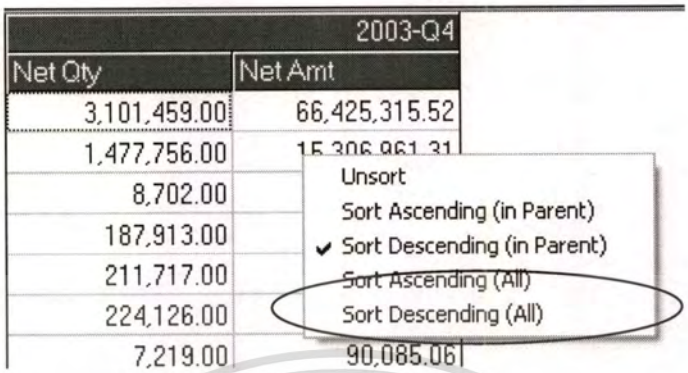


ซึ่งในภาพนี้เป็นการ คลิ้กขวาที่ Column ชื่อ 2003-Q4 และ Net Amount ซึ่งเมนูที่ขึ้นมานี้แสดงให้ทราบว่าข้อมูลที่แสดงไม่ได้มีการ Sort ใด ๆ อยู่ ซึ่งเมื่อทำการ Sort Descending แบบสนใจ Parent จะทำให้ได้ข้อมูลดังภาพ

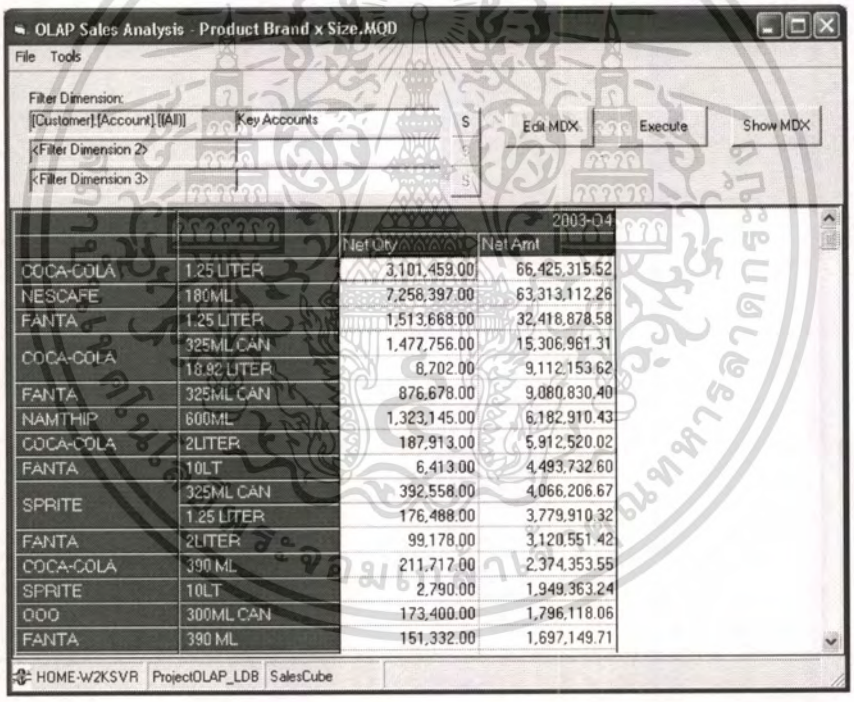


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และหากทำการเลือกใหม่เป็นการ Sort Descending แบบไม่สนใจ Parent



จะได้ข้อมูลดังภาพ



ซึ่งจะเห็นถึงความแตกต่างของการ Sort แบบสนใจ และไม่สนใจ Parent ของ OLAP ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานอื่น ๆ – การ Drill Up และ Drill Down

การ Drill Up และ Drill Down สามารถทำได้ที่ด้าน Row เท่านั้น และ Drill ได้เพียง Dimension เดียวเท่านั้น กล่าวคือ ถ้าด้าน Row ถูกเลือกมาจากสอง Dimension จะสามารถ Drill ได้ใน Dimension ที่สองเท่านั้น แต่ถ้า Row ถูกเลือกมาจาก Dimension เดียว ก็จะสามารถ Drill ได้เลย ซึ่งวิธีการ Drill นั้นทำได้โดยคลิ้กเม้าส์ปุ่มขวา ที่ Member ที่แสดงอยู่ในด้าน Row แล้วจะปรากฏเมนูขึ้นมาให้ทำการ Drill ดังภาพ

		2003-Q4	
		Net Qty	Net Amt
COCA-COLA	1.25 LITER		66,425,315.52
NESCAFE	180ML		63,313,112.26
FANTA	1.25 LITER		32,418,878.58
COCA-COLA	325ML CAN		15,306,961.31
	18.92 LITER		9,112,153.62
FANTA	325ML CAN		9,080,830.40
NAMTHIP	600ML	1,323,145.00	6,182,910.43

1.25 LITER -> Drill Up
1.25 LITER -> (All)
1.25 LITER -> Product Size
1.25 LITER -> Sub Size
1.25 LITER -> SKU

ในภาพนี้แสดงการคลิ้กขวาที่ Member ชื่อ 1.25 LITER ซึ่งสามารถ Drill Up หรือ Drill Down ลงสู่ Level ที่ต่ำกว่าได้ ซึ่งเมื่อทำการเลือก Drill Down ลงระดับ SKU ผลที่ได้จะแสดงได้ดังภาพ

OLAP Sales Analysis - Product Brand x Size.MQD

File Tools

Filter Dimension:

[Customer].[Account].[All] | Key Accounts | Edit MDX | Execute | Show MDX

<Filter Dimension 2>

<Filter Dimension 3>

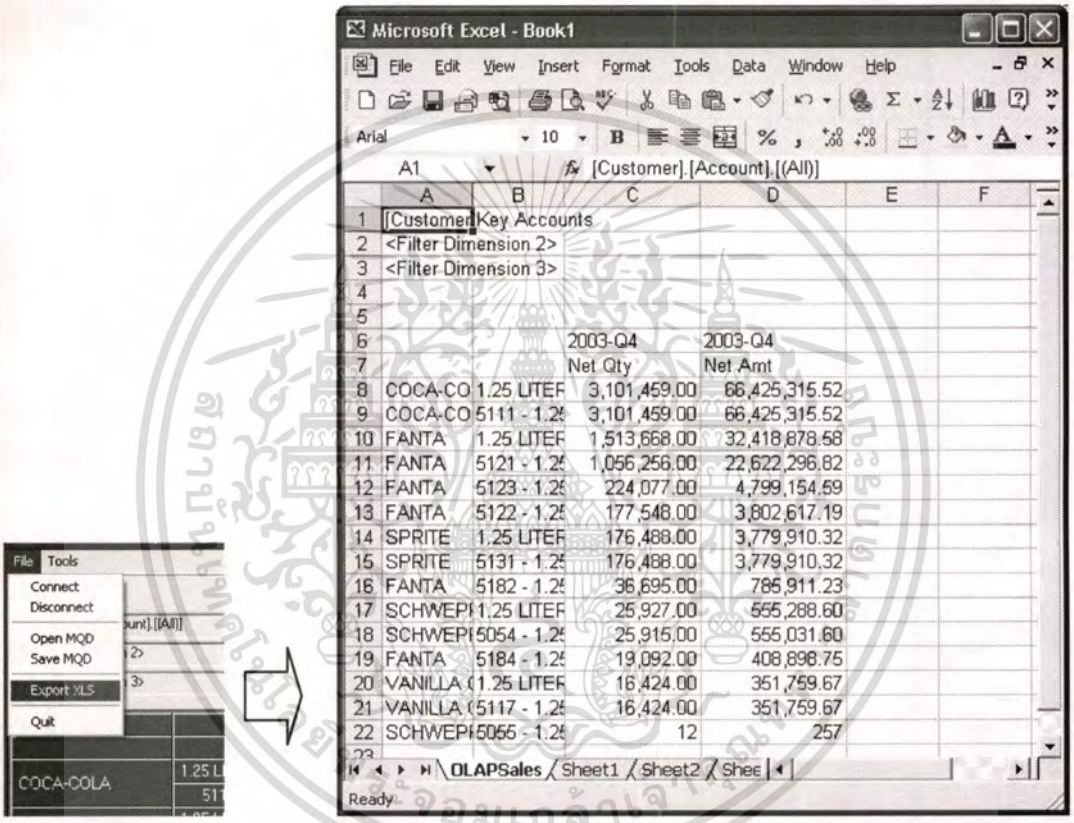
		2003-Q4	
		Net Qty	Net Amt
COCA-COLA	1.25 LITER	3,101,459.00	66,425,315.52
	5111 - 1.25LCOLA-COLA	3,101,459.00	66,425,315.52
FANTA	1.25 LITER	1,513,668.00	32,418,878.58
	5121 - 1.25LFANTA ORG	1,056,256.00	22,622,296.82
	5123 - 1.25LFANTA FRP	224,077.00	4,799,154.59
	5122 - 1.25LFANTA STR	177,548.00	3,802,617.19
SPRITE	1.25 LITER	176,488.00	3,779,910.32
	5131 - 1.25LSPRITE x1	176,488.00	3,779,910.32
FANTA	5182 - 1.25LFANTA BBR	36,695.00	785,911.23
SCHWEPPES	1.25 LITER	25,927.00	555,288.60
	5054 - 1.25LSCHWEPPES	25,915.00	555,031.60
FANTA	5184 - 1.25LFANTA NTR	19,092.00	408,898.75
VANILLA COKE	1.25 LITER	16,424.00	351,759.67
	5117 - 1.25LVANILLA COK	16,424.00	351,759.67
SCHWEPPES	5055 - 1.25LSCHWEPPES	12.00	257.00

HOME-W2KSVR | ProjectOLAP_LDB | SalesCube

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานอื่น ๆ – การ Export ไปยังโปรแกรม Excel

ความสามารถอีกอย่างที่โปรแกรมนี้สามารถทำได้คือการส่งข้อมูลที่อยู่ภายในตารางไปยังโปแกรม Microsoft Excel เพื่อนำไปทำงานต่อ ไม่ว่าจะเป็นการสร้างกราฟ หรือการนำไปจัดรูปแบบการพิมพ์ให้สวยงามได้ ซึ่งวิธีการส่งข้อมูลออกไปยัง Excel ทำได้โดย เลือกเมนู File แล้วเลือก Export XLS โดยที่เมนูนี้จะใช้งานได้ก็ต่อเมื่อมีข้อมูลอยู่ในตารางแล้วเท่านั้น



โดยโปรแกรม OLAP Sales นี้จะทำการเปิดโปรแกรม Excel ขึ้นมา แล้วนำข้อมูลที่แสดงอยู่ในตารางส่งไปไว้ใน Cell ต่าง ๆ ของ Excel รวมถึงข้อมูลที่อยู่ใน Filter Dimension ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายธนเดช จันทฑีโร
สถานที่เกิด	จังหวัดจันทบุรี
ระดับประถมศึกษา	โรงเรียนสฤษดิเดช จังหวัดจันทบุรี
ระดับมัธยมศึกษา	โรงเรียนเบญจมราชูทิศ จังหวัดจันทบุรี
ระดับอุดมศึกษา	ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2536
ประสบการณ์การทำงาน	ธนาคารกรุงเทพ จำกัด (มหาชน) บริษัท ไทยประกันชีวิต จำกัด บริษัท ทีโอเอเพ้นท์ (ประเทศไทย) จำกัด บริษัท ไทยน้ำทิพย์ จำกัด
E-mail Address	dej99@anet.net.th

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้