การพัฒนาโปรแกรมสร้างเอ็มดีเอ็กซ์ออนไลน์

นายวีรภัทร ศรีพิพัฒนกุล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2551

DEVELOPMENT OF ONLINE MDX GENERATOR PROGRAM

Weerapat Sripipattanakul

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Master of Engineering in Computer Engineering

Suranaree University of Technology

Academic Year 2008

การพัฒนาโปรแกรมสร้างเอ็มดีเอ็กซ์ออนไลน์

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

_____

(รศ. ดร.กิตติศักดิ์ เกิดประสพ)
ประธานกรรมการ

_____

(ผศ. ดร.พิชโยทัย มหัทธนาภิวัฒน์)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

_____

(รศ. ดร.นิตยา เกิดประสพ)
กรรมการ

_____

(อ. ดร.ปรเมศวร์ ห่อแก้ว)
กรรมการ

_____

(ศ. ดร.ไพโรจน์ สัตยธรรม)
รองอธิการบดีฝ่ายวิชาการ

_____

(รศ. น.อ. ดร.วรพจน์ ขำพิศ)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

วีรภัทร ศรีพิพัฒนกุล : การพัฒนาโปรแกรมสร้างเอ็มดีเอ็กซ์ออนไลน์ (DEVELOPMENT OF ONLINE MDX GENERATOR PROGRAM) อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.พิชโยทัย มหัทธนาภิวัฒน์, 85 หน้า.

ปัจจุบัน ข้อมูลเชิงวิเคราะห์ได้กลายมาเป็นปัจจัยสำคัญสำหรับการทำงานในองค์กรต่าง ๆ การทำสารสนเทศข้อมูลในองค์กรนั้นเป็นส่วนสำคัญส่วนหนึ่งที่ทำให้ผู้บริหารขององค์กรสามารถวิเคราะห์ผลดำเนินงานจากข้อมูลเชิงวิเคราะห์เพื่อนำไปใช้ช่วยประกอบการตัดสินใจในการวางแผนการดำเนินงานในอนาคตได้อย่างมีประสิทธิภาพ และข้อมูลที่นำมาใช้สำหรับงานวิเคราะห์นั้นเก็บอยู่ในรูปของคลังข้อมูล (Data Warehouse) และลูกบาศก์ข้อมูล (Data Cube)

การที่จะนำข้อมูลจากลูกบาศก์ข้อมูลที่มีอยู่นั้นออกมาใช้ได้อย่างมีประสิทธิภาพต้องอาศัยความรู้และความเชี่ยวชาญในการใช้คำสั่งที่จะเรียกข้อมูล นั่นคือเอ็มดีเอ็กซ์ (Multi-Dimensional Expressions) ซึ่งมีความยากซับซ้อนสำหรับผู้ที่ไม่มีความรู้ความเข้าใจเกี่ยวกับลักษณะโครงสร้างของคลังข้อมูลและลูกบาศก์ข้อมูล ดังนั้นงานวิจัยนี้จึงนำเสนอโปรแกรมที่ช่วยสำหรับการออกแบบและสร้างเอ็มดีเอ็กซ์แบบออนไลน์ โดยโปรแกรมจะใช้ออกแบบและสร้างเอ็มดีเอ็กซ์ เพื่อลดความผิดพลาดและเวลาที่ใช้ในการออกแบบและสร้างเอ็มดีเอ็กซ์ สร้างความสะดวกให้กับผู้ใช้ที่ไม่มีความชำนาญในการใช้เอ็มดีเอ็กซ์ที่เริ่มต้นออกแบบและสร้างเอ็มดีเอ็กซ์ สำหรับการเรียกดูข้อมูลในลักษณะข้อมูลสรุปจากลูกบาศก์ข้อมูล

สาขาวิชา<u>วิศวกรรมคอมพิวเตอร์</u> ลายมือชื่อนักศึกษา _____

ปีการศึกษา 2551 ลายมือชื่ออาจารย์ที่ปรึกษา_____

WEERAPAT SRIPIPATTANAKUL : DEVELOPMENT OF ONLINE MDX GENERATOR PROGRAM. THESIS ADVISOR : ASST. PROF. PITCHAYOTHAI MAHATTHANAPIWAT, Ph.D., 85 PP.

DEVELOPMENT OF ONLINE MDX GENERATOR PROGRAM

Nowadays, analytical data is an important factor for organizations work. Decision maker can analyze overall operation from it to make the efficient work plans. Data for analytical works are in form of data warehouse and data cube.

Using data cube effectively is based on expertise and knowledge in query. Furthermore, MDX (Multi-Dimensional Expression) is quite complex for unskillful users to data warehouse and data cube construction. Therefore, this research purposes a program which aid for designing and creating online MDX, this program reduce a mistake and time lost in designing and creating MDX. It will facilitate for novel users to examine summary data from data cube.

School of Computer Engineering Student's Signature _____

Academic Year 2008 Advisor's Signature _____

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอกราบขอบพระคุณ บุคคล และกลุ่มบุคคลต่าง ๆ ที่ได้กรุณาให้คำปรึกษา แนะนำ ช่วยเหลืออย่างดียิ่ง ทั้งในด้านวิชาการ และด้านการดำเนินงานวิจัย ดังต่อไปนี้

- ผศ.ดร. พิชโยทัย มหัทธนาภิวัฒน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์
- รศ.ดร. กิตติศักดิ์ เกิดประสพ หัวหน้าสาขาวิชาวิศวกรรมคอมพิวเตอร์ รศ.ดร. นิตยา เกิดประสพ ผศ.ดร. คะชา ชาญศิลป์ ผศ. สมพันธุ์ ชาญศิลป์ และ อ.ดร. ปรเมศวร์ ห่อแก้ว อาจารย์ประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
- คุณจริยาพร ศรีวิไลลักษณ์ ที่ให้คำปรึกษาการจัดรูปแบบ และช่วยตรวจทานความถูกต้องของวิทยานิพนธ์
- คุณกัลญา พับโพธิ์ เลขานุการสาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ให้ความช่วยเหลือในการประสานงานด้านเอกสารต่าง ๆ ระหว่างศึกษา
- และเพื่อน ๆ มหาบัณฑิตทุกคนที่คอยให้กำลังใจตลอดมา

นอกจากนี้ ขอขอบคุณครู อาจารย์ทั้งในอดีตและปัจจุบันที่ให้ความรู้แก่ผู้วิจัยจนประสบความสำเร็จในชีวิต

ท้ายที่สุดที่จะลืมไม่ได้ ขอกราบขอบพระคุณ บิดา มารดา ที่ให้กำเนิด อบรม เลี้ยงดูด้วยความรัก และส่งเสริมการศึกษาเป็นอย่างดีมาโดยตลอด ทำให้ผู้วิจัยมีความรู้ ความสามารถ มีจิตใจที่เข้มแข็ง รวมทั้งเป็นกำลังใจที่ยิ่งใหญ่ให้แก่ผู้วิจัย จนทำให้ผู้วิจัยประสบความสำเร็จในชีวิตเรื่อยมา

วีรภัทร ศรีพิพัฒนกุล

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย) ..... ก
บทคัดย่อ (ภาษาอังกฤษ) ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ช
สารบัญรูป ..... ซ

บทที่

1 บทนำ 1
1.1 ความสำคัญและที่มาของปัญหา 1
1.2 วัตถุประสงค์ของการวิจัย 3
1.3 ขอบเขตของงานวิจัย 3
1.4 ประโยชน์ที่คาดว่าจะได้รับ 4
2 ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง 5
2.1 ความหมายและบทบาทความสำคัญของระบบคลังข้อมูล 5
2.1.1 คุณลักษณะเฉพาะของคลังข้อมูล 5
2.1.2 สถาปัตยกรรมคลังข้อมูล (Data Warehouse Architecture) 6
2.1.3 เทคนิคในการสร้างคลังข้อมูล 7
2.2 ฐานข้อมูลแบบหลายมิติ (Multi Dimensional Databases) 10
2.2.1 องค์ประกอบของฐานข้อมูลแบบหลายมิติ 11
2.2.2 โครงสร้างของฐานข้อมูลแบบหลายมิติ
(Multi dimensional database schema) 11
2.3 On-Line Analytical Processing : OLAP 15
2.3.1 รูปแบบการเก็บข้อมูลของ OLAP (OLAP storage) 15
2.3.2 ประโยชน์ของ OLAP 17

สารบัญ (ต่อ)

หน้า

2.4 Multi Dimensional eXpressions (MDX) 18
2.4.1 เอ็มดีเอ็กซ์อย่างง่าย (Simple MDX query) 18
2.4.2 แบบจำลองข้อมูลของเอ็มดีเอ็กซ์ (MDX Data Model) 20
2.4.3 การสร้างเอ็มดีเอ็กซ์อย่างง่าย (Simple MDX Construction) 21
2.4.4 ข้อคิดเห็นในเอ็มดีเอ็กซ์ (Comments in MDX) 25
2.5 Decision Support Object (DSO) and Active Data Objects Multi-Dimensional (ADOMD) 27
2.6 Unified Model Langauge (UML) 30
2.6.1 Diagram of UML 30
2.6.2 ยูสเคสไดอะแกรม (Use case diagram) 30
2.6.3 คลาสไดอะแกรม (Class diagram) 32
2.6.4 ซีเควนซ์ไดอะแกรม (Sequence diagram) 35
3 ระเบียบวิธีวิจัย 37
3.1 ขั้นตอนการวิจัย 37
3.2 แหล่งที่มาของข้อมูลที่ใช้ในการทดสอบโปรแกรม 38
3.3 การวิเคราะห์และออกแบบโปรแกรมสร้างเอ็มดีเอ็กซ์ 40
3.3.1 ลักษณะการทำงานในภาพรวมของโปรแกรม 41
3.3.2 ยูสเคสไดอะแกรม (Use case diagram) 42
3.3.3 คลาสไดอะแกรม (Class diagram) 46
3.3.4 ซีเควนซ์ไดอะแกรม (Sequence diagram) 49
3.4 การออกแบบหน้าตาโปรแกรม (User Interface Design) 55
4 การทดสอบและอภิปรายผล 57
4.1 สภาพแวดล้อมที่ใช้ในการทดสอบ 57
4.2 การทดสอบการเชื่อมต่อแหล่งข้อมูล 58
4.3 การทดสอบการสร้างเอ็มดีเอ็กซ์ 60
4.4 อภิปรายผล 70

สารบัญ (ต่อ)

หน้า

5 สรุปผลการวิจัยและข้อเสนอแนะ 71
5.1 สรุปผลการวิจัย 72
5.2 การประยุกต์งานวิจัย 72
5.3 ปัญหาและข้อเสนอแนะ 73
รายการอ้างอิง 74
ภาคผนวก
ภาคผนวก ก. บทความผลงานวิจัยที่นำเสนอในการประชุมทางวิชาการเพื่อเสนอผลงานวิจัยครั้งที่ 4 "มหาวิทยาลัยกับการพัฒนาบนฐานความรู้" โดยความร่วมมือกับ สกว. และเครือข่าย มหาวิทยาลัยราชภัฏอีสาน 76
ประวัติผู้เขียน 85

สารบัญตาราง

ตารางที่ **หน้า**

2.1 ความสามารถและข้อจำกัดของ Analysis Serviceใน Microsoft SQL Server 2000 14

2.2 ข้อมูลยอดขายและต้นทุนของร้านค้า 18

2.3 เซ็ตข้อมูลผลลัพธ์ของเอ็มดีเอ็กซ์ตัวอย่าง 19

2.4 ผลลัพธ์จากเซ็ต query 21

2.5 แสดงผลลัพธ์การทำงานของฟังก์ชัน CrossJoin() 22

2.6 ผลลัพธ์จากการเรียงของฟังก์ชัน Order() 25

สารบัญรูป

รูปที่ **หน้า**

1.1 กระบวนการทำงานระบบคลังข้อมูล 2
2.1 โครงสร้างแบบดาว 12
2.2 โครงสร้างแบบเกล็ดหิมะ 13
2.3 แบบจำลองของลูกบาศก์ข้อมูล OLAP 14
2.4 ประเภทของ OLAP Storage 16
2.5 The ADOMD Connection object has Catalog and Cellset 29
2.6 ตัวอย่างยูสเคสไดอะแกรม 31
2.7 เครื่องหมายลูกศรแสดงความสัมพันธ์ระหว่างยูสเคส 31
2.8 สัญลักษณ์คลาส 32
2.9 ตัวอย่างคลาสไดอะแกรม 33
2.10 สัญลักษณ์ Robustness diagram 34
2.11 ตัวอย่าง Robustness diagram 35
2.12 ตัวอย่าง Sequence diagram 36
3.1 แผนภาพขั้นตอนการทำงานระหว่าง Client กับ Server 41
3.2 แผนภาพแสดงสถาปัตยกรรมการติดต่อข้อมูลสำหรับ OLAP Services 42
3.3 Use case diagram ของโปรแกรม 43
3.4 Class diagram (Robustness diagram) 46
3.5 แบบจำลองเชิงกิจกรรมการเลือกแหล่งข้อมูล 49
3.6 แบบจำลองเชิงกิจกรรมการสร้างเอ็มดีเอ็กซ์ 51
3.7 แบบจำลองเชิงกิจกรรมการดูผลการทำงานของเอ็มดีเอ็กซ์ 52
3.8 แบบจำลองเชิงกิจกรรมการดูโครงสร้างลูกบาศก์ข้อมูล 53
3.9 แบบจำลองเชิงกิจกรรมการแก้ไขเอ็มดีเอ็กซ์ 54
3.10 User interface ของโปรแกรม Online MDX Generator 56
4.1 การแสดงโครงสร้างลูกบาศก์ข้อมูลบนโครงสร้างต้นไม้ 59
4.2 หน้าโปรแกรมการสร้างเอ็มดีเอ็กซ์รายงานที่ 1 60

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

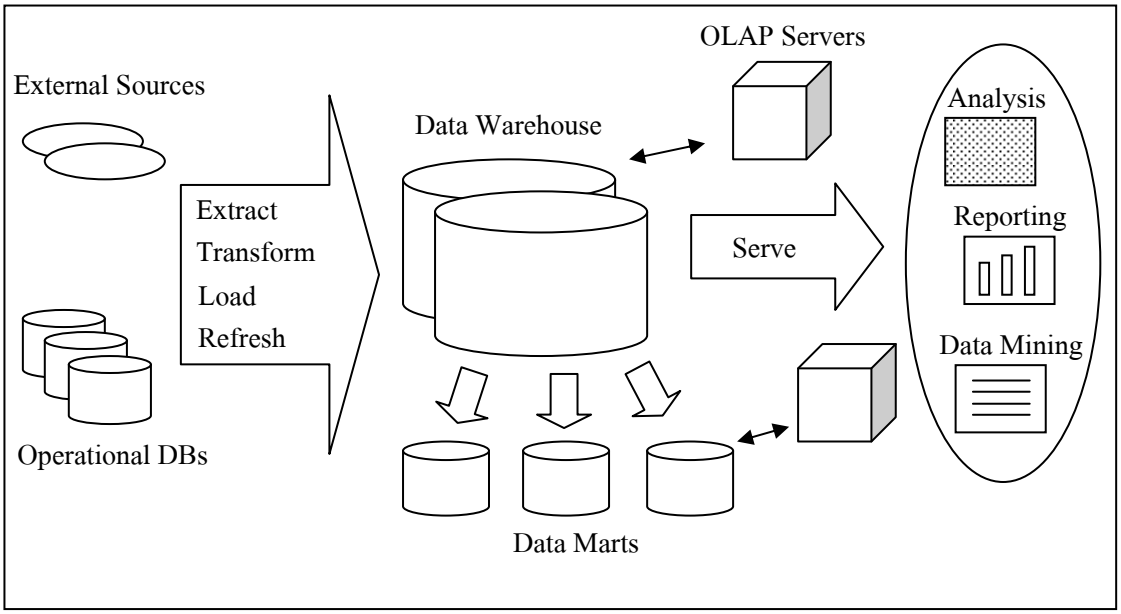
4.3 ผลลัพธ์การทำงานของเอ็มดีเอ็กซ์ที่ 1 61
4.4 หน้าโปรแกรมการสร้างเอ็มดีเอ็กซ์รายงานที่ 2 61
4.5 ผลลัพธ์การทำงานของเอ็มดีเอ็กซ์ที่ 2 62
4.6 หน้าโปรแกรมการสร้างเอ็มดีเอ็กซ์รายงานที่ 3 63
4.7 ผลลัพธ์การทำงานของเอ็มดีเอ็กซ์ที่ 3 63
4.8 หน้าโปรแกรมการสร้างเอ็มดีเอ็กซ์รายงานที่ 4 64
4.9 ผลลัพธ์การทำงานของเอ็มดีเอ็กซ์ที่ 4 65
4.10 หน้าโปรแกรมการสร้างเอ็มดีเอ็กซ์รายงานที่ 5 66
4.11 ผลลัพธ์การทำงานของเอ็มดีเอ็กซ์ที่ 5 67
4.12 หน้าโปรแกรมการสร้างเอ็มดีเอ็กซ์รายงานที่ 6 68
4.13 ผลลัพธ์การทำงานของเอ็มดีเอ็กซ์ที่ 6 69

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ปัจจุบัน คอมพิวเตอร์ได้เข้ามามีบทบาทในชีวิตประจำวันของมนุษย์มากขึ้น ข้อมูลได้กลายมาเป็นปัจจัยสำคัญสำหรับการทำงานในองค์กรทุก ๆ องค์กร ทั้งขนาดเล็ก ขนาดกลางและขนาดใหญ่ การใช้ระบบสารสนเทศในองค์กรนั้นเป็นส่วนสำคัญส่วนหนึ่งที่ทำให้ผู้บริหารขององค์กรสามารถวิเคราะห์ผลดำเนินงานที่ผ่านมาว่าเป็นไปในทิศทางใด และนำข้อมูลที่ได้จากการทำระบบสารสนเทศไปใช้เป็นเครื่องมือประกอบการตัดสินใจในการวางแผนการดำเนินงานในอนาคต รวมทั้งวางแผนการจัดสรรการใช้ทรัพยากรที่มีอยู่ให้เกิดประโยชน์สูงสุดและมีประสิทธิภาพดียิ่งขึ้น ดังนั้นในปัจจุบันพบว่าการทำระบบสารสนเทศเพื่อนำข้อมูลมาใช้เป็นแนวทางวิเคราะห์ในการวางแผนดำเนินงานขององค์กรจึงได้รับความนิยมและเผยแพร่เพิ่มมากขึ้นเรื่อย ๆ การที่มีการนำระบบสารสนเทศมาใช้ในองค์กร ข้อมูลที่นำมาใช้สำหรับงานวิเคราะห์นั้นส่วนมากแล้วถูกเก็บอยู่ในรูปของคลังข้อมูล (Data Warehouse) และลูกบาศก์ข้อมูล (Data Cube) ประกอบกับการใช้เทคโนโลยีที่เรียกว่า On-Line Analytical Processing (OLAP)

คลังข้อมูล คือฐานข้อมูลขนาดใหญ่เป็นศูนย์กลางที่รวบรวมข้อมูลจากแหล่งข้อมูลต่าง ๆ ที่มีความสอดคล้องกัน ซึ่งอาจมีโครงสร้างข้อมูลต่างกันโดยทำการถ่ายโอนข้อมูลจากแหล่งข้อมูลต่าง ๆ มาเก็บรวมกันด้วยแพลตฟอร์มที่กำหนดเป็นมาตรฐานไว้ที่ศูนย์กลางที่เดียว แล้วแบ่งออกเป็นคาต้ามาร์ทย่อย ๆ ตามแต่ละความต้องการเพื่อความสะดวกและความรวดเร็วในการทำงาน แล้วจากนั้นนำข้อมูลที่เก็บในรูปแบบที่พร้อมสำหรับการทำงานเชิงวิเคราะห์ผ่านกระบวนการคำนวณไปอยู่ในรูปแบบของฐานข้อมูลแบบหลายมิติหรือลูกบาศก์ข้อมูลเพื่อใช้ออกเป็นรายงานเชิงวิเคราะห์ที่สามารถนำเสนอข้อมูลเชิงวิเคราะห์รูปแบบของข้อมูลที่หลากหลาย แสดงกระบวนการทำงานได้ดังรูปที่ 1.1 เพื่อให้ผู้บริหารสามารถรับข้อมูลที่ถูกต้องแม่นยำต่อการตัดสินใจ ซึ่งแนวคิดนี้เรียกว่า Business Intelligence (BI) ลักษณะการออกแบบจะตรงกันข้ามกับการออกแบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งใช้หลักการ Normalization เพื่อพยายามแยกข้อมูลออกมาเก็บไว้ในตารางข้อมูลย่อย ๆ เพื่อลดความซ้ำซ้อนของข้อมูล ความล่าช้าและขาดประสิทธิภาพ แต่การออกแบบคลังข้อมูลนั้นใช้ในหลักการ De-Normalization พยายามรวมข้อมูลเข้าให้ได้มากที่สุดในตารางข้อมูล เพื่อทำให้มีความเร็วในการประมวลผลสำหรับหาผลลัพธ์ในลักษณะผลรวมของข้อมูล



รูปที่ 1.1 กระบวนการทำงานระบบคลังข้อมูล

OLAP คือกระบวนการประมวลผลทางคอมพิวเตอร์ที่ช่วยให้ผู้ใช้สามารถวิเคราะห์ข้อมูลในมิติต่าง ๆ ได้ง่ายขึ้น ที่มีลักษณะเด่นในการเรื่องความเร็วในการเรียกดูข้อมูล มุมมองข้อมูลที่สามารถแสดงได้หลากหลายตามความต้องการของผู้ใช้ สามารถทำการแบ่ง ลงลึกถึงรายละเอียดของข้อมูล (Drill Down) หรือแสดงในลักษณะภาพรวม (Roll Up) ลูกบาศก์ข้อมูลได้มาจากกระบวนการนี้โดยการนำข้อมูลที่อยู่ในคลังข้อมูลมาผ่านกระบวนการคำนวณในลักษณะการหาผลรวมของแต่ละฟิลด์ที่มีความสัมพันธ์กัน เมื่อผ่านกระบวนการคำนวณเรียบร้อยจะได้ผลลัพธ์ที่เรียกว่าลูกบาศก์ข้อมูล โครงสร้างในลูกบาศก์ข้อมูลประกอบด้วยตารางข้อมูล 2 ประเภท คือ ตารางข้อมูลหลัก (Fact table) กับตารางข้อมูลมิติ (Dimension table) และลูกบาศก์ข้อมูลมีรูปแบบโครงสร้างหลัก ๆ 2 รูปแบบคือโครงสร้างแบบดาว (Star schema) หรือโครงสร้างแบบเกล็ดหิมะ (Snow flake schema) ขึ้นอยู่กับการเลือกใช้ให้เหมาะกับลักษณะข้อมูลและการออกแบบสำหรับงานนั้น ๆ

การทำเหมืองข้อมูล หรือ การค้นหาความรู้จากฐานข้อมูล (Knowledge Discovery in Databases) คือ เทคนิคหาความรู้ รูปแบบ และความสัมพันธ์ของข้อมูลที่ซ่อนอยู่จากข้อมูลจำนวน

เมื่อมีการนำระบบคลังข้อมูลมาใช้ การจะนำข้อมูลจากลูกบาศก์ข้อมูลที่มีอยู่นั้นออกมาใช้ได้อย่างมีประสิทธิภาพ ต้องอาศัยความรู้และความเชี่ยวชาญในการใช้คำสั่งที่เรียกข้อมูล คือภาษาคำสั่ง เอ็มดีเอ็กซ์ (Multi-Dimensional eXpressions : MDX) เป็นภาษาคำสั่งที่สร้างขึ้นมาโดยบริษัท Microsoft เป็นภาษาลักษณะข้อคำถามคล้ายกับ SQL (Structured Query Language) ที่ใช้กับฐานข้อมูลเชิงสัมพันธ์ เอ็มดีเอ็กซ์ใช้เรียกดูข้อมูลจากลูกบาศก์ข้อมูลที่มีความยากซับซ้อนสำหรับผู้ที่ไม่มีความรู้ความเข้าใจเกี่ยวกับลักษณะโครงสร้างของคลังข้อมูลและลูกบาศก์ข้อมูล ดังนั้น ในงานวิจัยนี้จึงได้นำเสนอโปรแกรมที่ช่วยสำหรับการออกแบบและสร้างเอ็มดีเอ็กซ์แบบออนไลน์ โดยโปรแกรมจะทำหน้าที่ช่วยอำนวยความสะดวกในการออกแบบและสร้างเอ็มดีเอ็กซ์ เพื่อลดความผิดพลาดและเวลาที่ใช้ในการออกแบบและสร้างเอ็มดีเอ็กซ์ บทสรุปของงานวิจัยนี้ ได้สร้างโปรแกรมประยุกต์บนเว็บ (Web Application) เพื่อใช้เรียนรู้และออกแบบเอ็มดีเอ็กซ์ สร้างความสะดวกให้กับผู้ใช้ที่ไม่มีความชำนาญในการใช้เอ็มดีเอ็กซ์ที่เริ่มต้นออกแบบและสร้างเอ็มดีเอ็กซ์สำหรับการเรียกดูข้อมูลในลักษณะข้อมูลสรุปจากลูกบาศก์ข้อมูล

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อใช้ในการศึกษาลักษณะและโครงสร้างของเอ็มดีเอ็กซ์
2. เพื่อพัฒนาโปรแกรมประยุกต์บนบราวเซอร์ ใช้ออกแบบและสร้างเอ็มดีเอ็กซ์ ช่วยอำนวยความสะดวกแก่ผู้ใช้ที่ต้องการใช้งานข้อมูลเชิงวิเคราะห์จากลูกบาศก์ข้อมูล
3. เพื่ออำนวยความสะดวกในการเรียกดูข้อมูลจากลูกบาศก์ข้อมูล เพื่อมาใช้นำเสนอในรูปแบบข้อมูลเชิงวิเคราะห์
4. เพื่อให้สามารถทำงานกับลูกบาศก์ข้อมูลได้ง่ายและมีประสิทธิภาพมากขึ้น

1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้มีจุดมุ่งหมายที่จะพัฒนาการใช้ข้อมูลจากลูกบาศก์ข้อมูล นำเสนอในรูปแบบรายงานเชิงวิเคราะห์ด้วยการนำเสนอผ่านโปรแกรมประยุกต์บนบราวเซอร์ โดยที่ลดความยุ่งยากในการใช้เอ็มดีเอ็กซ์ ขอบเขตของงานจะประกอบด้วยการพัฒนาส่วนประกอบต่าง ๆ ดังนี้

1. ศึกษาลักษณะและโครงสร้างของเอ็มดีเอ็กซ์
2. ทำการพัฒนาโปรแกรมประยุกต์บนเว็บ MDX Generator ที่สามารถใช้ออกแบบและสร้างเอ็มดีเอ็กซ์
3. เทคโนโลยีที่ใช้พัฒนา ได้แก่ Multi Dimensional eXpressions, ASP.NET (C#), Microsoft SQL Server 2000 Service Pack 3, ADOMD.NET, Internet Information Service (IIS)
4. สามารถทำงานได้ดีบนเบราเซอร์ Internet Explorer 6.0 ขึ้นไป

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. สร้างแนวคิดเกี่ยวกับนำเสนอในลักษณะข้อมูลเชิงวิเคราะห์ และช่วยอำนวยความสะดวกแก่ผู้ใช้ที่เริ่มต้นใช้งานข้อมูลเชิงวิเคราะห์จากลูกบาศก์ข้อมูล
2. สามารถนำข้อมูลเชิงวิเคราะห์มาใช้ประโยชน์ได้หลากหลายมากขึ้น
3. สามารถนำโปรแกรมไปใช้พัฒนาต่อยอดโปรแกรมประยุกต์ทางธุรกิจที่ต้องการใช้การวิเคราะห์ข้อมูลแบบหลายมุมมอง (Multi Dimensions Thinking) ช่วยประกอบในการตัดสินใจดำเนินงานได้อย่างมีประสิทธิภาพ

บทที่ 2
ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะเป็นการนำเสนอวรรณกรรมและงานวิจัยที่เกี่ยวข้องกับงานวิจัยนี้ โดยในหัวข้อที่ 2.1 จะกล่าวถึงความหมายและบทบาทความสำคัญของระบบคลังข้อมูล หัวข้อที่ 2.2 จะกล่าวถึงฐานข้อมูลแบบหลายมิติ (Multi dimensional database) หรือลูกบาศก์ข้อมูล (Cube) หัวข้อที่ 2.3 จะกล่าวถึงการประมวลผลเชิงวิเคราะห์แบบออนไลน์ (On-Line Analytical Processing : OLAP) หัวข้อที่ 2.4 จะกล่าวถึงเอ็มดีเอ็กซ์ (Multi-Dimensional eXpressions : MDX) หัวข้อที่ 2.5 จะกล่าวถึง Decision Support Object (DSO) และ Active Data Object Multi-Dimensional (ADOMD.NET)

2.1 ความหมายและบทบาทความสำคัญของระบบคลังข้อมูล

Surajit Chaudhuri and Umeshwar Dayal (1996) เสนอ An Overview of Data Warehousing and OLAP Technology ได้อธิบายว่า คลังข้อมูล หมายถึงฐานข้อมูลขนาดใหญ่ขององค์กรหรือหน่วยงานหนึ่ง ๆ ซึ่งเก็บรวบรวมข้อมูลจากฐานข้อมูลระบบงานประจำวัน หรือเรียกอีกอย่างว่า operational database และฐานข้อมูลอื่นภายนอกองค์กร หรือเรียกว่า external database โดยข้อมูลที่ถูกจัดเก็บในคลังข้อมูลนั้น มีวัตถุประสงค์ในการนำมาใช้งานและมีลักษณะของการจัดเก็บแตกต่างไปจากข้อมูลในฐานข้อมูลระบบงานอื่น โดยข้อมูลในคลังข้อมูลจะถูกนำมาใช้เพื่อสนับสนุนการตัดสินใจบริหารงานของผู้บริหาร โดยเฉพาะการเป็นข้อมูลพื้นฐานให้กับระบบงาน เพื่อการบริหารงานอื่น เช่น ระบบ DSS และระบบ CRM เป็นต้น

2.1.1 คุณลักษณะเฉพาะของคลังข้อมูล

1. การแบ่งโครงสร้างตามเนื้อหา (Subject oriented) หมายถึงคลังข้อมูลถูกออกแบบมาเพื่อมุ่งเน้นไปในแต่ละเนื้อหาที่สนใจ ไม่ได้เน้นไปที่การทำงานหรือกระบวนการแต่ละอย่าง โดยเฉพาะเหมือนอย่างฐานข้อมูลปฏิบัติการในส่วนของรายละเอียดข้อมูลที่จัดเก็บในระบบทั้งสองแบบก็จะแตกต่างกันไปตามความต้องการใช้งานด้วยเช่นกัน คลังข้อมูลจะไม่จำกัดเก็บข้อมูลที่ไม่มีส่วนเกี่ยวข้องกับการประมวลผลเพื่อสนับสนุนการตัดสินใจ ในขณะที่ข้อมูลนั้นจะถูกเก็บไว้ในฐานข้อมูลปฏิบัติการถ้าหากฐานข้อมูลนั้นมีส่วนที่เกี่ยวข้องกับกระบวนการทำงานทั้งหมด

2. การรวมเป็นหนึ่ง (Integration) ซึ่งถือได้ว่าเป็นคุณลักษณะที่สำคัญที่สุดของคลังข้อมูล คือการรวบรวมข้อมูลจากหลายฐานข้อมูลปฏิบัติการเข้าด้วยกัน และทำให้ข้อมูลมีมาตรฐานเดียวกัน เช่นกำหนดให้มีค่าตัวแปรของข้อมูลในเนื้อหาเดียวกันให้เป็นแบบเดียวกันทั้งหมด

3. ความสัมพันธ์กับเวลา (Time variance) หมายถึง ข้อมูลในคลังข้อมูลจะต้องจัดเก็บโดยกำหนดช่วงเวลาเอาไว้ โดยจะสัมพันธ์กับการดำเนินธุรกิจของหน่วยธุรกิจนั้น เพราะในการตัดสินด้านการบริหารจำเป็นต้องมีข้อมูลเปรียบเทียบในแต่ละช่วงเวลา แต่ละจุดของข้อมูลจะเกี่ยวข้องกับจุดของเวลาและข้อมูลแต่ละจุดสามารถเปรียบเทียบกันได้ตามแกนของเวลา

4. ความเสถียรของข้อมูล (Nonvolatile) หมายถึง ข้อมูลในคลังข้อมูลจะไม่เปลี่ยนแปลงบ่อย ไม่ว่าจะเป็นการเพิ่มเติมข้อมูลใหม่ หรือการปรับปรุงแก้ไขข้อมูลเดิมที่บรรจุอยู่แล้ว ผู้ใช้ทำได้เพียงการเข้าถึงข้อมูลเท่านั้น

2.1.2 สถาปัตยกรรมคลังข้อมูล (Data Warehouse Architecture)

สถาปัตยกรรมคลังข้อมูลเป็นโครงสร้างมาตรฐานที่ใช้บ่อย เพื่อให้เข้าใจแนวคิดและกระบวนการของคลังข้อมูลซึ่งโดยทั่วไปแล้วคลังข้อมูลแต่ละระบบอาจจะมีรูปแบบที่ไม่เหมือนกันได้ เพื่อให้เหมาะสมกับแต่ละองค์กร ทั้งนี้ส่วนประกอบต่าง ๆ ภายในสถาปัตยกรรมคลังข้อมูลที่สำคัญได้แก่

1. Operational database หรือ external database layer ทำหน้าที่จัดการกับข้อมูลในระบบงานปฏิบัติการหรือแหล่งข้อมูลภายนอกองค์กร

2. Information access layer เป็นส่วนที่ผู้ใช้ปลายทางติดต่อผ่านโดยตรง ซึ่งประกอบด้วยฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการแสดงผลเพื่อวิเคราะห์โดยมีเครื่องมือช่วยเป็นตัวกลางที่ผู้ใช้ใช้ติดต่อกับคลังข้อมูล โดยในปัจจุบันเครื่องมือที่ได้รับความนิยมเพิ่มขึ้นอย่างรวดเร็วนั้นคือ Online Analytical Processing Tool หรือ OLAP tool ซึ่งเป็นเครื่องมือที่มีความสามารถในการวิเคราะห์ที่ซับซ้อน และแสดงข้อมูลในรูปแบบหลายมิติ

3. Data access layer เป็นส่วนต่อประสานระหว่าง Information access layer กับ operational layer

4. Data directory (metadata) layer เพื่อให้เข้าใจถึงข้อมูลได้ง่ายขึ้น และเป็นการเพิ่มความเร็วในการเรียกและดึงข้อมูลของคลังข้อมูล

5. Process management layer ทำหน้าที่จัดการกระบวนการทำงานทั้งหมด

6. Application messaging layer เป็นมิดเดิลแวร์ ทำหน้าที่ในการส่งข้อมูลภายในองค์กรผ่านทางเครือข่าย

7. Data warehouse (physical) layer เป็นแหล่งเก็บข้อมูลของทั้ง information data และ external data ในรูปแบบที่ง่ายแก่การเข้าถึงและยืดหยุ่นได้

8. Data staging layer เป็นกระบวนการการแก้ไข และดึงข้อมูลจาก external database

2.1.3 เทคนิคในการสร้างคลังข้อมูล

การเคลื่อนที่ของข้อมูลในคลังข้อมูล

ข้อมูลที่จัดเก็บภายในคลังข้อมูลมีการเคลื่อนที่ของข้อมูล (Information flow) 5 ประเภท ดังนี้

1. Inflow คือการนำข้อมูลจากฐานข้อมูลอื่นเข้าสู่คลังข้อมูลทั้งฐานข้อมูลภายในและภายนอกองค์กร โดยในขั้นนี้อาจมีการเปลี่ยนแปลงโครงสร้างข้อมูล การทำ De-Normalize การลบหรือการเพิ่มฟิลด์เพื่อให้ข้อมูลทั้งหมดอยู่ในเนื้อหาที่สนใจเดียวกัน ในขั้นตอนนี้อาจใช้เครื่องมือที่เรียกว่า data warehouse tool

2. Upflow เมื่อข้อมูลที่เราต้องการอยู่ในคลังข้อมูลแล้ว ในบางครั้งอาจต้องมีการเพิ่มคุณค่าให้กับข้อมูลด้วยเพื่อให้ข้อมูลอยู่ในรูปแบบที่เป็นประโยชน์มากที่สุดต่อการนำเครื่องมือมาใช้ ซึ่งได้แก่การจัดกลุ่มข้อมูลหาค่าทางสถิติที่ซับซ้อน จัดข้อมูลให้อยู่ในรูปแบบหรือเทมเพลตมาตรฐาน

3. Downflow เป็นขั้นตอนของการปรับปรุงเปลี่ยนแปลงข้อมูลเก่า และไม่อยู่ในเนื้อหาที่องค์กรสนใจออกไปจากคลังข้อมูลขององค์กร

4. Outflow เป็นขั้นตอนที่ผู้ใช้เรียกใช้ข้อมูลในคลังข้อมูลผ่านเครื่องมือต่าง ๆ โดยการเรียกใช้อาจมีเพียงขอเรียกเป็นครั้งคราวเป็นประจำทุกวันหรือประจำทุกเดือน หรือแม้กระทั่งต้องการแบบทันที

5. Metaflow ข้อมูลที่จัดเก็บในคลังข้อมูลจะถูกทำข้อมูลไว้อีกชุดหนึ่ง เป็นแหล่งที่มาของข้อมูลนั้น หรือแม้กระทั่งที่อยู่ของข้อมูลนั้นในคลังข้อมูลและข้อมูลอื่นที่เกี่ยวข้อง

วิธีการออกแบบฐานข้อมูลสำหรับคลังข้อมูล

Ralph Kimball (1996) ได้เสนอ Nine Decisions in the Design of a Data Warehouse เรียกว่าระเบียบวิธี 9 ขั้น หรือ Nine-Step Methodology โดยวิธีการนี้เริ่มจากการออกแบบจากส่วนย่อยที่แสดงถึงแต่ละระบบงานขององค์กร หรือเรียกอีกอย่างหนึ่งว่าดาต้ามาร์ท (data mart) โดยเมื่อออกแบบแต่ละส่วนสำเร็จแล้ว จึงนำมารวมกันเป็นคลังข้อมูล ขององค์กรในขั้นสุดท้าย ซึ่งขั้นตอนทั้ง 9 ขั้นตอน มีรายละเอียดดังนี้

1. กำหนดดาต้ามาร์ท คือการเลือกว่าจะสร้างดาต้ามาร์ทของระบบงานใดบ้าง และระบบงานใดเป็นระบบงานแรกโดยองค์กรจะต้องสร้าง E-R model ที่รวมระบบงานทุกระบบขององค์กรไว้ แสดงการเชื่อมโยงของแต่ละระบบงานอย่างชัดเจน และสิ่งที่ต้องคำนึงถึงในการเลือกระบบงานที่จะเป็นดาต้ามาร์ทแรกนั้น มี 3 ปัจจัยที่เกี่ยวข้อง ได้แก่ จะต้องสามารถพัฒนาออกมาได้ทันตามเวลาที่ต้องการ โดยอยู่ในงบประมาณที่กำหนดไว้และต้องตอบปัญหาทางธุรกิจให้แก่องค์กรได้ ดังนั้นดาต้ามาร์ทแรกควรจะเป็นของระบบงานที่นำรายได้เข้ามาสู่องค์กรได้ เช่น ระบบงานขาย เป็นต้น

2. กำหนด fact table ของดาต้ามาร์ท คือการกำหนดเนื้อหาหลักที่ควรจะเป็นของดาต้ามาร์ท โดยการเลือกเอนติตี้หลักและกระบวนการที่เกี่ยวกับเอนติตี้นั้นออกมาจาก E-R model ขององค์กร นั้นหมายถึงว่าจะทำให้เราทราบถึง dimension table ที่ควรจะมีด้วย

3. กำหนดแอตทริบิวต์ที่จำเป็นในแต่ละ dimension table คือการกำหนดแอตทริบิวต์ที่บอกหรืออธิบายรายละเอียดของ dimension ได้ ทั้งนี้แอตทริบิวต์ที่เป็น primary key ควรจะเป็นค่าที่คำนวณได้ กรณีที่มีดาต้ามาร์ทมากกว่าหนึ่งดาต้ามาร์ทมี dimension เหมือนกัน นั่นหมายถึงว่ามีค่าของแอตทริบิวต์ใน dimension นั้นจะต้องเหมือนกันทุกประการ นั้นไม่อาจจะแก้ไขปัญหาการจัดเก็บข้อมูลซ้ำซ้อน อันนำมาสู่ความแตกต่างกันของข้อมูลชุดเดียวกัน ปัญหานี้จึงเป็นการดีที่จะมีการใช้ dimension table ร่วมกันในแต่ละ fact table ที่จำเป็นต้องมี dimension ดังกล่าว โดยเรียก dimension table ลักษณะแบบนี้ว่า conformed และเรียก fact table ว่า fact constellation เราสามารถกำหนดข้อดีของการใช้ dimension table ร่วมกันได้ดังนี้

(1) มั่นใจได้ว่าในแต่ละรายงานจะออกมาสอดคล้องกัน

(2) สามารถสร้างดาต้ามาร์ทในเวลาต่าง ๆ กันได้

(3) สามารถเข้าถึงดาต้ามาร์ทโดยผู้พัฒนากลุ่มอื่น ๆ

(4) สามารถรวบรวมดาต้ามาร์ทหลาย ๆ อันเข้าด้วยกัน

(5) สามารถออกแบบคลังข้อมูลร่วมกันได้

4. กำหนดแอตทริบิวต์ที่จำเป็นใน fact table โดยแอตทริบิวต์หลักใน fact table จะมาจาก primary key ในแต่ละ dimension table นอกจากนี้แล้ว ยังสามารถมีแอตทริบิวต์ที่จำเป็นอื่น ๆ ประกอบอยู่ด้วย เช่น แอตทริบิวต์ที่ได้จากการคำนวณค่าเบื้องต้นที่จำเป็นสำหรับการคงอยู่ของแอตทริบิวต์อื่นใน fact table เรียกอีกอย่างหนึ่งว่า measure การกำหนดแอตทริบิวต์นี้ไม่ควรจะเลือกแอตทริบิวต์ที่คำนวณไม่ได้ เช่น เป็นตัวหนังสือหรือไม่ใช่ตัวเลข เป็นต้น และไม่ควรที่จะเลือกเอาแอตทริบิวต์ที่ไม่เกี่ยวข้องกับเนื้อหาของ fact table ที่สนใจด้วย

5. จัดเก็บค่าการคำนวณเบื้องต้นใน fact table คือการจัดเก็บที่ได้จากการคำนวณให้เป็นแอตทริบิวต์ใน fact table ถึงแม้ว่าจะสามารถหาค่าได้จากแอตทริบิวต์อื่น ๆ ก็ตาม ทั้งนี้เพื่อให้

6. เขียนคำอธิบาย dimension table ทั้งนี้ก็เพื่อให้ผู้ใช้สามารถใช้งานดาต้ามาร์ทได้อย่างมีประสิทธิภาพเพราะเกิดความเข้าใจอย่างดีในส่วนต่าง ๆ

7. กำหนดระยะเวลาในการจัดเก็บข้อมูลในฐานข้อมูล โดยอาจจะเป็นการจัดเก็บเพียงช่วงระยะเวลา 1-2 ปี หรือนานกว่านั้น ขึ้นอยู่กับความต้องการขององค์กร เนื่องจากองค์กรแต่ละประเภทมีความต้องการในการจัดเก็บข้อมูลต่างช่วงเวลากัน ทั้งนี้ขึ้นอยู่กับความจำเป็นหรือข้อกำหนดในการดำเนินธุรกิจมีข้อสังเกตอยู่ 2 ประการที่น่าสนใจและสำคัญสำหรับการออกแบบแอตทริบิวต์ในเรื่องของการจัดเก็บข้อมูล ดังนี้

(1) ข้อมูลที่ถูกจัดเก็บไว้นานเกินไปมักเกิดปัญหาการอ่าน หรือแปลข้อมูลนั้น ๆ จากแฟ้มหรือเทปเก่า

(2) เมื่อมีการนำรูปแบบเก่าของ dimension table มาใช้อาจเกิดปัญหาการเปลี่ยนแปลงของ dimension อย่างช้า ๆ ได้

8. การติดตามปัญหาการเปลี่ยนแปลงของ dimension อย่างช้า ๆ คือ การเปลี่ยนเอาแอตทริบิวต์ของ dimension table เก่ามาใช้แล้วส่งผลกระทบต่อข้อมูลปัจจุบันของ dimension table โดยสามารถแบ่งประเภทของปัญหาที่เกิดได้ 3 ประเภท ดังนี้

(1) เกิดการเขียนทับข้อมูลใหม่โดยข้อมูลเก่า

(2) เกิดเรคอร์ดใหม่ ๆ ขึ้นใน dimension

(3) เกิดเรคอร์ดที่มีทั้งค่าเก่าและใหม่ปนกันไป

9. กำหนด query เป็นการออกแบบด้านกายภาพเพื่อให้ผู้ใช้เกิดความสะดวกในการใช้งานและสามารถทำงานได้อย่างมีประสิทธิภาพเมื่อดำเนินการทั้ง 9 ขั้นตอนสำหรับแต่ละดาต้ามาร์ทเสร็จแล้วจึงจะนำทั้งหมดมารวมกันเป็นภาพของคลังข้อมูลขององค์กรต่อไป

การแปลงข้อมูลเข้าสู่ดาต้ามาร์ท

เมื่อออกแบบฐานข้อมูลสำหรับแต่ละดาต้ามาร์ทเสร็จแล้ว ขั้นตอนต่อไปที่สำคัญยิ่งก็คือการนำข้อมูลจากแหล่งข้อมูลไปแปลงให้อยู่ในแพลตฟอร์มของฐานข้อมูลที่ได้ออกแบบไว้ นั่นก็คือการแปลงข้อมูลหรือ Extraction Transformation and Loading (ETL) โดยที่คุณภาพของการแปลงข้อมูลเป็นสิ่งสำคัญมากสำหรับการสร้างคลังข้อมูล จะแตกต่างกันไปตามคลังข้อมูลที่แต่ละองค์กรต้องการ โดยที่การแปลงข้อมูลหมายรวมตั้งแต่การวิเคราะห์แหล่งข้อมูล กำหนดการส่งข้อมูล รวบรวมหรือสร้างข้อมูลภายนอก วางแผนและสร้างเส้นทางของการแปลงข้อมูล และตรวจสอบความถูกต้องของข้อมูลที่ได้สามารถสรุปเป็นขั้นตอนได้ดังนี้

1. วิเคราะห์แหล่งข้อมูล เช่น ปริมาณของข้อมูล จำนวนและชนิดของการเข้าถึงแหล่งข้อมูล แพลตฟอร์มและภาษาโปรแกรมที่ใช้ เป็นต้น

2. ย้ายข้อมูลที่ต้องการจากระบบเดิมมาไว้ในบริเวณที่ใช้ปรับแต่งข้อมูล หรือเรียกบริเวณนี้ว่า staging area เพื่อนำมาเลือกเฉพาะส่วนที่ต้องการแปลงข้อมูลและตรวจสอบความถูกต้อง หรือการทำความสะอาดข้อมูล

3. กำหนด primary key ของ fact table และ dimension table และกำหนด foreign key ระหว่าง fact table กับ dimension table

4. ย้ายข้อมูลที่ทำความสะอาดแล้วจาก staging area ลงสู่เซิร์ฟเวอร์ของดาต้ามาร์ท

5. สร้าง metadata ของแต่ละดาต้ามาร์ท โดยเก็บรายละเอียดของข้อมูลการปรับปรุงและส่งออกไปไว้ในดาต้ามาร์ท

6. ตรวจสอบความถูกต้องของข้อมูล ซึ่งจะต้องกระทำตลอดทั้งกระบวนการแปลงข้อมูล ทำได้ดังนี้

(1) ตรวจสอบผลรวมทั้งหมดของจำนวนข้อมูลที่ดึงมาจากแหล่งข้อมูลที่เพิ่มเข้าไป

(2) ตรวจแก้ข้อมูลในระบบเดิมของแหล่งข้อมูล หรือในเส้นทางของการแปลง ซึ่งควรจะเก็บข้อมูลในการตรวจแก้ไว้ใน metadata ของการแปลงข้อมูลด้วย

(3) ตรวจสอบค่าของข้อมูลให้ถูกต้องในกระบวนการรวบรวมข้อมูล

(4) ตรวจสอบผลรวมของข้อมูลหลังจากย้ายข้อมูลลงสู่ดาต้ามาร์ทแล้ว

2.2 ฐานข้อมูลแบบหลายมิติ (Multi Dimensional Databases)

ฐานข้อมูลแบบมิติเป็นฐานข้อมูลที่ใช้ในการจัดเก็บและจัดการข้อมูลให้สามารถแสดงผลตามมิติของข้อมูลต่าง ๆ ได้ พัฒนาต่อจากฐานข้อมูลเชิงสัมพันธ์ให้มีความสามารถขึ้นเพื่อรองรับการประมวลเชิงวิเคราะห์ หรือ OLAP นั่นเอง มีผู้ผลิตหลายรายที่นำเสนอ Multidimensional databases หรือ MDBs ซึ่งเก็บข้อมูลในรูปแบบที่เรียกว่า ลูกบาศก์ข้อมูล (Cube) ลูกบาศก์ข้อมูลถูกสร้างโดยเป็นการรวมกันของแอตทริบิวต์ของ dimension ที่มีการคำนวณค่าไว้เรียบร้อยแล้ว

2.2.1 องค์ประกอบของฐานข้อมูลแบบหลายมิติ

องค์ประกอบของฐานข้อมูลแบบหลายมิติประกอบไปด้วยตารางที่สำคัญ 2 ประเภท คือ

1. Fact table เป็นตารางหลักของฐานข้อมูลแบบหลายมิติซึ่งมีลักษณะคล้ายกันกับตารางประเภททรานแซคชั่นของ OLTP โดยส่วนใหญ่แล้วจะมีขนาดมากกว่า 80 เปอร์เซนต์ของข้อมูลทั้งหมด โดยภายในประกอบด้วยคอลัมน์ที่สำคัญ 2 ประเภท คือ

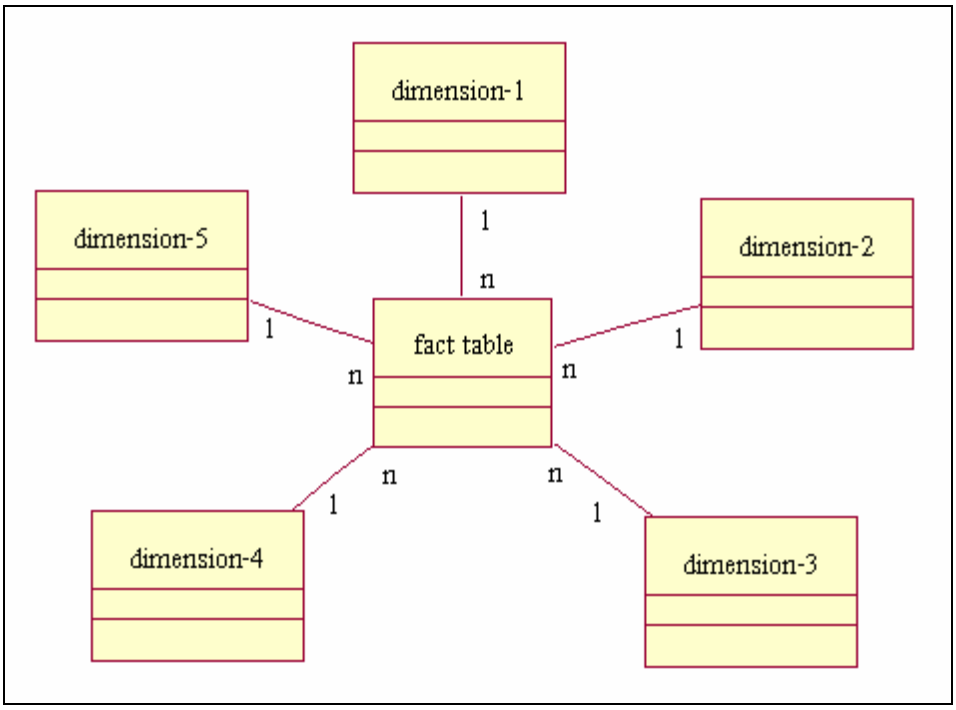
- Fact table key เป็นคอลัมน์ที่ใช้เชื่อมโยงข้อมูลไปยัง dimension table ต่าง ๆ ดังนั้นจำนวนคอลัมน์ของ fact table key จะเพิ่มขึ้นตามจำนวนของ dimension table ไปโดยปริยาย และคอลัมน์ทั้งหมดนี้ยังสามารถนำมาใช้สร้างให้เป็น primary key ของตารางอีกด้วย

- Measure เป็นคอลัมน์ที่มีประเภทของข้อมูลเป็นตัวเลข ทำหน้าที่เก็บจำนวนหรือปริมาณที่เกิดขึ้นของแต่ละทรานแซคชั่น

2. Dimension Tables เป็นตารางของแกนต่าง ๆ ของฐานข้อมูลแบบหลายมิติโดยภายในประกอบด้วยคอลัมน์ที่ใช้เป็นคีย์ เพื่อเชื่อมโยงไปยัง fact table key และคอลัมน์ที่ให้ความหมายเพิ่มเติมแก่เอนติตี้

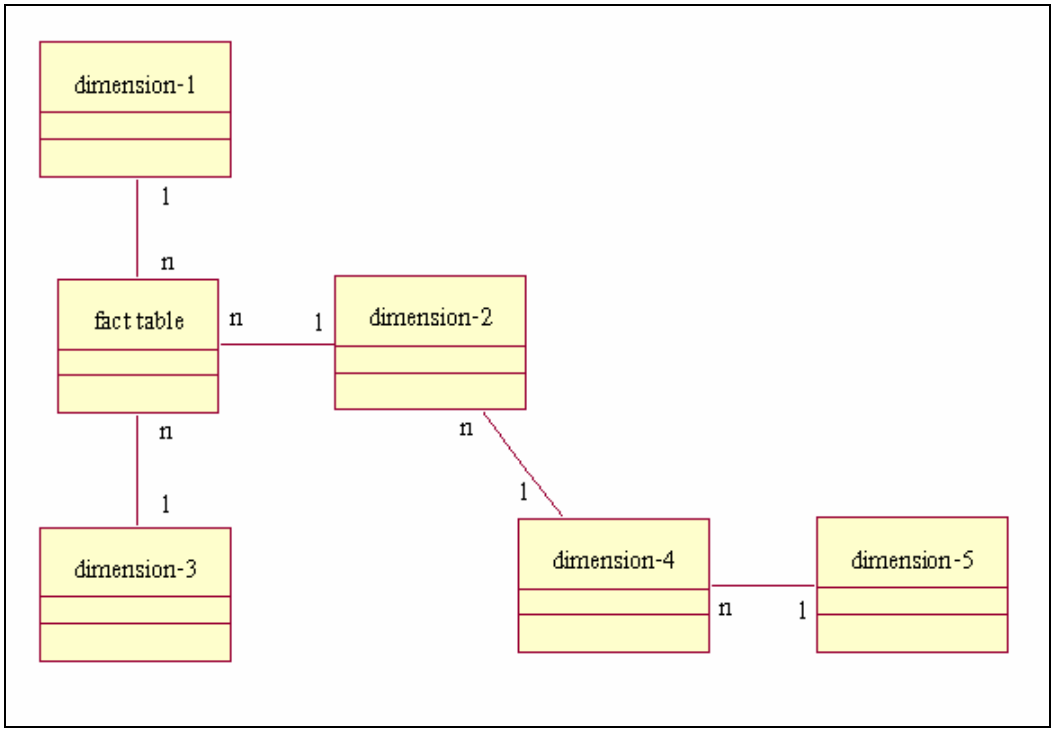
2.2.2 โครงสร้างของฐานข้อมูลแบบหลายมิติ (Multi dimensional database schema)

เทคนิคที่ใช้มี 2 รูปแบบ คือโครงสร้างแบบดาวและโครงสร้างแบบเกล็ดหิมะ โครงสร้างแบบดาว (Star schema) จะมีลักษณะที่มี fact table อยู่ตรงกลางล้อมรอบด้วย dimension table ที่เก็บรายละเอียดของ fact ที่ไม่ใช่ normalized center ดังรูปที่ 2.1



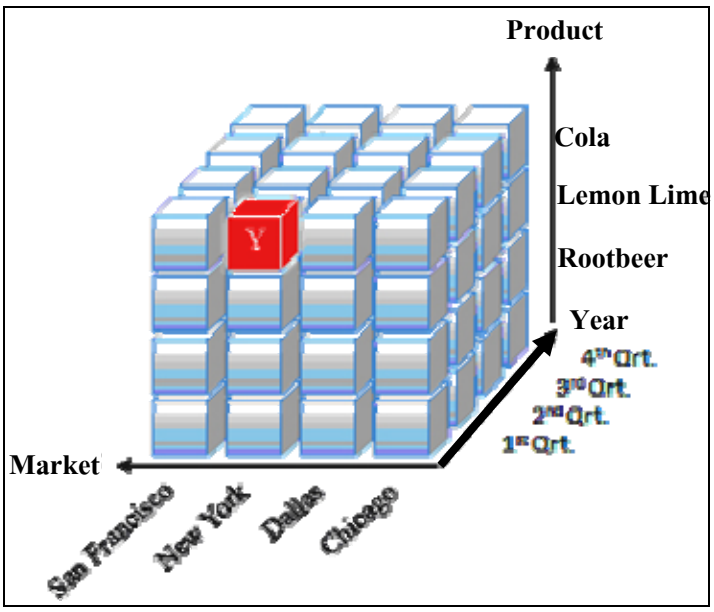
รูปที่ 2.1 โครงสร้างแบบดาว

โครงสร้างแบบเกล็ดหิมะ (Snowflake schema) มีความแตกต่างจาก Star schema ตรงที่ dimension table จะเก็บข้อมูลที่อยู่ในรูป normal form จากโครงสร้างแบบดาวอาจจะเกิดปัญหาในการออกแบบได้ โครงสร้างแบบเกล็ดหิมะจึงเกิดขึ้นมาเนื่องจากปัจจัยดังนี้ แสดงได้ดังรูปที่ 2.2



รูปที่ 2.2 โครงสร้างแบบเกล็ดหิมะ

ถึงแม้ว่าการยกตัวอย่างเพื่อให้มองเห็นภาพและทำความเข้าใจได้ง่ายนั้นจะสามารถแสดงข้อมูลเพื่ออธิบายเพียง 3 มิติ ดังรูปที่ 2.3 โดยแต่ละแกนเป็นตัวแทนของตารางแกนมิติซึ่งในที่นี้คือ Market Dimension, Product Dimension, Year Dimension ตามลำดับ ส่วนข้อมูลที่อยู่ในแต่ละ Cell คือค่าที่ได้จากการคำนวณของแต่ละแกนมิติ ตัวอย่างเช่น หากลูกบาศก์ข้อมูลเป็นที่เก็บข้อมูลของจำนวนสินค้าที่ขายได้ ดังนั้นหากต้องการทราบสินค้า Cola ที่ขายได้ในช่วงไตรมาสที่ 1 (1st Qtr.) ของ Market New York จะได้คำตอบเป็นรูปลูกบาศก์ที่ถูกแรเงาไว้ ซึ่งเกิดจากการแต่ละแกนของมิติมาตัดกัน



รูปที่ 2.3 แบบจำลองของลูกบาศก์ข้อมูล OLAP

ลูกบาศก์ข้อมูล OLAP สามารถประกอบกันเพื่อแสดงข้อมูลได้มากถึง 64 dimensions สำหรับข้อจำกัดอื่น ๆ สำหรับ Multi dimensional database ที่ Analysis Services ของ Microsoft SQL Server 2000 สามารถรองรับได้ เป็นดังตารางที่ 2.1

ตารางที่ 2.1 ความสามารถและข้อจำกัดของ Analysis Service ใน Microsoft SQL Server 2000

ความสามารถ	ข้อจำกัด
จำนวน Database ต่อ Server	ไม่จำกัด
จำนวน Cube ต่อ Database	ไม่จำกัด
จำนวน Cube ต่อ Virtual Cube	64
จำนวน Dimension ต่อ Cube	128
จำนวน Dimension ต่อ Database	65,535
จำนวน Measure ต่อ Cube	1,024
จำนวน Calculated member ต่อ Cube	65,535
จำนวน Level ต่อ Dimension	64

2.3 On-Line Analytical Processing : OLAP

John Horner, Il-Yeol Song, and Peter P.Chen (2004) เสนอ An Analysis of Additivity in OLAP Systems ได้อธิบายถึง OLAP หรือการประมวลผลเชิงวิเคราะห์แบบออนไลน์ว่าคือ กระบวนการประมวลผลทางคอมพิวเตอร์ที่ช่วยให้ผู้ใช้สามารถวิเคราะห์ข้อมูลในมิติต่าง ๆ ได้ง่ายยิ่งขึ้น การดำเนินการกับ OLAP เนื่องจากฐานข้อมูลแบบหลายมิติถูกคิดค้นขึ้นมาเพื่อรองรับการเรียกใช้ข้อมูลเชิงวิเคราะห์ ซึ่งจะต้องมีการดำเนินการกับข้อมูลในรูปแบบต่าง ๆ ดังนี้

1. Roll up and Drill Down คือการเปลี่ยนแปลงระดับความละเอียดของการพิจารณาข้อมูล โดยที่ Roll up และ Drill Down มีความแตกต่างกัน ดังนี้

- Drill Down คือการเพิ่มความละเอียดในการพิจารณาข้อมูลจากระดับที่หยาบไปสู่ระดับที่ละเอียดมากขึ้น

- Roll up หรือเรียกอีกอย่างหนึ่งว่า Consolidation คือกระบวนการตรงกันข้ามกับ Drill Down หมายถึงการเปลี่ยนแปลงระดับความละเอียดของการพิจารณาข้อมูล จากระดับที่ละเอียดขึ้นมาสู่ระดับที่หยาบมากขึ้น

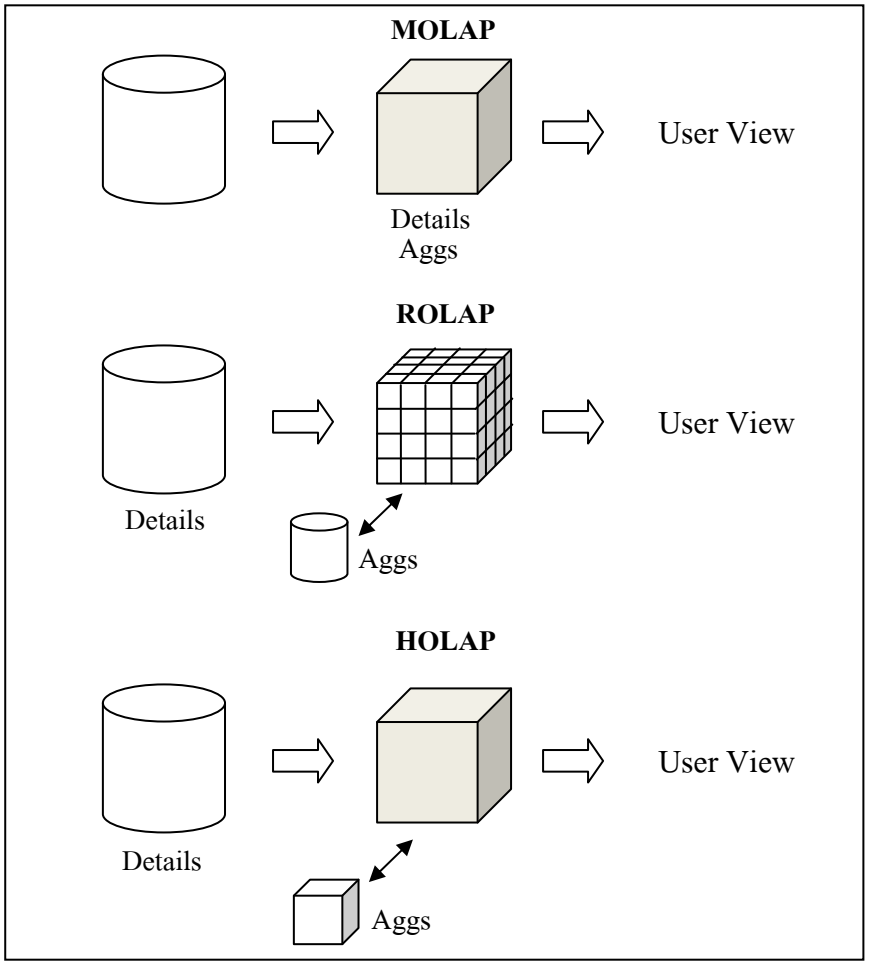
2. Slice and Dice หรือบางครั้งอาจจะเรียกได้อีกอย่างหนึ่งว่า “Pivoting” คือการแยกข้อมูลออกเป็นส่วนเพื่อพิจารณาเฉพาะส่วนที่ต้องการโดยเฉพาะโดยเกณฑ์ในการแยกจะใช้ข้อมูลของมิติใด ๆ

- Slice คือการเลือกพิจารณาผลลัพธ์บางส่วนที่เราสนใจ โดยการเลือกเฉพาะค่าที่ถูกกำกับด้วยข้อมูลบางค่าของแต่ละมิติเท่านั้น

- Dice คือการพลิกหรือหมุนของแกนมิติข้อมูล ให้ตรงตามความต้องการของผู้ใช้งาน

2.3.1 รูปแบบการเก็บข้อมูลของ OLAP (OLAP storage)

ผลลัพธ์ที่ได้จากคำนวณของลูกบาศก์ข้อมูลจะถูกเก็บไปยัง OLAP Storage เพื่อสามารถนำกลับมาใช้ได้ตลอดเวลา สำหรับเครื่องมือที่ใช้ในงานวิจัยนี้คือ Microsoft SQL Server 2000 Analysis Services สนับสนุนรูปแบบการข้อมูล ของ OLAP 3 ประเภท มีลักษณะของรูปแบบอธิบายได้ด้วยรูปที่ 2.4



รูปที่ 2.4 ประเภทของ OLAP Storage

1. Multi-Dimensional OLAP (MOLAP) Tatsuo Tsuji, Akihiro Hara and Ken Higuchi (2006) เสนอ An Extendible Multidimensional Array System for MOLAP กล่าวว่า MOLAP เป็นรูปแบบการเก็บข้อมูลของ OLAP ที่มีประสิทธิภาพสูง เนื่องจากข้อมูลทั้งที่เป็น details และ aggregates ถูกเก็บให้อยู่ในลักษณะของ Multi-Dimensional ดังนั้น จึงทำให้สามารถตอบ query ได้รวดเร็วที่สุด นอกจากนี้ยังได้ประโยชน์ในแง่ของระบบจัดเก็บข้อมูลที่มีอัลกอริทึมของการบีบอัดที่มีประสิทธิภาพ แต่ต้องเสียพื้นที่ฮาร์ดดิสก์เพิ่มขึ้น เนื่องจากเป็นการนำข้อมูลจากคลังข้อมูลมาเก็บไว้บน Analysis Services อีกที เสมือนเป็นการซ้ำซ้อนของข้อมูลอีกทางหนึ่ง

2. Relational OLAP (ROLAP) Konstantinos Morfonios, Stratis Konakas, Yannis Ioannidis and Nikolaos Kotsis (2007) เสนอ ROLAP Implementations of the Data Cube ได้กล่าวถึง ROLAP ว่าเป็นการกำหนดโครงสร้างของข้อมูลไว้บน Analysis Services เท่านั้น แต่

3. Hybrid OLAP (HOLAP) Owen Kaser and Daniel Lemire (2003) เสนอ Attribute Value Reordering for Efficient Hybrid OLAP ได้กล่าวถึง HOLAP ว่าเป็นการผสมผสานระหว่าง MOLAP และ ROLAP กล่าวคือข้อมูลของ details จะยังคงอยู่ที่ฐานข้อมูลเชิงสัมพันธ์ ส่วนข้อมูลที่เป็น aggregates จะถูกสร้างให้อยู่ในลักษณะของ Multi-Dimensional ดังนั้นจึงทำให้สามารถตอบสนองต่อ query ได้ในระดับปานกลาง ในขณะที่เสียพื้นที่น้อยกว่า MOLAP ส่วนเรื่องของเวลาที่ใช้ในการประมวลผลจะพอ ๆ กับ MOLAP เนื่องจากจำนวนข้อมูลจะถูกโหลดเข้าสู่หน่วยความจำในปริมาณที่เท่า ๆ กัน ดังนั้นการอ่านหรือเขียนข้อมูลจึงเป็นไปในลักษณะเดียวกัน

2.3.2 ประโยชน์ของ OLAP

การนำระบบประมวลผลเชิงวิเคราะห์แบบออนไลน์มาใช้จะช่วยให้ระบบสนับสนุนการตัดสินใจทำงานได้ดียิ่งขึ้น ดังต่อไปนี้

1. ช่วยในวิเคราะห์และเปรียบเทียบข้อมูลในมุมมองต่าง ๆ ทำให้ผู้ตัดสินใจมีมุมมองเกี่ยวกับการเปรียบเทียบข้อมูลมากยิ่งขึ้น ซึ่งเป็นการเพิ่มประสิทธิภาพในการตัดสินใจ
2. ช่วยให้ผู้ใช้ (ผู้ตัดสินใจ) สามารถคัดเลือกข้อมูลสำหรับตัดสินใจได้อย่างมีประสิทธิภาพ เนื่องจากระบบประมวลผลเชิงวิเคราะห์แบบออนไลน์ได้ทำการวิเคราะห์และเปรียบเทียบข้อมูลในมุมองต่าง ๆ ที่หลากหลาย
3. ช่วยให้ผู้ใช้แต่ละคนสามารถสร้างข้อมูลตามมุมมองของตนเองได้ เพื่อนำข้อมูลเหล่านั้นไปใช้งานเฉพาะด้าน
4. สามารถสอบถามข้อมูลได้อย่างรวดเร็ว แม้ในฐานข้อมูลที่มีขนาดใหญ่และมีความซับซ้อนมากก็ตาม

เนื่องจากความสามารถในการเปลี่ยนมุมมองในการเปรียบเทียบข้อมูลจากฐานข้อมูลแบบหลายมิติของการประมวลผลข้อมูลเชิงวิเคราะห์แบบออนไลน์ ทำให้ได้รับข้อมูลในมุมมองใหม่ ๆ สำหรับประกอบการตัดสินใจ ซึ่งเป็นการเพิ่มทางเลือกในการตัดสินใจมีประสิทธิภาพมากยิ่งขึ้น

2.4 Multi Dimensional eXpressions (MDX)

George Spofford (2001) กล่าวไว้ใน MDX Solution With Microsoft SQL Server Analysis Service ว่างานในเชิงวิเคราะห์สามารถใช้ฐานข้อมูลเชิงสัมพันธ์และ SQL ทำงานได้ แต่ SQL นั้นมีความยุ่งยากและประมวลผลช้าสำหรับงานวิเคราะห์เชิงมิติ เอ็มดีเอ็กซ์เป็นภาษาที่ช่วยปลดล็อคความสามารถของฐานข้อมูล OLAP เอ็มดีเอ็กซ์ถูกสร้างโดยบริษัทไมโครซอฟต์และเริ่มใช้เมื่อปี ค.ศ. 1998 เอ็มดีเอ็กซ์ได้รับการตอบรับอย่างรวดเร็วและถูกนำไปใช้ในวงการโปรแกรมประยุกต์ในงานเชิงวิเคราะห์ เช่น Applix, Microsoft, Microstrategy, SAS, SAP และ Whitelight เป็นต้น เอ็มดีเอ็กซ์ ถูกกำหนดให้เป็นภาษาส่วนประกอบของ OLE DB (Object Linking and Embedding, Database) สำหรับทำงานกับ OLAP อยู่ในส่วนของ COM (Component Object Model) interfaces ซึ่งโปรแกรมจะต้องติดต่อผ่านตัวจัดการข้อมูล OLAP ใช้เป็น query สำหรับการเรียกดูข้อมูลจากลูกบาศก์ข้อมูล ภาพรวมของ OLE DB สำหรับ OLAP ประกอบด้วยโครงสร้างข้อมูลและข้อกำหนดที่ใช้ query เพื่อการแลกเปลี่ยนข้อมูลที่ทำงานกับตัวจัดการข้อมูล

2.4.1 เอ็มดีเอ็กซ์อย่างง่าย (Simple MDX query)

กำหนดข้อมูลตัวอย่างของ sampleCube ดังตารางที่ 2.2 ข้อมูลร้านค้าประกอบด้วยมิติข้อมูลดังต่อไปนี้

1. มิติร้านค้า (Stores dimension) มีสมาชิก Store-A และ Store-B
2. มิติเวลา (Time dimension) มีสมาชิก June-2007 July-2007 และ August-2007
3. ค่ายอดขายและต้นทุน (Sales and Costs : Measure)

ตารางที่ 2.2 ข้อมูลยอดขายและต้นทุนของร้านค้า

Stores	Time	Sales	Costs
Store-A	June-2007	1200	1000
Store-A	July-2007	1300	1000
Store-A	August-2007	1300	1100
Store-B	June-2007	1000	800
Store-B	July-2007	1000	900
Store-B	August-2007	1300	1000

ตัวอย่าง เอ็มดีเอ็กซ์ที่ใช้เรียกดูข้อมูลโดยมีการระบุเงื่อนไขข้อมูล ผลลัพธ์ของเอ็มดีเอ็กซ์นี้ สามารถแสดงได้ดังตารางที่ 2.3

```
SELECT
  { [Measures].[Sales], [Measures].[Costs] } on columns,
  { [Time].[June-2007], [Time].[July-2007] } on rows
FROM  sampleCube
WHERE ( [Stores].[Store-A] )
```

ตารางที่ 2.3 เซ็ตข้อมูลผลลัพธ์ของเอ็มดีเอ็กซ์ตัวอย่าง

Store: Store-A		**Measures**	
Time		**Sales**	**Costs**
	June-2007	1200	1000
	July-2007	1300	1000

Tapio Niemi, Jyrki Nummenmaa, and Peter Thanisch (2001) ได้เสนอ Constructing OLAP Cubes Based on Queries เป็นเทคนิคการสร้างลูกบาศก์ข้อมูลอัตโนมัติจาก query

Tapio Niemi and Marko Niinimaki (2002) ได้เสนอ Constructing an OLAP Cube from Distributed XML Data การสร้างลูกบาศก์ข้อมูลจากแหล่งข้อมูลในรูปแบบ XML

Kazi A.Zaman and Donovan A. Schneider (2005) ได้นำเสนอ Modeling and Querying Multidimensional Data Sources in Siebel Analytics: A Federated Relational System เป็นการอธิบายถึง model และ query ของแหล่งข้อมูลแบบหลายมิติที่ทำงานกับระบบวิเคราะห์ของ Siebel ที่นำโปรโตคอล XMLA มาใช้เป็นมาตราฐานในการทำงาน

จากที่ทั้งหมดได้กล่าวถึงโครงสร้างของเอ็มดีเอ็กซ์ว่าโดยพื้นฐานส่วนใหญ่ของเอ็มดีเอ็กซ์จะมีรูปแบบโครงสร้างอย่างง่าย ๆ ประกอบด้วย

```
SELECT [<axis specification> [, <axis specification>, …]]
FROM [<cube specification>]
[WHERE <slicer specification>]
```

อนุประโยค SELECT ถูกใช้ในการกำหนดแกนมิติ (Axis-dimension) และอนุประโยค WHERE ถูกใช้ในการกำหนดแกนการซอยข้อมูล (Slicer dimensions) ส่วนอนุประโยค FROM คือใส่ชื่อของลูกบาศก์ข้อมูลที่ต้องการจะเรียกข้อมูล เอ็มดีเอ็กซ์มีลักษณะคล้ายกับ SQL ที่มีอนุประโยค SELECT FROM และ WHERE ในส่วนของการเรียกดูข้อมูลมีความแตกต่าง คือ SQL อนุญาตให้เลือกเฉพาะส่วนคอลัมน์ของตารางข้อมูล ขณะที่ผลลัพธ์ query ของเอ็มดีเอ็กซ์สามารถเลือกหลายมิติหรือเป็นการรวมของมิติโดยจะแสดงในลักษณะแกนข้อมูล จากที่กล่าวมาสามารถสรุปได้ 5 ข้อที่สำคัญ ดังนี้

1. เอ็มดีเอ็กซ์ใช้เครื่องหมายปีกกา ({}) ในการเปิดปิดเซ็ตของข้อมูลเป็นส่วนของมิติ
2. ในเอ็มดีเอ็กซ์ต้องระบุว่าจะให้มิติต่าง ๆ จากฐานข้อมูลนั้นแสดงในแนวแกนใด 3 แกนที่ใช้คือ คอลัมน์ (columns) แถว (rows) และหน้า (pages)
3. คอลัมน์จะต้องมาก่อนแถวเสมอ และแถวจะต้องมาก่อนหน้าเสมอเช่นกัน
4. อนุประโยค FROM ในเอ็มดีเอ็กซ์ใช้ระบุแหล่งข้อมูลคือลูกบาศก์ข้อมูล จะคล้ายกับ FROM ของ SQL ที่ใช้ระบุชื่อตารางข้อมูล
5. อนุประโยค WHERE ใช้ในการกำหนดเงื่อนไขของข้อมูลที่ต้องการ โดยสามารถระบุในแกนคอลัมน์ แถว หรือแกนอื่น ๆ

2.4.2 แบบจำลองข้อมูลของเอ็มดีเอ็กซ์ (MDX Data Model)

เอ็มดีเอ็กซ์ใช้แบบจำลองข้อมูลที่อ้างถึงมิติและลูกบาศก์ข้อมูล การทำความเข้าใจกับแบบจำลองข้อมูลจะทำให้ดึงความสามารถที่มีของงานวิเคราะห์ออกมาใช้ได้อย่างมีประสิทธิภาพ และทำให้เข้าใจว่าตัวดำเนินการพื้นฐานต่าง ๆ มีวิธีทำงานอย่างไร โดยแบ่งได้ดังนี้

- Tuples อ้างถึงการ slice ของลูกบาศก์ข้อมูล เป็นการรวมกันของสมาชิกของบางมิติ ด้วยสมาชิก 1 สมาชิกของแต่ละมิติ single member เป็น tuple แบบง่าย เช่น [Time].[June-2007] tuple ที่มีการรวมระหว่าง store member กับ time member เช่น ([Stores].[Store-A], [Time].[June-2007])

- Sets หรือเรียกว่า Collections คือกลุ่มของ tuples sets นั้นอาจจะมี tuple มากกว่า 1 tuple หรือมีแค่ 1 tuple หรือไม่มีเลย จะไม่เหมือนกับเซ็ตในทางคณิตศาสตร์ เซ็ตของเอ็มดีเอ็กซ์สามารถมี tuple เดียวกันซ้ำได้ ดังนั้นการเรียงลำดับมีความสำคัญมาก รูปแบบการเขียนถูกกำหนดไว้มีไม่มาก โดยทั่ว ๆ ไปเซ็ตถูกกำหนดให้ปิดโดยเครื่องหมายปีกกา เช่น {[Time].[June-2007], [Time].[July-2007]} เมื่อนำไปใช้สามารถแสดงได้ดังตัวอย่างต่อไปนี้ และผลลัพธ์ดังตารางที่ 2.4

```
SELECT
{  ([Time].[June-2007], [Stores].[Store-A]),
   ([Time].[July-2007], [Stores].[Store-B])
} on columns
FROM sampleCube
WHERE ([Measures].[Costs])
```

ตารางที่ 2.4 ผลลัพธ์จากเซ็ต query

	Time Stores	
	June-2007	**July-2007**
Measure: Costs	**Store-A**	**Store-B**
	1000	900

2.4.3 การสร้างเอ็มดีเอ็กซ์อย่างง่าย (Simple MDX Construction)

ในหัวข้อที่แล้วได้รู้พื้นฐานของ Tuples และ Sets แล้ว สำหรับหัวข้อนี้จะอธิบายวิธีการสร้างเอ็มดีเอ็กซ์ ตัวดำเนินการ (operator) และฟังก์ชัน (function) ที่ใช้เป็นประจำของเอ็มดีเอ็กซ์ที่ใช้มีดังนี้

- เครื่องหมายจุลภาค (,) และโคลอน (:)

Tuples ต่าง ๆ สามารถใช้เครื่องหมายจุลภาคในการแบ่งเซ็ต ดังตัวอย่าง

{[Time].[June-2007], [Time].[July-2007], [Time].[August-2007]}

จากตัวอย่าง ผลลัพธ์ที่ได้คือข้อมูลในเดือนมิถุนายน, เดือนกรกฎาคม, และเดือนสิงหาคม ค.ศ. 2007

ถ้าสมาชิกที่อยู่ในระดับเดียวกันสามารถใช้เครื่องหมายโคลอนได้ โดยจะมีความหมายว่าทุกสมาชิกที่อยู่ระหว่างสมาชิกทั้งสอง เป็นความหมายเดียวกันกับที่ใช้ในโปรแกรม Excel ดังตัวอย่าง

{[Time].[June-2007] : [Time].[August-2007]}

จากตัวอย่าง ผลลัพธ์ที่ได้คือข้อมูลที่อยู่ในช่วงเดือนมิถุนายนถึงสิงหาคม ค.ศ. 2007

. **- Members**

การดึงเซ็ตข้อมูลของสมาชิกในมิติ (dimensions) ลำดับชั้น (hierarchies) หรือระดับ (levels) เป็นจุดเริ่มต้นโดยทั่วไปของการทำงาน ตัวดำเนินการ .Members ทำหน้าที่ดึงเซ็ตข้อมูลทั้งหมดที่มีอยู่จากมิติ ลำดับชั้นหรือระดับทางซ้ายมือ เช่น

[Customer].Members ผลลัพธ์ที่ได้คือเซ็ตของข้อมูลสมาชิกทั้งหมดของมิติ customer

[Product].[Product Category].Members ผลลัพธ์ที่ได้คือเซ็ตของข้อมูลสมาชิกทั้งหมดของระดับ product category ของมิติ Product

- CrossJoin()

มีบ่อยครั้งที่ต้องการผลลัพธ์ที่เป็น cross product ของสมาชิกหรือ tuples ใน 2 เซ็ตข้อมูลที่ต่างกันที่มีความเป็นไปได้ทั้งหมด CrossJoin() เป็นฟังก์ชันที่ใช้สำหรับรวมข้อมูลในลักษณะ cross product มีวิธีการใช้ดังตัวอย่าง

```
SELECT
CrossJoin(
        {[Time].[January-2007]:[Time].[December-2007]},
        {[Product].[Product Category].Members}
      ) on columns,
      { [Stores].[East Regoin], [Stores].[West Region]} on rows
FROM sampleCube
```

ผลลัพธ์ที่ได้จากการทำงานของ MDX นี้ แสดงได้ดังตารางที่ 2.5

ตารางที่ 2.5 แสดงผลลัพธ์การทำงานของฟังก์ชัน CrossJoin()

	Jan 2007	**Jan 2007**	**Jan 2007**	…	**Dec 2007**	**Dec 2007**	**Dec 2007**
	Toys	**Games**	**Clothing**	…	**Toys**	**Games**	**Clothing**
East	45	60	120	…	350	500	800
West	60	30	90	…	400	400	750

ฟังก์ชัน CrossJoin() นั้นสามารถรับอาร์กิวเมนต์เป็นเซ็ตข้อมูล 2 เซ็ตข้อมูลเท่านั้น ถ้าต้องการทำ cross product 3 เซ็ตข้อมูลหรือมากกว่านั้น สามารถทำได้ดังนี้

```
CrossJoin(
  [Time].Members,
  CrossJoin([Scenario].Members, [Product].Members)
)
```

หรือ

```
CrossJoin(
  CrossJoin([Time].Members,[Scenario].Members),
  [Product].Members
)
```

CrossJoin() เป็นฟังก์ชันมาตรฐานของเอ็มดีเอ็กซ์ ใน Microsoft OLAP หรือ Analysis Service มีเทคนิคที่ไม่ได้เป็นมาตรฐานสำหรับการ cross product เช่นเดียวกันกับฟังก์ชัน CrossJoin() สามารถทำได้โดยใช้เครื่องหมาย asterisk (*) ดังนี้

```
{[Time].Members} * {[Scenario].Members} * {[Product].Members}
```

อีกหนึ่งรูปแบบที่ถูกใช้เป็นประจำของฟังก์ชัน CrossJoin() คือการรวม single member บน 1 มิติ กับเซ็ตของสมาชิกข้อมูลของต่างมิติ สามารถทำได้ดังนี้

```
CrossJoin(
  {[Product].[Toothpaste]},
  [Stores].[Store-A]:[Stores].[Store-E]
)
```

- Filter()

Filter() เป็นฟังก์ชันที่คล้ายกับ CrossJoin() และโคลอน (:) ช่วยในการสร้างเซ็ตข้อมูล Filter() ใช้กรองข้อมูลตามเงื่อนไขที่กำหนด ทำงานโดยกำหนดหนึ่งเซ็ตข้อมูลและเงื่อนไข โดยผลลัพธ์ที่ส่งกลับมาจะเป็นสับเซ็ตที่เงื่อนไขเป็นจริง ดังตัวอย่างต่อไปนี้

```
Filter(
  {[Product].[Product Category].Members},
  [Measures].[Sales] >= 5000
)
```

ผลลัพธ์ที่ได้จะเป็นเซ็ตข้อมูลสมาชิกของ product category ที่มีค่า sales มากกว่าหรือเท่ากับ 500 นอกจากการใช้เครื่องหมายเปรียบเทียบแล้ว เครื่องหมายตรรกะอื่น ๆ การสามารถใช้ในการกรองข้อมูลได้เช่นกัน ดังตัวอย่างต่อไปนี้

```
Filter(
  {[Product].[Product Category].Members},
  ([Measures].[Sales] >= 1.2 * [Measures].[Costs])
  AND [Measures].[Sales] > 150
)
```

ผลลัพธ์ที่ได้จะเป็นเซ็ตข้อมูลสมาชิกของ product category ที่ค่า sales มากกว่าหรือเท่ากับ 1.2 คูณด้วย costs และค่า sales ที่มีค่ามากกว่า 150 Filter() สามารถใช้กับเซ็ตข้อมูลที่ไม่ได้มีแค่สมาชิกของหนึ่งมิติ ดังตัวอย่างต่อไปนี้ที่ใช้เซ็ตข้อมูลของ product Category และ store-A ที่มีความสัมพันธ์กันและมีค่ามากกว่าหรือเท่ากับ 500

```
Filter(
  CrossJoin(
    [Product].[Product Category].Members, [Store].[Store-A].Members
  ),
  [Measures].[Sales] >= 500 )
```

- **Order()**

ฟังก์ชัน Order() จำเป็นที่จะต้องใช้ เพื่อที่จะให้ tuples ในเซ็ตข้อมูลมีการจัดเรียงบนพื้นฐานความสัมพันธ์ของค่าข้อมูล ฟังก์ชัน Order() รับอาร์กิวเมนต์เป็นเซ็ตข้อมูล เงื่อนไขที่ทำการเรียง ส่วนเสริม และสุดท้ายเป็นตัวหนดทิศทางการเรียงซึ่งจะมีเรียงจากน้อยไปมาก (ascending) หรือมากไปน้อย (descending) ผลลัพธ์ที่ส่งกลับมาเป็นเซ็ตข้อมูลต้นฉบับที่ทำการเรียงเรียบร้อยแล้ว ดังตัวอย่างต่อไปนี้

```
Order(
  [Product].[Product Category].Members,
  ([Measures].[Profit], [Time].[2000], [Customer].[All Customers]),
  BDESC )
```

จากตัวอย่างเป็นการใช้ฟังก์ชัน Order() ที่ทำการเรียงข้อมูล product category จากมากไปน้อย มีเงื่อนไขว่าต้องเป็น profit ของปี 2000 จาก all customer

การใช้ Order() ทำการเรียงข้อมูลหลาย tuples เช่น สนใจ profit ของ product และ store สำหรับเอ็มดีเอ็กซ์ตัวอย่างแสดงได้ดังต่อไปนี้

```
Order(
 Filter(
  CrossJoin(
    [Product].[Product Category].Members,
    [Store].[City].Members
   ),
   [Measures].[Sales] >= 500
 ),
 ([Measures].[Profit], [Time].[2007], [Customer].[All Customers]),
 BDESC
)
```

ตารางที่ 2.6 ผลลัพธ์จากการเรียงของฟังก์ชัน Order()

Product	**Store**	*Associated Sales*	*Associated Profit*
Leather Jackets	Las Vegas, NV	*6,000*	*3,500*
Silk Blouses	Honolulu, HI	*5,000*	*1,700*
Attache Cases	Nyack, NJ	*2,500*	*1,300*
Golf Umbrellas	Nyack, NJ	*1,900*	*1,200*
Golf Umbrellas	Augusta, GA	*8,000*	*1,150*

2.4.4 ข้อคิดเห็นในเอ็มดีเอ็กซ์ (Comments in MDX)

การมาของ Analysis Services และ OLE DB ของ OLAP ทำให้เอ็มดีเอ็กซ์รองรับการเขียนข้อคิดเห็นได้ หนึ่งในการใช้เอ็มดีเอ็กซ์คือผู้ใช้ใช้ติดต่อเกี่ยวกับความหมายของ query ต่าง ๆ และการคำนวณและการที่สามารถแทรกความคิดเห็นได้นั้นมีความสำคัญมาก 3 รูปแบบของการเขียนแสดงความคิดเห็นในเอ็มดีเอ็กซ์ มีดังนี้

1. เครื่องหมาย slash และ asterisk (/* ข้อความ */) ข้อความหรือโค้ดที่อยู่ระหว่างเครื่องหมาย /* และ */ ตัว parsing ของเอ็มดีเอ็กซ์จะไม่นำไปทำงานด้วย ใช้ได้ทั้งความคิดเห็นที่มี 1 บรรทัดและมากกว่านั้น แสดงได้ดังตัวอย่างต่อไปนี้

```
SELECT /* Put products
on columns */
[Product].Members  on columns
FROM Cube
```

2. เครื่องหมาย slash 2 ตัว (// ข้อความ) ข้อความหรือโค้ดที่อยู่หลังเครื่องหมาย // ในบรรทัดนั้น ตัวแปลภาษาจะมองส่วนนั้นว่าเป็นความคิดเห็น เอ็มดีเอ็กซ์จะไม่นำไปทำงาน ใช้กับความคิดเห็น 1 บรรทัด แสดงได้ดังตัวอย่างต่อไปนี้

```
SELECT
 // Put products on columns
[Product].Members  on columns
FROM Cube
```

3. เครื่องหมาย dash 2 ตัว (-- ข้อความ) มีความหมายและการใช้งานเช่นเดียวกับข้อที่ 2 แสดงได้ดังตัวอย่างต่อไปนี้

```
SELECT
-- Put products on columns
[Product].Members  on columns
FROM Cube
```

2.5 Decision Support Object (DSO) and Active Data Objects Multi-Dimensional (ADOMD)

Wen-Yuan Liu และ Shu-Fen Fang (2002) ได้นำเสนอ OLAP Realization Technology Research Based on MDX โดยใช้เทคนิคการใช้ Visual Basic (VB) ร่วมกับเอ็มดีเอ็กซ์โดยนำ VB ซึ่งเป็นภาษาโปรแกรมที่ได้รับความนิยม เนื่องจากเป็นภาษาที่ไม่ซับซ้อน ง่ายต่อการเรียนรู้ มีฟังก์ชันการทำงานที่มีประสิทธิภาพ มีโปรแกรมเมอร์จำนวนมากนิยมใช้ VB ในการทำงาน ดังนั้นงานวิจัยชิ้นนี้เลือกที่จะสร้างโปรแกรมประยุกต์ OLAP ด้วย VB และเอ็มดีเอ็กซ์ งานวิจัยชิ้นนี้จึงมีเป้าหมายที่จะนำ OLAP มาทำให้เป็นโปรแกรมประยุกต์ที่สร้างขึ้นด้วย Multi Dimensional eXpressions (MDX) Visual Basic และ OLAP ทำให้ผู้ใช้ที่ไม่ชำนาญได้การใช้ OLAP สามารถที่ใช้ได้ง่ายขึ้น โดยมีนำเสนอการใช้เทคนิคดังนี้

การเข้าถึงแหล่งข้อมูล OLAP ด้วย Decision Support Object (DSO)

DSO ใช้อ้างถึงลูกบาศก์ข้อมูล (cube) มิติ (dimension) สมาชิก (member) ระดับ (level) หรือค่า measure ด้วย DSO object สามารถเข้าถึงแหล่งข้อมูล OLAP ได้ดังโค้ดโปรแกรมภาษา Visual Basic ต่อไปนี้

```
Dim  dsoServer  As DSO.Server                    'server object
Dim  dsoSelectedDB  As DSO.MDStore              'data source
Dim  dsoSelectedCube  As DSO.MDStore            'cube
dsoServer.Connect        "Yiyu"                 'connecting to server
Set dsoSelectedDB  = dso.Server.MDStores(1)     'select a data source
Set dsoSelectedCube = dsoSelectedDB.MDStores(1)     'select a data cube
```

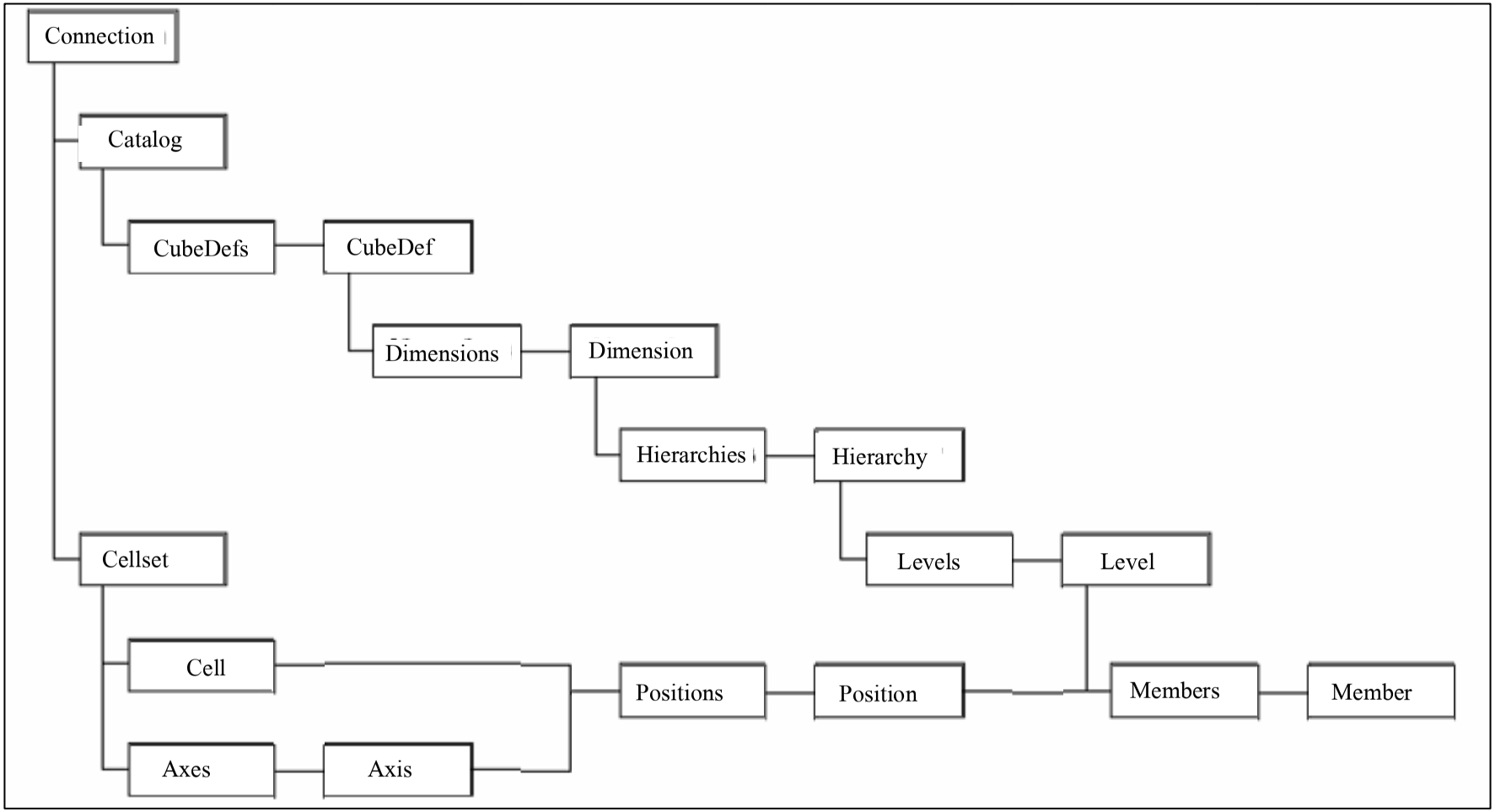
การทำงานกับ MDX ด้วย Active Data Objects Multi-Dimensional (ADOMD)

ADOMD คือ ตัวจัดการข้อมูลของ .NET framework ที่ออกแบบมาเพื่อใช้ติดต่อกับ Microsoft SQL Server Analysis Service (SSAS) สามารถแสดงตัวอย่างการใช้งานได้ดังโค้ดโปรแกรมต่อไปนี้

```
Dim Cat  As  ADOMD.Catalog          'OLAP data source object
Dim Cst  As  ADOMD.Cellset          'OLAP cube object
Set  Cat = new ADOMD.Catalog        'apply room for connection
Set  Cst = new ADOMD.Cellset        'apply room for cube
Cat.ActiveConnection = "provider=msolap:data source=yiyu;Initial Catalog=Cwgl;"
                                    'setting connect parameters
Cst.Source = "Select {[Measures].[Njffse]} on columns,
{[Periods].[All Periods].[1999].[Quarter 1]} on rows From Mxz"
                                    'set MDX sentence
Set  Cst.ActiveConnection = Cat.ActiveConnection    'connect cube to data source
Cst.Open                                            'get query result
```

ADOMD เป็นคลาสอินเตอร์เฟสที่สำคัญ ใช้ในการทำงานใช้สร้างโปรแกรมประยุกต์ ที่ทำงาน OLAP เพราะว่า ADOMD นั้นใช้มาตรฐานของอินเตอร์เฟส COM สามารถทำงานได้กับภาษาโปรแกรมมิงหลายภาษา เช่น C#, VB, C++, Java, VBScript and Jscript ในรูปที่ 2.5 แสดงโครงสร้างของ ADOMD ซึ่งมีอ็อบเจ็ค Connection และ 2 อ็อบเจ็คที่อยู่ภายในคือ Catalog และ Cellset ซึ่งทั้งสองเป็นสิ่งสำคัญของส่วนเพิ่มของ ADO ในเชิงหลายมิติ อ็อบเจ็ค Catalog ทำหน้าที่

ประกอบด้วย CubeDefs คือคอลเล็คชันของลูกบาศก์ข้อมูล Dimensions คือคอลเล็คชันของมิติข้อมูล Hierarchies คือคอลเล็คชันของลำดับชั้นของข้อมูล Level คือคอลเล็คชันของระดับของข้อมูลและ Members คือคอลเล็คชันของสมาชิกข้อมูล ส่วนอีกอ็อบเจ็คคือ Cellset นั้นใช้รับผลลัพธ์ของ query ใน Microsoft's ADOMD อ็อบเจ็ค Cellset ถูกกำหนดและออกแบบมาเพื่อรับข้อมูลในลักษณะโครงสร้างของข้อมูลเชิงหลายมิติ คือประกอบด้วย Cell และ Axes ที่เป็นผลลัพธ์ของการ query สามารถแสดงเป็นโครงสร้างต้นไม้ได้ดังรูปที่ 2.5



รูปที่ 2.5 The ADOMD Connection object has Catalog and Cellset

2.6 Unified Model Langauge (UML)

UML คือ โมเดลมาตรฐานที่ใช้หลักการออกแบบ OOP (Object Oriented Programming) รูปแบบของภาษา UML จะมีเครื่องหมาย (Notation) ซึ่งเป็นสัญลักษณ์ที่นำไปใช้ในโมเดลต่าง ๆ UML จะมีข้อกำหนดกฎระเบียบต่าง ๆ ในการโปรแกรม โดยกฎระเบียบต่าง ๆ จะมีความหมายต่อการเขียนโปรแกรม ดังนั้นการใช้ UML จะต้องทราบความหมายของการใช้เครื่องหมาย (Notation) ต่าง ๆ เช่น Generalize, association dependency class และ package สิ่งเหล่านี้มีความจำเป็นอย่างยิ่งต่อการตีความของการออกแบบ ระบบ ก่อนนำไปประยุกต์กับระบบงานจริง

UML ถูกนำมาใช้ในการพัฒนาซอฟต์แวร์อย่างกว้างขวาง และทำให้การทำงานที่มีคุณภาพกล่าว คือ

1. ช่วยลดระยะเวลาในการพัฒนาระบบงาน (Shortest Development life cycle)
2. ช่วยเพิ่มความสามารถในการทำงาน (Increase productivity)
3. ช่วยเพิ่มคุณภาพของระบบงาน (Improve software quality)
4. รองรับระบบงานเดิม (Support legacy system)
5. ช่วยในการสื่อสารระหว่างทีมผู้พัฒนาระบบงาน (Improve team connectivity)

2.6.1 Diagram of UML

ไดอะแกรมของ UML ที่ใช้ในงานออกแบบและวิเคราะห์ระบบเชิงวัตถุ

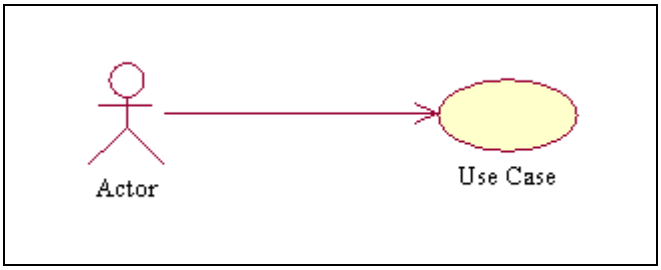
(Object-Oriented Analysis and Design) แบ่งได้เป็น 2 ประเภทคือ Static Diagram และ Dynamic Diagram

- Static Diagram คือ ไดอะแกรมที่แสดงภาพในเชิงสถิตย์ (Static) ของระบบ โดยจะแสดงถึงการมีอยู่ของคบาสและความสัมพันธ์ระหว่างคลาสแต่จะไม่แสดงถึงกิจกรรมที่จะเกิดขึ้น ไดอะแกรมที่ใช้ เช่น Use Case Diagram, Class Diagram

- Dynamic Diagram คือ ไดอะแกรมที่แสดงภาพในเชิงกิจกรรม (Dynamic) ของระบบ โดยจะแสดงถึงสิ่งที่เกิดขึ้นจากกิจกรรมของคลาสต่าง ๆ ที่มีในระบบ ไดอะแกรมที่ใช้ เช่น Sequence Diagram, Statechart Diagram

2.6.2 ยูสเคสไดอะแกรม (Use case diagram)

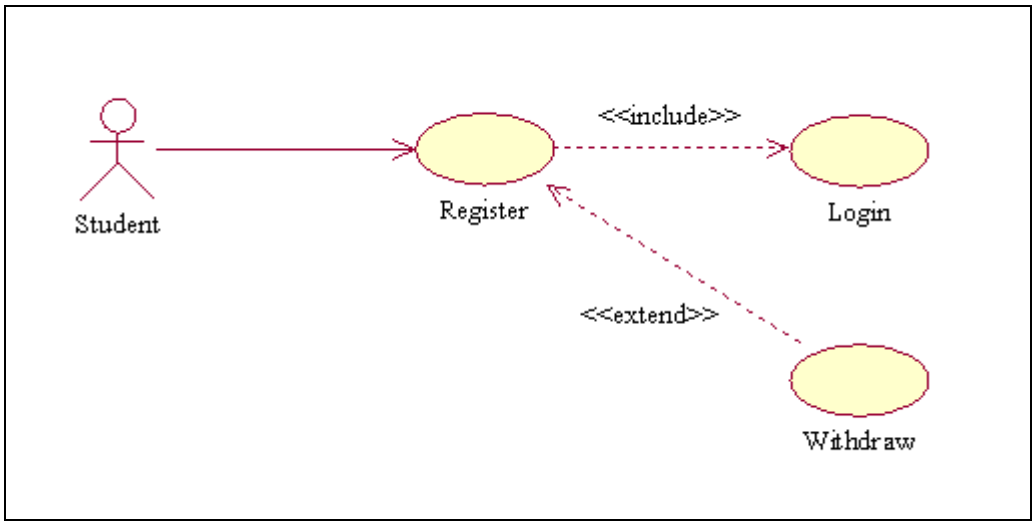
โดยทั่วไป ในขั้นตอนการวิเคราะห์ความต้องการของระบบตามแนวทางเชิงวัตถุ จะใช้ use case ไดอะแกรมเป็นเครื่องมือในการจำลองหน้าที่ของระบบที่ผู้ใช้ต้องการ เนื่องจาก use case ไดอะแกรมเป็นภาพที่ใช้แสดงถึงหน้าที่การทำงานที่ระบบสามารถทำได้เรียกว่า ยูสเคส (use



รูปที่ 2.6 ตัวอย่างยูสเคสไดอะแกรม

ข้อสังเกตของสัญลักษณ์ UML สำหรับยูสเคสจะเป็นรูปวงรี ซึ่งมีคำอธิบายใต้สัญลักษณ์ เป็นการกระทำหรือเหตุการณ์ สำหรับผู้แสดงใช้สัญลักษณ์รูปตัวคนและมีคำอธิบายบทบาทของผู้แสดงนั้น ลูกศรที่ขีดจากผู้แสดงไปยังสัญลักษณ์วงรีแสดงถึงความสัมพันธ์เพราะเป็นการเชื่อมโยงผู้แสดงเข้ากับยูสเคส

ยูสเคสสามารถมีปฏิสัมพันธ์กับยูสเคสอื่น ๆ ได้ โดยเมื่อผลของยูสเคสอันหนึ่งเข้ารวมกันกับยูสเคสอีกอันหนึ่ง คือ ยูสเคสอันที่สองใช้งานยูสเคสอันแรก ซึ่ง UML ได้กำหนดความสัมพันธ์ด้วยลูกศรเส้นประ แสดงได้ดังรูปที่ 2.7



รูปที่ 2.7 เครื่องหมายลูกศรแสดงความสัมพันธ์ระหว่างยูสเคส

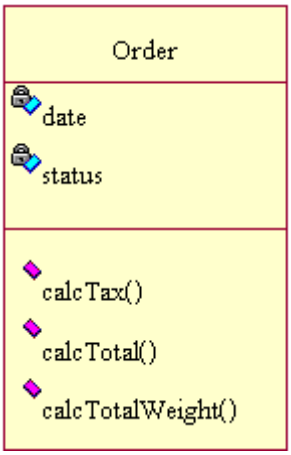
จากรูปที่ 2.7 เครื่องหมายลูกศรที่ใช้แสดงความสัมพันธ์ระหว่างยูสเคสมี 2 แบบ คือ แบบ include และแบบ extend

แบบ include หมายความว่า ยูสเคสที่หนึ่งจะทำงานได้จำเป็นจะต้องทำงานของยูสเคสที่สองด้วยเสมอ จากตัวอย่างคือยูสเคส Register จำเป็นจะต้องทำยูสเคส Login ด้วยเสมอ

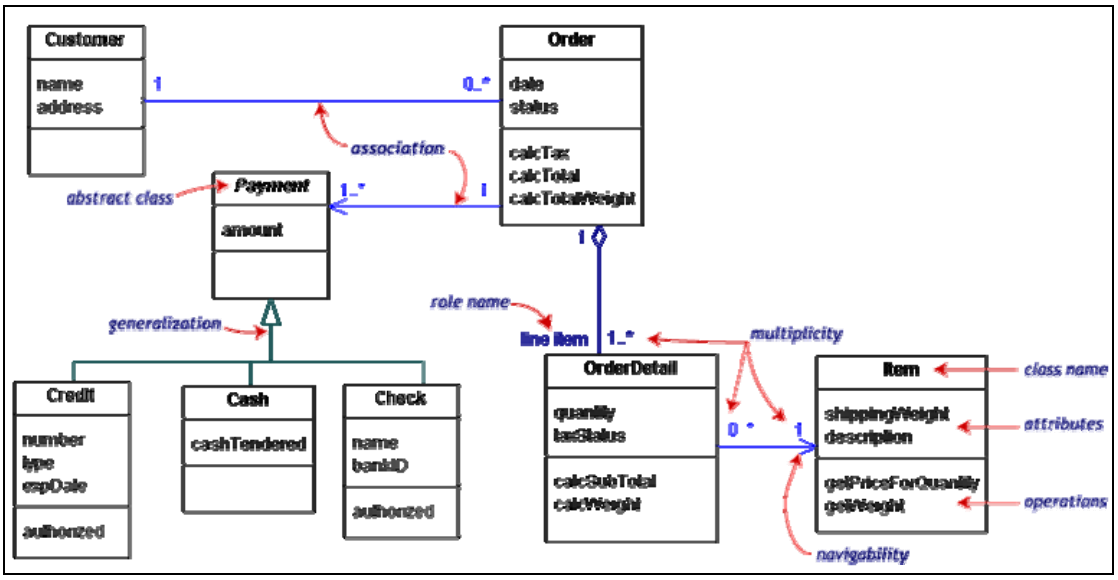
แบบ extend หมายความว่า ยูสเคสที่สองเป็นทางเลือกของยูสเคสที่หนึ่งเมื่อยูสเคสที่อาจจะทำหรือไม่ทำยูสเคสที่สองก็ได้ จากตัวอย่างคือเมื่อทำงานยูสเคส Register แล้วจะทำยูสเคส Withdraw หรือไม่ก็ได้

2.6.3 คลาสไดอะแกรม (Class diagram)

เป็นแผนภาพที่ใช้แสดงกลุ่มของคลาส โครงสร้างของคลาสและอินเตอร์เฟส ตลอดจนแสดงความสัมพันธ์ระหว่างคลาส การเริ่มต้นสร้างคลาสไดอะแกรมนั้นจะต้องค้นหาอ็อบเจ็คในยูสเคสไดอะแกรมก่อน โดยค้นหาจากคำอธิบายรายระเอียดของยูสเคส ชื่อคลาสหาได้จากคำนาม (Noun) ส่วนแอทริบิวต์ของคลาสหาได้จากคำคุณศัพท์ (Adjective) และเมธอดหรือโอเปอเรชันหาได้จากคำกิริยา (Verb) แสดงตัวอย่างรูปที่ 2.8



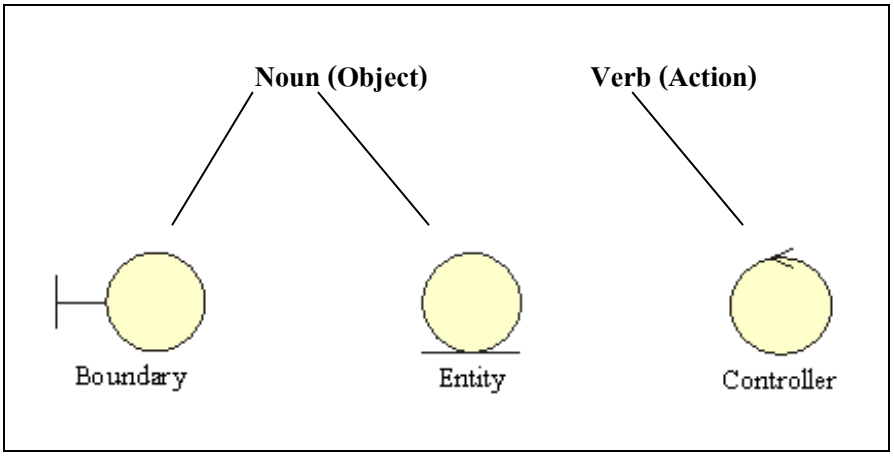
รูปที่ 2.8 สัญลักษณ์คลาส



รูปที่ 2.9 ตัวอย่างคลาสไดอะแกรม

จากรูปที่ 2.9 จะเห็นว่าคลาสไดอะแกรมนอกจากจะแสดงให้เห็นถึงคลาสทั้งหมดแล้ว ยังแสดงให้เห็นถึงความสัมพันธ์ระหว่างคลาส ไม่ว่าจะเป็นความสัมพันธ์แบบ Association, Generalization, Navigability และอื่น ๆ รวมถึงแสดงรูปแบบความสัมพันธ์ระหว่างคลาส (Multipicility) ด้วย

Doug Rosenberg and Matt Stephens กล่าวใน Use Case Driven Object Modeling with UML - Theory and Practice (Apress, 2007) ว่า Robustness diagram มีลักษณะเป็นแผนภาพผสมระหว่าง class diagram กับ activity diagram ใช้อธิบายถึงลักษณะพฤติกรรมของ use case แต่ว่าหลีกเลี่ยงที่จะใช้รูปที่เป็นคลาสกรอบสี่เหลี่ยมที่อธิบายพฤติกรรมของคลาสได้น้อย อย่างไรก็ตาม Robustness diagram ยังมีลักษณะที่คล้ายกับ activity diagram หรือ flowchart ในความหมายที่หนึ่งอ็อบเจ็คเรียกอ็อบเจ็คถัดไป การเรียกกันของอ็อบเจ็คใช้เส้นระหว่างอ็อบเจ็คเครื่องหมายว่าอ็อบเจ็คติดต่อกันอยู่ โดยจะใช้สัญลักษณ์แบ่งคลาสออกเป็นแต่ละประเภทซึ่งประกอบด้วย 3 ประเภท คือ คลาสเบาน์ดารี (Boundary class) คลาสคอนโทรลเลอร์ (Controller class) และคลาสเอ็นติตี้ (Entity class) เพื่อง่ายต่อการวิเคราะห์และออกแบบเพื่อพัฒนาระบบ ดังรูปที่ 2.10



รูปที่ 2.10 สัญลักษณ์ Robustness diagram

คลาสเบาน์ดารี เป็นคลาสทำหน้าที่ติดต่อกับผู้แสดง (Actor) เช่น หน้าตาโปรแกรมหรือหน้าเว็บ

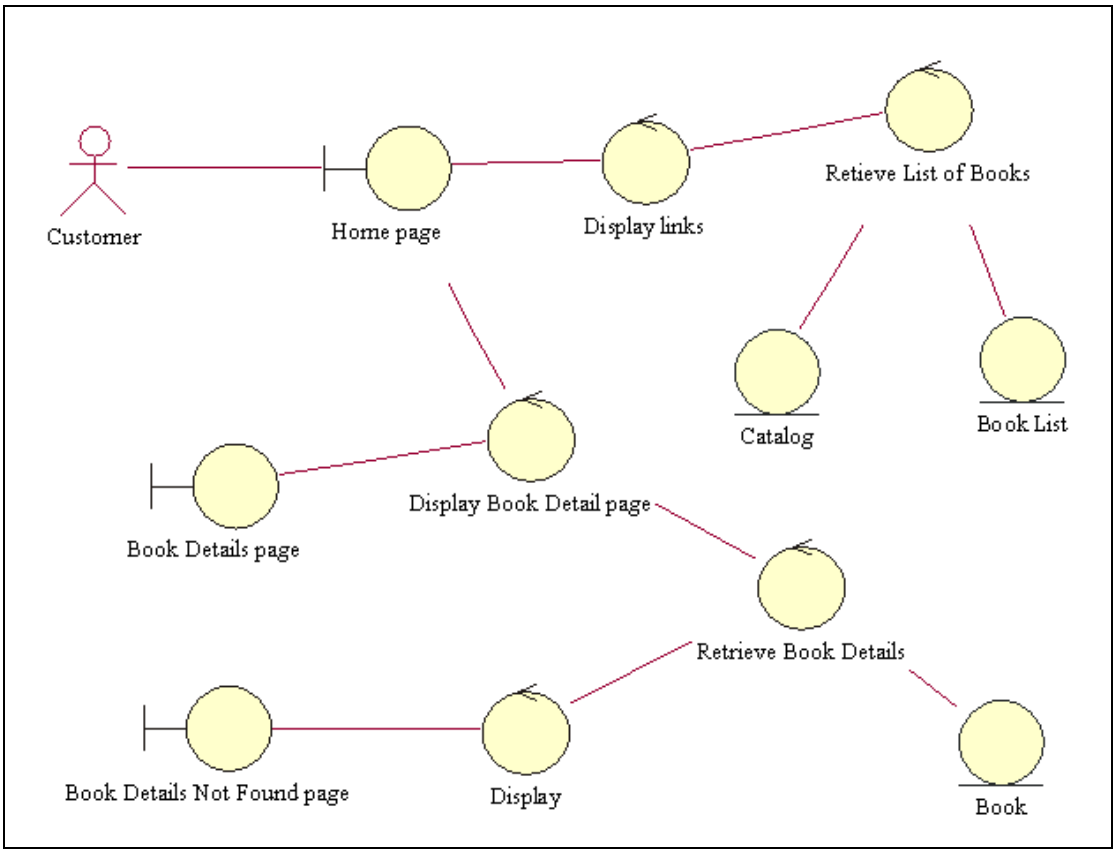
คลาสคลอนโทรลเลอร์ ทำหน้าที่เกี่ยวกับการทำงานต่าง ๆ เช่น การคำนวณ

คลาสเอนติตี้ เป็นคลาสที่เป็นโมเดลหรือต้นแบบของฐานข้อมูล

การทำงานของ Robustness diagram มีข้อดีในแง่ของการที่มองเบาน์ดารีและเอนติตี้เป็นอ็อบเจ็คที่เป็นคำนาม (Noun) และคลาสคอนโทรลเลอร์เป็นพฤติกรรมหรือกิริยา (Verb) เมื่อเป็นเช่นนั้นแล้วทำให้มีข้อกำหนดหลัก 3 ข้อในการเขียน Robustness diagram ดังนี้

- คำนาม (Noun) สามารถติดต่อกิริยา (Verb) ได้
- คำนาม (Noun) ไม่สามารถติดต่อกับคำนามอื่นโดยตรงได้
- กิริยา (Verb) สามารถติดต่อกับกิริยาอื่นได้

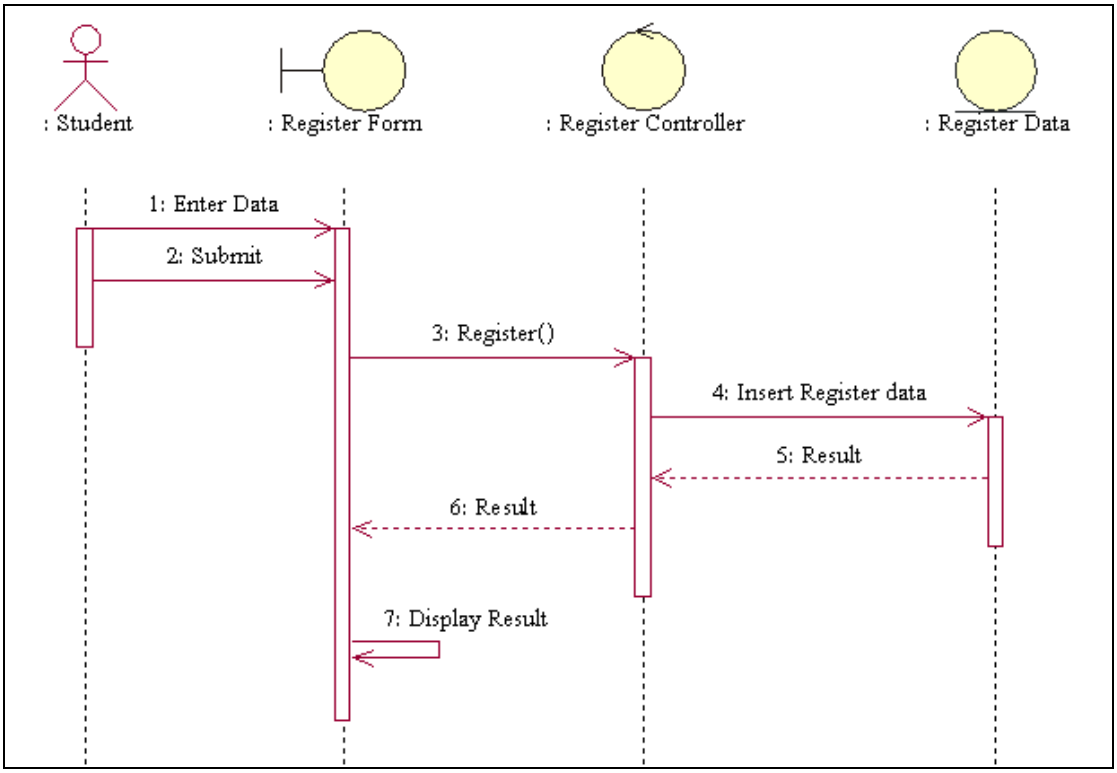
จากข้อกำหนดทั้ง 3 ข้อ สามารถแสดงเป็น Robustness diagram ตัวอย่างเป็น diagram ของร้านขายหนังสือโดยจะมีคลาสเบาน์ดารี 3 คลาสได้แก่ Home page, Book detail page and Book detail not found page มีคลาสคอนโทรลเลอร์ 4 คลาส ได้แก่ Display link, Retreive list of books, Display book details page and Retrieve book details และมีคลาสเอนติตี้ 3 คลาสได้แก่ Catalog, Book list and Book



รูปที่ 2.11 ตัวอย่าง Robustness diagram

2.6.4 ซีเควนซ์ไดอะแกรม (Sequence diagram)

เป็นแผนภาพที่แสดงให้เห็นถึงการปฏิสัมพันธ์ (Interaction) ระหว่างอ็อบเจ็คหรือคลาส โดยเฉพาะการส่ง Message ระหว่างอ็อบเจ็คตามลำดับของเวลา (Sequence) ที่เกิดเหตุการณ์ขึ้น จะมีสัญลักษณ์แสดงให้เห็นลำดับของการส่ง Message ตามเวลาส่งอย่างชัดเจน แสดงตัวอย่างไดอะแกรมได้ดังรูปที่ 2.12



รูปที่ 2.12 ตัวอย่าง Sequence diagram

จากรูปที่ 2.12 คือลำดับการทำงานของการลงทะเบียนโดยมีกระบวนการดังนี้

(1) นักเรียนกรอกข้อมูลที่หน้าการลงทะเบียนแล้ว

(2) ยืนยันการลงทะเบียน

(3) หน้าลงทะเบียนเรียกเมธอดจากคลาสคอนโทรลเลอร์

(4) ทำการบันทึกข้อมูลการลงทะเบียน

(5) คลาสเอนติตี้ส่งผลลัพธ์การลงทะเบียนกลับมาที่คลาสคอนโทรลเลอร์

(6) คลาสคอนโทรลเลอร์ส่งผลลัพธ์การมาที่คลาสเบาน์ดารี่

(7) แสดงผลการลงทะเบียนที่หน้าลงทะเบียน

การวิเคราะห์ระบบนั้นจะเน้นที่ความต้องการของระบบในแง่ของฟังก์ชันความสามารถของระบบ โดยเริ่มจากการตรวจสอบยูสเคสทั้งหมดของระบบจากยูสเคสไดอะแกรมแล้วกำหนดถึง อ็อบเจ็คและคลาสต่าง ๆ เมื่อขั้นตอนการวิเคราะห์ทำได้อย่างเหมาะสม ผลลัพธ์ที่ได้จากกระบวนการวิเคราะห์ระบบคือ คลาสและโอเปอเรชันการทำงานหลัก ๆ เป็นคลาสไดอะแกรมของระบบ จากนั้นนำคลาสและโอเปอเรชันมาเขียนในรูปของไดอะแกรมเชิงกิจกรรมเพื่ออธิบายขั้นตอนการทำงานได้อย่างชัดเจน

บทที่ 3
ระเบียบวิธีวิจัย

การทำงานในปัจจุบันมีการนำคลังข้อมูล (Data Warehouse) และฐานข้อมูลแบบหลายมิติ (Multidimensional Database) เข้ามาใช้สำหรับงานสารสนเทศภายในองค์กรต่าง ๆ อย่างแพร่หลาย ทำให้เอ็มดีเอ็กซ์เป็นสิ่งสำคัญ เพราะเป็นภาษาที่ใช้ในการเรียกดูข้อมูลจากฐานข้อมูลแบบหลายมิติ งานวิจัยนี้มีจุดมุ่งหมายที่จะศึกษาและใช้งานเอ็มดีเอ็กซ์ โดยสร้างโปรแกรมประยุกต์บนบราวเซอร์ที่ทำหน้าติดต่อเรียกดูข้อมูลกับลูกบาศก์ข้อมูล เพื่อสร้างคำสั่งเอ็มดีเอ็กซ์อำนวยความสะดวกให้กับผู้ใช้ที่ไม่มีความชำนาญในการใช้เอ็มดีเอ็กซ์ที่จะเรียกดูข้อมูล โปรแกรมที่พัฒนาขึ้นโดยใช้ภาษา ASP.NET (C# Language) ประกอบกับ Active Data Objects Multi-Dimensional (ADOMD.NET) ในการทำงานกับลูกบาศก์ข้อมูล รายละเอียดเนื้อหาในบทนี้ประกอบด้วยระเบียบวิธีวิจัย ปรากฏในหัวข้อ 3.1 แหล่งที่มาของข้อมูลในหัวข้อที่ 3.2 และหัวข้อที่ 3.3 เป็นรายละเอียดการออกแบบและวิเคราะห์ระบบของโปรแกรมสร้างเอ็มดีเอ็กซ์บนบราวเซอร์ โดยใช้ UML diagram หัวข้อที่ 3.4 เป็นรายละเอียดการออกแบบหน้าตาของโปรแกรม

3.1 ขั้นตอนการวิจัย

การค้นคว้าวิจัยแบ่งออกเป็น ขั้นตอนดังนี้

1) ศึกษาและรวบรวมสรุปงานวิจัยที่เกี่ยวข้อง

2) ศึกษาการทำงานของฟังก์ชันของเอ็มดีเอ็กซ์ เพื่อนำไปใช้กับโปรแกรม MDX Generator

3) ออกแบบและวิเคราะห์การทำงานของระบบ โดยใช้ UML diagram (use case diagram, class diagram, sequence diagram) เพื่อการทำงานที่เป็นระบบสามารถที่จะปรับแก้ได้สะดวก

4) ออกแบบหน้าตาโปรแกรม (Graphic User Interface) ให้สามารถตอบสนองความต้องการของผู้ใช้ให้ครอบคลุมที่สุด

5) พัฒนาโปรแกรมประยุกต์บนบราวเซอร์ Online MDX Generator ด้วยภาษา ASP.NET (C# Language)

6) ทดสอบและพัฒนาโปรแกรมประยุกต์บนบราวเซอร์ ตรวจสอบความถูกต้องของผลลัพธ์ที่ได้จากการทำงาน เครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบเป็นคอมพิวเตอร์ Pentium 4 3.12 GHz RAM 512 MB ทำงานเป็นเครื่องให้บริการ (server)

7) รายงานผลการทดสอบ สรุปรวมผลการทำงานทั้งหมด อภิปรายผลและจัดทำรายงานการวิจัย

3.2 แหล่งที่มาของข้อมูลที่ใช้ในการทดสอบโปรแกรม

แหล่งข้อมูลที่ใช้ทดสอบโปรแกรมในงานวิจัยนี้ได้มาจากโปรแกรม Microsoft SQL Server 2000 คือฐานข้อมูล FoodMarts 2000 ที่เป็นข้อมูลจำลองการซื้อขายสินค้าชนิดต่าง ๆ ของร้านค้าในประเทศแคนาดา เม็กซิโกและสหรัฐอเมริกา ประกอบด้วยลูกบาศก์ข้อมูลจำนวน 6 ลูกบาศก์ข้อมูลและแต่ละลูกบาศก์ข้อมูลประกอบด้วย dimensions ต่าง ๆ ซึ่งแสดงรายละเอียดได้ดังต่อไปนี้

1. Budget

Dimension	Description
Account	มิติข้อมูลบัญชี
Category	มิติข้อมูลตามประเภท
Measures	มิติข้อมูลค่าที่ใช้คำนวณ
Store	มิติข้อมูลร้านค้า
Time	มิติข้อมูลเวลา

2. HR

Dimension	Description
Department	มิติข้อมูลแผนก
Employees	มิติข้อมูลพนักงาน
Measures	มิติข้อมูลค่าที่ใช้คำนวณ
Pay Type	มิติข้อมูลประเภทการจ่าย
Position	มิติข้อมูลตำแหน่ง
Store	มิติข้อมูลร้านค้า
Store Type	มิติข้อมูลประเภทร้านค้า
Time	มิติข้อมูลเวลา

3. Sales

Dimension	Description
Customers	มิติข้อมูลลูกค้า
Education Level	มิติข้อมูลระดับการศึกษา
Gender	มิติข้อมูลเพศ
Marital Status	มิติข้อมูลสถานะการสมรส
Measures	มิติข้อมูลค่าที่ใช้คำนวณ
Product	มิติข้อมูลสินค้า
Promotion Media	มิติข้อมูลสื่อการส่งเสริม
Promotions	มิติข้อมูลการส่งเสริม
Store	มิติข้อมูลร้านค้า
Store Size in SQFT	มิติข้อมูลขนาดร้านค้าใน SQFT
Store Type	มิติข้อมูลประเภทร้านค้า
Time	มิติข้อมูลเวลา

4. Trained Cube

Dimension	Description
Customer Pattern	มิติข้อมูลรูปแบบลูกค้า
Customers	มิติข้อมูลลูกค้า
Education Level	มิติข้อมูลระดับการศึกษา
Gender	มิติข้อมูลเพศ
Marital Status	มิติข้อมูลสถานะการสมรส
Measures	มิติข้อมูลค่าที่ใช้คำนวณ
Product	มิติข้อมูลสินค้า
Promotion Media	มิติข้อมูลสื่อการส่งเสริม
Promotions	มิติข้อมูลการส่งเสริม
Store	มิติข้อมูลร้านค้า
Time	มิติข้อมูลเวลา
Yearly Income	มิติข้อมูลรายได้เป็นปี

5. Warehouse

Dimension	Description
Measures	มิติข้อมูลที่ใช้คำนวณ
Product	มิติข้อมูลสินค้า
Store	มิติข้อมูลร้านค้า
Store Size in SQFT	มิติข้อมูลขนาดร้านค้าใน SQFT
Store Type	มิติข้อมูลประเภทร้านค้า
Time	มิติข้อมูลเวลา
Warehouse	มิติข้อมูลคลังข้อมูล

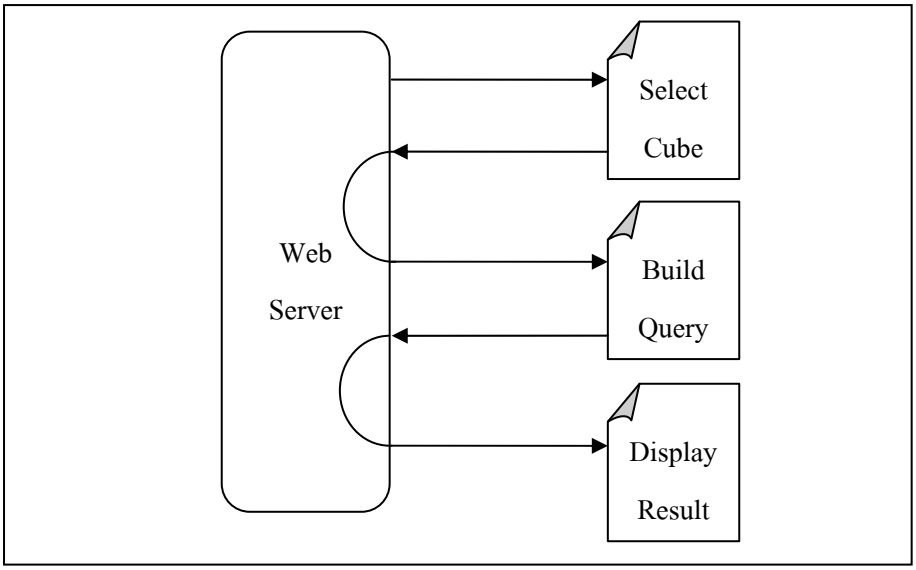
6. Warehouse and Sales

Dimension	Description
Customers	มิติข้อมูลลูกค้า
Education Level	มิติข้อมูลระดับการศึกษา
Gender	มิติข้อมูลเพศ
Marital Status	มิติข้อมูลสถานะการสมรส
Measures	มิติข้อมูลค่าที่ใช้คำนวณ
Product	มิติข้อมูลสินค้า
Promotion Media	มิติข้อมูลสื่อการส่งเสริม
Promotions	มิติข้อมูลการส่งเสริม
Store	มิติข้อมูลร้านค้า
Time	มิติข้อมูลเวลา
Warehouse	มิติข้อมูลคลังข้อมูล
Yearly Income	มิติข้อมูลรายรับเป็นปี

3.3 การวิเคราะห์และออกแบบโปรแกรมสร้างเอ็มดีเอ็กซ์

เนื้อหาในส่วนนี้จะกล่าวถึงภาพโดยรวมของการทำงานของโปรแกรม โดยนำหลักการของวิศวกรรมซอฟต์แวร์เข้ามาใช้กับการทำงานและการออกแบบโปรแกรมเชิงวัตถุโดยใช้ UML diagram ที่ประกอบด้วย use case diagram, class diagram (robustness) and sequence diagram ในการออกแบบและวิเคราะห์การทำงานของโปรแกรม

3.3.1 ลักษณะการทำงานในภาพรวมของโปรแกรม



รูปที่ 3.1 แผนภาพขั้นตอนการทำงานระหว่าง Client กับ Server

จากรูปที่ 3.1 แสดงขั้นตอนการทำงานโดยหลัก ๆ ของโปรแกรม Online MDX Generator ซึ่งมีขั้นตอนดังนี้

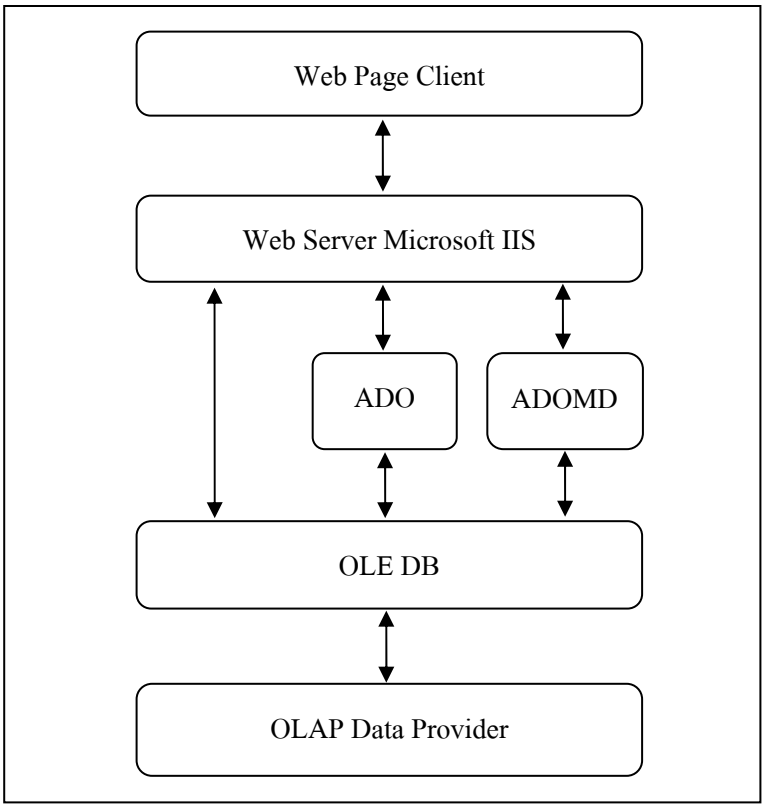
(1) เชื่อมต่อแหล่งข้อมูล ฐานข้อมูล และลูกบาศก์ข้อมูล และแสดงผลสู่ผู้ใช้ทางหน้าเว็บ โดยแสดงข้อมูลโครงสร้างลูกบาศก์ข้อมูลในรูปแบบของโครงสร้างต้นไม้

(2) ใช้ html form component อนุญาตให้ผู้ใช้สามารถเลือก dimension ที่ต้องการเรียกดูข้อมูลลงในส่วนที่กำหนดไว้

(3) จัดการเกี่ยวกับการใช้ฟังก์ชันการทำงานของเอ็มดีเอ็กซ์

(4) สั่งคำสั่งเอ็มดีเอ็กซ์ทำงานและแสดงผลลัพธ์สู่ผู้ใช้

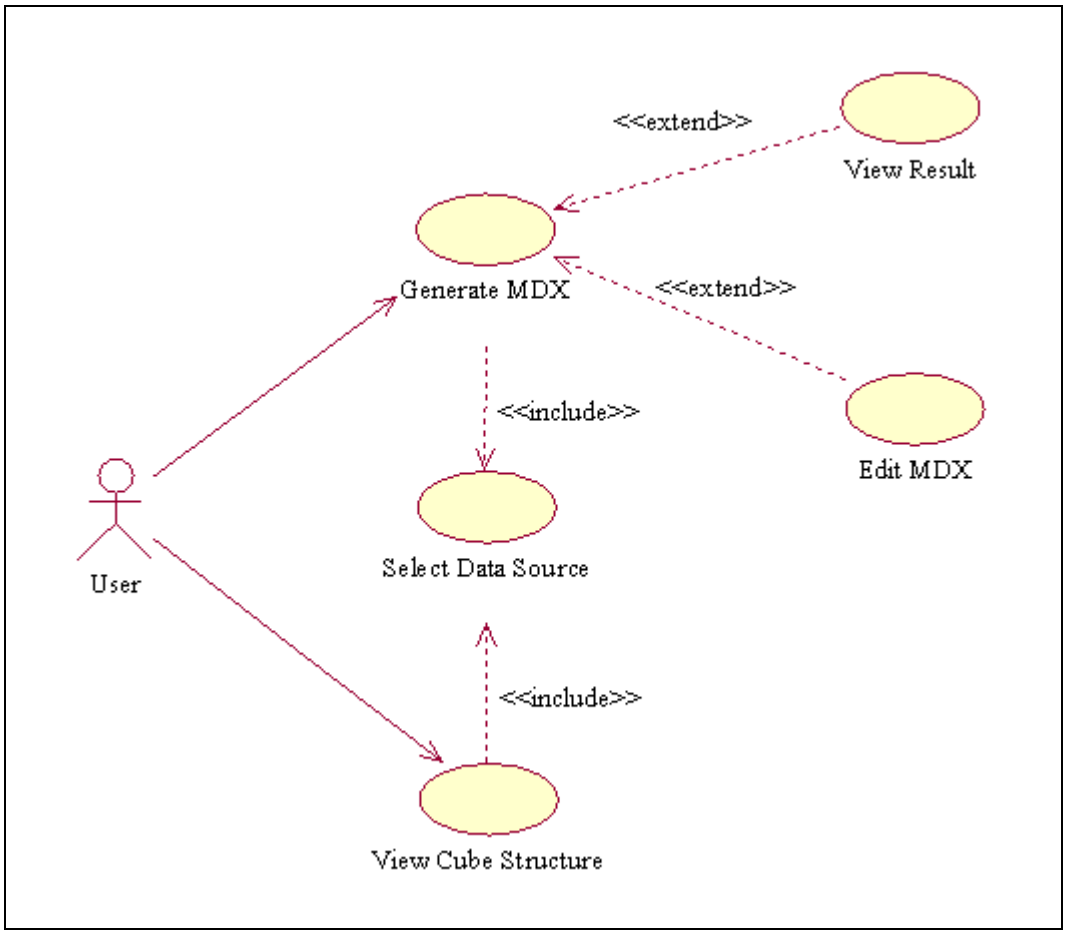
ในการที่โปรแกรมประยุกต์บนบราวเซอร์ทำงานกับ OLAP มีสถาปัตยกรรมการติดต่อข้อมูลสำหรับ OLAP Services นั้น ไม่ได้ทำงานกับตัวจัดการข้อมูล OLAP โดยตรง แต่การทำงานทั้งหมดจะผ่านทางเว็บ Server (IIS) ซึ่งเป็นการกำหนดที่ทำให้ไม่จำเป็นที่จะสร้าง ADO connection ระหว่างหน้าหน้าโปรแกรมกับตัวจัดการข้อมูลวิธีการแบบนี้มีข้อดีคือ ให้ความสำคัญกับมุมมองของความรักษาปลอดภัย โดยผู้ใช้ทุกคนจะถูกกำหนดเป็นผู้มาเยือน (Guest) ทำให้เป็นการง่ายต่อการกำหนดสิทธิ์ผู้ใช้ในการเข้าถึงข้อมูล ดังรูปที่ 3.2



รูปที่ 3.2 แผนภาพแสดงสถาปัตยกรรมการติดต่อข้อมูลสำหรับ OLAP Services

3.3.2 ยูสเคสไดอะแกรม (Use case diagram)

การทำงานของโปรแกรมเมื่อแบ่งหน้าที่ของระบบเป็นแต่ละกรณีของการทำงาน สามารถแบ่งความสามารถหลัก ๆ ในการทำงานของโปรแกรมที่ต้องการได้ 5 กรณี คือ Select Data Source, Generate MDX, Edit MDX, View Structure and View Result มี Actor ที่ทำงานเกี่ยวข้องระบบ 1 Actor คือ OLAP Administrator โดย Generate MDX และ View Cube Structure จะทำงานได้นั้นต้องทำ Select Data Source สำเร็จก่อน และเมื่อทำ Generate MDX สำเร็จจึงจะสามารถทำการ View Result หรือ Edit MDX ได้ ซึ่งทั้งหมดสามารถเขียนความสัมพันธ์ในรูปแบบจำลองหน้าที่ของระบบที่ผู้ใช้ต้องการ ได้ดังรูปที่ 3.4



รูปที่ 3.3 Use case diagram ของโปรแกรม

จากรูปที่ 3.3 สามารถอธิบายรายละเอียดและขั้นตอนการทำงานของแต่ละหน้าที่ของระบบที่ต้องการได้ดังต่อไปนี้

1. Use case: Select Data Source

Name:	Select Data Source
Actor:	User
Description:	เป็น use case ที่ให้ผู้ใช้เลือกแหล่งข้อมูลที่จะทำงานในการเรียกดูข้อมูล
Related use case:	-
Pre-Condition:	-
Post-Condition:	แสดงโครงสร้างของลูกบาศก์ข้อมูลบนโครงสร้างต้นไม้

ขั้นตอนการทำงานของ Use case: Select Data Source

Actor Actions	System Responses
1. กรอกชื่อของเครื่อง Server และชื่อของฐานข้อมูล	
2. กดปุ่ม OK	3. เชื่อมต่อ server และฐานข้อมูล

2. Use case: Generate MDX

Name:	Generate MDX
Actor:	User
Description:	เป็น use case ที่ทำหน้าที่สร้างและแสดงเอ็มดีเอ็กซ์ออกมาทางจอภาพ
Related use case:	Select Data Source, View Data
Pre-Condition:	1. ต้องทำ Select Data Source สำเร็จ
Post-Condition:	ได้เอ็มดีเอ็กซ์ที่พร้อม execute query

ขั้นตอนการทำงานของ Use case: Generate MDX

Actor Actions	System Responses
1. เลือกมิติและค่าคำนวณที่ต้องการ	
2. กดปุ่ม Generate MDX	3. สร้างและแสดงเอ็มดีเอ็กซ์

3. Use case: View Cube Structure

Name:	View Cube Structure
Actor:	User
Description:	เป็น use case ที่ผู้ใช้ใช้ดูโครงสร้างของลูกบาศก์ข้อมูลว่าประกอบด้วยมิติ (dimensions) ค่าคำนวณ (measures) เป็นอย่างไร
Related use case:	Select Data Source
Pre-Condition:	1. ต้องทำ Select Data Source สำเร็จ
Post-Condition:	ขยายหรือยุบโครงสร้างต้นไม้แสดงรายละเอียดของลูกบาศก์ข้อมูล

ขั้นตอนการทำงานของ Use case: View Cube Structure

Actor Actions	System Responses
1. เลือกลูกบาศก์ข้อมูล	2. ระบบทำการโหลดข้อมูลใส่โครงสร้างต้นไม้
3. เลือกขยายหรือยุบในส่วนของโครงสร้างต้นไม้	4. แสดงข้อมูลเมื่อมีการขยายหรือยุบโครงสร้างต้นไม้

4. Use case: View Result

Name:	View Result
Actor:	User
Description:	เป็น use case ที่แสดงข้อมูลที่ได้จากการทำงานของคำสั่งเอ็มดีเอ็กซ์
Related use case:	Select Data Source, Generate MDX
Pre-Condition:	1. ต้องทำ Select Data Source สำเร็จ 2. ต้องทำ Generate MDX สำเร็จ
Post-Condition:	แสดงผลลัพธ์ของเอ็มดีเอ็กซ์ที่สร้างในรูปตารางข้อมูล

ขั้นตอนการทำงานของ Use case: View Data

Actor Actions	System Responses
1. กดปุ่ม View Data	2. สร้างหน้าแสดงข้อมูล

5. Use case: Edit MDX

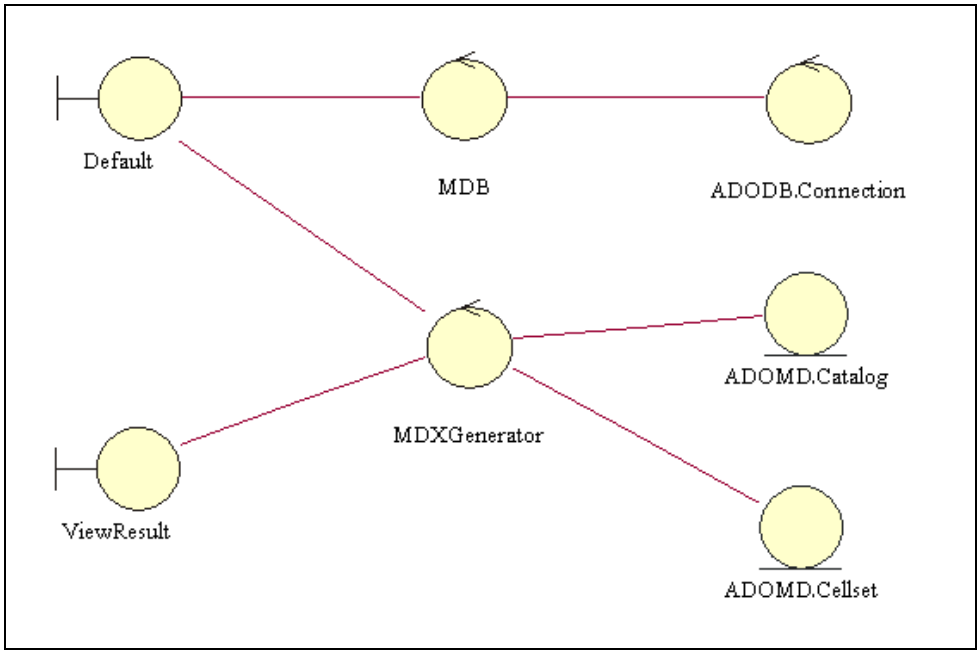
Name:	Edit MDX
Actor:	User
Description:	use case แก้ไขหรือเพิ่มเติมเอ็มดีเอ็กซ์ที่สร้างขึ้น อาจจะต้องการใช้เอ็มดีเอ็กซ์ที่ซับซ้อนกว่าที่โปรแกรมจะสามารถสร้างให้ได้ โดยแก้ไขผ่านทางตัวแก้ไขข้อความ (Text editor)
Related use case:	Select Data Source, Generate MDX
Pre-Condition:	1. ต้องทำ Select Data Source สำเร็จ 2. ต้องทำ Generate MDX สำเร็จ
Post-Condition:	แก้ไขหรือเพิ่มเติมเอ็มดีเอ็กซ์

ขั้นตอนการทำงานของ Use case: Edit MDX

Actor Actions	System Responses
1. กดปุ่ม Edit	
2. แก้ไขเอ็มดีเอ็กซ์ที่โปรแกรมสร้างขึ้น	

3.3.3 คลาสไดอะแกรม (Class diagram)

จาก Use case ที่ออกแบบไว้ในข้างต้นสามารถทำการออกแบบ Class diagram ต่อได้ โดยออกแบบ Class diagram ในลักษณะที่เรียกว่า Robustness diagram ที่มีการใช้สัญลักษณ์แทนประเภทของคลาสเพื่อง่ายต่อการทำความเข้าใจและนำไปพัฒนา โดยมีคลาสที่เป็นคลาสเบาน์ดารี คือ Default and ViewResult คลาสคอนโทรลเลอร์ คือ MDB and MDXGenerator และใช้คลาสของ ADOMD.NET ที่เป็น COM Web References เข้ามาในการเชื่อมต่อ ทำงานกับฐานข้อมูลและลูกบาศก์ข้อมูล ดังรูปที่ 3.4



รูปที่ 3.4 Class diagram (Robustness diagram)

จากรูปที่ 3.5 สามารถอธิบายได้ดังนี้ คลาส Default ทำหน้าที่เป็นหน้าหลักของโปรแกรม โดยเป็นส่วนที่ติดต่อกับผู้ใช้ เป็น user interface ที่เป็นหน้าเว็บไซต์แสดงข้อมูลของฐานข้อมูล

ลูกบาศก์ข้อมูล ข้อมูล dimension และสมาชิกของ dimension คลาส Default จะติดต่อกับคลาส MDB และคลาส MDXGenerator ที่เป็นคลาสคอนโทรลเลอร์ คลาส MDB ทำหน้าที่ติดต่อกับฐานข้อมูลโดยอาศัยการทำงานของคลาส ADOMD แล้วส่งค่าของฐานข้อมูลและลูกบาศก์ข้อมูลกับมาแสดงผลที่คลาส Default การทำงานของคลาส MDXGenerator จะรับข้อมูลจากการที่ผู้ใช้ได้เลือกข้อมูล dimension ต่าง ๆ ลงในส่วนที่ของ Columns, Rows and Slicer ที่เป็น ListBox Control จากนั้น MDXGenerator จะทำการสร้างเอ็มดีเอ็กซ์จากข้อมูลที่ผู้ใช้เลือกโดยอาศัยการทำงานของคลาส CubeDef ที่ใช้ทำงานในส่วนของโครงสร้างต่าง ๆ ของลูกบาศก์ข้อมูลและคลาส Cellset ทำงานเป็นอ็อบเจ็คที่เก็บผลลัพธ์ของการทำงานของเอ็มดีเอ็กซ์ แล้วส่งค่ากลับมาแสดงผลที่คลาส Default ส่วนคลาส ViewResult เป็นคลาสเบาน์ดารีที่ทำหน้าที่แสดงผลลัพธ์เป็นตารางข้อมูลที่ได้จากการทำงานของเอ็มดีเอ็กซ์ รายละเอียดของแต่ละคลาสคอนโทรลเลอร์แสดงได้ดังต่อไปนี้

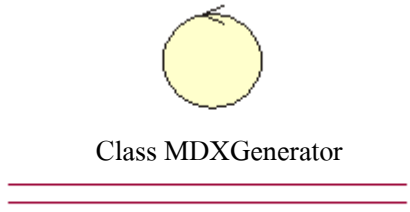
Class MDB

MDB

ConnectMDB(server : String) : ADOMD.ConnectionClass
ConnectMDB(server : String, databaseName : String) : ADOMD.ConnectionClass
DisconnectMDB(conObj : ADOMD.ConnectionClass) : Boolean
LoadTreeView(cubeName : String)

Methods	
	• **ConnectMDB()** Signature: ConnectMDB(server : String) : ADOMD.ConnectionClass Description: เมธอดเชื่อมต่อกับแหล่งข้อมูลโดยไม่ได้ระบุชื่อฐานข้อมูล • **ConnectMDB()** Signature: ConnectMDB(server : String, databaseName : String) : ADOMD.ConnectionClass Description: เมธอดเชื่อมต่อกับแหล่งข้อมูลโดยระบุฐานข้อมูล • **DisconnectMDB()** Signature: DisconnectMDB(conObj : ADOMD.ConnectionClass) : Boolean

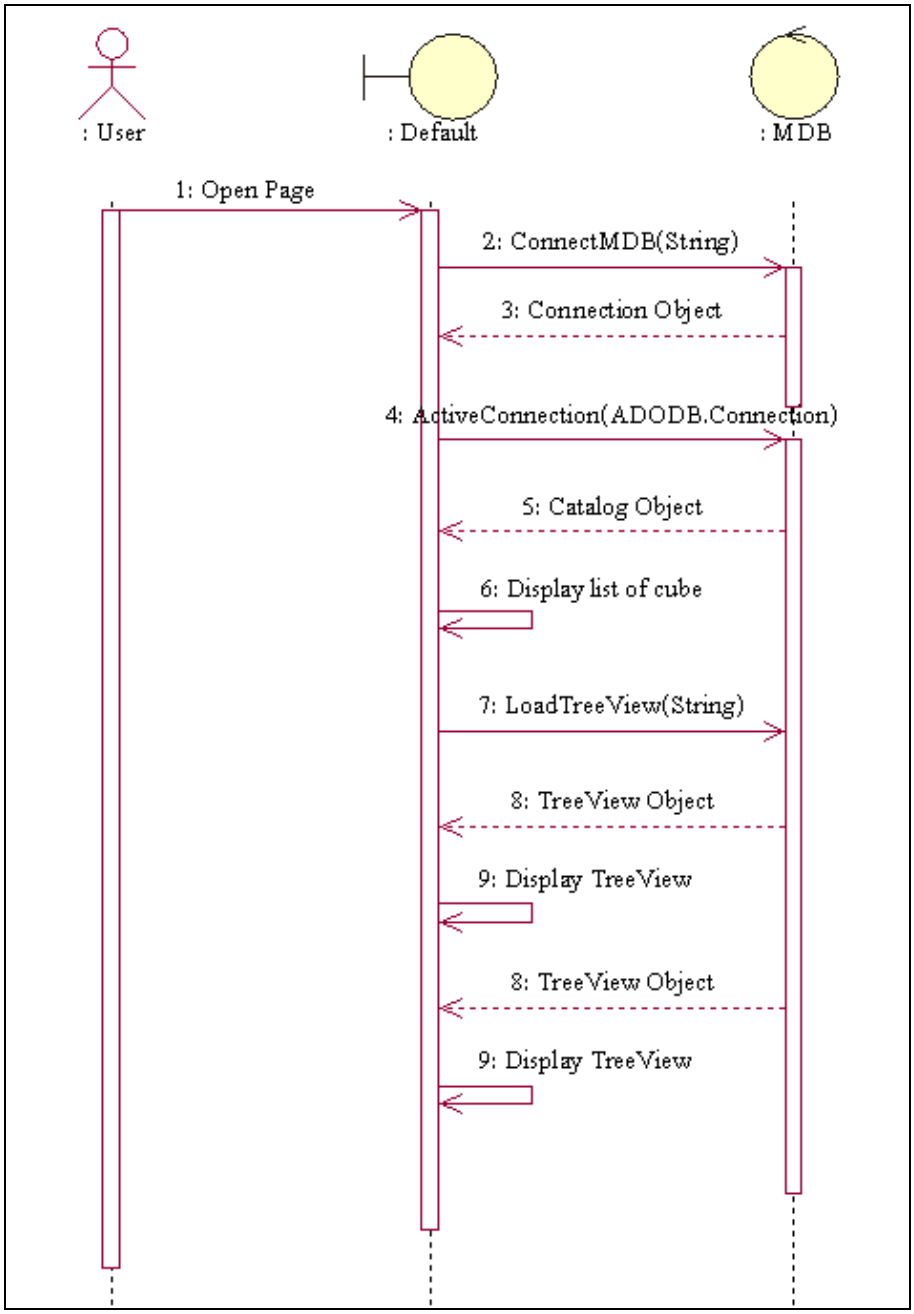
	Boolean Description: เมธอดหยุดการเชื่อมต่อกับแหล่งข้อมูล • **ActiveConnection(ADODB.Connection)** Signature: ActiveConnection(ADODB.Connection) : ADOMD.Catalog Description: เมธอดกำหนดสถานะการเชื่อมต่อกับแหล่งข้อมูล • **LoadTreeView()** Signature: LoadTreeView(cubeName : String) Description: เมธอดโหลดข้อมูลโครงสร้างลูกบาศก์ข้อมูลให้อยู่ในโครงสร้างต้นไม้

Class MDXGenerator	
	
Method	• **GenerateMDX()** Signature: GenerateMDX() : String Description: เมธอดสร้างเอ็มดีเอ็กซ์ • **Run()** Signature: Run(mdx : String) : CellSet Description: เมธอดสั่งเอ็มดีเอ็กซ์ทำงาน

3.3.4 ซีเควนซ์ไดอะแกรม (Sequence diagram)

แบบจำลองเชิงกิจกรรมของโปรแกรม MDX Online Generator อธิบายการทำงานของโปรแกรมตามแต่ละหน้าที่ของที่ผู้ใช้ต้องการได้ดังนี้

(1) Use case: Select Data Source

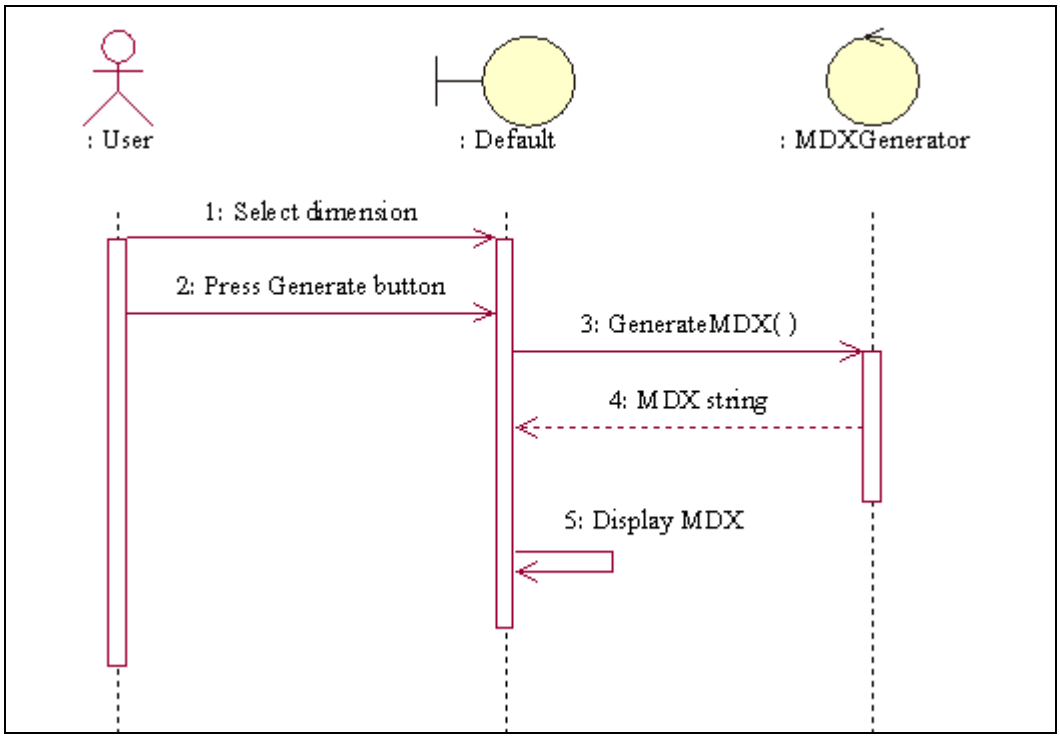


รูปที่ 3.5 แบบจำลองเชิงกิจกรรมการเลือกแหล่งข้อมูล

จากรูปที่ 3.5 สามารถอธิบายรายละเอียดขั้นตอนการทำงานของแบบจำลองเชิงกรรมได้ดังนี้

1. ผู้ใช้เปิดหน้า Default
2. ปรแกรมเปิดการเชื่อมต่อกับแหล่งข้อมูล โดยเรียกเมธอด ConnectMDB() จากคลาส MDB
3. คลาส MDB เรียกใช้เมธอด Open() จากคลาส ADODB.Connection เพื่อเปิดการเชื่อมต่อ
4. คลาส ADODB.Connection ส่ง Object การเชื่อมต่อกลับมาที่คลาส MDB
5. คลาส MDB เรียกใช้เมธอด OpenSchema() โดยผ่าน Object ที่ส่งกลับมา เพื่อเรียกข้อมูลฐานข้อมูล
6. คลาส ADODB.Connection ส่งข้อมูลฐานข้อมูลกลับมาในรูปของ RecordSet
7. คลาส MDB ส่ง RecordSet ให้หน้า Default
8. คลาส Default แสดงข้อมูลรายชื่อฐานข้อมูลที่หน้าเว็บ
9. เรียกใช้เมธอด ActiveConnection() เพื่อกำหนดการทำงานให้กับ Catalog object
10. เมธอด ActiveConnection() ส่ง Catalog object กลับมาให้คลาส Default
11. คลาส Default แสดงข้อมูลรายชื่อลูกบาศก์ข้อมูลที่หน้าเว็บ
12. คลาส Default เรียกเมธอด LoadTreeView() จากคลาส MDB เพื่อทำการโหลดข้อมูลโครงสร้างลูกบาศก์ใส่ในโครงสร้างต้นไม้
13. คลาส MDB ส่ง TreeView object กลับมาที่คลาส Default
14. คลาส Default แสดงโครงสร้างลูกบาศก์ข้อมูลบนโครงสร้างต้นไม้

(2) Use case: GenerateMDX

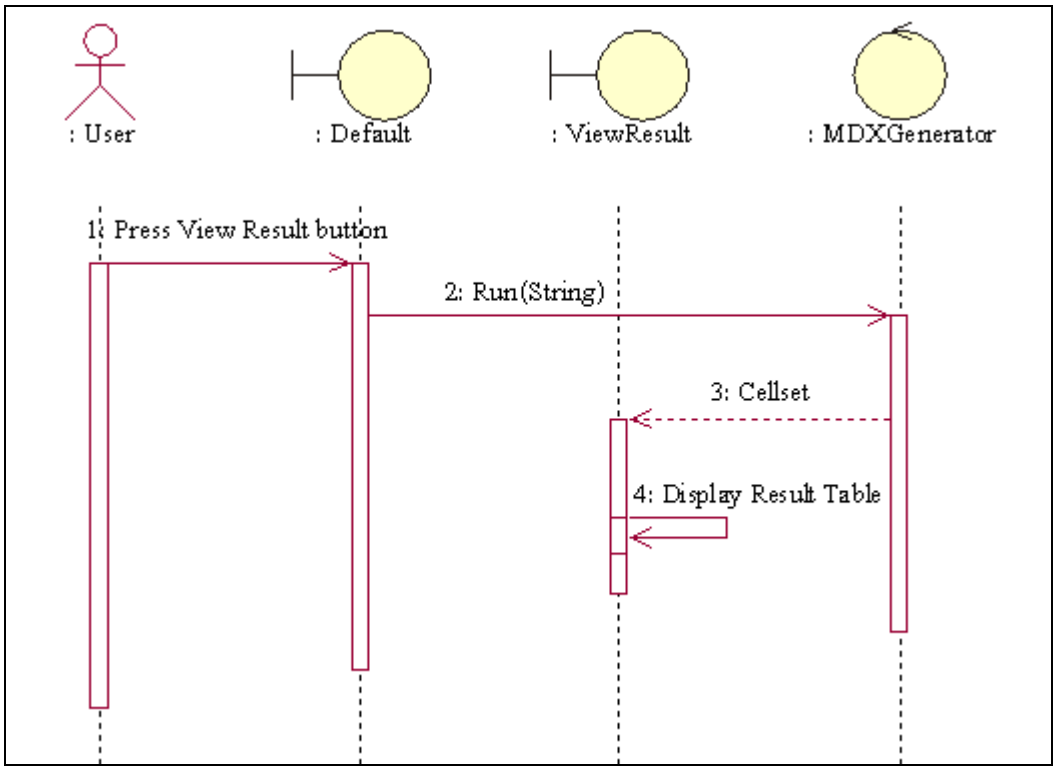


รูปที่ 3.6 แบบจำลองเชิงกิจกรรมการสร้างเอ็มดีเอ็กซ์

จากรูปที่ 3.6 สามารถอธิบายรายละเอียดขั้นตอนการทำงานของแบบจำลองเชิงกรรม ได้ดังนี้

1. ผู้ใช้เลือก dimension ต่าง ๆ ที่ต้องการ
2. ผู้ใช้คลิกปุ่ม Generate MDX
3. คลาส Default เรียกใช้เมธอด GenerateMDX() จากคลาส MDXGenerator
4. คลาส MDXGenerator ส่งเอ็มดีเอ็กซ์กลับมาที่คลาส Default
5. คลาส Default แสดงเอ็มดีเอ็กซ์ที่หน้าเว็บ

(3) Use case: ViewResult

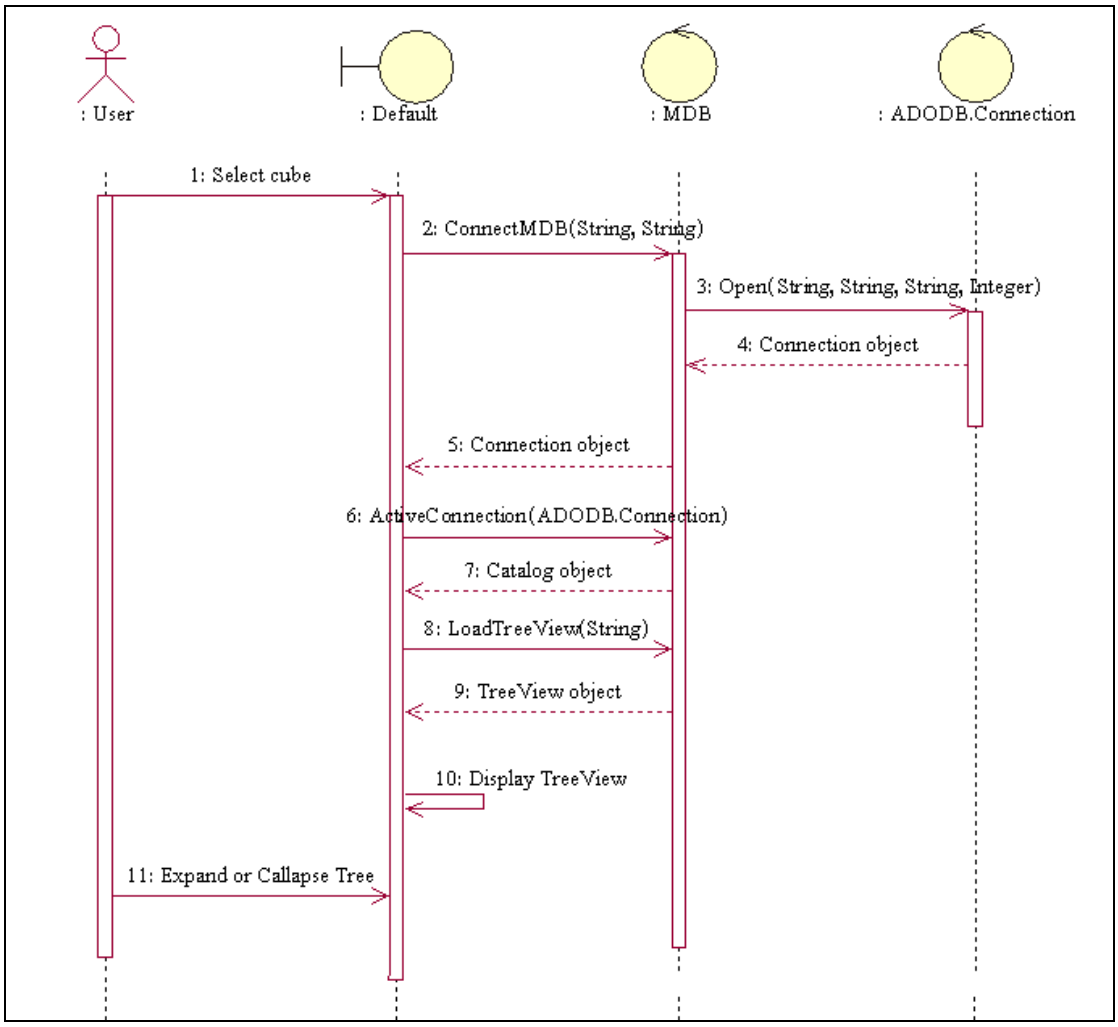


รูปที่ 3.7 แบบจำลองเชิงกิจกรรมการดูผลการทำงานของเอ็มดีเอ็กซ์

จากรูปที่ 3.7 สามารถอธิบายรายละเอียดขั้นตอนการทำงานของแบบจำลองเชิงกรรมได้ดังนี้

1. ผู้ใช้คลิกปุ่ม View Result
2. คลาส Default เรียกใช้เมธอด Run จากคลาส MDXGenerator
3. คลาส MDXGenerator ส่ง Cellset ผลลัพธ์จากการทำงานของเอ็มดีเอ็กซ์กลับมาให้คลาส ViewResult
4. หน้า viewResult แสดงตารางผลลัพธ์

(4) Use case: ViewStructure



รูปที่ 3.8 แบบจำลองเชิงกิจกรรมการดูโครงสร้างลูกบาศก์ข้อมูล

จากรูปที่ 3.8 สามารถอธิบายรายละเอียดขั้นตอนการทำงานของแบบจำลองเชิงกรรมได้ดังนี้

1. ผู้ใช้เลือกชื่อลูกบาศก์ข้อมูล
2. คลาส Default เรียกเมธอด ConnectionMDB() จากคลาส MDB โดยส่งชื่อลูกบาศก์ข้อมูลไปด้วย
3. คลาส MDB เรียกใช้เมธอด Open() จากคลาส ADODB.Connection เพื่อกำหนดการเชื่อมต่อ

4. คลาส ADODB.Connection ส่ง Connection object กลับมาที่คลาส MDB

5. คลาส MDB ส่ง Connection object กลับมาที่คลาส Default

6. คลาส Default เรียกใช้เมธอด ActiveConnection() จากคลาส MDB เพื่อกำหนดการทำงานให้กับ Catalog object

7. เมธอด ActionConnection() ส่ง Catalog object กลับมาให้คลาส Default

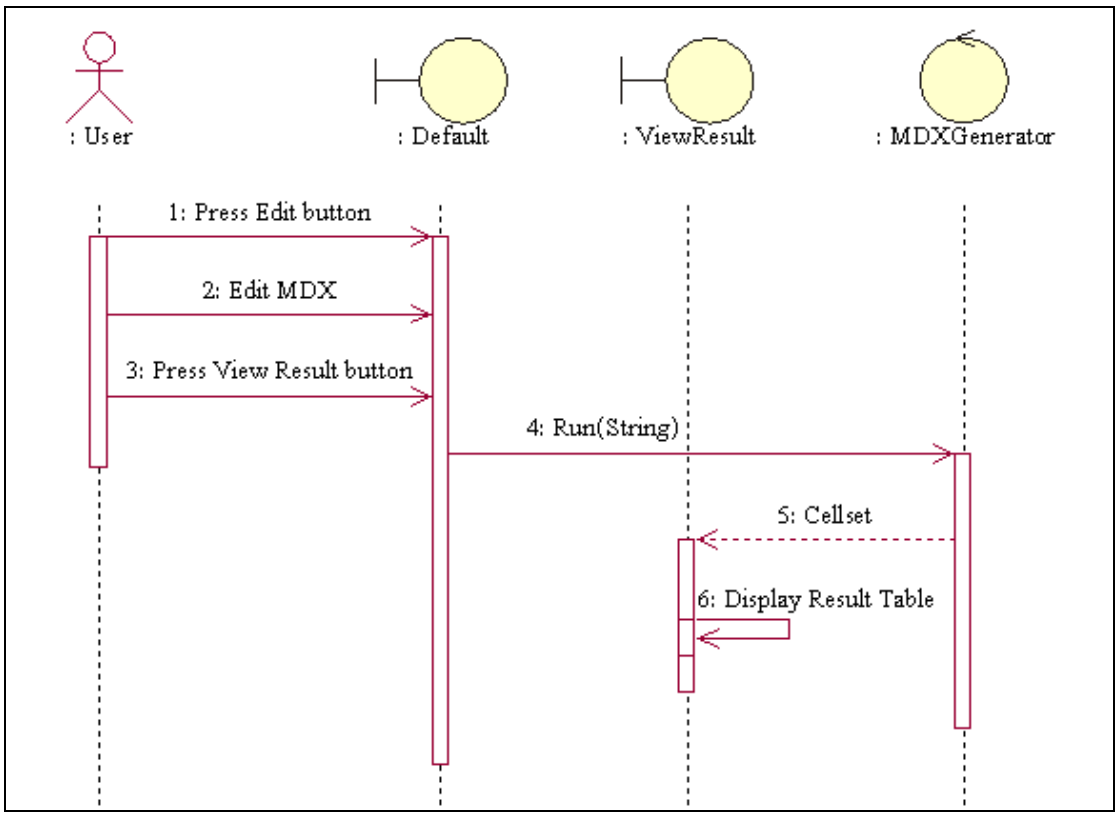
8. คลาส Default เรียกเมธอด LoadTreeView() จากคลาส MDB เพื่อทำการโหลดข้อมูลโครงสร้างลูกบาศก์ใส่ในโครงสร้างต้นไม้

9. คลาส MDB ส่ง TreeView object กลับมาที่คลาส Default

10. คลาส Default แสดงโครงสร้างลูกบาศก์ข้อมูลบนโครงสร้างต้นไม้

11. ผู้ใช้เลือกกระจายหรือยุบโครงสร้างต้นไม้เพื่อดูรายละเอียด

(5) Use case: EditMDX



รูปที่ 3.9 แบบจำลองเชิงกิจกรรมการแก้ไขเอ็มดีเอ็กซ์

จากรูปที่ 3.9 สามารถอธิบายรายละเอียดขั้นตอนการทำงานของแบบจำลองเชิงกรรม ได้ดังนี้

1. ผู้ใช้คลิกปุ่ม Edit
2. แก้ไขเอ็มดีเอ็กซ์ในช่อง TextBox
3. คลิกปุ่ม View Result
4. คลาส Default เรียกใช้เมธอด Run() จากคลาส MDXGenerator
5. คลาส MDXGenerator ส่งค่า Cellset ผลลัพธ์การทำงานเอ็มดีเอ็กซ์ไปที่คลาส ViewResult
6. คลาส ViewResult แสดงผลลัพธ์ในรูปของตารางข้อมูล

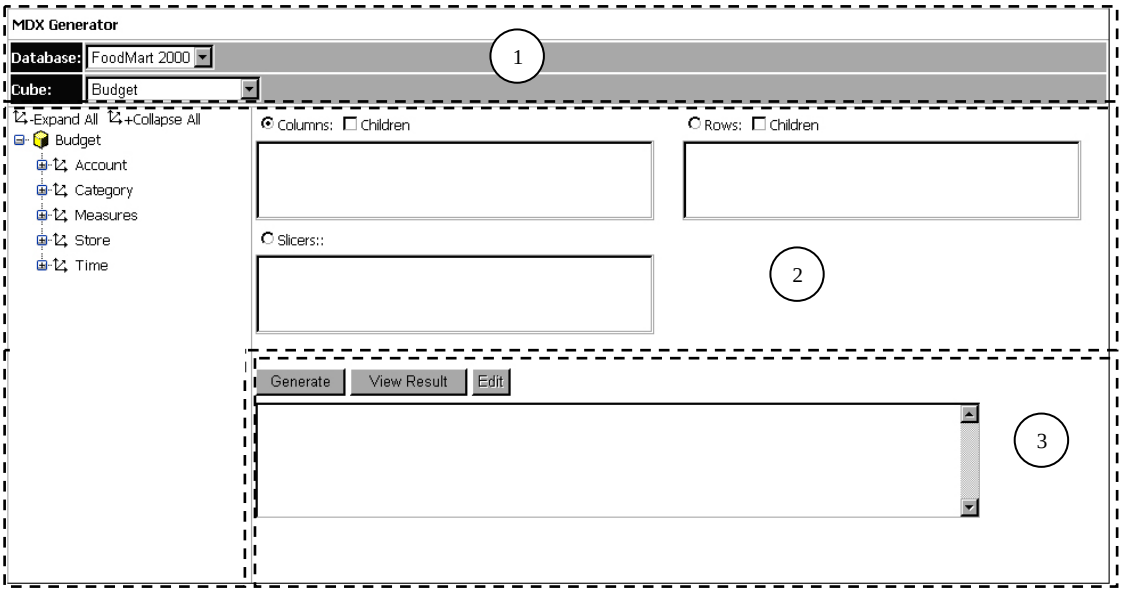
3.4 การออกแบบหน้าตาโปรแกรม (User Interface Design)

การพัฒนาโปรแกรม Online MDX Generator โดยนำการออกแบบในส่วนของ Class diagram และ Sequence diagram มาพัฒนาพร้อมกับการออกแบบหน้าตาของโปรแกรม (User interface) เป็นหน้าเว็บ การออกแบบหน้าตาของโปรแกรมเพื่อใช้ติดต่อตอบโต้กับผู้ใช้จะใช้การติดต่อกับผู้ใช้ผ่านทางหน้าเว็บเพจที่มีส่วนประกอบต่าง ๆ แบ่งเป็นส่วนตามการวิเคราะห์จาก Use case การออกแบบหน้าตาโปรแกรมที่จะสามารถจัดการกับการใช้ของผู้ใช้เพื่อการกำหนดพารามิเตอร์ต่าง ๆ ของ query และการสร้างผลลัพธ์ของการสร้างเอ็มดีเอ็กซ์ User interface ของโปรแกรม แบ่งออกเป็นส่วนต่าง ๆ ตามหน้าที่การทำงาน 3 ส่วน

ส่วนที่ 1: การเชื่อมต่อกับแหล่งข้อมูลโดยจะรับชื่อแหล่งข้อมูลเพื่อใช้ในการเชื่อมต่อกับแหล่งข้อมูล (ในที่นี้กำหนดค่าเริ่มต้นให้เชื่อมต่อที่ localhost)

ส่วนที่ 2: การจัดการเกี่ยวกับการเลือกมิติ (dimension) เพื่อสร้างเอ็มดีเอ็กซ์ โดยกำหนดให้ผู้ใช้สามารถเลือก dimension จากโครงสร้างต้นไม้ลงในส่วนของ Columns, Rows or Slicers

ส่วนที่ 3: การจัดการกับเอ็มดีเอ็กซ์ เช่น การสั่งให้เอ็มดีเอ็กซ์ทำงาน การดูตัวอย่างข้อมูลจากการทำงานของเอ็มดีเอ็กซ์ การแก้ไขเอ็มดีเอ็กซ์ แสดงได้ดังรูปที่ 3.11



รูปที่ 3.10 User interface ของโปรแกรม Online MDX Generator

บทที่ 4
การทดสอบและอภิปรายผล

ในบทนี้กล่าวถึงการทดสอบโปรแกรมสร้างเอ็มดีเอ็กซ์ออนไลน์บนบราวเซอร์ โดยในหัวข้อที่ 4.1 จะกล่าวถึงสภาพแวดล้อมที่ใช้ในการทดสอบ การทดสอบเริ่มจากการเชื่อมต่อแหล่งข้อมูล ฐานข้อมูล ลูกบาศก์ข้อมูล และแสดงผลสู่ผู้ใช้ทางหน้าเว็บ ปรากฏรายละเอียดในหัวข้อที่ 4.2 การสร้างเอ็มดีเอ็กซ์จากตัวอย่างรายงานจากฐานข้อมูลทดสอบ FoodMarts 2000 เพื่อทดสอบว่าสามารถช่วยอำนวยความสะดวกให้แก่ผู้ใช้ได้มากน้อยเพียงใดปรากฏรายละเอียดในหัวข้อที่ 4.3 และหัวข้อที่ 4.4 จะเป็นการอภิปรายผลเปรียบเทียบการทำงานระหว่างโปรแกรมสร้างเอ็มดีเอ็กซ์จากงานวิจัยนี้กับโปรแกรมจากงานวิจัย OLAP Realization Technology Research Based on MDX ของ Wen-Yuan Liu กับ Shu-Fen Fang ในแง่ของความยืดหยุ่นของรูปแบบความสามารถของการสร้างเอ็มดีเอ็กซ์

4.1 สภาพแวดล้อมที่ใช้ในการทดสอบ

เพื่อให้การทดสอบนี้เป็นเหมือนการใช้งานจริงผ่านเครือข่าย ซึ่งเครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบระบบมีรายละเอียดดังนี้

1) เครื่องให้บริการ (Server)

ฮาร์ดแวร์

- คอมพิวเตอร์แบบพีซี Pentium IV 3.2 กิกะเฮิร์ส
- หน่วยความจำ 512 เมกกะไบต์
- ฮาร์ดดิสก์ 80 กิกะไบต์

ซอฟต์แวร์

- ระบบปฏิบัติการ Windows Server 2003
- เว็บเซิร์ฟเวอร์ Internet Information Services
- ฐานข้อมูล Microsoft SQL Server 2000
- โปรแกรม Analysis Service
- ภาษาโปรแกรมมิง ASP.NET (C#)

2) เครื่องรับบริการ (Client)

ฮาร์ดแวร์

- คอมพิวเตอร์แบบพีซี Pentium IV 3.2 กิกะเฮิร์ส
- หน่วยความจำ 512 เมกกะไบต์
- ฮาร์ดดิสก์ 80 กิกะไบต์

ซอฟต์แวร์

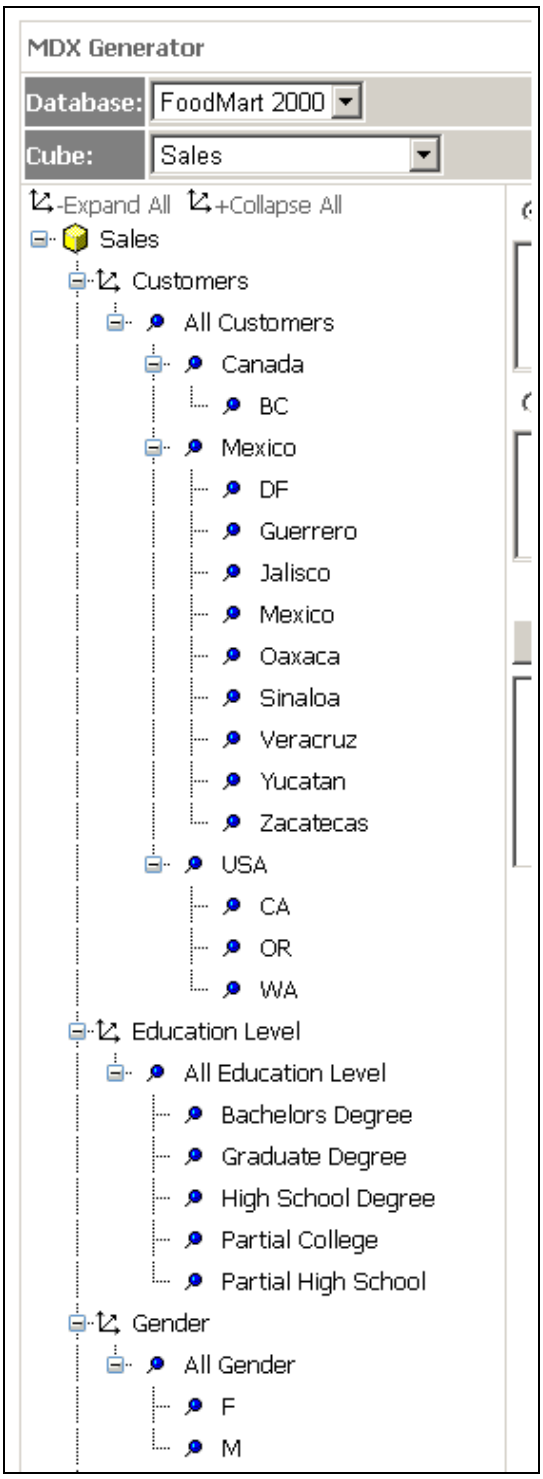
- ระบบปฏิบัติการ ไมโครซอฟต์วินโดวส์ เอ็กซ์พี
- เว็บเบราว์เซอร์ Internet Explorer 6.0

4.2 การทดสอบการเชื่อมต่อแหล่งข้อมูล

การเชื่อมต่อแหล่งข้อมูลจะใช้ COM (Component Object Model) ที่ชื่อว่า ADOMD.NET เป็นอินเตอร์เฟสช่วยในการเชื่อมต่อกับแหล่งข้อมูลที่เป็นฐานข้อมูลแบบหลายมิติ โดยได้ทำการสร้างเมธอดขึ้นมาเพื่อสร้างการเชื่อมต่อ เมธอดจะรับอาร์กิวเมนต์ 2 ค่า คือ ชื่อเครื่องให้บริการ (server) กับชื่อฐานข้อมูลแล้วจะส่งค่าอ็อบเจ็คการเชื่อมต่อกลับมาให้ ดังตัวอย่างส่วนของโค้ดต่อไปนี้

```
// Class MDB
using  ADOMD;
using  ADODB;
….
ADODB.Connection con;
public ADODB.Connection Connect(String server,String database)
{
constr = "Datasource=" + server + "; Provider=msolap; Initial Catalog=" + database +
";";
 try
{
con = new ADODB.Connection();
con.Open(constr, "sa", "", 0);
}catch(Exception ex){}
 return con;  }
```

โปรแกรมจะทำการเชื่อมต่อกับแหล่งข้อมูลอัตโนมัติเมื่อเริ่มทำงานแล้วดึงข้อมูลของลูกบาศก์ข้อมูลมาแสดงในรูปแบบโครงสร้างต้นไม้ดังรูปที่ 4.1



รูปที่ 4.1 การแสดงโครงสร้างลูกบาศก์ข้อมูลบนโครงสร้างต้นไม้

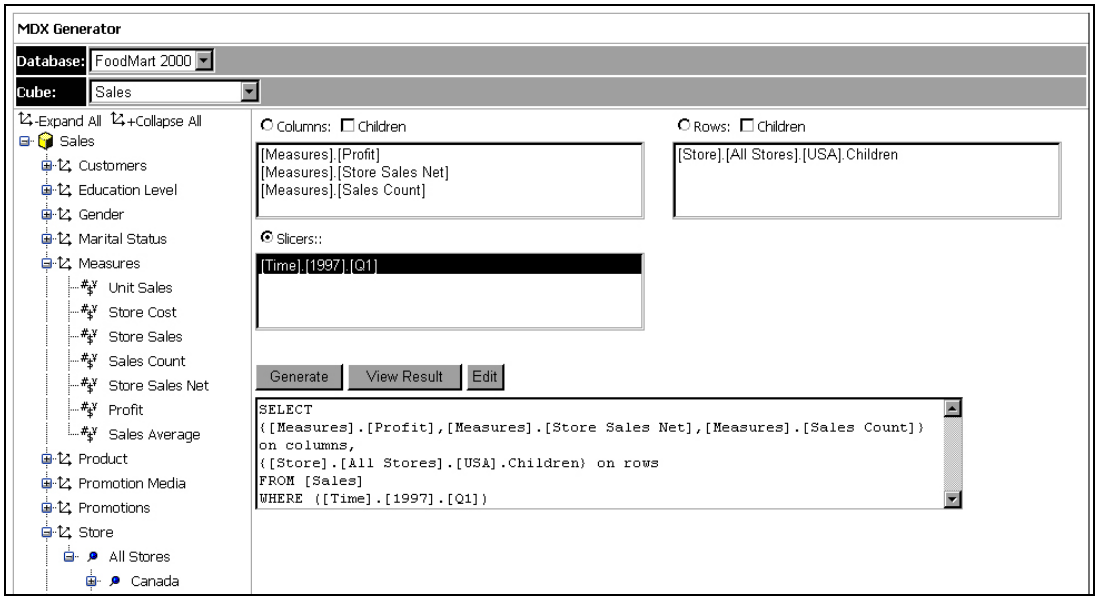
4.3 การทดสอบการสร้างเอ็มดีเอ็กซ์

ผลการทดสอบสร้างเอ็มดีเอ็กซ์ด้วยการสร้างรายงานตัวอย่าง จากฐานข้อมูลตัวอย่าง FoodMart 2000 โดยได้ออกแบบและสร้างด้วยโปรแกรมสร้างเอ็มดีเอ็กซ์ออนไลน์บนบราวเซอร์ ที่มาของเอ็มดีเอ็กซ์ตัวอย่างได้มาจากบทความ MDX samples of foodmart เป็นการวิเคราะห์ในมุมมองเชิงธุรกิจจริง ดังต่อไปนี้

รายงานที่ 1 รายงานวิเคราะห์ข้อมูลสรุปผลกำไร ยอดขาย และหน่วยการขายของร้านค้าของรัฐต่าง ๆ ในสหรัฐอเมริกาในปี 1997 ไตรมาสที่ 1 การทำงานโดยเลือกมิติข้อมูล Measure Profit, Store Sales Net and Sale Count ลงในส่วนคอลัมน์ จากนั้นเลือกมิติข้อมูล Store USA ลงในส่วนของแถว แล้วเลือกมิติข้อมูล Time ปี 1997 ไตรมาสที่ 1 ลงในส่วนของการกรองข้อมูล สามารถแสดงผลลัพธ์การทำงานของโปรแกรมแสดงได้ดังรูปที่ 4.2 และผลลัพธ์ที่ได้จากทำงานของเอ็มดีเอ็กซ์นี้ ดังรูปที่ 4.3

MDX:

```
SELECT
{[Measures].[Profit],[Measures].[Store Sales Net],[Measures].[Sales Count]} on columns,
{[Store].[All Stores].[USA].Children} on rows
FROM [Sales]
WHERE ([Time].[1997].[Q1])
```



รูปที่ 4.2 หน้าโปรแกรมการสร้างเอ็มดีเอ็กซ์รายงานที่

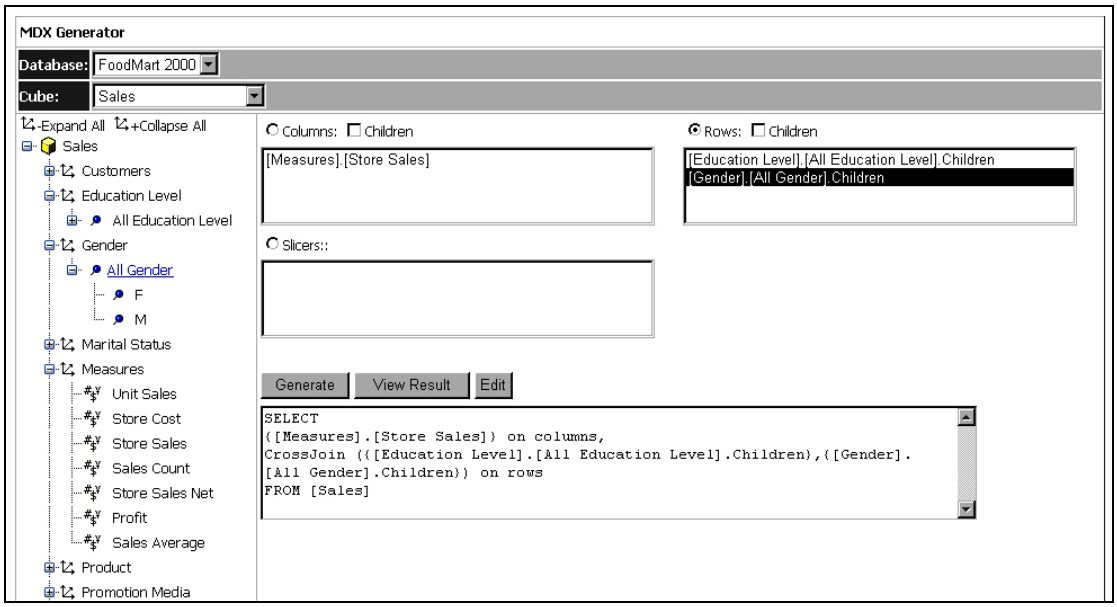
	Profit	Store Sales Net	Sales Count
CA	21,744.11	21,744.11	5498
OR	24,089.22	24,089.22	6184
WA	38,042.78	38,042.78	9906

รูปที่ 4.3 ผลลัพธ์การทำงานของเอ็มดีเอ็กซ์ที่ 1

รายงานที่ 2 รายงานวิเคราะห์ข้อมูลการซื้อของลูกค้าตามระดับการศึกษาและเพศ การทำงานโดยเลือกมิติข้อมูล Measure Store Sales ลงในส่วนคอลัมน์ จากนั้นเลือกมิติข้อมูล Education Level และมิติข้อมูล Gender ลงในส่วนของแถว สามารถแสดงผลลัพธ์การทำงานของโปรแกรมแสดงได้ดังรูปที่ 4.4 และผลลัพธ์ที่ได้จากทำงานของเอ็มดีเอ็กซ์นี้ ดังรูปที่ 4.5

MDX:

```
SELECT
{[Measures].[Store Sales]} on columns,
CrossJoin ({[Education Level].[All Education Level].Children},
{[Gender].[All Gender].Children}) on rows
FROM [Sales]
```



รูปที่ 4.4 หน้าโปรแกรมการสร้างเอ็มดีเอ็กซ์รายงานที่ 2

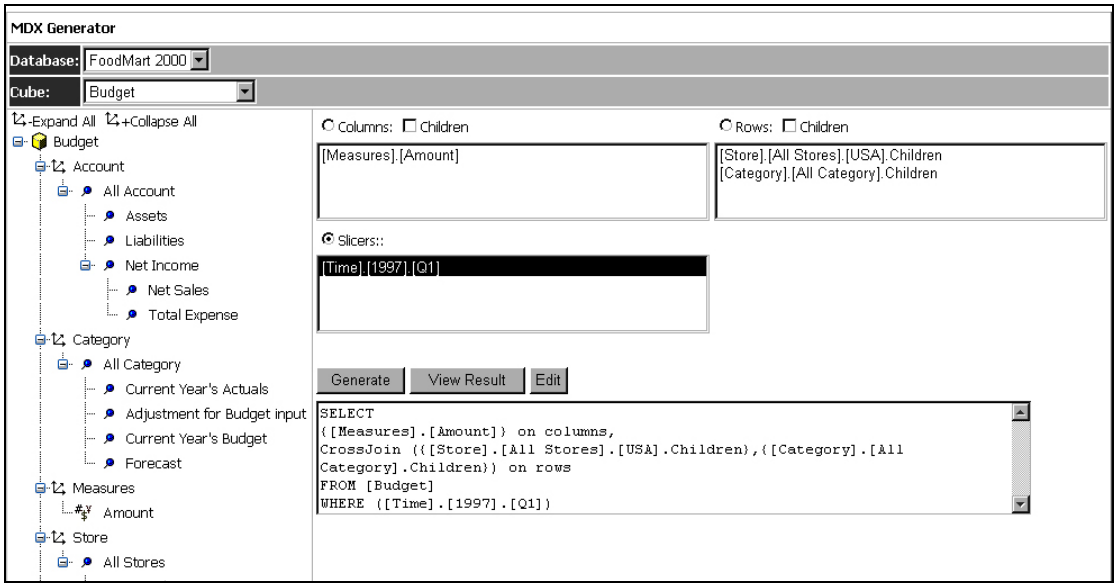
		Store Sales
Bachelors Degree	F	$72,119.26
	M	$73,386.15
Graduate Degree	F	$17,641.64
	M	$15,254.55
High School Degree	F	$81,112.23
	M	$85,627.41
Partial College	F	$27,175.97
	M	$25,183.22
Partial High School	F	$82,177.11
	M	$85,560.59

รูปที่ 4.5 ผลลัพธ์การทำงานของเอ็มดีเอ็กซ์ที่ 2

รายงานที่ 3 รายงานวิเคราะห์ข้อมูลตามรายการแนะนำสินค้าแต่ละประเภทสินค้ากับผลกำไร การทำงานโดยเลือกมิติข้อมูล Measure Amount ลงในส่วนคอลัมน์ จากนั้นเลือกมิติข้อมูล Store USA มิติข้อมูล Category ลงในส่วนของแถว แล้วเลือกมิติข้อมูล Time ปี 1997 ไตรมาสที่ 1 ลงในส่วนของการกรองข้อมูล สามารถแสดงผลลัพธ์การทำงานของโปรแกรมแสดงได้ดังรูปที่ 4.6 และผลลัพธ์ที่ได้จากทำงานของเอ็มดีเอ็กซ์นี้ ดังรูปที่ 4.7

MDX:

```
SELECT
{[Measures].[Amount]} on columns,
CrossJoin ({[Store].[All Stores].[USA].Children},{[Category].[All Category].Children}) on rows
FROM [Budget]
WHERE ([Time].[1997].[Q1])
```



รูปที่ 4.6 หน้าโปรแกรมการสร้างเอ็มดีเอ็กซ์รายงานที่ 3

		Amount
CA	Current Year's Actuals	$14,408.28
	Adjustment for Budget input	$36,175.20
	Current Year's Budget	$36,175.20
	Forecast	$36,175.20
OR	Current Year's Actuals	$33,776.57
	Adjustment for Budget input	$40,170.29
	Current Year's Budget	$40,170.29
	Forecast	$40,170.29
WA	Current Year's Actuals	$49,883.93
	Adjustment for Budget input	$63,282.86
	Current Year's Budget	$63,282.86
	Forecast	$63,282.86

รูปที่ 4.7 ผลลัพธ์การทำงานของเอ็มดีเอ็กซ์ที่ 3

รายงานที่ 4 รายงานแสดงประเภทสินค้าที่ทำยอดขาย 10 อันดับแรก การทำงานโดยเลือกมิติข้อมูล Measure Unit Sales ลงในส่วนคอลัมน์ จากนั้นเลือกมิติข้อมูล Product Food ลงในส่วนของแถว แล้วคลิกปุ่ม Generate เพื่อสร้างเอ็มดีเอ็กซ์แล้วคลิกปุ่ม Edit เพื่อแก้ไขเอ็มดีเอ็กซ์โดยใช้งานฟังก์ชัน TopCount() ที่ทำงานกรองข้อมูลตามจำนวนที่กำหนด สามารถแสดงผลลัพธ์การทำงานของโปรแกรมแสดงได้ดังรูปที่ 4.8 และผลลัพธ์ที่ได้จากทำงานของเอ็มดีเอ็กซ์นี้ ดังรูปที่ 4.9

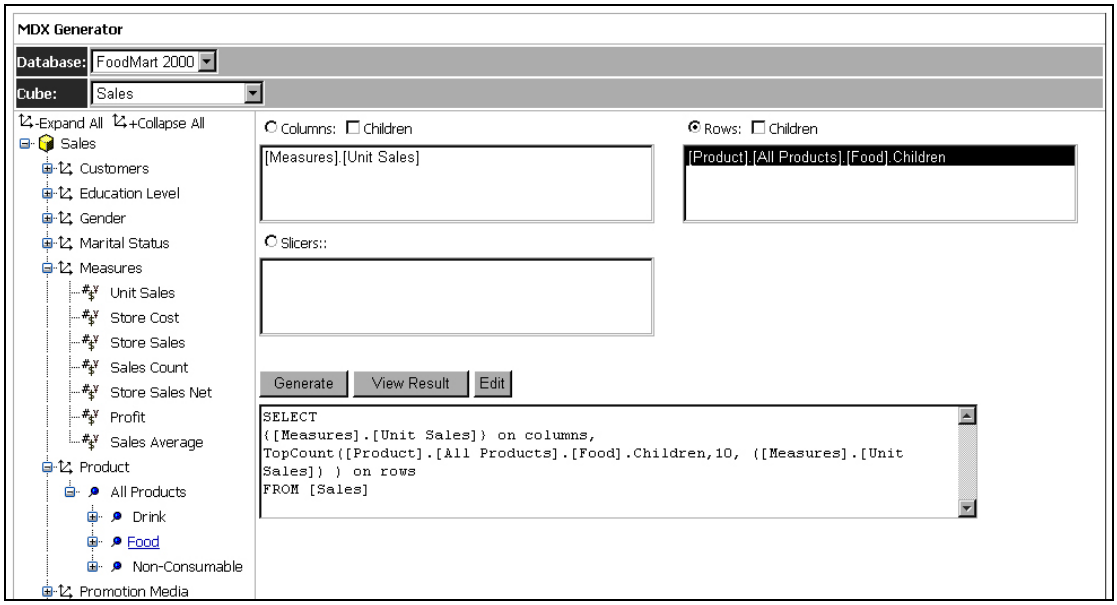
MDX:

SELECT {[Measures].[Unit Sales]} on columns,

TopCount([Product].[Product Category].Members, 10, ([Measures].[Unit Sales])) on rows

FROM [Sales]

WHERE ([Time].[1997].[Q1])



รูปที่ 4.8 หน้าโปรแกรมการสร้างเอ็มดีเอ็กซ์รายงานที่ 4

	Unit Sales
Produce	37,792.00
Snack Foods	30,545.00
Frozen Foods	26,655.00
Baking Goods	20,245.00
Canned Foods	19,026.00
Dairy	12,885.00
Deli	12,037.00
Baked Goods	7,870.00
Snacks	6,884.00
Starchy Foods	5,262.00

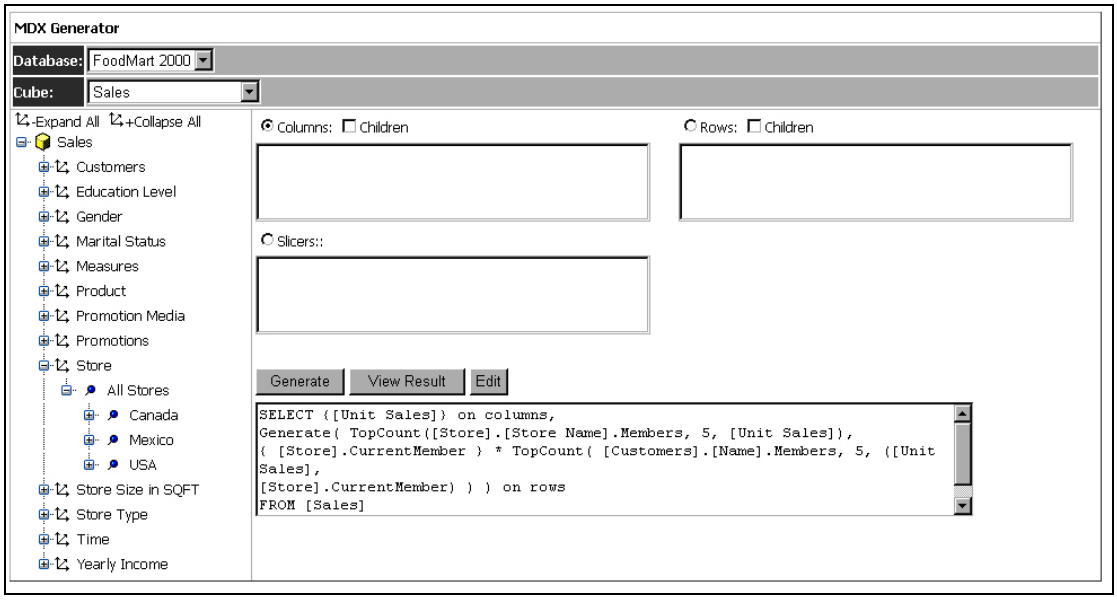
รูปที่ 4.9 ผลลัพธ์การทำงานของเอ็มดีเอ็กซ์ที่ 4

รายงานที่ 5 รายงานลูกค้า 5 อันดับแรกของแต่ละร้านค้า การทำงานเป็นการทดสอบการทำงานส่วนของการสร้างรายงานในรูปของตารางข้อมูลที่ได้จากเอ็มดีเอ็กซ์ที่มีความซับซ้อนโดยอาศัยโหมด Edit ด้วยการกรอกคำสั่งที่ใช้ฟังก์ชันต่าง ๆ ของเอ็มดีเอ็กซ์ลงในช่องรับข้อความที่เตรียมไว้

MDX:

```
SELECT {[Unit Sales]} on columns,
Generate( TopCount([Store].[Store Name].Members, 5, [Unit Sales]),
{ [Store].CurrentMember } * TopCount( [Customers].[Name].Members, 5, ([Unit Sales],
[Store].CurrentMember) ) ) on rows
FROM [Sales]
```

สามารถแสดงผลลัพธ์การทำงานของโปรแกรมแสดงได้ดังรูปที่ 4.10 และผลลัพธ์ที่ได้จากทำงานของเอ็มดีเอ็กซ์นี้ ดังรูปที่ 4.11



รูปที่ 4.10 หน้าโปรแกรมการสร้างเอ็มดีเอ็กซ์รายงานที่ 5

		Unit Sales
Store 13	Bobby Lundin	229.00
	Timothy Beard	213.00
	Barbara Chan	195.00
	Erin Smith	191.00
	LeeAnn Otte	186.00
Store 17	Jeremy Styers	337.00
	Patricia Gervasi	313.00
	William Wade	277.00
	Diane Brown	261.00
	Tammy Wilkinson	261.00
Store 11	Janeen Broxton	178.00
	Jim Velasquez	131.00
	Rosemary Stephens	129.00
	Patrick Arroyo	128.00
	Rory Wren	128.00
Store 7	Andy Schilleroff	89.00
	Michael Sample	87.00
	Lillian McCormick	84.00
	Beatrix Stevens	79.00
	Terry Schumer	77.00
Store 24	Lawrence Villamar	102.00
	Monica Ardon	101.00
	Edward Keith	98.00
	John Truszkowski	98.00
	Yolanda Bowers	94.00

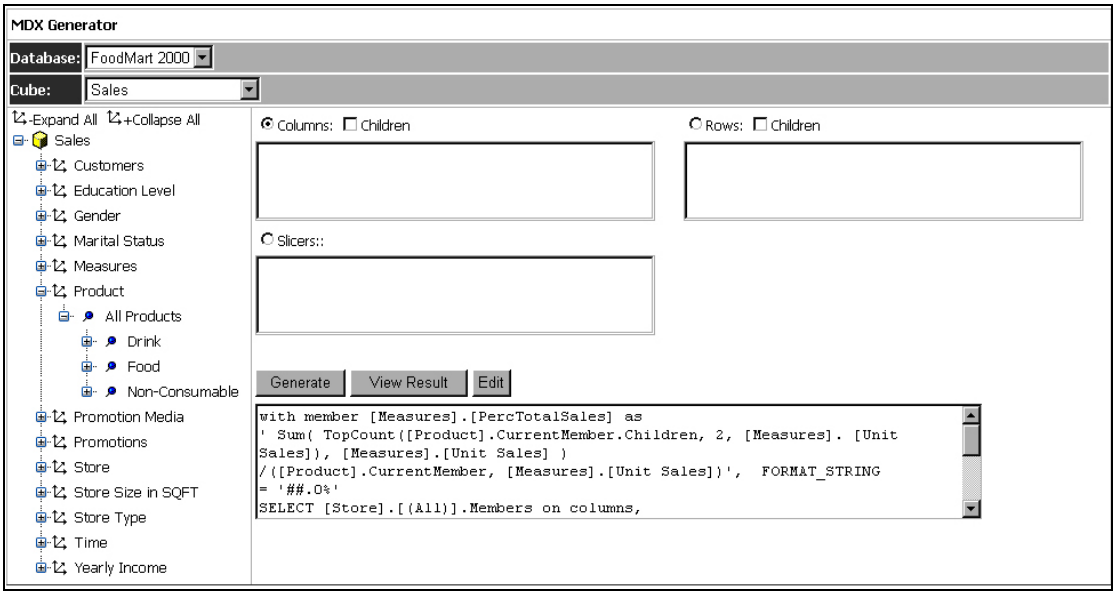
รูปที่ 4.11 ผลลัพธ์การทำงานของเอ็มดีเอ็กซ์ที่ 5

รายงานที่ 6 รายงานสินค้าที่มียอดขายสูงสุด 2 อันดับของแต่แบรนด์ยี่ห้อ ใช้ทดสอบการสร้างตารางข้อมูลของผลลัพธ์ที่ได้มาจากการทำงานของเอ็มดีเอ็กซ์ มีการใช้ฟังก์ชันกับข้อมูลที่ทำให้ผลลัพธ์มีลักษณะการนำเสนอในมุมมองมากกว่าสองมิติข้อมูล ซึ่งโปรแกรมสามารถสร้างตารางข้อมูลจากอ็อบเจ็ค cellset ผลลัพธ์ได้เป็นอย่างดี

MDX:

```
with member [Measures].[PercTotalSales] as
' Sum( TopCount([Product].CurrentMember.Children, 2, [Measures]. [Unit Sales]), [Measures].[
Unit Sales] )
/([Product].CurrentMember, [Measures].[Unit Sales])',  FORMAT_STRING = '##.0%'
 SELECT [Store].[(All)].Members on columns,
Generate( [Product].[Brand Name].Members,
Union(
TopCount( [Product].CurrentMember.Children, 2, [Measures].[Unit Sales] )
* {[Measures].[Unit Sales]}, { ([Product].CurrentMember, [Measures].[PercTotalSales]) }
)  ) on rows
FROM [Sales]
```

สามารถแสดงผลลัพธ์การทำงานของโปรแกรมแสดงได้ดังรูปที่ 4.12 และผลลัพธ์ที่ได้จากทำงานของเอ็มดีเอ็กซ์นี้ ดังรูปที่ 4.13



รูปที่ 4.12 หน้าโปรแกรมการสร้างเอ็มดีเอ็กซ์รายงานที่ 6

		All Stores
Good Imported Beer	Unit Sales	154.00
Good Light Beer	Unit Sales	115.00
Good	PercTotalSales	100.0%
Pearl Light Beer	Unit Sales	210.00
Pearl Imported Beer	Unit Sales	175.00
Pearl	PercTotalSales	100.0%
Portsmouth Imported Beer	Unit Sales	187.00
Portsmouth Light Beer	Unit Sales	175.00
Portsmouth	PercTotalSales	100.0%
Top Measure Light Beer	Unit Sales	161.00
Top Measure Imported Beer	Unit Sales	145.00
Top Measure	PercTotalSales	100.0%
Walrus Light Beer	Unit Sales	187.00
Walrus Imported Beer	Unit Sales	174.00
Walrus	PercTotalSales	100.0%
Good Chardonnay	Unit Sales	192.00
Good Chablis Wine	Unit Sales	163.00
Good	PercTotalSales	37.6%
Pearl Merlot Wine	Unit Sales	228.00
Pearl Chardonnay Wine	Unit Sales	210.00
Pearl	PercTotalSales	36.3%
Portsmouth Chardonnay Wine	Unit Sales	214.00
Portsmouth Merlot Wine	Unit Sales	184.00
Portsmouth	PercTotalSales	39.2%
Top Measure Merlot Wine	Unit Sales	186.00
Top Measure Light Wine	Unit Sales	172.00
Top Measure	PercTotalSales	37.6%

รูปที่ 4.13 ผลลัพธ์การทำงานของเอ็มดีเอ็กซ์ที่ 6

จากตัวอย่างรายงานที่ 4 ถึงตัวอย่างรายงานที่ 6 ที่ใช้ในการทดสอบการทำงานของโปรแกรมจะมีการใช้ฟังก์ชันต่าง ๆ ในการทำงาน โดยจะใช้การทำงานโหมด Edit สำหรับการ

ใช้เอ็มดีเอ็กซ์เพื่อให้การทำงานมีความยืดหยุ่น สามารถสร้างเอ็มดีเอ็กซ์เรียกดูข้อมูลได้หลากหลาย และโปรแกรมยังสามารถดึงข้อมูลมาแสดงในรูปแบบของตารางข้อมูลบนหน้าเว็บได้

4.4 อภิปรายผล

เมื่อผ่านการทดสอบเรียบร้อยได้ผลการทำงานของโปรแกรมประยุกต์สร้างเอ็มดีเอ็กซ์บนบราวเซอร์ พบว่าโปรแกรมสามารถอำนวยความสะดวกแก่ผู้ใช้ที่ไม่มีความเชี่ยวชาญเกี่ยวกับการใช้คำสั่งเอ็มดีเอ็กซ์ได้เป็นอย่างดี โดยกระบวนการทั้งหมดเริ่มต้นตั้งแต่การเชื่อมต่อกับแหล่งข้อมูล แสดงรายละเอียดต่าง ๆ จนได้ผลลัพธ์เป็นเอ็มดีเอ็กซ์ แล้วยังสามารถแสดงตัวอย่างข้อมูลในรูปแบบรายงานจากเอ็มดีเอ็กซ์ที่โปรแกรมสร้างขึ้นมานั้นผู้ใช้ไม่จำเป็นต้องเขียนโค้ดเอ็มดีเอ็กซ์เพื่อเรียกดูข้อมูลด้วยตัวเอง มีความง่ายในการใช้งานด้วยการที่โปรแกรมนำเสนอข้อมูลของลูกบาศก์ข้อมูลแบบโครงสร้างต้นไม้ที่แสดงข้อมูลแกนมิติข้อมูล (dimension) ลำดับชั้น (Hierarchy) และสมาชิก (member) ทั้งหมดที่ประกอบอยู่ในลูกบาศก์ข้อมูล ทำให้ผู้ใช้มองเห็นโครงสร้างและสามารถลงลึกถึงรายละเอียดของลูกบาศก์ข้อมูล นอกจากนั้นโปรแกรมยังอนุญาตให้สามารถทำการเลือกแกนมิติข้อมูลหรือสมาชิกลงในส่วนของคอลัมน์ แถว หรือส่วนของการกรองข้อมูล โดยเลือกจากโครงสร้างต้นไม้ได้เลย การสร้างคำสั่งภาษาเอ็มดีเอ็กซ์สำหรับการเรียกดูข้อมูลในลักษณะข้อมูลสรุปได้ มีความสามารถกรองนำในส่วนที่ไม่มีข้อมูลออกได้

เมื่อเปรียบเทียบการออกแบบหน้าตาโปรแกรมและรูปแบบความสามารถการทำงานของงานวิจัย OLAP Realization Technology Research Based on MDX โปรแกรมสร้างเอ็มดีเอ็กซ์ออนไลน์จากงานวิจัยนี้จะมีข้อดีกว่าในเรื่องของความสามารถในการเรียกดูข้อมูลได้ระเอียดมากกว่าเนื่องในงานวิจัยดังกล่าวได้มีการกำหนดให้สามารถเรียกดูข้อมูลได้ครั้งละ 2 มิติเท่านั้น นอกจากนั้นยังมีความสามารถในการใช้ฟังก์ชันครอสจอย (CrossJoin()) เพื่อทำการแสดงข้อมูลที่จุดตัดต่าง ๆ ของแกนมิติข้อมูลทำให้เกิดมุมมองของข้อมูลที่แตกต่างจากไปจากเดิมและเกิดมุมมองต่อข้อมูลหลากหลายมากขึ้น เป็นเครื่องมือช่วยในการตัดสินใจทำให้การวิเคราะห์ข้อมูลทำได้อย่างมีประสิทธิภาพมากขึ้น

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

ข้อมูลคือสิ่งที่สำคัญในโลกธุรกิจในปัจจุบัน สถานประกอบการมีแนวคิดที่ต้องการบรรลุเป้าหมายด้านการดำเนินธุรกิจ ดังนั้นจึงมีกระบวนการมากมายที่เข้าไปเกี่ยวข้องเพื่อให้ได้มาซึ่งเป้าหมายนั้น ๆ การทำสารสนเทศข้อมูลในสถานประกอบการเป็นส่วนสำคัญที่จะทำให้ผู้บริหารวิเคราะห์ข้อมูลเพื่อนำไปใช้เป็นแนวทางในการวางแผนการดำเนินงานของบริษัทได้ เนื่องจากโดยทั่วไปแล้วข้อมูลที่ถูกเก็บฐานข้อมูลมักถูกเก็บในลักษณะของตาราง โดยในแต่ละตารางข้อมูลจะถูกเก็บเป็นแถวของข้อมูลดิบ ไม่สามารถนำไปวิเคราะห์เพื่อหาความสัมพันธ์ได้ โดยข้อมูลต่าง ๆ ที่นำมาวิเคราะห์นั้นอยู่ในรูปของฐานข้อมูลและถูกสร้างเป็นคลังข้อมูลซึ่งการใช้คลังข้อมูลในการเตรียมข้อมูลเพื่อการวิเคราะห์มีจุดเด่นในเรื่องความเร็วในการประมวลผลและสามารถนำเสนอได้หลายมุมมองของข้อมูล ดังนั้นจึงเป็นเหตุผลสำคัญที่จะต้องทำการเก็บข้อมูลให้อยู่ในรูปของตารางความสัมพันธ์ หรือลูกบาศก์ข้อมูลเพื่อนำข้อมูลเหล่านั้นมาวิเคราะห์หาแนวโน้มหรือค่าทางเศรษฐกิจได้

เมื่อมีคลังข้อมูล ควรจะมีการนำข้อมูลที่มีออกมาใช้ได้อย่างมีประสิทธิภาพ ข้อมูลที่มีค่าเหล่านั้นก็อาจจะไม่ได้ช่วยให้มีประโยชน์ต่อองค์กร การนำข้อมูลที่มีอยู่นั้นออกมาใช้ได้อย่างมีประสิทธิภาพต้องอาศัยความรู้และความเชี่ยวชาญในการใช้คำสั่งที่จะเรียกข้อมูล เอ็มดีเอ็กซ์เป็นภาษาคำสั่งที่ใช้สำหรับการเรียกข้อมูลจากลูกบาศก์ข้อมูลที่มีความยาก ความซับซ้อน สำหรับผู้ที่ไม่มีความรู้เกี่ยวกับโครงสร้างของคลังข้อมูลและลูกบาศก์ข้อมูล ดังนั้น เพื่อเป็นการง่ายสำหรับผู้ใช้ที่ไม่มีความเชี่ยวชาญเอ็มดีเอ็กซ์ โครงสร้างของคลังข้อมูลและลูกบาศก์ข้อมูล สามารถที่จะเรียกดูข้อมูลเชิงวิเคราะห์มาใช้ได้โดยง่ายยิ่งขึ้น

บทสรุปของงานวิจัยนี้ ได้นำเสนอโปรแกรมประยุกต์สร้างคำสั่งเอ็มดีเอ็กซ์บนบราวเซอร์ ที่ทำหน้าที่ติดต่อเรียกใช้ข้อมูลกับลูกบาศก์ข้อมูล เพื่อสร้างความสะดวกให้กับผู้ใช้ที่ไม่มีความชำนาญในการใช้เอ็มดีเอ็กซ์ ที่ต้องการสร้างคำสั่งเอ็มดีเอ็กซ์สำหรับการเรียกดูข้อมูลในลักษณะข้อมูลสรุปจากลูกบาศก์ข้อมูล

5.1 สรุปผลการวิจัย

ผลการทดลองสรุปในประเด็นที่สำคัญ 2 ประเด็น ดังต่อไปนี้

1) การออกแบบหน้าตาของโปรแกรม เพื่อที่จะสามารถจัดการกับการใช้งานของผู้ใช้ในการกำหนดพารามิเตอร์ต่าง ๆ ของเอ็มดีเอ็กซ์และการสร้างผลลัพธ์ของการสร้างคำสั่งเอ็มดีเอ็กซ์นั้นมีความยากที่จะสามารถทำให้การทำงานได้ยืดหยุ่นตามความต้องการทั้งหมด ดังนั้น จึงจำเป็นต้องกำหนดขอบเขตของการทำงานในแต่ละฟังก์ชัน คือ ในงานวิจัยนี้ส่วนที่ติดต่อกับผู้ใช้จะแสดงรายชื่อมิติของลูกบาศก์ข้อมูลนั้น ๆ และอนุญาตให้ผู้ใช้เลือกได้ถึงข้อมูลระดับสมาชิกสามารถเลือกเจาะลึกลงไปได้ และถ้าผู้ใช้ต้องการที่จะดูข้อมูลที่มีรูปแบบที่ซับซ้อนมาก ๆ สามารถทำได้ในส่วนของการสร้างคำสั่งเอ็มดีเอ็กซ์เอง โดยโปรแกรมอำนวยความสะดวกด้วยการแสดงโครงสร้างของลูกบาศก์ข้อมูลนั้นทางโครงสร้างต้นไม้

2) ความสามารถฟังก์ชันการทำงานของโปรแกรมในการสร้างเอ็มดีเอ็กซ์ จากการดำเนินงานที่ผ่านมาพบว่าความซับซ้อนของตัวคำสั่งและฟังก์ชันการทำงานต่าง ๆ ที่ต้องอาศัยความรู้ความเข้าใจในโครงสร้างของลูกบาศก์ข้อมูลพอสมควร และขอบเขตของงานถ้าต้องการรายงานข้อมูลที่มีความซับซ้อนมากนั้นจะทำให้การที่จะสร้างตัวโปรแกรมสร้างเอ็มดีเอ็กซ์ให้ยืดหยุ่นนั้นยากที่จะสามารถสร้างให้ได้ครอบคลุมเงื่อนไขทั้งหมด ซึ่งในงานวิจัยนี้ได้อนุญาตให้สามารถเลือกแกนมิติมากกว่า 2 แกนมิติ เพื่อให้เกิดความหลากหลายในการเรียกดูข้อมูลจากการใช้งานฟังก์ชัน CrossJoin แต่การสร้างเอ็มดีเอ็กซ์ซับซ้อนมากจะมีผลต่อการแสดงผลในรูปของตารางข้อมูลซับซ้อนตามด้วย

5.2 การประยุกต์งานวิจัย

ในปัจจุบันการทำสารสนเทศเป็นสิ่งจำเป็นสำหรับสถานประกอบการหรือองค์กรต่าง ๆ เทคโนโลยีเกี่ยวกับการทำสารสนเทศมีการใช้งานอย่างแพร่หลาย เช่น การสร้างรายงานสรุปผลการประกอบการต่าง ๆ สามารถนำงานวิจัยชิ้นนี้ไปประยุกต์ใช้กับการทำงานในลักษณะของการสร้างรายงานเชิงวิเคราะห์ อำนวยความสะดวกต่อการทำงานและทำให้เกิดรายงานในมุมมองที่ต่างไปจากรายงานจากข้อมูลแบบเดิม จึงสามารถเป็นเครื่องมือช่วยสำหรับประกอบการตัดสินใจที่มีประสิทธิภาพมากขึ้น และเมื่อมีการพัฒนาต่อไปที่ดีขึ้นแล้วอาจนำเอาความรู้ความสามารถมานำเสนอรายงานได้มุมมองต่อข้อมูลที่หลากหลายเพิ่มมากขึ้น และมีประสิทธิภาพที่ดีต่อไปในอนาคตได้

5.3 ปัญหาและข้อเสนอแนะ

1) การที่จะพัฒนาโปรแกรมสร้างเอ็มดีเอ็กซ์สำหรับผู้ใช้ที่ไม่มีความชำนาญในการใช้คำสั่งเอ็มดีเอ็กซ์นั้น จะต้องกำหนดขอบเขตความสามารถการทำงานให้ชัดเจน เช่น การกำหนดระดับความลึกของข้อมูลที่เลือกได้ ความสามารถการเลือกแกนมิติ เพื่อเป็นการควบคุมการเข้าถึงข้อมูล และควรมีการกำหนดสิทธิ์ในการเข้าถึงข้อมูล โดยการล็อกอินเข้าสู่ระบบในรูปแบบการใช้สำหรับผู้ดูแลระบบก่อนเพื่อเป็นการความมั่นคงของข้อมูล

2) การทดสอบและใช้งานโปรแกรมสร้างเอ็มดีเอ็กซ์ยังคงเป็นขั้นเริ่มต้นของการนำไปพัฒนาต่อเพื่อปรับใช้ให้เหมาะกับการทำงานของแต่ละองค์กรหรือธุรกิจ ฟังก์ชันต่าง ๆ ของเอ็มดีเอ็กซ์บนโปรแกรมสามารถทำได้ แต่เนื่องจากโปรแกรมยังมีข้อจำกัดในการใช้คำสั่ง ฟังก์ชัน ที่อำนวยสะดวกยังไม่ดีเท่าที่ควร คือผู้ใช้จำเป็นจะต้องกรอกคำสั่งนั้น ๆ ในโหมด Edit ในการพัฒนาขั้นถัดไปควรจะอำนวยความสะดวกในการใช้ฟังก์ชันการทำงานของเอ็มดีเอ็กซ์ให้ง่ายต่อการใช้งาน เพื่อจะสามารถทำงานกับข้อมูลได้มีประสิทธิภาพที่ดียิ่งขึ้น

รายการอ้างอิง

Jens-Peter Dittrich, Donald Kossmann and Alexander Kreutz. (2005). **Bridging the Gap between OLAP and SQL**. Norway : Proceeding of the 31st VLDB Conference, Trondheim, Norway.

John Horner, Il-Yeol Song and Peter P.Chen. (2004). **An Analysis of Additivity in OLAP Systems**.

Kazi A. Zaman and Donovan A. Schneider. (2005). **Modeling and Querying Multidimensional Data Source in Siebel Analytics: A Federated Rational System**. USA : SIGMOD 2005, June 14-16, 2005, Baltimore, Maryland, USA.

Konstantinos Morfonios, Stratis Konakas, Yannis Ioannidis and Nikolaos Kotsis. (2007). **ROLAP Implementations of the Data Cube.**

Owen Kaser and Daniel Lemier. (2003). **Attribute Value Reordering of Efficient Hybrid OLAP**. Ralph Kimball. (1996). **N**ine Decisions in the Design of Data Warehouse.

Surajit Chaudhuri and Umeshwar Dayal. (1997). **An Overview of Data Warehousing and OLAP Technology**. ACM : March 1997, ACM SIGMOD Record, Volume 26 Issue 1.

Tapio Niemi, Jyrki Nummenmaa and Peter Thanisch. (2001). **Constructing OLAP Cubes Based on Queries**. ACM : November 2001, DOLAP '01: Proceedings of the 4th ACM international workshop on Data warehousing and OLAP**.**

Tapio Niemi and Marko Niinimaki. (2002). **Constructing an OLAP Cube from Distributed XML Data.**

Tatsuo Tsuji, Akihiro Hara and Ken Higuchi. (2006). **An Extendible Multidimensional Array System for MOLAP.**

Wen-Yuan Liu and Shu-Fen Fang.(2002). OLAP Realization Technology Research Based on MDX. China : **Proceeding of 1st International Conference on Machine Learning and Cybernetics**, Beijing, 4-5 November 2002.

Teknas.(2004). **Cubes, MDX, Analysis Services and ADOMD in C#.** [Online]. Available: http://www.codeproject.com/cs/database/CubesAdomd.asp [Nov 1, 2007]

Carl Nolan. (1999). **Manipulate and Query OLAP Data Using ADOMD and Multidimensional Expressions.** [Online]. Available: http://www.microsoft.com/msj/0899/mdx/mdx.aspx [Nov 1, 2007]

Doug Rosenberg and Matt Stephens. (2007). **Use Case Driven Object Modeling with UML - Theory and Practice.** Apress.

George Spofford. (2001). **MDX Solutions with MS SQL Server analysis services.** Canada : WILEY.

William Pearson. (2002). **MDX at First Glance: Introduction to SQL Server MDX Essentials.** [Online]. Available : http://www.databasejournal.com/features/mssql/article.php/10894_1495511_3 [Nov 1, 2007]

ภาคผนวก ก

บทความผลงานวิจัยที่นำเสนอในการประชุมทางวิชาการเพื่อเสนอผลงานวิจัย ครั้งที่ 4 "มหาวิทยาลัยกับการพัฒนาบนฐานความรู้" โดยความร่วมมือกับ สกว. และเครือข่าย มหาวิทยาลัยราชภัฏอีสาน ณ มหาวิทยาลัยราชภัฏนครราชสีมา จ. นครราชสีมา

ชื่อโครงการวิจัย

(ภาษาไทย) การพัฒนาโปรแกรมสร้างเอ็มดีเอ็กซ์ออนไลน์

(ภาษาอังกฤษ) DEVELOPMENT OF ONLINE MDX GENERATOR

ชื่อผู้วิจัย (ภาษาไทย) นายวีรภัทร ศรีพิพัฒนกุล

(ภาษาอังกฤษ) Mr.Weerapat Sripipattanakul

ชื่ออาจารย์ที่ปรึกษา (ภาษาไทย) ผู้ช่วยศาสตราจารย์ ดร. พิชโยทัย มหัทธนาภิวัฒน์

(ภาษาอังกฤษ) Assistant Professor Dr.Pichayothai Mahatthanapiwat

ที่อยู่ สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี นครราชสีมา 30000

ทำการวิจัยเสร็จเรียบร้อย พ.ศ. 2550

บทคัดย่อ

ปัจจุบัน ข้อมูลเชิงวิเคราะห์ได้กลายมาเป็นปัจจัยสำคัญสำหรับการทำงานในองค์กรต่าง ๆ การทำสารสนเทศข้อมูลในองค์กรนั้นเป็นส่วนสำคัญส่วนหนึ่งที่ทำให้ผู้บริหารขององค์กรสามารถวิเคราะห์ผลดำเนินงานจากข้อมูลเชิงวิเคราะห์เพื่อนำไปใช้ช่วยประกอบการตัดสินใจในการวางแผนการดำเนินงานในอนาคตได้อย่างมีประสิทธิภาพ และข้อมูลที่นำมาใช้สำหรับงานวิเคราะห์นั้นเก็บอยู่ในรูปของคลังข้อมูล (Data Warehouse) และลูกบาศก์ข้อมูล (Data Cube)

การที่จะนำข้อมูลจากลูกบาศก์ข้อมูลที่มีอยู่นั้นออกมาใช้ได้อย่างมีประสิทธิภาพต้องอาศัยความรู้และความเชี่ยวชาญในการใช้คำสั่งที่จะเรียกข้อมูล นั่นคือเอ็มดีเอ็กซ์ (Multi-Dimensional Expressions) ซึ่งมีความยากซับซ้อนสำหรับผู้ที่ไม่มีความรู้ความเข้าใจเกี่ยวกับลักษณะโครงสร้างของคลังข้อมูลและลูกบาศก์ข้อมูล ดังนั้นงานวิจัยนี้จึงนำเสนอโปรแกรมที่ช่วยสำหรับการออกแบบและสร้างเอ็มดีเอ็กซ์แบบออนไลน์ โดยโปรแกรมจะใช้ออกแบบและสร้างเอ็มดีเอ็กซ์ เพื่อลดความผิดพลาดและเวลาที่ใช้ในการออกแบบและสร้างเอ็มดีเอ็กซ์ สร้างความสะดวกให้กับผู้ใช้ที่ไม่มีความชำนาญในการใช้เอ็มดีเอ็กซ์ที่เริ่มต้นออกแบบและสร้างเอ็มดีเอ็กซ์ สำหรับการเรียกดูข้อมูลในลักษณะข้อมูลสรุปจากลูกบาศก์ข้อมูล

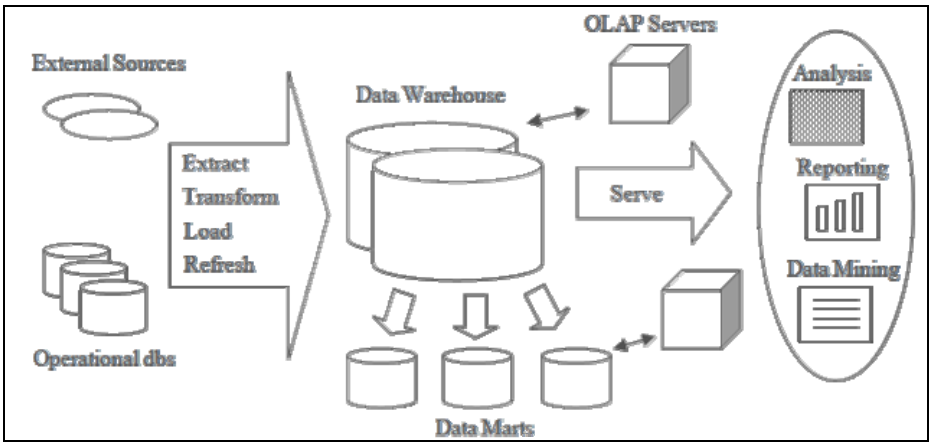
Abstract

Nowadays, analytical data is an important factor for organizations work. Decision maker can analyze overall operation from it to make the efficient work plans. Data for analytical works are in form of data warehouse and data cube.

Using data cube effectively is based on expertise and knowledge in query. Furthermore, MDX (Multi-Dimensional Expression) is quite complex for unskillful users to data warehouse and data cube construction. Therefore, this research purposes a program which aid for designing and creating online MDX, this program reduce a mistake and time lost in designing and creating MDX. It will facilitate for novel users to examine summary data from data cube.

1. ความสำคัญและความเป็นมาของปัญหา

ปัจจุบัน คอมพิวเตอร์ได้เข้ามามีบทบาทในชีวิตประจำวันของมนุษย์มากขึ้น ข้อมูลได้กลายมาเป็นปัจจัยสำคัญสำหรับการทำงานในองค์กรทุก ๆ องค์กร ทั้งขนาดเล็ก ขนาดกลางและขนาดใหญ่ การใช้ระบบสารสนเทศในองค์กรนั้นเป็นส่วนสำคัญส่วนหนึ่งที่ทำให้ผู้บริหารขององค์กรสามารถวิเคราะห์ผลดำเนินงานที่ผ่านมาว่าเป็นไปในทิศทางใด และนำข้อมูลที่ได้จากการทำระบบสารสนเทศไปใช้เป็นเครื่องมือประกอบการตัดสินใจในการวางแผนการดำเนินงานในอนาคต รวมทั้งวางแผนการจัดสรรการใช้ทรัพยากรที่มีอยู่ให้เกิดประโยชน์สูงสุดและมีประสิทธิภาพดียิ่งขึ้น ดังนั้นในปัจจุบันพบว่าการทำระบบสารสนเทศเพื่อนำข้อมูลมาใช้เป็นแนวทางวิเคราะห์ในการวางแผนดำเนินงานขององค์กรจึงได้รับความนิยมและเผยแพร่เพิ่มมากขึ้นเรื่อย ๆ การที่มีการนำระบบสารสนเทศมาใช้ในองค์กร ข้อมูลที่นำมาใช้สำหรับงานวิเคราะห์นั้นส่วนมากแล้ว จะใช้ในลักษณะของคลังข้อมูล (Data Warehouse) และลูกบาศก์ข้อมูล (Data Cube) ประกอบกับการใช้เทคโนโลยีที่เรียกว่า On-Line Analytical Processing (OLAP) โดยมีภาพรวมกระบวนการทำงานของระบบคลังข้อมูลแสดงดังรูปที่ 1



รูปที่ 1 ภาพรวมการทำงานของระบบคลังข้อมูล

การจะนำข้อมูลจากลูกบาศก์ข้อมูลที่มีอยู่นั้นออกมาใช้ได้อย่างมีประสิทธิภาพต้องอาศัยความรู้และความเชี่ยวชาญในการใช้คำสั่งที่เรียกข้อมูล คือ เอ็มดีเอ็กซ์ (Multi-Dimensional eXpressions : MDX) เป็นภาษาคำสั่งที่สร้างขึ้นมาโดยบริษัท Microsoft เป็นภาษาลักษณะข้อคำถามคล้ายกับ SQL (Structured Query Language) โดยพื้นฐานส่วนใหญ่ของเอ็มดีเอ็กซ์จะมีรูปแบบโครงสร้างอย่างง่าย ๆ ดังตัวอย่าง

SELECT [<axis specification> [, <axis specification>, …]]

FROM [<cube specification>]

[WHERE <slicer specification>]

อนุประโยค SELECT ถูกใช้ในการกำหนดแกนมิติ (Axis-dimension) และอนุประโยค WHERE ถูกใช้ในการกำหนดแกนการซอยข้อมูล (Slicer dimensions) ส่วนอนุประโยค FROM คือใส่ชื่อของลูกบาศก์ข้อมูลที่ต้องการจะเรียกข้อมูลเอ็มดีเอ็กซ์

เอ็มดีเอ็กซ์ใช้เรียกดูข้อมูลจากลูกบาศก์ข้อมูลที่มีความยากซับซ้อนสำหรับผู้ที่ไม่มีความรู้ความเข้าใจเกี่ยวกับลักษณะโครงสร้างของคลังข้อมูลและลูกบาศก์ข้อมูล ดังนั้น ในงานวิจัยนี้จึงได้นำเสนอโปรแกรมที่ช่วยสำหรับการออกแบบและสร้างเอ็มดีเอ็กซ์แบบออนไลน์ โดยโปรแกรมจะทำหน้าที่ช่วยอำนวยความสะดวกในการออกแบบและสร้างเอ็มดีเอ็กซ์ เพื่อลดความผิดพลาดและเวลาที่ใช้ในการออกแบบและสร้างเอ็มดีเอ็กซ์ เพื่อใช้เรียนรู้และออกแบบเอ็มดีเอ็กซ์ สร้างความสะดวกให้กับผู้ใช้ที่ไม่มีความชำนาญในการใช้เอ็มดีเอ็กซ์ที่เริ่มต้นออกแบบและสร้างเอ็มดีเอ็กซ์สำหรับการเรียกดูข้อมูลในลักษณะข้อมูลสรุปจากลูกบาศก์ข้อมูล

2. วัตถุประสงค์ของการวิจัย

1. เพื่อใช้ในการศึกษาลักษณะและโครงสร้างของเอ็มดีเอ็กซ์
2. เพื่อพัฒนาโปรแกรมประยุกต์บนบราวเซอร์ ใช้ออกแบบและสร้างเอ็มดีเอ็กซ์ ช่วยอำนวยความสะดวกแก่ผู้ใช้ที่เริ่มต้นใช้งานข้อมูลเชิงวิเคราะห์จากลูกบาศก์ข้อมูล
3. เพื่ออำนวยความสะดวกในการเรียกดูข้อมูลจากลูกบาศก์ข้อมูล เพื่อมาใช้นำเสนอในรูปแบบข้อมูลเชิงวิเคราะห์
4. เพื่อให้สามารถทำงานกับฐานข้อมูลแบบหลายมิติได้มีประสิทธิภาพมากขึ้น

3. ประโยชน์ของการวิจัย

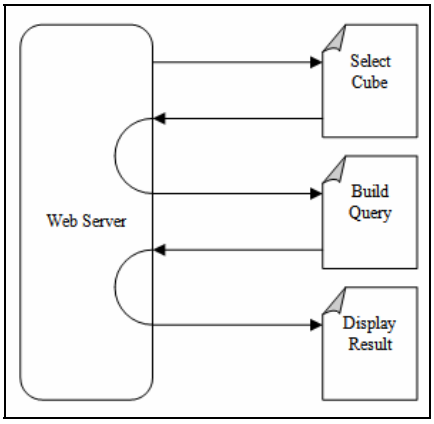
1. สร้างแนวคิดเกี่ยวกับนำเสนอในลักษณะข้อมูลเชิงวิเคราะห์ และช่วยอำนวยความสะดวกแก่ผู้ใช้ที่เริ่มต้นใช้งานข้อมูลเชิงวิเคราะห์จากลูกบาศก์ข้อมูล
2. สามารถนำข้อมูลเชิงวิเคราะห์มาใช้ประโยชน์ได้หลากหลายมากขึ้น

3. สามารถนำโปรแกรมไปใช้พัฒนาต่อยอดโปรแกรมประยุกต์ทางธุรกิจที่ต้องการใช้การวิเคราะห์ข้อมูลแบบหลายมุมมอง (Multi-Dimensions Thinking) ช่วยประกอบในการตัดสินใจดำเนินงานได้อย่างมีประสิทธิภาพ

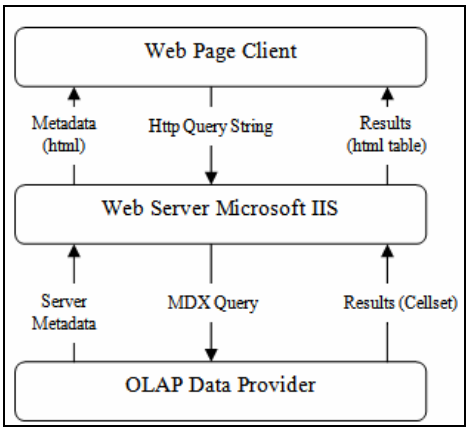
4. วิธีดำเนินการวิจัย

การค้นคว้าวิจัยจะแบ่งเป็นขั้นตอนดังนี้ ศึกษาและรวบรวมสรุปงานวิจัยที่เกี่ยวข้อง ออกแบบและวิเคราะห์โครงสร้างของระบบ สร้างโปรแกรมประยุกต์บนบราวเซอร์ และเชื่อมต่อกับฐานข้อมูลแบบหลายมิติ ทดสอบและพัฒนาโปรแกรมประยุกต์บนบราวเซอร์ ตรวจสอบความถูกต้องของ ผลลัพธ์ที่ได้จากการทำงานของโปรแกรมประยุกต์บนบราวเซอร์ แหล่งที่มาของข้อมูลที่ใช้ในการทดสอบ ฐานข้อมูลแบบหลายมิติหรือลูกบาศก์ข้อมูลที่ใช้ทดสอบในโครงการนี้ได้มาจากโปรแกรม Microsoft SQL Server 2000 เป็นลูกบาศก์ข้อมูลที่เป็นข้อมูลจำลองการซื้อขายสินค้าชนิดต่าง ๆ ของร้านค้าในรัฐต่าง ๆ ของแคนาดา เม็กซิโกและสหรัฐอเมริกา

การออกแบบและวิเคราะห์โครงสร้างของระบบ การทำงานของโปรแกรมสร้างเอ็มดีเอ็กซ์ออนไลน์ จะทำงานตามขั้นตอนดังนี้ เชื่อมต่อแหล่งข้อมูล ฐานข้อมูล และลูกบาศก์ข้อมูล และแสดงผลสู่ผู้ใช้ทางหน้าเว็บ ใช้ html form component อนุญาตให้ผู้ใช้สามารถเลือกพารามิเตอร์ต่าง ๆ ของ เอ็มดีเอ็กซ์ จัดการเกี่ยวกับการใช้ฟังก์ชันการทำงานของเอ็มดีเอ็กซ์ สั่งเอ็มดีเอ็กซ์ทำงานและแสดงผลลัพธ์สู่ผู้ใช้สามารถแสดงขั้นตอนการทำงานได้ดังแผนภาพรูปที่ 2 และภาพรวมการทำงานของระบบดังรูปที่ 3



รูปที่ 2 ขั้นตอนการทำงานของระบบ



รูปที่ 3 ภาพรวมการทำงานของระบบ

การออกแบบหน้าตาของโปรแกรม เพื่อใช้ติดต่อตอบโต้กับผู้ใช้จะใช้การติดต่อผ่านทางหน้าเว็บเพจโดยจะแบ่งหน้าเว็บเพจออกเป็น 3 ส่วน ดังรูปที่ 7 แต่ละส่วนจะมีหน้าที่การทำงานดังนี้

ส่วนที่ 1: การเชื่อมต่อกับแหล่งข้อมูลโดยจะรับชื่อแหล่งข้อมูลเพื่อใช้ในการเชื่อมต่อกับแหล่งข้อมูล

ส่วนที่ 2: การจัดการเกี่ยวกับการเลือกมิติ (Dimensions) เพื่อสร้างเอ็มดีเอ็กซ์

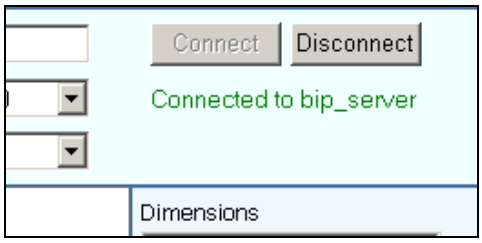
ส่วนที่ 3: การจัดการกับเอ็มดีเอ็กซ์ เช่น การสั่งให้เอ็มดีเอ็กซ์ทำงาน การสร้างหน้ารายงาน

การสร้าง User Interface ที่จะสามารถจัดการกับการใช้ของผู้ใช้ในการกำหนดพารามิเตอร์ต่าง ๆ ของข้อคำถามและการสร้างผลลัพธ์ของการสร้างเอ็มดีเอ็กซ์ ในส่วนของ Simple MDX (ส่วนที่2) จะแสดงรายชื่อของมิติของลูกบาศก์ข้อมูลนั้น ๆ และอนุญาตให้ผู้ใช้เลือกได้เฉพาะข้อมูลสมาชิกในระดับสูงสุดซึ่งเป็นข้อมูลสรุป แต่ถ้าผู้ใช้ต้องการที่จะดูข้อมูลที่ลึกลงไปสามารถทำได้ในส่วนของการสร้าง เอ็มดีเอ็กซ์เอง โดยสามารถดูโครงสร้างของลูกบาศก์ข้อมูลนั้นทางโครงสร้างต้นไม้ การสร้างการเชื่อมต่อกับแหล่งข้อมูลและลูกบาศก์ข้อมูลนั้น อันดับแรกในการสร้าง OLAP Web Client คือการสร้างการเชื่อมต่อด้วยตัวจัดการข้อมูล OLAP โดยใช้ ADOMD ตั้งค่าฐานข้อมูล จัดการชุดข้อมูลของรายชื่อลูกบาศก์ข้อมูลจากการเลือกของผู้ใช้ ขั้นตอนการทำงาน เริ่มจาก Web Server สร้างการเชื่อมต่อกับแหล่งข้อมูล โดยผู้ใช้กรอกข้อมูลแหล่งข้อมูล จากนั้น Web Server หาฐานข้อมูลในแหล่งข้อมูลแล้ว Web Server สร้างหน้าเว็บเพจที่มีรายชื่อลูกบาศก์ข้อมูลเพื่อให้ผู้ใช้เลือก

5. ผลการวิจัย

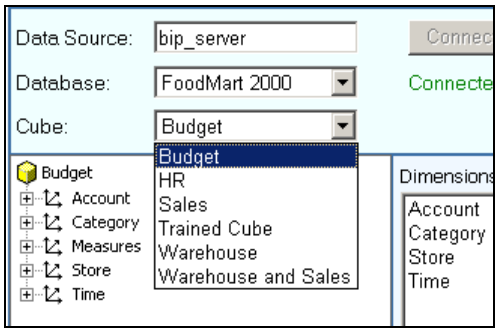
5.1 การทดสอบผลการทำงานของโปรแกรมสร้างเอ็มดีเอ็กซ์ออนไลน์

การเชื่อมต่อแหล่งข้อมูล ฐานข้อมูล และลูกบาศก์ข้อมูล และแสดงผลสู่ผู้ใช้ทางหน้าเว็บ ซึ่งทำได้ง่ายสร้างความสะดวกแก่ผู้ใช้ ด้วยการเลือกแหล่งข้อมูลจากรายชื่อแหล่งข้อมูลที่โปรแกรมแสดงรายการ การเชื่อมต่อกับแหล่งข้อมูล ถ้าเชื่อมต่อกับแหล่งข้อมูลสำเร็จ จะมีสถานะเชื่อมต่อกับแหล่งข้อมูลปรากฏขึ้น ดังรูปที่ 4



รูปที่ 4 แสดงสถานะเชื่อมต่อกับแหล่งข้อมูล

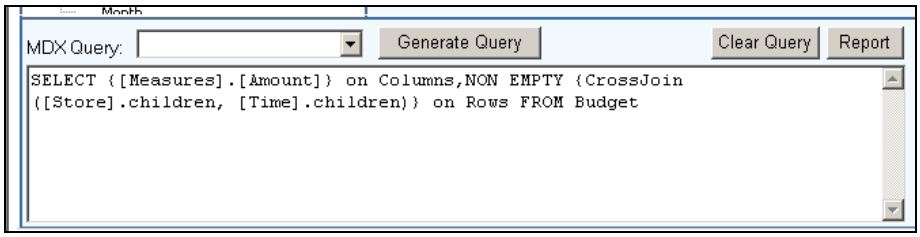
เมื่อทำการเชื่อมต่อสำเร็จโปรแกรมจะมีรายชื่อของฐานข้อมูลและรายชื่อของลูกบาศก์ข้อมูลในฐานข้อมูลนั้น ๆ แสดงขึ้นมา และแสดงข้อมูลของลูกบาศก์ข้อมูลในรูปโครงสร้างต้นไม้ ดังรูปที่ 5



รูปที่ 5 รายชื่อของฐานข้อมูลและรายชื่อของลูกบาศก์ข้อมูลในฐานข้อมูล

5.2 การสร้างเอ็มดีเอ็กซ์

พบว่าสามารถอำนวยความสะดวกแก่ผู้ใช้ที่เริ่มต้นใช้เอ็มดีเอ็กซ์ได้ดี ในการสร้างเอ็มดีเอ็กซ์สำหรับการเรียกดูข้อมูลในลักษณะข้อมูลสรุปได้ โดยมีสามารถกรองในส่วนที่ไม่มีข้อมูลออกได้โดยใช้ Non Empty Keyword ความสามารถในการใช้ฟังก์ชัน CrossJoin เพื่อทำการ Cross-product ข้อมูลทำให้เกิดมุมมองใหม่กับข้อมูล ด้วยการเลือกมิติ (Dimension) ให้ไปอยู่ในส่วนของคอลัมน์หรือแถวตามต้องการ หลังจากได้มิติตามแนวแกนที่ต้องการ แล้วกดปุ่ม Generate Query เพื่อสร้างเอ็มดีเอ็กซ์ ได้ผลลัพธ์ดังรูปที่ 6



รูปที่ 6 ผลลัพธ์การสร้างคำสั่งเอ็มดีเอ็กซ์

6. บทสรุปและข้อเสนอแนะ

สรุปในประเด็นที่สำคัญได้ดังนี้

การออกแบบหน้า User Interface ที่จะสามารถจัดการกับการใช้ของผู้ใช้ในการกำหนดพารามิเตอร์ต่าง ๆ ของข้อคำถามและการสร้างผลลัพธ์ของการสร้าง เอ็มดีเอ็กซ์นั้นมีความยากที่จะสามารถทำให้การทำงานได้ตามที่ต้องการทั้งหมด ดังนั้น จึงต้องมีการกำหนดขอบเขตของการทำงานในแต่ละกระบวนการทำงาน คือ ในโครงการวิจัยนี้ ในส่วนของ Simple MDX (ส่วนที่ 2) จะแสดงรายชื่อของมิติของลูกบาศก์ข้อมูลนั้น ๆ และอนุญาตให้ผู้ใช้เลือกได้เฉพาะข้อมูลสมาชิกในระดับสูงสุดซึ่งเป็นข้อมูลสรุป แต่ถ้าผู้ใช้ต้องการที่จะดูข้อมูลที่ลึกลงไปสามารถทำได้ในส่วนของการสร้างเอ็มดีเอ็กซ์เอง โดยสามารถดูโครงสร้างของลูกบาศก์ข้อมูลนั้นทางโครงสร้างต้นไม้

การสร้างเอ็มดีเอ็กซ์จากการดำเนินงานที่ผ่านมา ความซับซ้อนของเอ็มดีเอ็กซ์และฟังก์ชันต่าง ๆ ที่ต้องอาศัยความรู้ความเข้าใจในโครงสร้างของฐานข้อมูลแบบหลายมิติพอสมควร และขอบเขตของงานถ้าต้องการรายงานข้อมูลที่มีความซับซ้อนมากนั้น จะทำให้การที่จะสร้างตัวโปรแกรมสร้าง เอ็มดีเอ็กซ์ให้ยืดหยุ่นจะทำได้ยากจึงทำให้สามารถที่จะทำได้เฉพาะการสร้างเรียกข้อมูลในลักษณะที่เป็นข้อมูลสรุป

การพัฒนาต่อยอดนั้นสามารถนำไปพัฒนาประยุกต์ใช้กับการทำงานในส่วนของการออกรายงานสรุปต่าง ๆ ขององค์กรที่ใช้คลังข้อมูลอยู่แล้วหรือต้องการใช้คลังข้อมูล ทำให้สามารถสร้างรายงานที่มีมุมมองของข้อมูลแตกต่างออกไป ด้วยการขั้นตอนการทำงานที่ง่ายของโปรแกรม ทำให้เป็นเครื่องมือช่วยให้ใช้ประกอบการตัดสินใจของผู้บริหารมีประสิทธิภาพเพิ่มขึ้น

บรรณานุกรม

Jens-Peter Dittrich, Donald Kossmann and Alexander Kreutz. (2005). **Bridging the Gap between OLAP and SQL.** Norway : Proceeding of the 31st VLDB Conference, Trondheim, Norway, 2005.

Kazi A. Zaman and Donovan A. Schneider. (2005). **Modeling and Querying Multidimensional Data Source in Siebel Analytics: A Federated Rational System**. USA : SIGMOD 2005, June 14-16, 2005, Baltimore, Maryland, USA.

Surajit Chaudhuri and Umeshwar Dayal. (1997). **An Overview of Data Warehousing and OLAP Technology.** ACM : March 1997, ACM SIGMOD Record, Volume 26 Issue 1.

Tapio Niemi. (2001). **Constructing OLAP Cubes Based on Queries.** ACM : November 2001, DOLAP '01: Proceedings of the 4th ACM international workshop on Data warehousing and OLAP.

Wen-Yuan Liu and Shu-Fen Fang.(2002). **OLAP Realization Technology Research Based on MDX.** China : Proceeding of 1st International Conference on Machine Learning and Cybernetics, Beijing, 4-5 November 2002.

Teknas.(2004) **Cubes, MDX, Analysis Services and ADOMD in C#.** [Online]. Available: http://www.codeproject.com/cs/database/CubesAdomd.asp [Nov 1, 2007]

Carl Nolan. (1999). **Manipulate and Query OLAP Data Using ADOMD and Multidimensional Expressions.** [Online]. Available: http://www.microsoft.com/msj/0899/mdx/mdx.aspx [Nov 1, 2007]

George Spofford. (2001). **MDX Solutions with MS SQL Server analysis services**. Canada : WILEY.

William Pearson. (2002). **MDX at First Glance: Introduction to SQL Server MDX Essentials.** [Online]. Available : http://www.databasejournal.com/features/mssql/article.php/10894_1495511_3 [Nov 1, 2007]

ประวัติผู้เขียน

นายวีรภัทร ศรีพิพัฒนกุล เกิดเมื่อวันที่ 10 กันยายน พ.ศ. 2526 ที่อำเภอเมือง จังหวัดนครราชสีมา เริ่มเข้าศึกษาระดับชั้นอนุบาล 1- ชั้นประถมศึกษาปีที่ 6 ที่โรงเรียนสุริยาอุทัย ตำบลในเมือง อำเภอพิมาย จังหวัดนครราชสีมา หลังจากสำเร็จการศึกษาได้เข้าศึกษาต่อในระดับมัธยมศึกษาตอนต้น-ตอนปลาย ที่โรงเรียนราชสีมาวิทยาลัย อำเภอเมือง จังหวัดนครราชสีมา หลังจากสำเร็จการศึกษาได้เข้าศึกษาต่อในระดับปริญญาตรี ในปีการศึกษา 2545 ที่มหาวิทยาลัยเทคโนโลยีสุรนารี สังกัดสำนักวิชาวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์ และสำเร็จการศึกษาเมื่อปี พ.ศ. 2548 ภายหลังสำเร็จการศึกษาในระดับปริญญาตรี ได้เข้าศึกษาต่อในระดับปริญญาโท สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในปีการศึกษา 2549

ในระหว่างการศึกษาได้รับความอนุเคราะห์อย่างดียิ่งจากคณาจารย์ในสาขาวิชา และได้รับความไว้วางใจ ให้เป็นผู้ช่วยสอนปฏิบัติการ 5 รายวิชา คือ (1) Computer Programming (2) Object-Oriented Technology (3) Event-Driven Programming (4) System Analysis และ (5) Software Engineering เป็นเวลา 2 ปี