

ศึกษาวิธีการสร้างคลังข้อมูลเพื่อสนับสนุนการตัดสินใจ

Building A Data Warehouse for Decision Support Study

โดย

นางสาวเพ็ญศรี บั๊กกะสีนัง

รหัส 38626049



\*H001534\*

อาจารย์ที่ปรึกษา

รศ.ดร.ศุภมิตร จิตตะยโสธร

วัน เดือน ปี	10 7 S.ศ. 2549
เลขทะเบียน	01534
เลขเรียกหนังสือ	สท ท899ด 2540
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2540

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	ศึกษาวิธีการสร้างคลังข้อมูลเพื่อสนับสนุนการตัดสินใจ
นักศึกษา	นางสาวเพ็ญศรี ปักกะสีนัง
อาจารย์ที่ปรึกษา	รศ.ดร.ศุภมิตร จิตตะยโสธร
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
พ.ศ.	2540

### บทคัดย่อ

ปัจจุบันนี้องค์กรจะประสบความสำเร็จได้ต้องขึ้นอยู่กับปัจจัยหลายอย่าง ข้อมูลเป็นปัจจัยหลักที่ช่วยให้องค์กรประสบความสำเร็จ ซึ่งนับวันข้อมูลจะมีมากขึ้นจนเป็นปัญหาในการจัดเก็บและการเรียกใช้ข้อมูลเหล่านั้นให้เกิดประสิทธิภาพสูงสุด เพื่อแก้ปัญหาดังกล่าวได้มีการนำเทคโนโลยีใหม่ ๆ มาช่วยในการจัดการกับข้อมูลที่มีปริมาณมาก ๆ โดยเลือกเฉพาะหัวข้อที่ต้องการและนำมาสรุปเป็นข้อมูลสำหรับผู้บริหารเพื่อการตัดสินใจ เทคโนโลยีดังกล่าว เรียกว่าเทคโนโลยี "คลังข้อมูล" หรือ Data warehouse ซึ่งเป็นเทคโนโลยีที่กำลังได้รับความนิยมจากทุกองค์กรในปัจจุบัน

การศึกษานี้เป็นการนำเสนอหลักการของเทคโนโลยีคลังข้อมูล(Data warehouse) แนวทางในการสร้างคลังข้อมูล การใช้ Software Tools (Pilot Decision Support) ช่วยในการสร้างและวิเคราะห์ข้อมูล ในคลังข้อมูล โดยอาศัยเทคโนโลยี Multidimensional Analysis พร้อมทั้งเปรียบเทียบข้อดีและข้อเสียของการใช้งาน และกรณีศึกษาการสร้างคลังข้อมูลสำหรับสนับสนุนการตัดสินใจของผู้บริหารในสถานศึกษา : สถาบันราชภัฏยะเชิงเทรา

Title	Building A Data Warehouse for Decision Support Study
Student	Miss Pensri Pukkasenung
Advisor	Assoc.Prof.Dr.Suphamit Chittayasothorn
Level of Study	Master of Science in Information Technology
Major	Information Science
Year	1997

### ABSTRACT

The Success of Organizations depends on a variety of factors. Data is a major factor accounting for Success. Proliferation of data has created problems in finding the most effective way of storing and retrieving data. New technologies have been used to manipulate large amounts of data, to sort out the required topics and summarize the data for administrative decision - making support. The proposed technology is called " Data Warehouse " , guidelines in constructing the " Data Warehouse " , and instructions on how to use Software tools (Pilot Decision Support ) in constructing and analyzing data in the warehouse . A mulitdimensional analysis is has been performed , and the comparative strengths and weaknesses of this technology have been studied . It is a case study of data warehouse construction for Rajabhat Institute Chachoengsao in the context of support for administrative decision -making .

## กิตติกรรมประกาศ

โครงการศึกษาและพัฒนาระบบงานนี้สำเร็จได้ด้วยความช่วยเหลือและสนับสนุนจากหลายฝ่ายเป็นอย่างดี ผู้ศึกษาจึงใคร่ขอขอบพระคุณบุคคลต่อไปนี้

1. บิดา มารดา และพี่น้อง ทุกคน ที่คอยให้กำลังใจและห่วงใยในการศึกษาเล่าเรียน
2. อธิการบดีสถาบันราชภัฏยะเชิงเทรา ที่สนับสนุนการศึกษาให้เป็นไปด้วยความเรียบร้อย
3. รศ.ดร.ศุภมิตร จิตตะยโสธร อาจารย์ที่ปรึกษาโครงการพัฒนาระบบงานที่กรุณาให้คำแนะนำในด้านการศึกษาปัญหา แนะนำแนวทางในการศึกษาและแก้ไขปัญหาต่างๆ ที่เกิดขึ้น ตลอดจนให้ความสะดวกในการใช้ซอฟต์แวร์สำหรับพัฒนาระบบ
4. คุณเทอดศักดิ์ ปรีชา ตำแหน่ง Product Manager ของบริษัท Abstract ซึ่งให้ความช่วยเหลือและแนะนำการใช้งานโปรแกรม Pilot Decision Support เป็นอย่างดี
5. ภาควิชาคอมพิวเตอร์ศึกษา สถาบันราชภัฏยะเชิงเทราที่ให้ความสะดวกในการใช้อุปกรณ์สำหรับพัฒนาระบบงาน
6. นางสาวทิพวรรณ วิภาตผลิน ที่ช่วยสนับสนุนงานด้านเอกสาร

นางสาวเพ็ญศรี บั๊กกะสีนัง

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	1
บทคัดย่อภาษาอังกฤษ .....	2
กิตติกรรมประกาศ.....	3
สารบัญ.....	4
สารบัญตาราง .....	7
สารบัญภาพ.....	8
บทที่	
1. บทนำ.....	10
1.1 ความเป็นมา .....	10
1.2 วัตถุประสงค์ .....	11
1.3 ขั้นตอนการศึกษา.....	12
1.4 ขอบเขตของการศึกษา.....	13
1.5 ประโยชน์ที่คาดว่าจะได้รับ .....	13
2. ระบบสนับสนุนการตัดสินใจ (Decision Support System).....	14
2.1 ระบบสนับสนุนการตัดสินใจคืออะไร.....	14
2.2 วิวัฒนาการของระบบสนับสนุนการตัดสินใจ.....	14
2.3 วงชีวิตของระบบสนับสนุนการตัดสินใจ (Decision Support Life Cycle) .....	18
3. แนวคิดเกี่ยวกับคลังข้อมูล (Data Warehouse) .....	22
3.1 แนวคิดเกี่ยวกับคลังข้อมูล .....	22
3.2 ลักษณะของคลังข้อมูล .....	27
3.3 ประโยชน์ของคลังข้อมูล.....	29
4. สถาปัตยกรรมของคลังข้อมูล ( Data Warehouse Architecture ) .....	30
4.1 สถาปัตยกรรมข้อมูล (Data Architecture).....	30
4.2 สถาปัตยกรรมของคลังข้อมูล .....	31

4.3 การประยุกต์ใช้งานรูปแบบต่าง ๆ .....	35
5. แนวทางในการสร้างคลังข้อมูล .....	40
5.1 องค์ประกอบที่สำคัญในการสร้าง Data warehouse.....	40
5.2 การจัดการกับข้อมูลเพื่อนำเข้าสู่ Data warehouse .....	42
5.3 การออกแบบฐานข้อมูลสำหรับ Data warehouse .....	45
5.4 การสร้างลำดับชั้นของข้อมูล .....	52
5.5 แนวทางในการเรียกใช้ข้อมูล .....	54
6. ทฤษฎีการออกแบบที่เกี่ยวข้องกับการสร้างคลังข้อมูล (Data Warehouse) .....	57
6.1 ระบบฐานข้อมูล .....	57
6.2 รูปแบบเชิงสัมพันธ์ .....	61
6.3 การจัดการฐานข้อมูลเชิงสัมพันธ์ Relational Database ( RDBMS) .....	63
6.4 รูปแบบหลายมิติ .....	65
6.5 ระบบการจัดการฐานข้อมูลหลายมิติ Multidimensional Database (MDBMS) ..	76
6.6 ความสัมพันธ์ระหว่าง Multidimension Database กับ Data Warehouse .....	78
6.7 เปรียบเทียบระหว่าง Relational Database กับ Data Warehouse.....	80
7. ศึกษาเครื่องมือ (Tools) ที่ช่วยในการสร้างคลังข้อมูลเพื่อการตัดสินใจ (Pilot Decision Support System ) .....	81
7.1 บทบาทของ Tools ในการพัฒนา Data Warehouse.....	81
7.2 ส่วนประกอบของ Pilot Decesion Support .....	81
7.3 การทำงานของ Pilot Decision Support.....	85
8. วิเคราะห์ และเปรียบเทียบ Tools ที่ใช้ในการสร้าง Data warehouse .....	93
8.1 โครงสร้างข้อมูล (Data Structure) .....	93
8.2 ความถูกต้องของข้อมูล (Data Integrity) .....	93
8.3 การจัดการข้อมูล (Data manipulate).....	94
9. กรณีศึกษาการสร้าง Data Warehouse สำหรับสนับสนุนการตัดสินใจของผู้บริหาร ในสถานศึกษา : สถาบันราชภัฏฉะเชิงเทรา .....	95

9.1 ความเป็นมาของปัญหา.....	95
9.2 วัตถุประสงค์.....	96
9.3 ขอบเขตของการศึกษา.....	96
9.4 ขั้นตอนดำเนินงาน.....	96
10. สรุปผลการศึกษา.....	113
10.1 ผลการศึกษา.....	113
10.2 ประโยชน์ที่ได้รับ.....	113
10.3 ข้อเสนอแนะ.....	114
บรรณานุกรม.....	115
ประวัติผู้เขียน.....	118



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่	หน้า
2.1 เปรียบเทียบการจัดการข้อมูลสำหรับการประมวลผล OLTP กับ OLAP.....	16
6.1 แสดงการแทนข้อมูลด้วยจุด Coordinate.....	70
6.2 ตารางเปรียบเทียบ Database กับ Datawarehouse.....	79



## สารบัญภาพ

รูปที่	หน้า
2.1	วิวัฒนาการของการประมวลผลข้อมูล..... 14
2.2	การแยกโปรแกรมออกจากฐานข้อมูล ..... 15
2.3	ความต้องการใช้ข้อมูล ..... 15
2.4	พัฒนาการของการใช้ฐานข้อมูลตั้งแต่อดีต-ปัจจุบัน ..... 18
2.5	ช่วงเวลาในการสรุปข้อมูลเข้าสู่ Data warehouse..... 19
3.1	การนำข้อมูลมาใช้ให้เกิดประโยชน์.....25
3.2	แนวคิดการใช้ข้อมูลสำหรับสนับสนุนการตัดสินใจ..... 26
3.3	การเลือกหัวข้อที่ต้องการ..... 27
3.4	การรวมข้อมูล ..... 28
3.5	การเรียกใช้ข้อมูลโดยไม่มี การเปลี่ยนแปลง..... 30
4.1	สถาปัตยกรรมของการใช้ข้อมูล..... 31
4.2	สถาปัตยกรรมของ Data Warehouse ..... 32
4.3	สถาปัตยกรรมพื้นฐานของการทำงานใน Data warehouse..... 36
4.4	สถาปัตยกรรมการทำงาน Data warehouse แบบ Enterprise ..... 37
4.5	สถาปัตยกรรมการทำงานแบบแยก Data warehouse..... 38
4.6	แสดงการรวมกันของข้อมูลใน Data warehouse..... 39
5.1	Star schema ..... 46
5.2	Entities และ Relationship..... 47
5.3	แสดงการกำหนดขอบเขตก่อนการรวมข้อมูล ..... 48
5.4	แสดงโครงสร้างของการรวม ERD จากหลาย ๆ มุมมองของผู้ใช้ เข้าด้วยกัน.. 48
5.5	แสดงโครงสร้างของ Data Item ใน ERD ..... 49
5.6	กำหนด Data item และความสัมพันธ์ระหว่าง entity..... 50

## สารบัญภาพ (ต่อ)

5.7 แสดงปัจจัยที่มีผลต่อประสิทธิภาพของฐานข้อมูล ใน Data Warehouse .....	51
5.8 แสดงลำดับชั้นของข้อมูล.....	53
5.9 แสดง Multiple Hierarchies .....	54
5.10 แสดง ระดับของ tools สำหรับการเรียกใช้งาน .....	55
5.11 แสดงสถาปัตยกรรม Three tier Client/Server.....	56
6.1 แสดงความสัมพันธ์ระหว่างข้อมูลแบบโครงข่าย .....	57
6.2 แสดงความสัมพันธ์ระหว่างข้อมูลแบบลำดับชั้น.....	58
6.3 แสดงความสัมพันธ์ระหว่างข้อมูลแบบเชิงสัมพันธ์.....	58
6.4 สถาปัตยกรรมนิยามข้อมูล 3 ระดับ.....	59
6.5 แสดงรายละเอียดของ Relation .....	61
6.6 แสดงโครงสร้างข้อมูลขนาด 1 x 1 x 1 .....	66
6.7 แสดงโครงสร้างของ Multidimensional Model .....	67
6.8 แสดงการ slice ข้อมูล จาก cube.....	68
6.9 แสดงตารางก่อนนำเข้าสู่ Multidimensional Model.....	69
6.10 แสดงข้อมูลใน Multidimensional Model.....	69
6.11 แสดงการหมุนข้อมูล .....	72
6.12 แสดงถึงมุมมองทั้ง 6 ด้านที่สามารถเข้าถึงข้อมูลโดยการหมุนหลายมิติ.....	73
6.13 การมองข้อมูลเป็นกลุ่มย่อย .....	74
6.14 แสดงระดับความสัมพันธ์ของข้อมูล.....	75
6.15 การแบ่งข้อมูลเป็นกลุ่ม.....	76
7.1 แสดงการเข้าสู่ Pilot Analysis Server .....	82
7.2 แสดงจอภาพของ Pilot Designer .....	83
7.3 แสดงเมนูของ Pilot Desktop.....	83

## สารบัญภาพ (ต่อ)

7.4	สถาปัตยกรรมของ Pilot Decision Support .....	84
7.5	แสดงสถาปัตยกรรม Three-tier ของ Pilot Decision Support .....	84
7.6	แสดงจอภาพของการกำหนด Link Id .....	85
7.7	แสดงโครงสร้างของ Model.....	85
7.8	แสดงจอภาพของการสร้าง Model Builder.....	86
7.9	แสดงการสร้าง Model .....	87
7.10	เมนูของ Pilot Desktop .....	88
7.11	แสดงการ View ข้อมูลโดยใช้ Navigator .....	89
7.12	แสดงการมองข้อมูลเป็นกราฟ.....	89
7.13	แสดงการจัดลำดับข้อมูล .....	90
7.14	กราฟแสดงยอดขายแบ่งเป็นปี.....	90
7.15	แสดงจอภาพของการใช้ Exception .....	91
7.16	แสดงการคำนวณราคาที่ต้องการ .....	91
7.17	แสดงการเก็บข้อมูลใน Portfolio.....	92
9.1	แสดงความสัมพันธ์ระหว่าง Entity หรือ ERD ของระบบ .....	103
9.2	กำหนด Data Item ของแต่ละ Entity.....	104
9.3	Star Schema ของระบบทะเบียนการเงิน .....	105
9.4	Star Schema ของระบบใหม่.....	106
9.5	แสดงการสร้าง Model และ Dimension .....	110
9.6	แสดงการ View ข้อมูล.....	111
9.7	การแสดงผลข้อมูลด้วยกราฟ.....	111
9.8	แสดงกำหนดมุมมอง.....	112
9.9	แสดงการจัดลำดับข้อมูลโดยให้ Ranging Analysis .....	112

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในยุคปัจจุบันเป็นยุคของเทคโนโลยีสารสนเทศ ( Information Techonology ) ซึ่งเป็นยุคที่มีการแข่งขันกันอย่างรุนแรงในโลกของธุรกิจ ผู้ที่มีข้อมูลที่เหนือกว่า ถูกต้องกว่า และแม่นยำกว่า ย่อมได้เปรียบในการนำข้อมูลมาจัดการวางแผนกลยุทธ์ทางธุรกิจ หรือกำหนดทิศทางการของธุรกิจ ทำให้ธุรกิจนั้นๆ สามารถผ่านอุปสรรคและบรรลุถึงเป้าหมายได้เป็นอย่างดี เพราะฉะนั้นทุกองค์กรต่างก็พยายามที่จะนำเทคโนโลยีใหม่ๆ เข้ามาช่วยในการพัฒนาระบบงานเพื่อให้ได้ข้อมูลที่เหนือกว่าในทุก ๆ ด้าน ไม่ว่าจะเป็นระบบอินเทอร์เน็ต (Internet) หรือระบบคอมพิวเตอร์แบบ Client-Server หรือระบบเปิด (Open System) หรือการทำ Business Reengineering หรือการทำคลังข้อมูลเพื่อการบริหาร (Data Warehouse ) เทคโนโลยีดังกล่าวจึงถูกเลือกและนำมาใช้ เพื่อปรับเปลี่ยน กระบวนการ (Process) การทำงานขององค์กรให้ทันสมัยยิ่งขึ้น โดยเฉพาะเทคโนโลยีคลังข้อมูล (Data Warehouse ) ซึ่งเป็นเทคโนโลยีที่รวมหลาย ๆ เทคโนโลยีเข้าด้วยกันเพื่อประยุกต์ใช้ข้อมูลขององค์กรให้เกิดประสิทธิภาพสูงสุด กล่าวคือข้อมูลขององค์กรที่มีอยู่และใช้ประจำวันหรือ Operational Database นับวันจะมีมากขึ้นจนเป็นปัญหาสำหรับองค์กรที่จะต้องจัดการเก็บข้อมูล ซึ่งข้อมูลเหล่านั้นอาจเก็บรวบรวมอยู่ในรูปแบบต่างๆ ไม่ว่าจะเป็น เทปแม่เหล็ก ดิสก์เก็ตหรือ ในแผ่นซีดี นอกจากนี้ข้อมูลจำนวนมากเหล่านี้ยังไม่เหมาะที่จะนำมาใช้ในการวิเคราะห์ทางธุรกิจหรือช่วยผู้บริหารสำหรับการตัดสินใจ (Decision Support System) ทางธุรกิจได้ เพราะต้องใช้เวลาในการประมวลผลที่นานกว่าจะได้คำตอบที่ต้องการ และส่งผลกระทบต่อระบบการทำงานของเครื่องที่ใช้งานประจำวันด้วย เพื่อแก้ปัญหาดังกล่าวเพื่อที่จะให้สามารถนำข้อมูลที่มีอยู่มาใช้ตอบสนองความต้องการทางธุรกิจได้อย่างรวดเร็วและมีประสิทธิภาพ จึงทำให้เกิดแนวความคิดเกี่ยวกับ Data warehouse ขึ้นเพื่อตอบสนองงานในรูปแบบของคลังเก็บข้อมูลสำหรับผู้บริหาร โดย Data Warehouse จะมีการรวบรวมข้อมูล จากส่วนปฏิบัติงานต่าง ๆ จากหลาย ๆ แหล่งทั้งในอดีตและปัจจุบัน และเลือกเฉพาะหัวข้อที่ต้องการ มารวมไว้ด้วยกันเป็นคลังข้อมูลขององค์กร เพื่อให้ง่ายต่อการเรียกใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวความคิดเกี่ยวกับคลังข้อมูล (Data Warehouse) เป็นแนวคิดที่ได้รับความสนใจเป็นอย่างมากในทุกวงการที่จะนำแนวคิดดังกล่าวมาประยุกต์กับงานของตนเอง แต่ทั้งนี้ในการประยุกต์ใช้งานจริง ๆ ไม่ได้มีการกำหนดรูปแบบที่แน่นอนตายตัว ขึ้นอยู่กับสภาพแวดล้อมของแต่ละองค์กรที่จะนำมาประยุกต์ใช้ให้เหมาะสมกับองค์กรของตนเองให้มากที่สุด แต่ในการสร้าง Data Warehouse ของทุก ๆ องค์กรยังคงยึดหลักการในการสร้าง Warehouse ที่เหมือน ๆ กัน

## 1.2 วัตถุประสงค์ของการศึกษา

1.2.1 เพื่อเป็นแนวทางในการบริหารข้อมูลสำหรับองค์กร

1.2.2. เพื่อเป็นแนวทางในการสร้างคลังข้อมูลสำหรับองค์กร เพื่อให้สามารถใช้ข้อมูลให้เกิดประสิทธิภาพสูงสุด

1.2.3 เพื่อชี้ให้เห็นประโยชน์ของการสร้างคลังข้อมูล(Data Warehouse)สำหรับการบริหารซึ่งช่วยสนับสนุนการตัดสินใจของผู้บริหารได้ (Decision Support )

## 1.3 ขั้นตอนของการศึกษา

ขั้นตอนการศึกษา	เดือน( ปี 2540 )			
	ก.ค.	ส.ค.	ก.ย.	ต.ค.
ศึกษาแนวคิดเกี่ยวกับคลังข้อมูลเพื่อสนับสนุนการตัดสินใจ	←→			
ศึกษาทฤษฎีการออกแบบที่เกี่ยวข้องกับการสร้างคลังข้อมูล	←→			
ศึกษา Tools ที่ใช้ในการสร้างคลังข้อมูลเพื่อการตัดสินใจ Pilot Decision Support		←→		
ศึกษาระบบงานลงทะเบียนและการเงิน(ค่าลงทะเบียน)			←→	
สร้าง Model ระบบงานทะเบียนและการเงิน โดย Pilot Decision support				←→

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.4 ขอบเขตของการศึกษา

การศึกษาวิธีการสร้างคลังข้อมูลสำหรับสนับสนุนการตัดสินใจ มีขอบเขตในการศึกษาดังนี้

- 1.4.1. ศึกษาแนวคิดเกี่ยวกับระบบสนับสนุนการตัดสินใจ
- 1.4.2. ศึกษาแนวคิดเกี่ยวและทฤษฎีที่เกี่ยวข้องกับคลังข้อมูล
- 1.4.3. ศึกษาทฤษฎีการออกแบบที่เกี่ยวข้องกับคลังข้อมูล
- 1.4.4. ศึกษาเครื่องมือ Tools ที่ช่วยในการสร้างคลังข้อมูล
- 1.4.5. วิเคราะห์และเปรียบเทียบข้อดีข้อเสียของคลังข้อมูล
- 1.4.6. กรณีศึกษาการสร้างคลังข้อมูลสำหรับงานทะเบียน-การเงิน

(การเก็บค่าลงทะเบียน ) ของสถาบันราชภัฏฉะเชิงเทราโดยมีขอบเขตดังนี้

- 1.4.6.1 สร้างฐานข้อมูลสรุป
- 1.4.6.2 สรุปจำนวนนักศึกษา และสรุปจำนวนเงินค่าลงทะเบียน
- 1.4.6.3 เลือกดูข้อมูลตามหัวข้อที่ต้องการ
- 1.4.6.4 วิเคราะห์ข้อมูลสรุปจำนวนนักศึกษาและรายได้

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

องค์กรสามารถนำแนวความคิดเกี่ยวกับ Data Warehouse ไปประยุกต์ใช้กับระบบงานเพื่อบริหารข้อมูลที่มีอยู่จำนวนมาก ให้เป็นข้อมูลสำหรับการตัดสินใจ โดยสามารถเลือกรูปแบบของกัวิประยุกต์ใช้ Data Warehouse ได้หลาย ๆ รูปแบบ และปรับให้เข้ากับสภาพโดยทั่วไปขององค์กร ทำให้สามารถที่จะใช้ข้อมูลได้อย่างเต็มที่ โดยนำข้อมูลเหล่านั้นมาวิเคราะห์เพื่อหาแนวโน้ม และทำรายงานต่าง ๆ ตามที่ผู้บริหารต้องการได้ หรือผู้บริหารสามารถที่จะเลือกรูปแบบการมองข้อมูลในรูปแบบต่าง ๆ ตามต้องการได้ โดยไม่ต้องรอโปรแกรมจากโปรแกรมเมอร์ ทำให้สะดวกต่อการใช้งาน ทั้งนี้ขึ้นอยู่กับ Tools ที่เลือกใช้งานด้วยจะมีขีดความสามารถในการทำงานได้มากน้อยเพียงใด เพราะฉะนั้นในการเลือกใช้ Tools ก็ควรเลือก Tools ที่มีประสิทธิภาพในการทำงานสูง และง่ายต่อการใช้งาน

## บทที่ 2

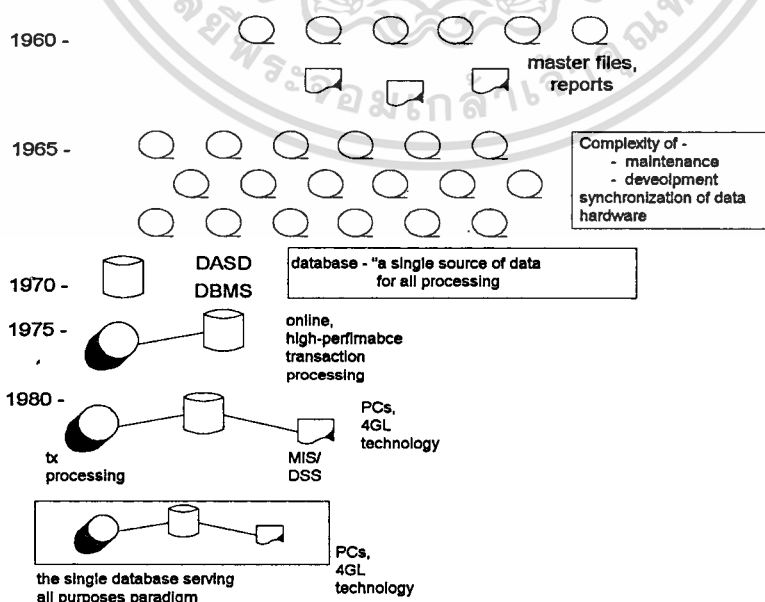
### ระบบสนับสนุนการตัดสินใจ (Decision Support System)

#### 2.1 ระบบสนับสนุนการตัดสินใจคืออะไร

ระบบสนับสนุนการตัดสินใจเป็นระบบที่ให้บริการข้อมูลข่าวสารแก่ผู้ใช้ ( User ) ให้สามารถวิเคราะห์ข้อมูลและสามารถตัดสินใจในสถานการณ์ที่อาจเกิดขึ้นได้อย่างเหมาะสม โดยอาศัย Application ที่มีประสิทธิภาพช่วย Access และวิเคราะห์ข้อมูล พร้อมทั้ง Model ในการเรียกดูข้อมูล และช่วยในการตัดสินใจ ตัวอย่าง การวิเคราะห์ยอดขายของสินค้าในระยะเวลา 2 ปี และดูว่ายอดขายเป็นอย่างไร สมควรที่จะผลิตต่อไปหรือยกเลิก และออกผลิตภัณฑ์ตัวใหม่ แทนการเก็บสะสมข้อมูลไว้จะช่วยให้ผู้บริหารสามารถตัดสินใจได้ ซึ่งเป็นจุดเด่นของระบบสนับสนุนการตัดสินใจ

#### 2.2 วิวัฒนาการของระบบสนับสนุนการตัดสินใจ

การใช้ข้อมูลในการตัดสินใจของมนุษย์มีมาตั้งนานแล้ว แต่การใช้ข้อมูลสำหรับการตัดสินใจโดยผ่านกระบวนการประมวลผลด้วยคอมพิวเตอร์ เพิ่งเริ่มต้นประมาณปี 1960 โดยเริ่มจากการประมวลผลข้อมูลที่เป็น Master File บันทึกข้อมูลใน Magnetic tap และมีการพัฒนามาเรื่อยๆจนเป็น Transaction ดังรูปที่ 2.1



เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 2.1 วิวัฒนาการของการประมวลผลข้อมูล อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

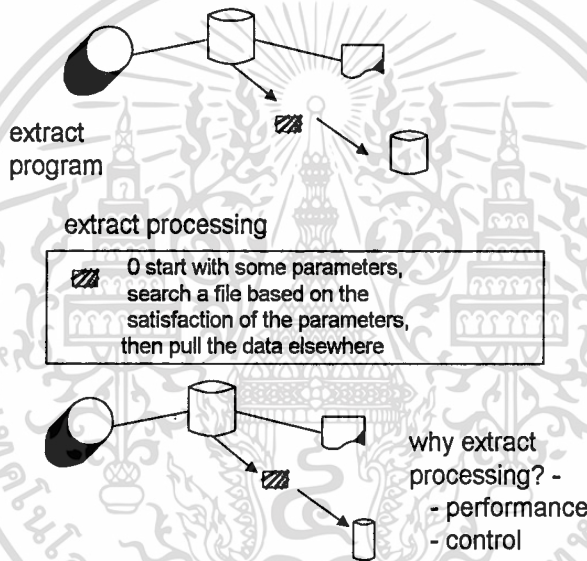
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน 1970 มีการประมวลผลแบบ Online transaction Processing เริ่มมีฐานข้อมูล (database )

ใน 1980 เกิด Technology ใหม่ มีระบบ Mis , DSS database ต้อง Support ทั้ง 2 ระบบ คือ Transaction Processing และ DSS วิเคราะห์และ Process ในเวลาเดียวกัน กล่าวคือฐานข้อมูลเดียวต้องให้บริการทุก ๆ งาน มีผลทำให้ประสิทธิภาพการทำงานลดลง

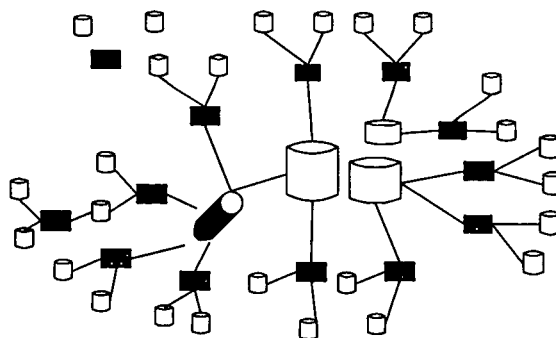
ต่อมาในปี 1985 มีการแยกโปรแกรมสำหรับการประมวลผลออกจากฐานข้อมูล เพื่อเพิ่มประสิทธิภาพและง่ายต่อการควบคุม ดังรูปที่ 2.2

1985 -



รูปที่ 2.2 การแยกโปรแกรมออกจากฐานข้อมูล

ในปี 1990 ความต้องการใช้โปรแกรมและข้อมูลมีมากขึ้นทำให้เกิดการกระจายเชื่อมต่อกันไปเรื่อย ๆ เกิดเป็นใยแมงมุม ดังรูปที่ 2.3



รูปที่ 2.3 ความต้องการใช้ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากความต้องการใช้ข้อมูลดังกล่าว ทำให้เกิดปัญหาต่าง ๆ ตามมามากมายเป็นต้นว่า

- ปัญหาเกี่ยวกับข้อมูลที่มีปริมาณมาก
- ปัญหาการใช้งาน Program
- ปัญหาในการทำรายงานสรุป 3 - 5 ปี
- ปัญหาในการจัดเก็บและจัดการกับข้อมูล

เพื่อแก้ปัญหาดังกล่าว จึงมีการเปลี่ยนวิธีการจัดการข้อมูลแบบเก่ามาเป็นระบบใหม่ เพื่อเป็นข้อมูลสำหรับสนับสนุนการตัดสินใจ โดยอาศัยข้อมูลจากฐานข้อมูลเก่ามาปรับเปลี่ยนให้อยู่ในรูปที่เหมาะสมสำหรับการเรียกใช้งาน

Primitive Data/Operation Data	Derived Data/DSS Data
1. เก็บรายละเอียดของข้อมูล	1. เก็บข้อมูลสรุป
2. สามารถทำการ Update ได้	2. ไม่สามารถ Update ข้อมูลได้
3. เข้าสู่ข้อมูลแบบ Unit at a time	3. เข้าสู่ข้อมูลแบบ Set at a time
4. ข้อมูลเป็น Transaction Driven	4. ข้อมูลเป็น Analysis Driven
5. ข้อมูลไม่มีความซ้ำซ้อน nonredundancy	5. ข้อมูลมีความซ้ำซ้อน Redundancy
6. โครงสร้างไม่เปลี่ยนแปลง static Structure	6. โครงสร้างมีความยืดหยุ่น Flexible
7. ประมวลผลฐานข้อมูลขนาดเล็ก	7. ประมวลผลฐานข้อมูลขนาดใหญ่
8. สนับสนุนการปฏิบัติงาน Day -to -day	8. จัดการได้ตามความต้องการ
9. โปรแกรมใช้งานเป็น Application oriented	9. โปรแกรมใช้งานเป็น Analysis oriented
10. การใช้ข้อมูล High availability	10. การใช้ข้อมูล Relax availability

ตารางที่ 2.1 เปรียบเทียบการจัดการข้อมูลสำหรับการประมวลผล OLTP กับ OLAP

จะเห็นว่าการใช้ข้อมูลตั้งแต่อดีตจนถึงปัจจุบันจะมีพัฒนาการที่ซับซ้อนมาก โดยเฉพาะ Data Warehouse ที่เป็นการพัฒนาการและผสมผสานความต่างของเทคโนโลยีระหว่าง OLAP และ OLTP เข้าด้วยกัน ซึ่งพอสรุปพัฒนาการของการใช้ข้อมูลได้ดังนี้

- Operational Report เป็นระดับของการพัฒนาการใช้ข้อมูลในระยะเริ่มแรกที่ต้องอาศัยการขอข้อมูลจากแหล่งหนึ่ง โดยไม่มีการเชื่อมโยงข้อมูลเข้าด้วยกัน

- End User Access to Production Systems เป็นการเชื่อมโยงข้อมูลระหว่างเครื่องคอมพิวเตอร์ด้วยกันและสามารถข้ามระบบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- DSS Overload the Production Environment เป็นการใช้ Operational System เป็นศูนย์กลางกระจายข้อมูลไปยังผู้ต้องการใช้ข้อมูลซึ่งหากมีความต้องการใช้ในเวลาที่พร้อม ๆ กัน จะทำให้การเรียกใช้ข้อมูลประสบความสำเร็จ ความยุ่งยาก เสียเวลาเพราะการดึงข้อมูลจะสามารถทำได้ครั้งละ 1 เครื่องเท่านั้น

- End User Access to Data Extracts เป็นการกระจายข้อมูลโดยอาศัย Data Extract และเป็นศูนย์กลางการกระจายข้อมูลระหว่าง Operational System กับ End user ทั้งนี้จะช่วยลดความซ้ำซ้อนของข้อมูลในองค์กร

- Same Data ,Different Results เป็นการเชื่อมโยงข้อมูลจากหลาย ๆ แหล่งมารวมกัน โดยอาศัย Data Extracts เพื่อป้องกันความผิดพลาด แต่บางครั้งก็มีการผิดพลาด

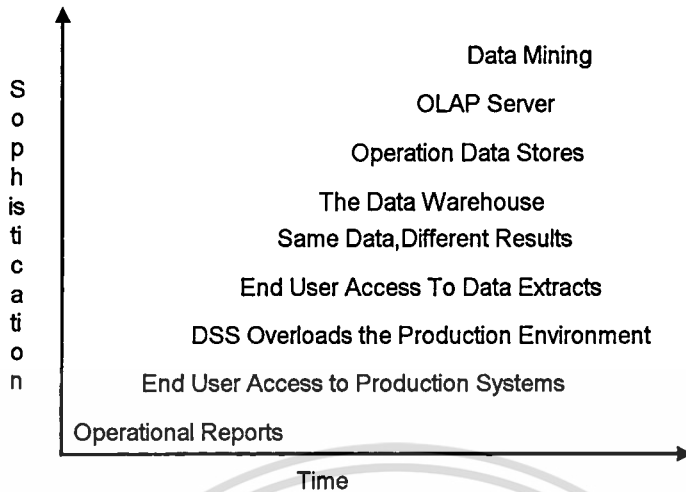
- The Data Warehouse เป็นการรวบรวมข้อมูลต่าง ๆ จากหลายเครื่องมารวมกันที่ Data warehouse เพื่อสะดวกแก่ผู้เรียกใช้

- Operation Data Stores เป็นลักษณะการดึงข้อมูลจาก Operation System โดยตรง โดยตัวฐานข้อมูลจะทำการเก็บข้อมูลเพื่อทำการ Update ข้อมูล

- Data Marts เป็นการแยกเก็บข้อมูล (Transaction) จาก Data warehouse โดยใช้เทคโนโลยี Multidimension Database เพื่อเก็บข้อมูลให้กับแผนกที่มีการเรียกใช้บ่อย เพื่อไม่ให้เกิดความซ้ำซ้อนในหน่วยงานที่เกี่ยวข้อง

- OLAP Server เป็นการพัฒนาสูงขึ้นที่สูงกว่าในระบบ Data Warehouse โดยมีการนำเอา OLAP Servers มาเพิ่มซึ่งจะมีความจำเป็นสำหรับองค์กรใหญ่ ๆ ที่มีการจัดเก็บระบบข้อมูลมาก ๆ

- Data Mining ถือได้ว่าเป็นระดับการจัดเก็บข้อมูลที่สูงสุด เนื่องจากเป็นวิธีคิดที่จะนำข้อมูลมาใช้ในการวิเคราะห์ ให้เกิดประโยชน์สูงสุด โดยเฉพาะอย่างยิ่งสำหรับการตัดสินใจของฝ่ายบริหาร ซึ่งระบบนี้เป็นขั้นตอนต่อไปของ data warehouse มีระบบการทำงานอัตโนมัติ สามารถตัดสินใจแทนคนได้โดยอาศัยกฎเกณฑ์ต่าง ๆ ที่กำหนดขึ้นแล้วป้อนใส่คอมพิวเตอร์



รูปที่ 2.4 พัฒนาการของการใช้ฐานข้อมูลตั้งแต่อดีต-ปัจจุบัน

ทุกระบบที่ปรับเปลี่ยนก็เพื่อที่จะหาวิธีการที่ดีที่สุดสำหรับจัดการกับข้อมูล เพื่อให้ นำข้อมูลมาใช้ให้เกิดประโยชน์สูงสุด โดยเฉพาะข้อมูลสำหรับสนับสนุนการตัดสินใจที่จำเป็นต้องมีความละเอียดสูง และพร้อมที่จะใช้งานได้ทุกเมื่อ

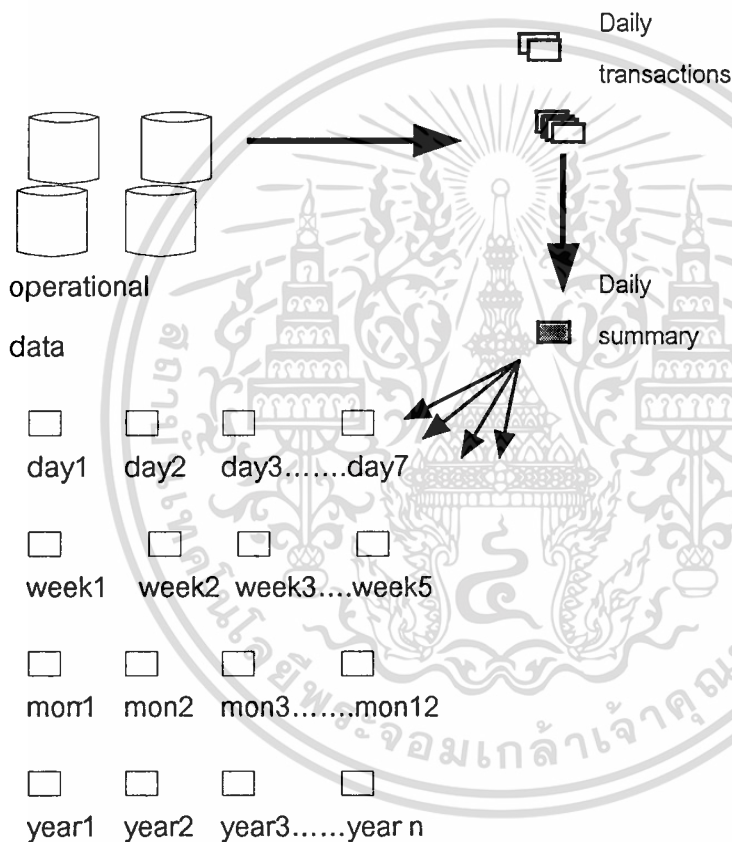
### 2.3 วงชีวิตของระบบสนับสนุนการตัดสินใจ ( Decision Support Life Cycle)

การพัฒนากระบบสำหรับการประมวลผลแบบ On-Line เป็นการประมวลผลข้อมูลที่เป็น Transaction หรือที่เรียกว่า On Line Transaction Processing (OLTP) มีวงจรสำหรับการพัฒนาที่เรียกว่า System Development Life Cycle (SDLC) ซึ่งมีหลายขั้นตอนตั้งแต่ Planning ,Analysis,Design,Development,Testing และ Implement ส่วนการพัฒนาคลังข้อมูลสำหรับการตัดสินใจก็จะมีวงจรในการพัฒนาข้อมูลเช่นเดียวกัน ซึ่งเรียกววงจรพัฒนาระบบสำหรับสนับสนุนการตัดสินใจว่า Decision Support Life Cycle (DSLCL) ซึ่งมีหลาย ๆ ขั้นตอนที่เหมือนและแตกต่างกัน กับ SDLC โดยเฉพาะข้อมูลที่ออกแบบเพื่อการตัดสินใจของระบบ DSLC กับข้อมูลที่ออกแบบเพื่อการประมวลผลแบบ On - Line Transaction ของระบบ SDLC

จุดเด่นของข้อมูลสำหรับการตัดสินใจคือ เป็นข้อมูลสรุป อาจจะเป็นวัน เป็นสัปดาห์ เป็นเดือน หรือเป็นปีดังรูปที่ 2.5 ทั้งนี้ขึ้นอยู่กับความต้องการของแต่ละองค์กร และมี Model สำหรับการเรียกดูและวิเคราะห์ข้อมูล ช่วยตอบคำถามที่ผู้บริหารต้องการ ข้อมูลเป็นข้อมูลที่รวบรวมมาจากฐานข้อมูลปฏิบัติงาน (Operational) ฐานข้อมูลมีขนาดใหญ่เก็บข้อมูลตั้งแต่ Mega Byte จนถึง Tera Byte ข้อมูลมาจากหลายรูปแบบ (Various Plat Forms) อาจเป็น Database, File และจาก Source ข้อมูลอื่น ๆ ต้องใช้เครื่องมือ (Tools) และเทคนิคพิเศษในการพัฒนาข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้สามารถเรียกใช้ข้อมูลให้เกิดประสิทธิภาพสูงสุด ในการพัฒนาข้อมูลสำหรับสนับสนุนการตัดสินใจหรือที่เรียกว่า Data warehouse ต้องพิจารณาถึงสิ่งแวดล้อมหลาย ๆ อย่าง หลาย ๆ กรณี เป็นต้นว่า โครงสร้างพื้นฐานด้านเทคนิคต่าง ๆ ที่เกี่ยวข้อง เครื่องมือ เทคโนโลยีที่จำเป็น หรือเทคโนโลยีที่ช่วยให้การพัฒนาระบบประสบความสำเร็จ เพราะในการทำ Data Warehouse งานหลัก ๆ ที่ ต้องทำ คือ การจัดการกับสิ่งแวดล้อมหลายรูปแบบ เป็นต้นว่า การรวมข้อมูล(Integrate ) การสรุปข้อมูล, Platform ,Data Structure, Database รวมทั้งการจัดการกับระบบ Hardware ที่ดีด้วย



รูปที่ 2.5 แสดงช่วงเวลาในการสรุปข้อมูลเข้าสู่ Data warehouse

วงชีวิตของระบบสนับสนุนการตัดสินใจ มีทั้งหมด 10 ขั้นตอน

ขั้นที่ 1 การวางแผน ( Planning )

การสร้าง Project plan และมีการกำหนดระยะเวลาการพัฒนาระบบทำให้มีความยากมากสำหรับ Project manager ใหม่ ที่จะต้องมาทำงานในรูปแบบของ DSLC project plan จะ

อธิบายรายละเอียดในการสร้าง Data Warehouse เช่น กำหนดขอบเขตการสร้าง กำหนดการใช้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลทั้งภายนอกและภายในองค์กร กำหนดระยะเวลาต่างๆ ทั้งขั้นตอนการสร้างระบบ การส่งระบบงาน

ขั้นที่ 2 รวบรวมความต้องการ (Gathering Data Requirement and Modeling)

ความเข้าใจทางธุรกิจอย่างถ่องแท้ รวบรวมความคิดเห็น และความต้องการของผู้ใช้ โดยสิ่งที่ได้จะนำมาประมวลแล้วจะถูกนำไปสร้างเป็น Modeling

ขั้นที่ 3 ออกแบบฐานข้อมูล และพัฒนา (Physical database design and development )

ในขั้นตอนนี้จะรวมถึงการออกแบบฐานข้อมูล และการ Denormalization ในขั้นตอนนี้จะต้องมีความรู้เกี่ยวกับ Decision Support Concept , Concept of hierarchies dimension and facts , star schemes

ขั้นที่ 4 Data Mapping and Transformation

จะทำการรวบรวม Database design จัดการนำข้อมูลในระดับปฏิบัติการ ทำการแปลงข้อมูลที่ซับซ้อนในอยู่ในรูปที่เข้าใจ และ ทำการ Mapping source ให้เข้ากับ การออกแบบฐานข้อมูลของ Data Warehouse

ขั้นที่ 5 Population the Data Warehouse

จะมีขั้นตอนการทำ Extracting , Transforming และ Loading data ไปใส่ในฐานข้อมูล

1. การพัฒนาวิธีการในการ Extract และ move data ตัวอย่างกลไกการ extract data เช่น simple extract mechanism, time stamp extract mechanism, refresh mechanism

2. การพัฒนา วิธีการในการ load ข้อมูลไปใส่ใน Data Warehouse

3. การพัฒนา โปรแกรม หรือ ใช้ data transformation tool ในการ Transform และ Integrate

4. การทดสอบ

ขั้นที่ 6 Automating Data Management Procedures

ในขั้นตอนนี้จะทำการ Extraction , Transformation และ Load data ลง Data Warehouse โดยอัตโนมัติ มีการสร้าง Backup และ Recovery

ขั้นที่ 7 Application Development Creating the starter set of reports

การสร้าง starter set of predefined reports พัฒนา Report ทดสอบ Report จัดทำเอกสาร คู่มือการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ขั้นที่ 8 การตรวจสอบ และทดสอบ (Data Validation and testing)

ต้องมีการกำหนดมาตรฐานสำหรับการได้มาซึ่งข้อมูลที่ต้องการ การประยุกต์ไปใช้กับงาน Data Warehouse ได้อย่างเหมาะสม

### ขั้นที่ 9 การฝึกอบรม ( Training )

ทำการฝึกอบรมผู้ใช้โดยที่เนื้อหาจะต้องมี ข้อมูลใน warehouse ตัวหน้าจอสําหรับเรียกดูข้อมูล ขั้นตอนการใช้งาน คอยช่วยแก้ปัญหาที่เกิดขึ้นเพื่อทำให้ระบบดำเนินการต่อไปได้

### ขั้นที่ 10 Rollout

ทำการลงระบบให้ผู้ใช้ได้ใช้งานจริง ฝึกอบรมบุคลากรเพื่อเป็นผู้ดูแลระบบ สร้างแนวทางในการพัฒนา Report ในอนาคต กำหนดวิธีการ Backup data

### สรุป

จากที่กล่าวมาจะเห็นว่าความต้องการใช้ข้อมูล ในการตัดสินใจทำให้เกิดวิธีการต่าง ๆ ที่ จะจัดการข้อมูลเหล่านั้นให้สามารถนำมาใช้ให้เกิดประโยชน์สูงสุดได้

อาจกล่าวได้ว่า ระบบสนับสนุนการตัดสินใจเป็นระบบที่ทำให้เกิด Data warehouse เพราะ Data warehouse เป็นที่เก็บข้อมูลสำหรับช่วยในการตัดสินใจ

## บทที่ 3

### แนวคิดเกี่ยวกับคลังข้อมูล (Data Warehouse)

#### 3.1 แนวคิดเกี่ยวกับคลังข้อมูล

เทคโนโลยีสารสนเทศ คือการประยุกต์ใช้เครื่องมือและอุปกรณ์ต่าง ๆ คือเครื่องคอมพิวเตอร์ เครื่องใช้สำนักงาน และอุปกรณ์โทรคมนาคม โดยที่คอมพิวเตอร์เป็นเครื่องมือที่ช่วยในการเก็บและประมวลข้อมูลให้เกิดสารสนเทศสำหรับผู้ใช้ ซึ่งสารสนเทศนั้นสามารถส่งและแลกเปลี่ยนโดยผ่านระบบเครือข่ายโทรคมนาคมต่าง ๆ

Data (ข้อมูล) หมายถึงความจริง(Fact) ซึ่งอาจอยู่ในรูปของตัวเลข ข้อความ ภาพ เสียง หรือรายละเอียดในรูปแบบอื่น ๆ เกี่ยวกับสิ่งต่าง ๆ ที่เราสนใจ

Information (สารสนเทศ) หมายถึง การนำข้อมูลมาประมวลโดยการผสมผสานความคิดเห็น วัฒนธรรม และสิ่งแวดล้อมลงไป

Warehousing หมายถึงการปฏิบัติทางกายภาพเกี่ยวกับการรับ การเก็บรักษา และการจ่ายพัสดุ

จากความสำคัญของข้อมูลในยุคปัจจุบัน ได้มีการนำเทคโนโลยีต่าง ๆ เข้ามาควบคุมข้อมูลเพื่อบริหารข้อมูลให้เกิดประโยชน์สูงสุด จึงทำให้เกิดเทคโนโลยีคลังข้อมูล (Data Warehouse) หรือคลังสารสนเทศ (Information Warehouse) ซึ่งเป็นเทคโนโลยีที่กำลังได้รับความสนใจจากองค์กรต่าง ๆ เป็นอย่างมาก ในที่นี้จะใช้ Data Warehouse ในความหมายเดียวกับ Information Warehouse

Data Warehousing ถือว่าเป็นเทคโนโลยีที่มีความสำคัญเป็นอย่างยิ่ง ในการวางแผนพื้นฐานกับองค์กร (Infrastructure) กล่าวคือทุกองค์กรจำเป็นต้องมีระบบฐานข้อมูลขององค์กร(Corporate Database Information) มีการจัดการข้อมูลที่ดี และนำข้อมูลเหล่านั้นเก็บไว้ในคลังข้อมูล(Data Warehouse) เพื่อให้เรียกใช้ข้อมูลได้อย่างมีประสิทธิภาพ

#### 3.1.1 ระบบฐานข้อมูลองค์กรกับคลังข้อมูล

(Corporate Database Information and Data Warehouse)

ระบบฐานข้อมูลขององค์กรเป็นส่วนที่มีความสำคัญเป็นอย่างยิ่งในการวางแผนพื้นฐานให้กับองค์กร ซึ่งเปรียบระบบฐานข้อมูลเหมือนกระดูกสันหลังของการดำเนินการขององค์กร องค์กรใดไม่มีการจัดข้อมูลที่ดีก็คงจะเหมือนคนตาบอด ซึ่งยากที่จะบรรลุถึงจุดหมาย เมื่อเทียบกับคนตาดีและยังมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล้องส่องทางไกลเห็นภาพได้ชัดเจนกว่าจึงสามารถตัดสินใจเดินทางไปสู่จุดมุ่งหมายอย่างราบรื่น

ฐานข้อมูลในองค์กรจะรวบรวมข้อมูลที่เกี่ยวข้องไว้ทั้งหมดในลักษณะที่สอดคล้องกับความต้องการ และโครงสร้างขององค์กร ซึ่งข้อมูลที่จะนำมาบันทึกในฐานข้อมูลนั้นประกอบด้วยข้อมูลภายใน องค์กร ข้อมูลภายนอก ข้อมูลทั้งในอดีตและปัจจุบัน จากข้อมูลดังกล่าวถึงแม้ว่าจะมีอยู่ในระบบก็ตาม แต่การเรียกใช้ข้อมูลมาตัดสินใจก็ใช้เวลาสูง และสิ้นเปลืองค่าใช้จ่าย ทั้งด้านบุคลากรและด้านการลงทุน ผู้บริหารไม่สามารถเรียกข้อมูลในรูปแบบที่ต้องการได้ด้วยตนเอง ต้องให้กลุ่มผู้เชี่ยวชาญหรือเจ้าหน้าที่คอมพิวเตอร์เป็นผู้ดำเนินการทุกครั้งไป และการดึงข้อมูลก็จำเป็นต้องเอาข้อมูลจากเครือข่ายคอมพิวเตอร์ในสถานที่ที่กำหนด เช่นเครื่องบางเครื่องสามารถเรียกดูข้อมูลได้เพียงเรื่องเกี่ยวกับเงินกู้เท่านั้น ไม่สามารถเรียกดูข้อมูลอื่น ๆ ได้เลย กล่าวคือไม่สามารถเห็นภาพของลูกค้ารายหนึ่งว่ามีเงินฝาก เงินกู้ หรือ สนใจบริการอื่นใดบ้างกับสถาบันการเงิน เป็นต้นและข้อมูลยังมีข้อจำกัดต้องเป็นข้อมูลชนิด Text Base เท่านั้น จากความสำคัญของฐานข้อมูล ซึ่งถือได้ว่าเป็นหัวใจหลักของธุรกิจ จึงได้มีการศึกษาถึงระบบการจัดการฐานข้อมูล และวิธีการที่จะแก้ปัญหาข้างต้น โดยวิธีการที่สามารถเข้าถึงข้อมูลได้ ไม่ว่าจะเป็นที่ใด (Any Where) ข้อมูลแบบใด (Any Information) และเมื่อไร (Any Time) การแก้ปัญหาดังกล่าวโดยผสมผสานระหว่างเทคโนโลยีข้อมูลแบบสัมพันธ์ (Relational Database Management System) และระบบฐานข้อมูลแบบออบเจกต์ (Object Oriented Database Management System) ทำให้สามารถเข้าถึงข้อมูลได้ทุกชนิด โดยอาศัยเทคโนโลยีแบบขนาน (Parallel Database Processing) ซึ่งเป็นระบบที่มีประสิทธิภาพสูงสุด ข้อมูลทั้งหมดนั้นจะถูกรวบรวมเก็บไว้ที่เดียวกันเรียกที่เก็บข้อมูลเหล่านั้นว่า คลังข้อมูล (Data Warehouse)

### 3.1.2 คลังข้อมูล (Data Warehouse)

Data Warehouse เป็นคำที่ได้รับความนิยมเป็นอย่างมาก บริษัทใหญ่ ๆ ต่างให้ความสนใจและทำการศึกษาเพื่อนำมาประยุกต์เข้ากับธุรกิจของตน ให้สามารถที่จะมีแต้มต่อทางธุรกิจ และผลตอบแทนการลงทุนด้านข้อมูลให้คุ้มค่าแก่บริษัทมากยิ่งขึ้น

Meta Group ซึ่งเป็นบริษัทด้านการวิจัยในประเทศสหรัฐอเมริกาได้ลงบทความวิจัยว่า 90 ถึง 95 เปอร์เซนต์ ของ Fortune 200 มีความสนใจและวางแผนการทำ Data Warehouse เหตุผลที่ Data Warehouse เป็นเรื่องที่น่าสนใจมากในเทคโนโลยีสารสนเทศในปัจจุบันและบริษัทต่าง ๆ สนใจเนื่องจากในโลกธุรกิจปัจจุบันมีการแข่งขันกันอย่างรุนแรง องค์กรธุรกิจจะต้องเข้าใจลูกค้า ของตนเองให้มากที่สุดและผู้ที่มีข้อมูลที่ถูกต้องการ รวดเร็วกว่า และแม่นยำกว่า ย่อมได้เปรียบในการเป็นผู้นำทางธุรกิจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Warehouse เป็นการนำเสนอแนวทางในการเข้าถึงข้อมูลในองค์กรได้อย่างมีประสิทธิภาพ ซึ่งความเป็นจริงแล้ว หลักการของ Data Warehouse มีมานานแล้วภายใต้ชื่อที่แตกต่างกันไป ซึ่ง Data Warehouse ช่วยสนับสนุนการทำงานแบบ Decentralized ,Non Uniform ของระบบคอมพิวเตอร์ ภายใต้คอมพิวเตอร์ที่มีรูปแบบที่แตกต่างกันไป ซึ่งเป็นการเพิ่มข้อดี หรือยกระดับให้กับคอมพิวเตอร์ที่เป็น Workstation โดยเฉพาะ Desktop Workstation การทำ Data Warehouse สามารถที่จะเข้าถึงข้อมูลได้ทั้งแบบท้องถิ่น (Local) หรือเป็น Remote สำหรับเครื่อง Mainfram หรือ Unix -BaseServer ในเครือข่ายได้

การจัดสภาพแวดล้อมของData Warehouse เป็นการใช้ผู้ใช้ หลาย ๆ คน และหลาย ๆ จุดประสงค์ สามารถเรียกใช้ข้อมูลใน Warehouse ได้ โดยข้อมูลมาจากแหล่งข้อมูลที่ต่าง ๆ กัน เป็นข้อมูลแบบ On-line Production System และต่างการประมวลผลกันนำมาเก็บใน Warehouse ให้อยู่ในรูปแบบที่ง่ายต่อการเรียกใช้ (Easy to use)

โดยสภาพแวดล้อมของData Warehouse สามารถสนับสนุนการทำงานของ Client โดยไม่จำกัดความแตกต่างของ Client Client Warehouse เหมาะสำหรับ 2 กลุ่มหลักคือ กลุ่ม University's central Administration office) และกลุ่ม อื่น ๆ ที่ทำงาน ใน Department ข้อมูลใน Data Warehouse สามารถสร้างเพื่อสนับสนุนแต่ละพื้นที่ได้ สมมติว่า 2 Client ที่แตกต่างกัน สามารถร้องขอข้อมูลเหมือนกัน เพื่อนำเสนอในพื้นที่ที่ต่างกันได้ และข้อมูลจากแหล่งที่ต่างกัน สามารถที่จะรวมกันและเก็บในพื้นที่เดียวกัน ได้ นี่เป็นจุดเด่นที่น่าสนใจของ Data Warehouse

Warehouse Model เป็นการปรับปรุงวิธีการจัดการ Information ในองค์กรซึ่งข้อมูลทุก ๆ สิ่ง ถูกเก็บไว้ที่เดียวกัน (Oneplace) อย่างเรียบร้อย เพื่อรอการประมวลผล เพราะฉะนั้นการทำ Data Warehouse จึงใช้การลงทุนสูงเพราะต้องมีการวางแผนสำหรับการเก็บข้อมูลเป็นอย่างดี เพราะข้อมูลมีแนวโน้มจะใหญ่ขึ้นเรื่อย ๆ การประมวลผลข้อมูลใน Warehouse ก็มากขึ้นเรื่อย ๆ การจัดเก็บข้อมูลใน Data Warehouse มีการจัดเก็บอย่างปลอดภัย การ Backup และ Recovery ทำโดยศูนย์กลางข้อมูล (Data Center) การรักษาความปลอดภัย(Security) ทำทั้งในระดับ Physical และ Logical

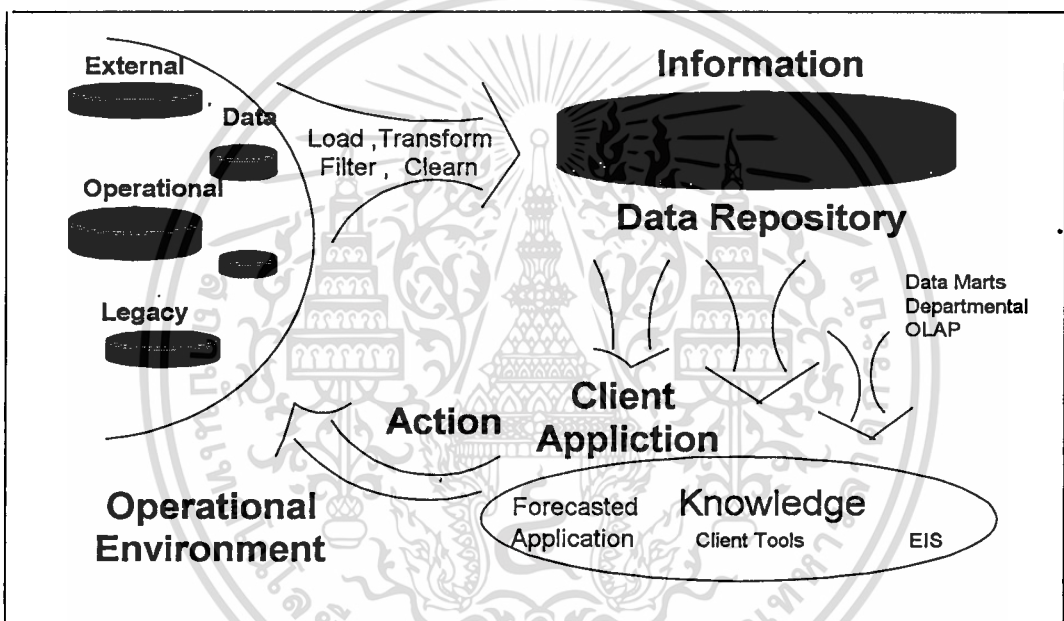
ปัจจุบันมี Software Tool ที่ช่วยจัดการข้อมูลใน Data Warehouse ที่นิยมใช้คือ DB2,Sybase,Informix,Oracle เป็นต้น

### 3.1.3 ความหมายของ Data Warehouse

Data Warehouse คือการรวบรวมข้อมูลจากฐานข้อมูลของส่วนปฏิบัติงานต่าง ๆ (Operational Database) หลาย ๆ แหล่งจากนั้นก็แปลงข้อมูล หรือสรุปข้อมูลเพื่อให้อยู่ในรูปแบบที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหมาะสมในการเก็บและง่ายต่อการนำกลับมาใช้ แล้วนำข้อมูลที่ผ่านการแปลงหรือสรุปแล้วนั้นเข้าไปเก็บใน Data Warehouse เพื่อใช้เป็นข้อมูลทางธุรกิจ สำหรับนักวางแผน นักวิเคราะห์ ผู้จัดการแผนก หรือผู้บริหารในการตัดสินใจด้านธุรกิจอย่างถูกต้องสำหรับองค์กรต่อไป

ดังนั้นการทำ Data Warehouse ก็จะเป็นการแปลงข้อมูลจากการปฏิบัติงานของแต่ละหน่วยงานไปเป็นข้อมูลทางธุรกิจสำหรับองค์กร โดยที่ผู้ดำเนินการออกแบบ Data Warehouse ต้องมีความเข้าใจในธุรกิจ ไม่ว่าจะเป็นลักษณะของลูกค้า พฤติกรรมผู้บริโภค ธุรกิจการแข่งขัน เพื่อให้การทำ Data Warehouse เป็นไปอย่างมีประสิทธิภาพสูงสุดต่อองค์กร



รูปที่ 3.1 การนำข้อมูลมาใช้ให้เกิดประโยชน์

### 3.1.4 จุดเด่นของคลังข้อมูล ( Data Warehouse )

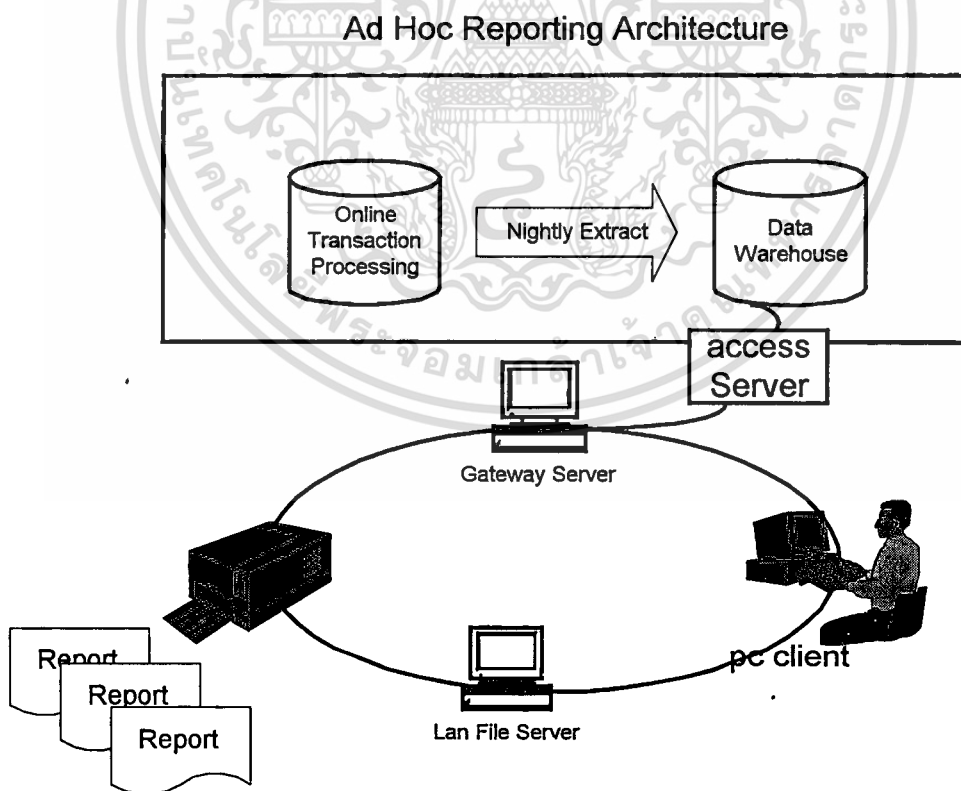
Data warehouse เป็นวิธีการใช้ข้อมูลสำหรับสนับสนุนการตัดสินใจ การทำรายงาน , และการวิเคราะห์ข้อมูล ในระบบธุรกิจความต้องการซอฟต์แวร์เพื่อช่วยวิเคราะห์และประมวลค่าข้อมูลเพิ่มขึ้นอย่างรวดเร็ว เพราะระบบธุรกิจมีการแข่งขันกันอย่างรุนแรงต่างองค์กรก็พยายามหาวิธีการที่จัดการกับข้อมูลเพื่อการตัดสินใจที่ดีที่สุด เพราะฉะนั้น Data warehouse จึงออกแบบฐานข้อมูลที่เน้นประสิทธิภาพในด้านการค้นหาข้อมูลสำหรับธุรกิจ คือข้อมูลสามารถอ่านได้อย่างเดียว ( Read - Only ) โดยจะ Load ข้อมูลจากฐานข้อมูลปฏิบัติงานเข้าสู่ Data warehouse ซึ่งในส่วนนี้จะแยกข้อมูล OLTP ออกจาก OLAP โดยเด็ดขาด หมายความว่าข้อมูล แบบ OLTP เป็นการประมวลผลเอกสารเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่เป็น Transaction มีการ Update แบบ immediately แบบทันทีทันใด มีการโต้ตอบ (interactive) ในขณะที่ OLAP คือ ข้อมูลใน Data warehouse ที่ใช้ในการวิเคราะห์ และทำรายงานสรุป โดยทั่วไป ข้อมูลจะมีการ Update เป็นคาบ เช่น ทุก ๆ คืน (nightly) , ทุก ๆ สัปดาห์ (weekly) และทุก ๆ เดือน (Monthly) เป็นต้น

OLTP ถูกออกแบบเพื่อประสิทธิภาพของการ Update ข้อมูลในรูปของ ( third normal form) 3n หรือสูงกว่าส่วน Data warehouse ออกแบบมาเพื่อ ประสิทธิภาพของการ Query ข้อมูลจะอยู่ในรูป denormalizing เพื่อหลีกเลี่ยงการ join และ นำเสนอข้อมูลที่มีความหมายสมบูรณ์ให้กับผู้ใช้

ดังนั้น data warehouse จึงเป็นโครงสร้างพื้นฐาน (infrastructure) สำหรับ Application การสนับสนุนการตัดสินใจที่ดีกว่า data warehouse ไม่ใช่แบบการตัดสินใจ แต่เป็นข้อมูลสำหรับ Tools เข้าไปเรียกใช้ โดยส่วนมากรูปแบบของ DSS ประกอบด้วยโปรแกรมสำหรับ Access ข้อมูล โดยเฉพาะ และข้อมูลที่ถูกออกแบบมา ( flat files ) เพื่อตอบคำถามเฉพาะกิจ ( ad hoc queries ) สำหรับการตัดสินใจ

ตัวอย่าง แนวคิดการใช้ Data warehouse ช่วยสนับสนุนการตัดสินใจ

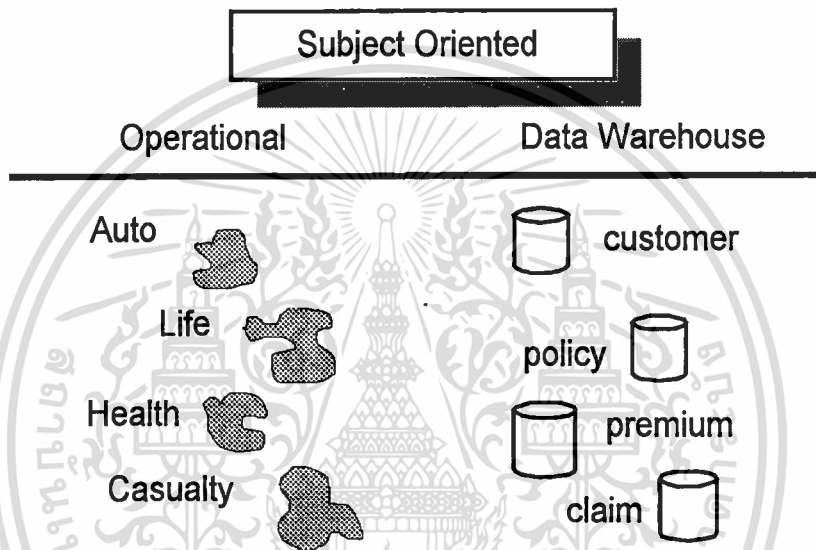


รูปที่ 3.2 แนวคิดการใช้ข้อมูลสำหรับสนับสนุนการตัดสินใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 ลักษณะของข้อมูลในคลังข้อมูล

3.2.1 Subject Oriented : ข้อมูลจะต้องถูกสร้างขึ้นจากหัวข้อ (Subject) เรื่องที่สนใจ เช่น ถ้าบริษัทประกันภัยต้องการใช้ Data Warehouse ฐานข้อมูลที่ได้จะต้องสร้างขึ้นจากประวัติลูกค้า(Customer) ,เบี้ยประกัน (Premium) และการเรียกร้อง (Claim) แทนที่จะแยกตามชนิดของผลิตภัณฑ์ สรุปก็คือข้อมูลที่สร้างขึ้นจะประกอบด้วยหัวข้อที่เก็บเฉพาะข่าวสารที่จำเป็นต่อกระบวนการตัดสินใจเท่านั้น ดังรูปที่ 3.3

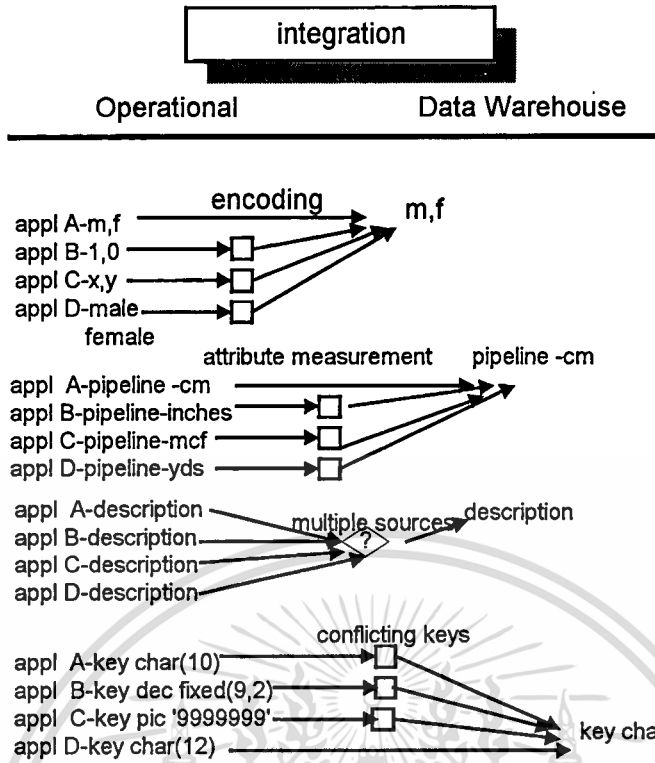


รูปที่ 3.3 การเลือกหัวข้อที่ต้องการ (Subject oriented)

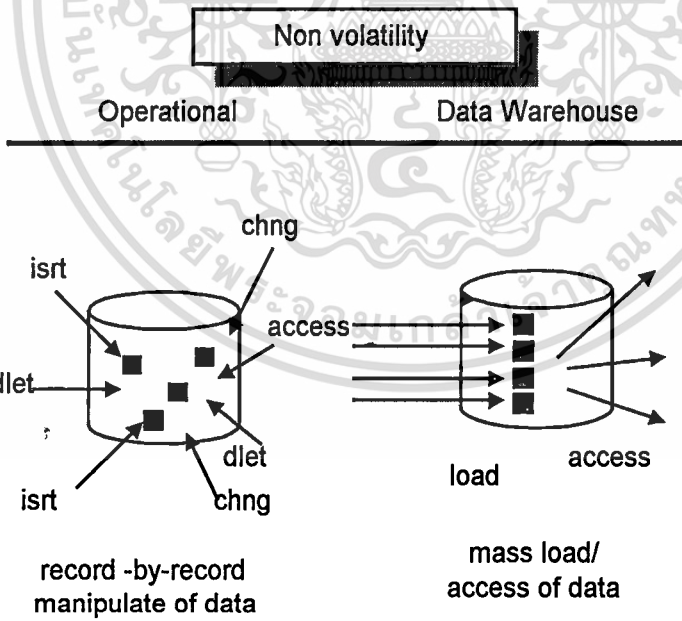
3.2.2 Integrated เป็นการรวบรวมข้อมูลจากแหล่งต่าง ๆ สร้างขึ้นเป็นฐานข้อมูลที่มีความสอดคล้องกัน ตัวอย่างเช่น ค่าของตัวแปรตัวเดียวกันในแต่ละฐานข้อมูลอาจจะต่างกัน (ฐานข้อมูลหนึ่งอาจใช้ "0" และ "1" แต่ตัวแปรเดียวกันนี้ในอีกฐานข้อมูลหนึ่งใช้ "T" กับ "F" เป็นต้น) ดังนั้นฐานข้อมูลที่สร้างขึ้นใหม่จะต้องได้รับการกำหนดค่าของตัวแปรให้เหมือนกันเป็นหนึ่งเดียวดังรูปที่ 3.4

3.2.3 Time Variant ข้อมูลที่ใช้ในการตัดสินใจ ที่เก็บจะต้องมีอายุประมาณ 5 ถึง 10 ปี เพื่อใช้เปรียบเทียบ ,หาแนวโน้มและทำนายผลลัพธ์ในอนาคตได้ ดังรูปที่ 3.5

3.2.4 Non-Volatile ข้อมูลจะไม่อัปเดตหรือถูกทำให้เปลี่ยนแปลงง่าย ๆ ผู้ใช้ฐานข้อมูลใน Data Warehouse ได้เพียงแค่อ่าน และเข้าถึงเท่านั้น ดังรูปที่ 3.6

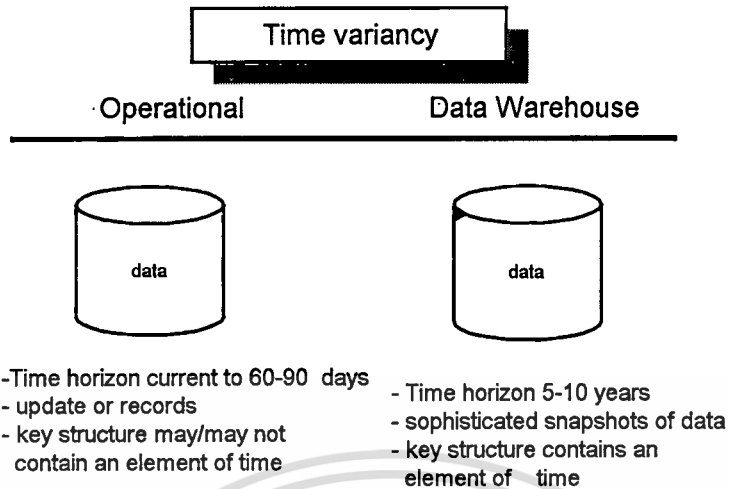


รูปที่ 3.4 การรวมข้อมูล (integration)



รูปที่ 3.5 การเรียกใช้ข้อมูลโดยไม่มีการเปลี่ยนแปลง (Non Volatility)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 ช่วงเวลาการเก็บข้อมูล (Time Variancy)

### 3.3 ประโยชน์ของคลังข้อมูล

ประโยชน์ของคลังข้อมูลมีมากมาย ซึ่งพอสรุปได้ดังนี้

- 3.2.1 ลดต้นทุนในการหาข้อมูลเพื่อการตัดสินใจ
- 3.2.2 ทำให้การดำเนินงานขององค์กรเป็นไปตามทิศทางจากข้อมูลที่ถูกต้องและแม่นยำ
- 3.2.3 เพิ่มคุณภาพให้ในการบริหารงานและการให้บริการแก่ลูกค้า
- 3.2.4 ทำให้การใช้ประโยชน์จากทรัพย์สินและบริหารงานอื่น ๆ ได้มีประสิทธิภาพ
- 3.2.5 ปรับองค์กรให้มีประสิทธิภาพสูงขึ้น

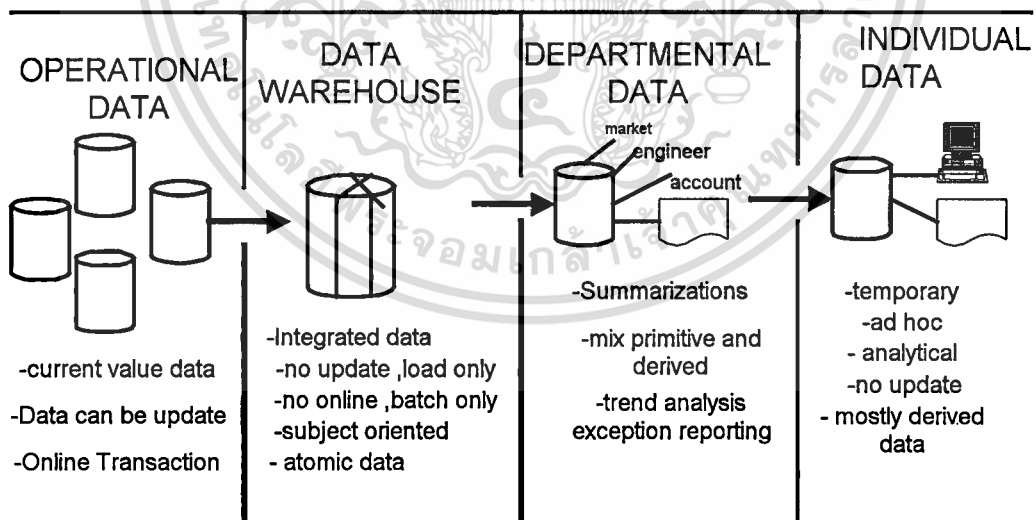
## บทที่ 4

### สถาปัตยกรรมของคลังข้อมูล ( Data Warehouse Architecture )

#### 4.1 สถาปัตยกรรมข้อมูล (Data Architecture)

การนำข้อมูลมาใช้ให้เกิดประโยชน์มีหลายรูปแบบ และหลายลักษณะที่แตกต่างกันออกไป โดยข้อมูลที่ใช้กันทั่ว ๆ สามารถที่จะแบ่งสถาปัตยกรรมของข้อมูลได้ 4 ระดับ ดังนี้

1. Operational Level เป็นส่วนของข้อมูลที่สนับสนุนการทำ Transaction Procssing
2. Data Warehouse เป็นการรวมข้อมูลจากระดับ Operational ตามหัวข้อที่ต้องการ และข้อมูลจะไม่มี Update
3. Departmental Level เป็นการแยกข้อมูลออกไปตามแผนกต่าง ๆ ซึ่งข้อมูลเฉพาะของแต่ละแผนกเพื่อใช้สำหรับการวิเคราะห์ และดูแนวโน้มของข้อมูล
4. Individual เป็นข้อมูลเฉพาะสำหรับการวิเคราะห์งานใดงานหนึ่งโดยเฉพาะส่วนมากจะใช้โดยนักวิเคราะห์ เหมาะสำหรับตอบคำถามเฉพาะกิจสำหรับผู้บริหาร (Ad hoc)



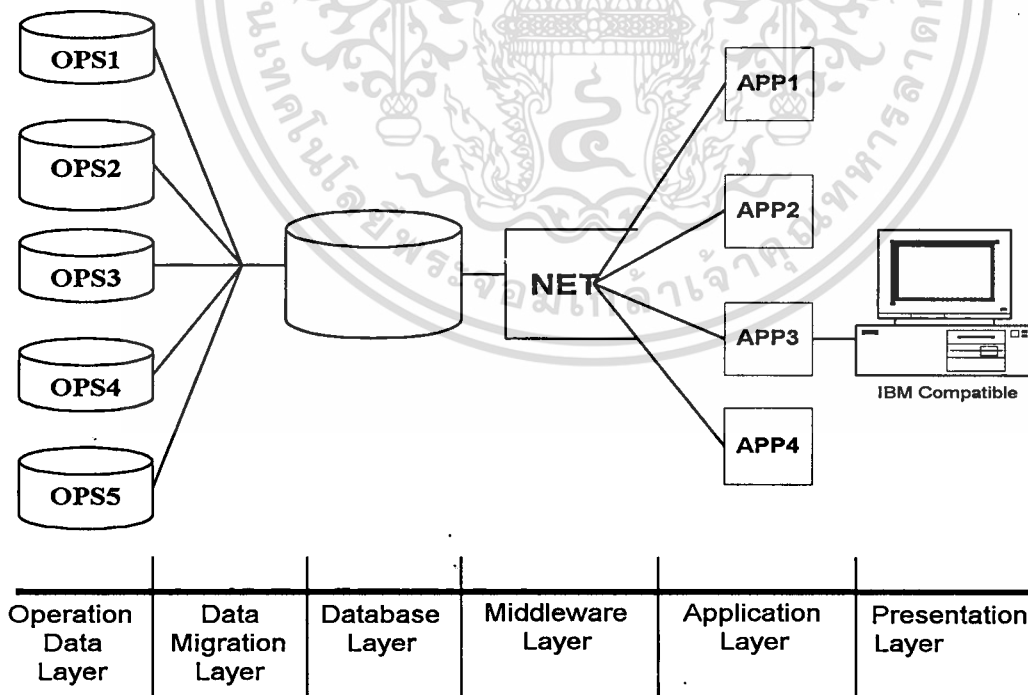
รูปที่ 4.1 แสดงสถาปัตยกรรมของการใช้ข้อมูล (Data Architecture)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 สถาปัตยกรรมของ Data Warehouse

การปฏิบัติการกับข้อมูลที่จะนำเข้าสู่ Data Warehouse มีขั้นตอนที่สลับซับซ้อน เพราะต้องมีการแปลงข้อมูล และตรวจสอบข้อมูล เพื่อให้เกิดประโยชน์ต่อการเรียกใช้มากที่สุดเพราะฉะนั้น จึงมีสถาปัตยกรรมของ data warehouse เป็นกระบวนการในการจัดการกับข้อมูลใน warehouse ตั้งแต่เริ่มต้นจนถึงการนำไปใช้งาน เพื่อให้การใช้งานข้อมูลใน Warehouse เกิดประสิทธิภาพสูงสุด โดยแบ่งสถาปัตยกรรมของ Data warehouse ออกเป็น 6 Layer ดังนี้

1. Operational Data Base/External Data Base Layer
2. Data Migration
3. Data Access Warehouse /database administration Layer
4. Middleware Layer
5. Decision support application Layer
6. Presentation interface Layer



สถาปัตยกรรมของ Data Warehouse

รูปที่ 4.2 สถาปัตยกรรมของ Data Warehouse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1. Operational Database/External Database

ชั้นของข้อมูลปฏิบัติงาน (Operational Data) นี้จะเป็นส่วนของข้อมูลที่เป็นแหล่งกำเนิดข้อมูล (Source) ไม่ว่าจะเป็นข้อมูลจากภายในองค์กร และภายนอกองค์กร จะนำเข้าไปเก็บในคลังข้อมูล (Data Warehouse) ในชั้นนี้โดยทั่ว ๆ ไปจะประกอบด้วยข้อมูลการปฏิบัติงานที่เป็นข้อมูลแบบ On-line Transaction โดยข้อมูลในส่วนนี้จะต้องมีการกำหนดความคงที่ของข้อมูล โดยมีการเช็คเงื่อนไขและกำหนดมาตรฐานของข้อมูลก่อนที่จะนำเข้าสู่ Data Warehouse เพื่อป้องกันการซ้ำซ้อนของข้อมูล จึงจะทำให้ข้อมูลในระบบการปฏิบัติงาน มีความถูกต้องและสมบูรณ์ และสามารถรักษาความคงสภาพของ(Integrity) ข้อมูลไว้ได้ เพราะข้อมูลมีความสำคัญมากจะต้องมีการควบคุม ความถูกต้อง เนื่องจากผู้ใช้ข้อมูลเหล่านี้ สามารถใช้งานและเปลี่ยนแปลงแก้ไขข้อมูลในฐานข้อมูลได้

### 2. Data Migration Layer

ในชั้นนี้เป็นชั้นที่มีการปฏิบัติการเกี่ยวกับข้อมูลเพื่อนำข้อมูลจากฐานข้อมูลปฏิบัติการเข้าสู่ฐานข้อมูลกลาง Data Warehouse กิจกรรมในส่วนนี้ ประกอบไปด้วย การดึงข้อมูล การปรับเปลี่ยนข้อมูล การย้ายข้อมูล จากแหล่งกำเนิดข้อมูล (Source) ไปยังเป้าหมายคือ Warehouse ในการเคลื่อนย้ายข้อมูล พร้อมกับการเปลี่ยนแปลงรูปแบบของข้อมูล ต้องแน่ใจได้ว่าข้อมูลที่จะนำเข้าไปเก็บใน Data Warehouse จะต้องมีความถูกต้อง Accurate , Integrated , Timely ,Valid เป็นต้น

ในการควบคุมข้อมูลในระบบให้มีคุณสมบัติดังกล่าวจะต้องใช้เทคนิคที่ซับซ้อนประกอบกับระบบสถาปัตยกรรมข้อมูล Tools และเทคนิคต่าง ๆ เข้ามาช่วยควบคุมข้อมูล เทคนิคต่าง ๆ ที่เกี่ยวข้องในการทำงาน Data Migration ประกอบด้วย Data Refresh ,Data Integrity ,Data Update ,Data Transformation, Transport Mechanism, Data Propagation ,Data Re-formatting ,Data Loading เป็นต้น

### 3. Data Access Warehouse /database administration Layer

ชั้นของการเข้าถึงข้อมูลใน Data Warehouse จะต้องมีการออกแบบและบริหารควบคุมกิจกรรมที่จะเกิดขึ้นใน Data Warehouse ซึ่งสิ่งนี้เป็นสิ่งที่สำคัญมาก ต้องอาศัย สถาปัตยกรรมของข้อมูลและมาตรฐานของข้อมูล เป็นตัวช่วยในการควบคุมหรือดูแลข้อมูลเหล่านี้ โดยข้อมูลที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะนำเข้าไปเก็บใน Data Warehouse จะต้องมีการรวมและแปลงข้อมูลก่อน เพื่อให้ง่ายต่อการเรียกใช้ ซึ่งมี 2 วิธีที่การจะควบคุมดูแลข้อมูลโดยใช้ หลักการของ Meta Data และ Granularity โดย Meta Data จะเก็บรวบรวมรายละเอียดของข้อมูลที่เก็บใน Data Warehouse รายละเอียดของข้อมูลประกอบด้วยส่วนต่าง ๆ ของข้อมูล เช่น โครงสร้างข้อมูล,ตาราง,ความหมายของข้อมูล,ตำแหน่งของข้อมูล ,สถานภาพต่าง ๆ ของข้อมูล เป็นต้น

ส่วน Data Granularity เป็นการควบคุมเกี่ยวกับการเรียกใช้ข้อมูล หรือการเปลี่ยนแปลงข้อมูล (Update) เมื่อถูกร้องขอข้อมูลจากส่วนต่าง ๆ ว่าจะมีวิธีการแบ่งการใช้ข้อมูลอย่างไร การ Lock ข้อมูล จะมีวิธีการ Lock อย่างไรบ้าง

#### 4. Middleware Layer

เป็นตัวกลางในการเข้าถึงข้อมูลใน Warehouse ระหว่าง Application ต่าง ๆ กับ ฐานข้อมูล โดยการเข้าถึงข้อมูลในส่วนนี้ จะสามารถเข้าถึงข้อมูลได้กว้าง (Wide) ไม่ว่าจะเป็นการเข้าถึงข้อมูลจาก เครื่องคอมพิวเตอร์ส่วนบุคคล PC ,Workstation , และ Terminal ซึ่งสามารถทำได้ การเข้าถึงข้อมูลมีการรวมเครื่องคอมพิวเตอร์เข้าด้วยกัน เป็นระบบ Local Area Network (LAN) และระบบ Lan หลาย ๆ วงสามารถเชื่อมต่อกันเป็นระบบ WAN (Wide Area Network)

การพัฒนาเทคโนโลยี LAN ให้มีประสิทธิภาพมากยิ่งขึ้น ในปัจจุบันได้มีหลักการของ Client / Server มาใช้ โดยจะมีเครื่องคอมพิวเตอร์ที่ทำหน้าที่ เป็น Client (ผู้ขอใช้บริการ) ร้องขอความต้องการไปยัง Server (ผู้ให้บริการ) จากนั้น Server จะส่งผลลัพธ์กลับมาถึง Client หลักการนี้จะมีการแยกข้อมูลออกจากโปรแกรม (Application) ซึ่งทำให้ Client สามารถร้องขอได้ไม่ว่าจะอยู่ที่ใด จากความสามารถของระบบ Client /Server จึงถูกเลือกมาใช้เพื่อช่วยในการเข้าถึงข้อมูลใน Data Warehouse เพราะข้อมูลใน Warehouse จะมีการเก็บข้อมูลไว้ที่เดียวกัน (Single Local And Physical) คือรวมเป็น Data Center และโปรแกรมที่เรียกใช้สามารถเก็บที่เครื่อง PC หรือ ที่ Lan Database Server บนโต๊ะผู้ใช้ หลาย ๆ คน ได้ ระบบนี้มีข้อดีคือมีความยืดหยุ่นในการเรียกใช้ข้อมูลจากระบบ คือ Client สามารถร้องขอข้อมูลจาก Database server ซึ่งเป็นระบบ Lan ในท้องถิ่น หรือจาก Remote Database (ที่อยู่ใน Data Warehouse) โดยผ่าน gateway เพราะฉะนั้นไม่ว่าจะอยู่ที่ใด ผู้ใช้ก็สามารถที่จะร้องขอข้อมูลที่เก็บไว้ในฐานข้อมูลต่าง ๆ ได้

## 5. Decision support application Layer

Application สำหรับช่วยสนับสนุนการตัดสินใจ จะต้องมีคุณสมบัติที่ดี คือสามารถที่จะขอข้อมูลต่าง ๆ ได้อย่างละเอียด ข้อมูลมีความถูกต้อง รวดเร็ว และง่ายต่อการเรียกใช้ คุณสมบัติของ Application โดยทั่ว ๆ ไปควรมีลักษณะดังนี้

- Presentation System ลักษณะการนำเสนอสามารถที่จะแสดงผลข้อมูล ข้อเสนอเทศ ที่ดูง่าย เข้าใจง่าย ไม่ว่าจะเป็นการนำเสนอแบบรายงาน หรือการทำสถิติวิเคราะห์ มีรูปแบบในการนำเสนอที่ครบถ้วน เช่น Textual ,Table ,Graph, และ Chart
- Interrogative System สามารถเข้าถึงข้อมูลได้โดยไม่กำหนดโครงสร้าง ในการค้นหา หรือการเข้าถึงข้อมูล การตอบรับต้องเป็นแบบโต้ตอบ (Ad hoc Queries)
- Simulation System สามารถที่จำลองสถานการณ์ต่าง ๆ ที่เกิดขึ้นได้ อาจเป็นการพยากรณ์ (Forecasts) และการวางแผนขององค์กร
- Function System มี Function เฉพาะสำหรับงานต่าง ๆ เพื่อสนับสนุนการจัดการ การประมวลผลงานในองค์กร
- Automatic System เป็นระบบที่สามารถทำได้อัตโนมัติ สามารถที่จะวินิจฉัยเหตุการณ์เบื้องต้นที่เกิดขึ้นกับระบบได้

## 6. Presentation interface Layer

เป็นการกำหนดการแสดงผลข้อมูลโดยมีมาตรฐานของ Software ที่จะเป็นตัวกำหนดการเข้าถึงข้อมูลโดยให้คอมพิวเตอร์ที่มีความแตกต่างกัน สามารถเข้าถึงข้อมูลในแหล่งเดียวกันได้

กรอบโต้ตอบ (Dialog Box) สำหรับผู้ใช้ที่จะใช้ติดต่อกับข้อมูล ใน Data Warehouse จะต้องมีความภาพ มีขอบเขตที่แน่นอน คือมีความแข็งแรงและมีรูปแบบที่คล่องตัวในการเรียกใช้รูปแบบของการโต้ตอบมีลักษณะดังนี้

- Command Line Interface  
คำสั่งโต้ตอบ แบบ Sql
- Menu Driven Interface  
ใช้เมนูในการควบคุม
- Query Language Interface  
ติดต่อโดยใช้ภาษาเรียกค้น Query
- Graphic Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ติดต่อโดยใช้รูปภาพเช่น Windows,Icon

- Multimedia Interface

Sound,Video,Animation,Text

- End User Profile

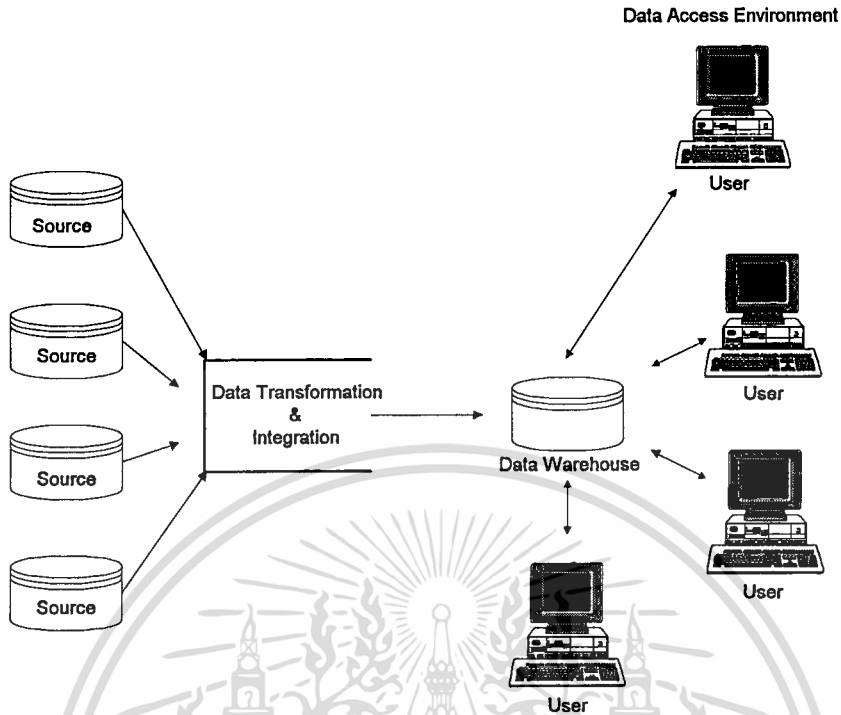
แสดงผลข้อมูลแบบซับซ้อนได้ ให้ผู้ใช้ที่มีความชำนาญน้อยก็สามารถเข้าสู่ข้อมูลได้

#### 4.3 การประยุกต์ใช้งานสถาปัตยกรรมในรูปแบบต่าง ๆ

ลักษณะพื้นฐานของ Data Warehouse จะใช้หลักการเดียวกันทั้งหมด แต่เมื่อใช้งานจริงจะมีการประยุกต์ใช้ในรูปแบบที่แตกต่างกันเพื่อให้เหมาะสมกับสภาพขององค์กรและลักษณะของระบบงาน โดยลักษณะของ สถาปัตยกรรม Data Warehouse ที่นิยมใช้มีดังนี้

4.3 1. สถาปัตยกรรมพื้นฐาน (Generic Data Warehouse Architecture ) แสดงได้ดังรูปที่ 4.3 ซึ่งแบ่งการทำงานออกเป็น 4 ส่วนดังนี้

1. Data ถูกดึงมาจาก Source System, Database หรือ File ซึ่ง Source fields ของ data warehouse อาจมาจากต่าง database ต่าง platforms ต่าง Type ต่าง format
2. Data จาก Source System จะถูก Integrate และ Transform ก่อนที่จะมาเข้า Data Warehouse
3. Data Warehouse เป็น read-only database ที่ถูกสร้างเพื่อใช้ใน decision support
4. Users เข้าถึงข้อมูลใน Data Warehouse โดย front end tool หรือ application



รูปที่ 4.3 สถาปัตยกรรมพื้นฐานของการใช้งานใน Data warehouse

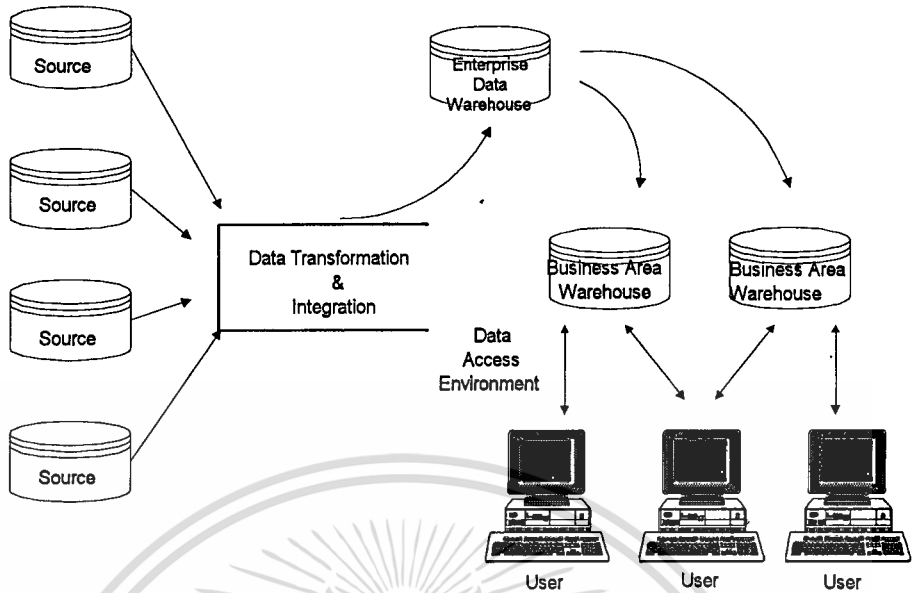
ในการประยุกต์ใช้งานจริงไม่มีวิธีที่แน่นอนในการ implement data warehouse อย่างไรก็ดี ตาม โครงสร้างพื้นฐานของ data warehouse จะต้องมียู่ ( must be in place )

#### 4.3.2 Enterprise Data warehouse Architecture

เป็นสถาปัตยกรรม ที่รวมข้อมูลทั้งหมดใน Center Data Warehouse จากนั้นก็มีการแบ่ง ข้อมูลใน Data Warehouse ให้เหมาะสมกับการใช้งานเช่น แบ่ง data Warehouse สำหรับงาน แผนกบัญชี หรือการตลาด เป็นต้น หรือที่มักจะได้ยินกันในความหมายของ Data Mart โดยมีจุด ประสงค์คือ

- เพื่อให้ทุก database ใช้ Enterprise Data Model เป็น base เดียวกัน
- เพื่อแยก process ของ source data transformation / integration ออกจาก process ของ database design และ denormalization ใช้ Enterprise Data Warehouse ในการ Mapping source ไปยัง target สำหรับ multiple Business Area warehouse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 สถาปัตยกรรมการใช้งาน Data warehouse แบบ Enterprise

ซึ่งมีขั้นตอนในการทำงานดังนี้

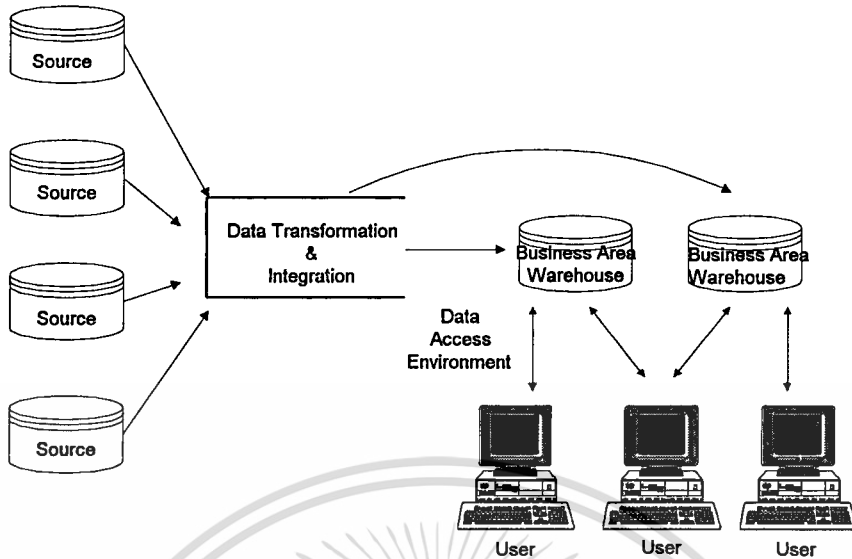
- Data ถูกดึงจาก Source System, Databases และ File
- Data จาก Source System จะถูก Transformation และ Integration ก่อนที่จะนำเข้าไปเก็บใน read-only Enterprise Data Warehouse
- Data เหล่านี้จะถูกจัดโครงสร้างใหม่ ออกแบบใหม่และเก็บเข้าไปใน Separate read only Business Area Warehouse ซึ่งใช้ใน Decision Support
- User เข้าถึงข้อมูลโดย front end tool หรือ application

สถาปัตยกรรมของ Data Warehouse แบบนี้ ได้ถูกปรับปรุง เพื่อให้อำนวยความสะดวกในการวางกลยุทธ์เริ่มต้น, สนับสนุนการตัดสินใจ และข้อจำกัดในด้านเทคนิคของบริษัท

#### 4.3.3 Subject area Warehouse Architecture

สถาปัตยกรรม แบบที่ 3 แสดงในรูปที่ 4.5 จะเห็นว่าไม่มีฐานข้อมูลขนาดใหญ่ในการเก็บข้อมูล แต่ก็ยังมีคุณสมบัติของ Data Warehouse อยู่ เนื่องจากมีข้อจำกัดในเรื่องของขนาดของ database จึงทำให้ต้องมีการออกแบบสถาปัตยกรรมแบบนี้ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 สถาปัตยกรรมการใช้งานแบบแยก Data warehouse

#### 4.3.4 Integrated Database Architecture

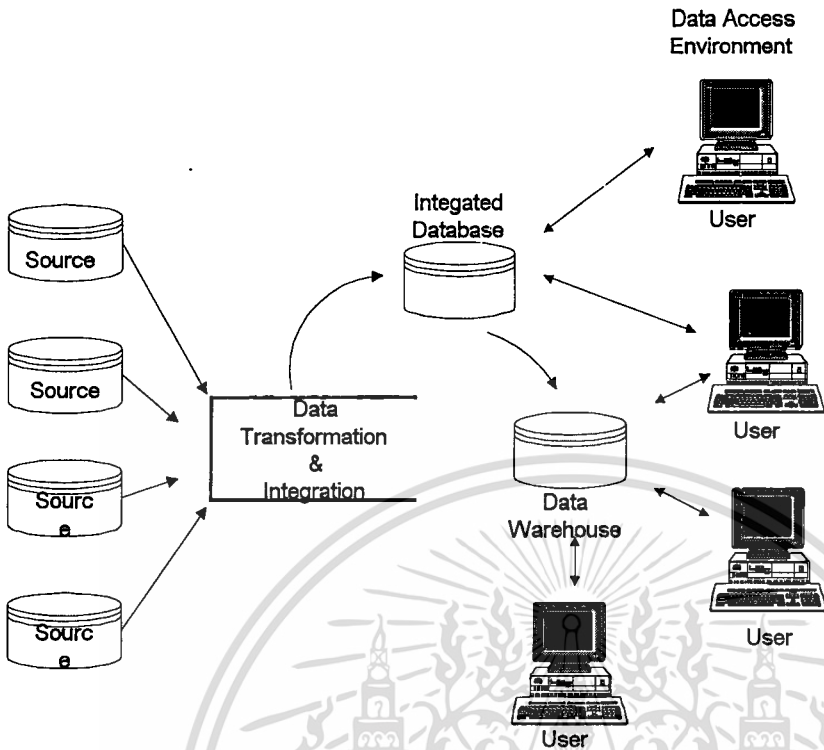
Architecture แบบนี้ มีวัตถุประสงค์ ในการตัดสินใจทุกวัน ( daily tactical decision making ) ในรูปที่ 4.6 มีขั้นตอนดังนี้

- Data ถูกดึงจาก Source System
- Data ถูก Transform และ Integration
- จากนั้น Data จะถูกนำไปไว้ที่ Integrated relational database ซึ่งจะถูกออกแบบเป็น third normal form สำหรับ information ที่ถูกนำเข้าไปใน Integrated database จะเป็น read-only

-New Data จะถูกใส่เพิ่มเข้าไปใน Database ถ้ามีฟีดใหม่ ๆ ที่จำเป็นในการตัดสินใจ ซึ่งจะถูกใส่โดย User

เหตุผลที่องค์กรใช้ architecture แบบนี้เนื่องจาก องค์กรมีความจำเป็นที่จะต้องรวบรวม data เพื่อการตัดสินใจในธุรกิจทุก ๆ วัน เพราะมันจะต้องใช้เวลามาก และมี error มากในการที่จะนำ information มาจาก application หลายร้อยตัว สิ่งที่ต้องการอันดับแรกขององค์กร คือ การมี Integrated Information ที่มีค่าในการดำเนินธุรกิจ การ cleaning และ adding ฟีดไปยัง database เป็นส่วนหนึ่งของ process นี้ ปัจจุบันบริษัทนี้กำลังใช้ Data Warehouse สำหรับ long-term strategic business decisions

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงการรวมกันของข้อมูลใน Data warehouse

จากรูปที่ 4.6 เนื่องจาก Integrated database อยู่ในรูป third normal form ดังนั้น data ที่ใส่เข้าไปใน Data Warehouse ต้องการการ redesign เพื่อที่จะได้นำไปสนับสนุนการตัดสินใจอย่างมีประสิทธิภาพ เนื่องจาก data เหล่านี้มีทั้ง read-only และ ฟิวเจอร์ใหม่ที่ถูกเพิ่มเข้าไปโดย User ดังนั้น data cleansing programs จำเป็นที่จะต้องเป็นส่วนหนึ่งของ data transformation process ในการจัดการ error ของ data ที่ user ใส่เพิ่มเข้าไป ก่อนที่จะนำไปใส่ใน Data Warehouse

## บทที่ 5

### แนวทางในการสร้างคลังข้อมูล

#### 5.1 องค์ประกอบที่สำคัญในการสร้าง Data Warehouse

ข้อมูลขององค์กรเหมือนกับสายแร่ที่ฝังอยู่ในภูเขา ซึ่งเป็นการยากที่ผู้บริหารจะนำข้อมูลมาเพื่อตัดสินใจ มีการวิเคราะห์กันว่า ในองค์กรที่ไม่มี Data warehouse เราอาจจะใช้ประโยชน์ของข้อมูลได้เพียง 10 % เท่านั้น จากการสร้างโปรแกรมที่จะดึงข้อมูลขึ้นมา และใช้เวลานานกว่าและซับซ้อนกว่าอีกด้วย ในองค์กรที่มี Data Warehouse ก็จะสามารถรวบรวมข้อมูลในอดีตเข้าเป็นฐานข้อมูลเพียงฐานเดียว แล้วใช้โปรแกรมที่งานในการดึงข้อมูลมาแสดงตามต้องการ ซึ่งการสร้าง Data warehouse จะมี 6 ส่วนหลัก ๆ ที่จะต้องพิจารณาดังนี้

##### 1. Data Modeling

เป็นการวิเคราะห์ข้อมูลจากฐานข้อมูลปฏิบัติต่าง ๆ (Operational Database) เพื่อการวางแผนเอาข้อมูลเข้ามาเก็บใน Data Warehouse และวางแผนในการสร้างโครงสร้างของข้อมูลใน Data warehouse การวิเคราะห์ข้อมูลนี้จะแบ่งเป็นงานย่อย ๆ คือ การสร้างมาตรฐานของการแปลงข้อมูล (Data Encoding) การปรับเปลี่ยนข้อมูล (Transforming ) การสรุปข้อมูล (Consolidating) ก่อนที่จะเข้าไปเก็บใน Data warehouse และในส่วนของโครงสร้าง จะต้องพิจารณาว่า ควรจะมี Data warehouse ส่วนกลางเพียงจุดเดียว หรือจะมี Data Warehouse ย่อย ๆ หลายจุด เพื่อผู้ใช้แต่ละประเภทงานหรือที่เรียกว่า Datamart ซึ่งก็มีข้อดี-ข้อเสีย คือถ้าเรามีข้อมูลแบ่งย่อยตามผู้ใช้หรือฟังก์ชันของงาน ก็จะสะดวกในการใช้ข้อมูล และลงทุนเริ่มต้นต่ำกว่า แต่ก็จะทำให้ไม่มีข้อมูลในลักษณะองค์กรโดยรวม เพื่อการวิเคราะห์ทางธุรกิจ (Enter-prise-wide Business Analysis) หรืออาจทำให้ลงทุนหลายครั้งในการทำ Datamart เพื่อแต่ละแผนก แทนที่จะลงทุนครั้งเดียวในส่วนกลาง ซึ่งผู้ออกแบบ Data Warehouse จำเป็นต้องพิจารณาข้อดี - ข้อเสียนี้ในการออกแบบ Data Warehouse

##### 2. การบริหาร Warehouse

จะต้องเลือก Tools ที่จะใช้ในการดึงข้อมูลและแปลงข้อมูล สร้างรหัสและบริหารข้อมูลใน Data Warehouse ซึ่ง Tools ดังกล่าวนี มีผู้ผลิตหลายรายเช่น Prism Solutions และ Carleton Corporation ได้ทำการพัฒนาขึ้นมา

3. ระบบฐานข้อมูลแบบขนานที่ออกแบบสถาปัตยกรรมโดยออกแบบจากแกนของฐานข้อมูล ระบบฐานข้อมูลแบบสัมพันธ์จะเป็นตัวกลางสำคัญสำหรับการทำ Data Warehouse เนื่องจากความสามารถในการรวบรวมและจัดการกับข้อมูลที่มีขนาดใหญ่หลายร้อย Gigabytes หรือหลาย Terabytes โดยที่เรายังคงสามารถใช้ข้อมูลได้โดยง่ายและรวดเร็ว

#### 4. ระบบเปิดที่สามารถขยายตัวได้อย่างมีประสิทธิภาพ

ระบบเปิดเป็นระบบที่ใช้กันมากในการทำ Data Warehouse ด้วยเหตุผลทั้งในแง่ของการลงทุน การเลือกใช้ Tools ที่มีอย่างแพร่หลาย คล่องตัวในการดำเนินการ และมีทางเลือกที่มากกว่า

#### 5. การเรียกใช้ข้อมูลจาก Data Warehouse

ผู้พัฒนาจะต้องสร้างโปรแกรมเพื่อเรียกข้อมูลสำหรับการตัดสินใจ (Decision Support) เพื่อให้ง่าย คล่องตัว และปรับเปลี่ยนความต้องการในการสอบถามได้ง่ายที่เรียกว่า Data Access Tools นอกจากนั้นยังมีการใช้ OLAP (ONLINE ANALYTICAL PROCESSING) ในการทำการวิเคราะห์ข้อมูลจาก Data Warehouse แบบ Multidimensional analysis ซึ่งทำให้สามารถที่จะเลือกวิเคราะห์ และนำเสนอข้อมูลในอดีตในหลายมุมมองในรูปแบบทางธุรกิจตามที่ต้องการได้เป็นอย่างดี

ซึ่งในปัจจุบันนี้มีโปรแกรมที่สร้างขึ้นเพื่อเป็นโปรแกรมพัฒนาระบบงานสำหรับองค์กร โดยสามารถทำรูปแบบการติดต่อแบบกราฟิก(GUI) เพื่อติดต่อกับผู้ใช้ในการเรียกดูข้อมูลจาก Data Warehouse

#### 6. Consulting Services

การให้คำปรึกษาในการดำเนินการก็เป็นส่วนหนึ่งที่สำคัญเช่นกัน อีกทั้งผู้ใช้จะต้องมีการวางแผน การฝึกอบรมและการศึกษาระบบเพื่อการบริหาร Data Warehouse และปรับปรุงต่อไป

นอกจากที่กล่าวมาทั้ง 6 ข้อนี้ผู้ออกแบบ Data Warehouse ควรมีความเข้าใจถึงธุรกิจนั้น และลูกค้าเป็นอย่างดี เพื่อที่จะสามารถออกแบบและประยุกต์ให้ Data Warehouse เป็นฐานข้อมูลทางธุรกิจที่มีประสิทธิภาพ และให้ภาพหรือมุมมองแก่ผู้บริหารองค์กร หรือผู้จัดการแผนกในการนำข้อมูลไปวิเคราะห์วางแผนธุรกิจให้ประสบความสำเร็จต่อไป

ตัวอย่างของบริษัทขนาดใหญ่ในประเทศสหรัฐอเมริกาที่ใช้ระบบ Data Warehouse

บริษัท MCI Communications Corporation (MCI) เป็นบริษัทยักษ์ใหญ่ทางด้านโทรคมนาคมของประเทศสหรัฐอเมริกา ได้เลือกระบบฐานข้อมูลของ Informix แบบ Online Extended Parallel Service เพื่อใช้เป็น Data Warehouse สำหรับแผนก Consumer Marketing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Group ซึ่งรับผิดชอบการให้บริการลูกค้าในส่วนของระบบ Telemarketing Direct Mail และการขาย ขณะนี้ในขั้นต้น บริษัท MCI ได้เก็บข้อมูลของลูกค้าทั่วประเทศสหรัฐอเมริกาประมาณ 600 Gigabytes โดยออกแบบใช้งานเป็นระบบข้อมูลเพื่อการตัดสินใจ (Decision Support) ซึ่งออกแบบให้สามารถนำข้อมูลมาใช้ในการตัดสินใจได้ง่ายและมีประสิทธิภาพ โดยเมื่อใช้เต็มโครงการแล้วระบบจะมีข้อมูลประมาณ 3 Terabytes (3,000 Gigabytes) และมีผู้ใช้ในระบบประมาณ 10,000 คน และกระจาย ในสาขาทั่วประเทศประมาณ 20 สาขา

บริษัท GTE Telephone Operations (GTE) ก็เป็นยักษ์ใหญ่ในวงการโทรคมนาคมเช่นกัน บริษัท GTE ได้ใช้โปรแกรมระบบงานทั้งงานการขาย การตลาด การสนับสนุนการค้า และการบริหารงานอื่น ๆ โดยมีการทำ Data Warehouse ซึ่ง GTE คาดว่าจะใช้งานทั้งระบบ โดยมีผู้ใช้งานในระบบ 50,000 คน และใช้ข้อมูล 18 Terabytes (=18,000 Gigabytes) บนเครื่อง AT&T, HP และ IBM

## 5.2 การจัดการกับข้อมูลเพื่อนำเข้าสู่ Data Warehouse มีลักษณะดังนี้

### 5.2.1 Data Modeling

การกำหนดรูปแบบของข้อมูลใน warehouse เป็นงานที่สำคัญมาก เพราะว่าข้อมูลในคลังข้อมูล จะเป็นฐานข้อมูลที่อ่านได้อย่างเดียว โดยอ่านรูปแบบข้อมูลจาก Data Description Language (DDL) ของต้นแบบข้อมูลสู่รูปแบบข้อมูลปลายทาง ซึ่งประโยชน์ในการทำ Data Model คือทำให้ข้อมูลมีความถูกต้อง เพราะฉะนั้นในการสร้าง Data Warehouse จึงต้องมี tools สำหรับช่วยในการกำหนด Data Model เพื่อการรวบรวมข้อมูล จากหลาย ๆ แหล่ง (Sources) สู่ฐานข้อมูลปลายทาง (Target) ที่สมบูรณ์ ซึ่งจะกล่าวถึงการกำหนด Data Modeling โดยละเอียดในหัวข้อที่ 5.3

### 5.2.2 Extracting, Transforming and Loading

ข้อมูลใน Data Warehouse เป็นข้อมูลขนาดใหญ่ที่รวบรวมมาจากหลาย ๆ ที่ โดยเลือกเฉพาะหัวข้อที่ต้องการ จุดประสงค์หลักของการรวมเพื่อประสิทธิภาพของการค้นหา (Query) เพราะฉะนั้นในการรวมหรือการแปลงข้อมูลเข้าสู่ Data Warehouse จึงมีความจำเป็นที่จะต้องมียุทธศาสตร์ที่ดี นั่นหมายถึง Tools ที่มีสมรรถนะที่จะจัดการกับข้อมูลที่มีปริมาณมากได้

Extracting Data เป็นการแยกข้อมูลบางส่วนจากฐานข้อมูลขนาดใหญ่ เพื่อประโยชน์ในการวิเคราะห์ โดยอาศัยเครื่องมือพิเศษที่ออกแบบมาเฉพาะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการทำ Transforming ข้อมูล คือการแปลงข้อมูลที่มีหลายรูปแบบ ให้เป็นรูปแบบมาตรฐานก่อนนำเข้าสู่ Warehouse ซึ่งต้องอาศัย Tools ที่ดีเช่นกัน ซึ่งในหัวข้อนี้พอสรุปได้ว่าในการนำข้อมูลเข้าสู่ Warehouse จะต้องมีการ Extract Tools เพื่อช่วยในการแยกข้อมูล แปลงข้อมูล และ อ่านข้อมูล (Load) โดยเลือก Tools ที่เหมาะสมกับการใช้งานกับข้อมูลขององค์กร ลักษณะของ Tools สำหรับการ Extraction และ Transforming สามารถที่จะอ่านข้อมูลจาก Source File , คำนวณ, สรุปยอดรวมและสามารถเปลี่ยนแปลง ข้อมูลให้เป็นมาตรฐานได้

### 5.2.3 Data Validation

การตรวจสอบความถูกต้องของข้อมูลใน Data Warehouse เป็นสิ่งที่สำคัญคือมีการตรวจสอบความถูกต้องโดยทำการเปรียบเทียบข้อมูลที่เหมือนกันในหลาย ๆ ที่ และเลือก 1 รายการเพื่อ Load เข้าสู่ Data Warehouse หรือแสดงรายการข้อมูลที่ซ้ำกันของ Report เช่นที่อยู่ในระบบเงินเดือน กับระบบบุคลากร ของคน ๆ เดียวกัน แตกต่างกันในกรณีนี้การเลือกข้อมูลที่สมบูรณ์เข้าสู่ Data Warehouse เป็นหน้าที่ของผู้ดูแลข้อมูลหรือเจ้าของข้อมูลว่าจะเลือกรายการใด หรือกรณีที่ข้อมูลมีขอบเขตที่แน่นอนสามารถที่จะเลือกข้อมูลเข้าสู่ Warehouse โดยอัตโนมัติได้ ในการตรวจสอบจะมีการเขียนโปรแกรมสำหรับอ่านค่าและเปรียบเทียบความถูกต้องของข้อมูล

### 5.2.4 Storing

ข้อมูลพื้นฐานใน Data Warehouse เป็นข้อมูลที่ถูกจัดเก็บอยู่ในรูปความสัมพันธ์ ที่จัดการโดย Relational DBMS แต่การเรียกใช้ข้อมูลเพื่อเกิดประสิทธิภาพโดยอาศัยระบบฐานข้อมูลแบบหลายมิติ (Multidimensional Database) ซึ่งเป็น Model ที่ออกแบบมาเพื่อการเรียกใช้ข้อมูลใน data warehouse โดยเฉพาะ โดยเพิ่มลักษณะพิเศษ จาก DBMS คืออ่านข้อมูลได้อย่างเดียว สามารถเข้าถึงข้อมูลแบบขนานได้ (Parallel) มีการทำ Indexing เพื่อการเข้าถึงข้อมูลที่มีประสิทธิภาพ

ประสิทธิภาพของการเข้าถึงข้อมูลใน Data Warehouse ขึ้นอยู่กับการออกแบบข้อมูลใน Warehouse ด้วย ซึ่งข้อมูลใน Data Warehouse จะเป็นข้อมูลที่อยู่ในรูป Denormalize โดยแปลงมาจากข้อมูลที่ทำกร Normalized แล้ว เพื่อประสิทธิภาพที่ดีกว่า การออกแบบโครงสร้างข้อมูลโดยวิธี Star Schema ซึ่งเป็นวิธีที่ได้รับความนิยมมาก เหมาะกับข้อมูลขนาดใหญ่ โดยเก็บใน Table เดียวกัน ข้อมูลซ้ำซ้อน (Redundantly) เพื่อลดการ Join บ่อย ๆ ทำให้เกิดผลดีคือ

- สร้างหรือออกแบบ Database เพื่อให้การตอบสนองเร็วขึ้น (Fast Response Times)

- ง่ายต่อการเพิ่มข้อมูล (Add)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- User เข้าถึงข้อมูลได้ง่าย
- ง่ายต่อการเข้าใจของ User และผู้พัฒนาระบบในแง่ของ Meta data

### 5.2.5 Query Tools

การเลือก tools ที่มีประสิทธิภาพในการค้นหาข้อมูล ควรเป็น Tools ที่มีคุณสมบัติที่ดีคือ มีความสามารถในการเรียกค้นสูง การ Interface ง่าย มีความสามารถด้านการคำนวณ การทำรายงาน (Report) ที่สมบูรณ์ ด้าน Graphics การเชื่อมต่อกับ tools อื่น ๆ (Export/Interoperate) และเป็นระบบเปิดเข้าถึงได้โดยผ่าน API (Application Program Interface) และอื่น ๆ เป็นต้น

โดยส่วนมาก Query Tools ไม่สามารถตอบสนองความต้องการของ user ได้ทั้งหมด แต่ผู้ใช้ก็ควรเลือก Tools ที่สามารถนำมาใช้งาน และตอบคำถาม (Ad hoc) ได้มากที่สุด ซึ่ง Tools ในปัจจุบันมีมากมายเป็นหลายร้อย บริษัทที่ผลิตออกมาจำหน่าย ขึ้นอยู่กับวิจารณ์ของผู้ใช้ที่เลือก Tools เหล่านั้น

### 5.2.6 Repository/Metadata

ลักษณะที่สำคัญของ Meta Data สำหรับการสร้าง Warehouse คือ

- เก็บ Directory of Content ที่ประกอบไปด้วยตำแหน่ง และรายละเอียดเกี่ยวกับ Data Warehouse Server,ฐานข้อมูลต่าง ๆ ตาราง ชื่อ และข้อสรุปต่าง ๆ
- กฎเกณฑ์อัตโนมัติสำหรับเจาะลึกเข้าไปในโครงสร้างย่อย ๆ ทางธุรกิจ เช่น การตลาด ,และชาร์ตต่างๆ ทางบัญชี
- ระบบรักษาความปลอดภัยสำหรับการใช้งานในระดับบุคคล,เวิร์กกรุป และองค์กรขนาดใหญ่ ซึ่งจะครอบคลุมการใช้งานในลักษณะต่าง ๆ เช่น การวิว,เปลี่ยนแปลง ,การคำนวณ , และการวิเคราะห์ในลักษณะอื่น ๆ ของ User
- ที่มาของข้อมูลที่คัดเลือกแล้ว (original source the winner)
- มีแนวทางในการสรุปข้อมูล (Algorithm for summarizing)

Meta Data เป็นเทคโนโลยีที่สำคัญที่ช่วยให้ data warehouse ประสบความสำเร็จเพราะเป็นส่วนที่เก็บรายละเอียดที่สำคัญของข้อมูลไว้ เก็บข้อมูลเกี่ยวกับData Dictionary,Location ,Structure ของข้อมูล เพื่อการติดต่อหรือเชื่อมต่อกับส่วนต่าง ๆ ของข้อมูลอื่น ๆ ที่จะนำเข้าไปเก็บใน Data Warehouse

### 5.2.7 Automating The Loading Process

การกำหนดช่วงเวลาในการ Load ข้อมูลจากต้นกำเนิดข้อมูล (Source) จากข้อมูล OLTP เข้าสู่ Data Warehouse โดยไม่กระทบกับการประมวลผลของ OLTP โดยอาจมีการกำหนดช่วงเวลาในการ Load เป็นอัตโนมัติเช่น ช่วงเวลากลางคืน, กลางวัน เป็นต้น

## 5.3 การออกแบบฐานข้อมูลสำหรับ Data Warehouse

การออกแบบฐานข้อมูลสำหรับ Data Warehouse จะมีความแตกต่างจากการออกแบบฐานข้อมูลสำหรับการประมวลผลแบบ On-Line ซึ่งอยู่ในรูปของ Relational Database Schema โดยจะมีการทำ Normalization เพื่อสร้างตาราง (Table) หลาย ๆ ตาราง เพื่อการ Join และเพื่อควบคุมความถูกต้องของข้อมูล (Integrity) ที่มีความซับซ้อนให้ง่ายต่อการ Manipulate ส่วนฐานข้อมูลที่ออกแบบเพื่อสนับสนุนการตัดสินใจจะมีลักษณะที่ตรงข้ามกันอย่างชัดเจน โดยมีโครงสร้างข้อมูลที่สามารถอ่านเข้าใจได้โดย user ข้อมูลอยู่ในรูป Denormalization ข้อมูลจะไม่เปลี่ยนแปลง (Static) สามารถเข้าถึง ค้นหาข้อมูลโดยใช้ SQL Query ทำให้หาข้อมูลได้ง่าย เพราะลดจำนวนครั้งของการ join ระหว่าง Table การทำ Backup ,Snapshot แน่ใจได้ว่าข้อมูลไม่สูญหาย

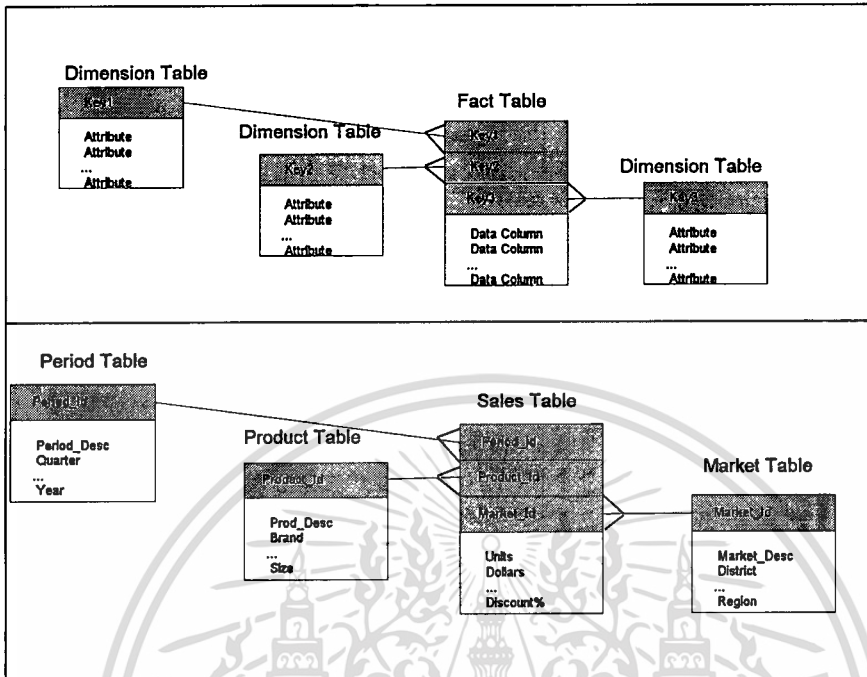
การออกแบบ Database สำหรับ Data Warehouse โดยใช้ Star Schema โดยอยู่ในขั้นตอนของการกำหนด Data Modeling ซึ่ง Star Schema เป็นการออกแบบโครงสร้างแบบง่าย ๆ มี 2-3 Table เพื่อการ join ข้อมูลที่รวดเร็ว ซึ่งเป็นการออกแบบที่ตรงกันข้ามกับการ Normalized Structure สำหรับฐานข้อมูลที่ประมวลผลแบบ On-Line Transaction เพื่อให้ง่ายต่อการใช้และมี Response time ดีขึ้น เพราะเป็นฐานข้อมูลที่อยู่ในรูป Denormalized เพราะฉะนั้นการออกแบบโดยใช้ Star Schema ได้กลายมาเป็นมาตรฐานสำหรับการออกแบบฐานข้อมูลสำหรับ Data Warehouse

### Star Schema

Star Schema ประกอบด้วยตาราง 2 ประเภท ดังรูปที่ 5.1 คือ Fact Table และ Dimension Tables Fact Tables หรือบางที่เรียกว่า Major Tables ซึ่งเป็นปริมาณข้อมูลจริง ๆ หรือข้อมูลที่ต้องการใช้จริง ประกอบไปด้วย หลาย ๆ คอลัมน์ และหลาย ๆ Rows ส่วน Dimension Tables หรือบางครั้งเรียกว่า Minor Tables เป็น Table สำหรับมองมิติของข้อมูลใน Fact table โดยใช้ SQL Query ช่วยในการ Join ระหว่าง Fact Table กับ Dimension Table ซึ่ง Star Schema จะเป็นขั้นตอนหนึ่งในการกำหนดแบบข้อมูล หรือที่เรียกว่า Data Modeling

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น เมื่อนำไปใช้โดยไม่ได้รับอนุญาตให้ถือว่าผิดกฎหมาย

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 Star schema

จากรูป แต่ละ table ต้องมี primary key ซึ่งเป็นตัวระบุลักษณะของแต่ละ row ใน star schema primary key ใน Fact table สามารถเรียกใช้ Foreign key ได้มากกว่า 1 key ในรูปแสดง ถึง 1 Fact Table และ 3 dimension table โดย Fact Table มี primary key ที่ประกอบไปด้วย Foreign Key 3 Key คือ Key1,Key2,Key3

จากตัวอย่าง Sales Table คือ Fact Table ส่วน Period\_Id,Product\_Id และ Market Table คือ foreign key

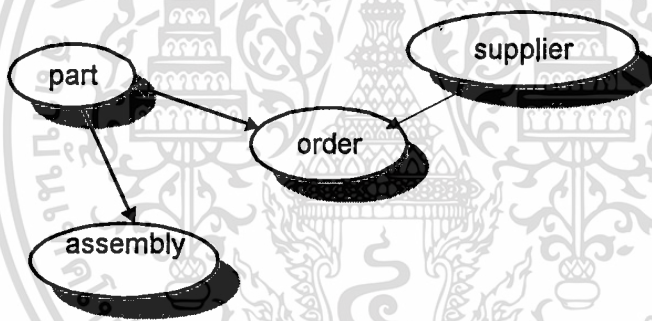
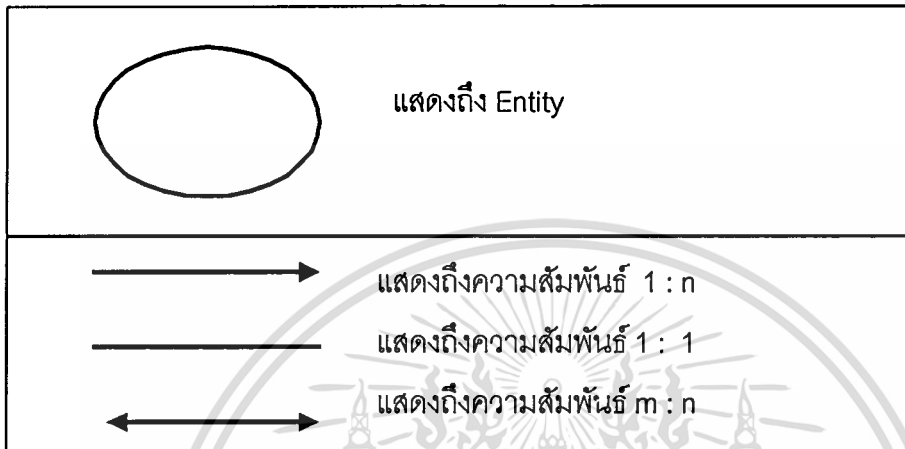
ข้อสังเกต ความสัมพันธ์ระหว่าง Foreign key ใน Fact Table กับ Primary key ใน Dimension table จะเป็นแบบ many - to - one

### Data Model

รูปแบบข้อมูลเป็นเหมือนแผนที่บอกทางไม่ ให้ชัดเจน ขาวรวมขี้ ฐ ลากแหล่ง ต่าง ๆ เข้าด้วยกัน ซึ่ง Data Model จะให้รายละเอียดเกี่ยวกับข้อมูล เป็นต้นว่า ความหมาย ความสัมพันธ์กับแหล่งอื่น ๆ อาจรวมถึงผู้ใช้ข้อมูลด้วย ซึ่งในการกำหนดรูปแบบข้อมูลนี้ สามารถแบ่งออกได้ 3 ระดับคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

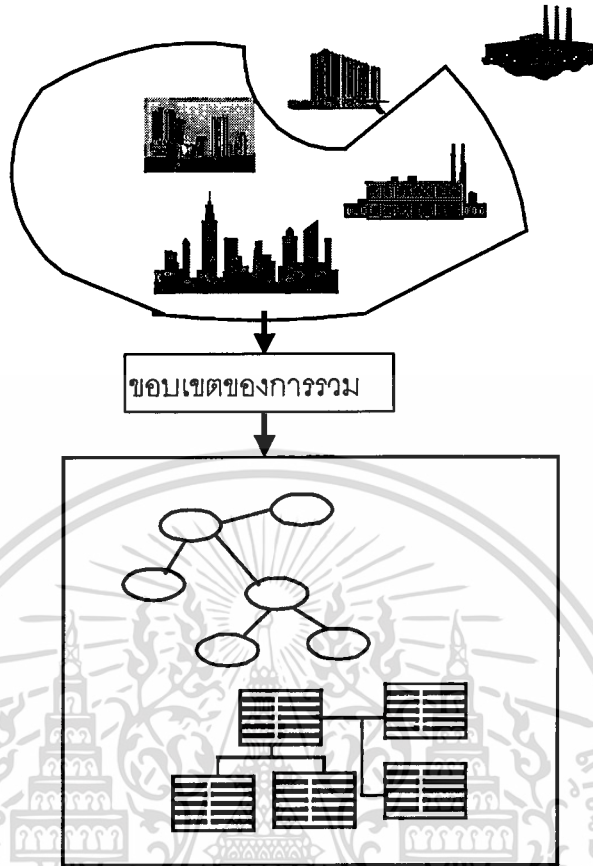
1. ระดับสูง (High Level) หรือที่เรียกว่า ERD : Entity Relationship Level เป็นระดับที่แสดงความสัมพันธ์ระหว่าง Entity และมีการกำหนดขอบเขตของ Data Model ที่จะนำเข้ามาใช้งาน โดยมีสัญลักษณ์ที่แสดงถึงความสัมพันธ์ของข้อมูลดังนี้



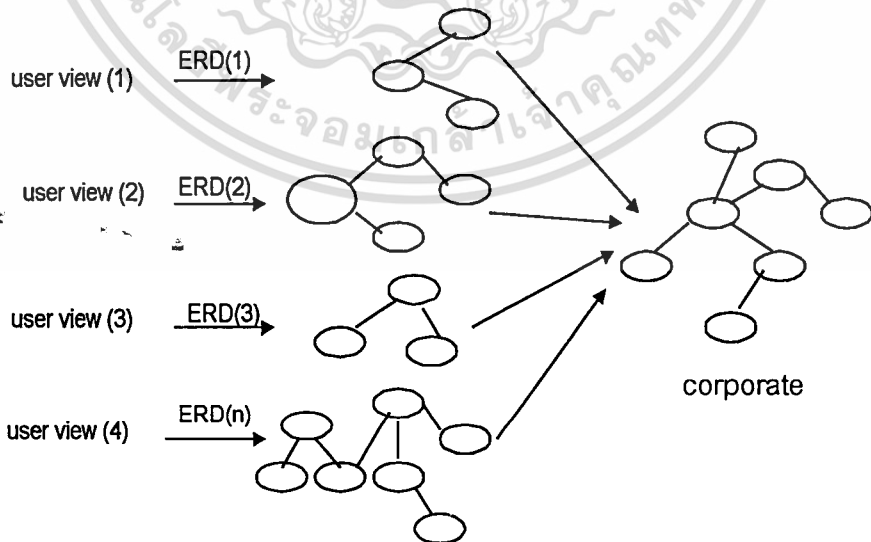
รูปที่ 5.2 Entities และ Relationship

วิธีการเขียน โดยเขียนชื่อ Entity ในวงกลม และใช้ เส้น แสดงถึงความสัมพันธ์ ตัวอย่างการเขียนความสัมพันธ์ระหว่าง entity Supplier, order, part, assembly ดังรูปที่ 5.2

ในการเขียน ERD แสดงถึงความสัมพันธ์ของข้อมูลนั้นผู้ออกแบบจะต้องมีการกำหนดขอบเขตการรวมข้อมูลด้วย หมายถึงว่าข้อมูลที่ต้องการใช้ มาจากที่ใดบ้าง ควรกำหนดขอบเขตเพื่อให้เห็นชัดเจน ก่อนทำการรวมข้อมูลเข้าสู่ Warehouse ดังตัวอย่างในรูปที่ 5.3



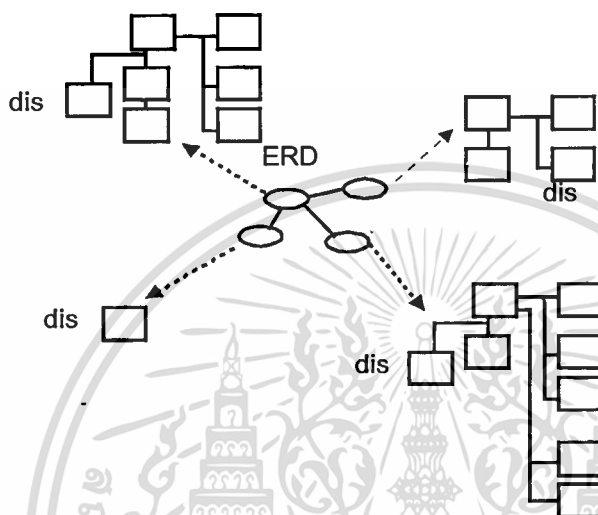
รูปที่ 5.3 แสดงการกำหนดขอบเขตก่อนการรวมข้อมูล



รูปที่ 5.4 แสดงโครงสร้างของการรวม ERD จากหลาย ๆ มุมมองของผู้ใช้ เข้าด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

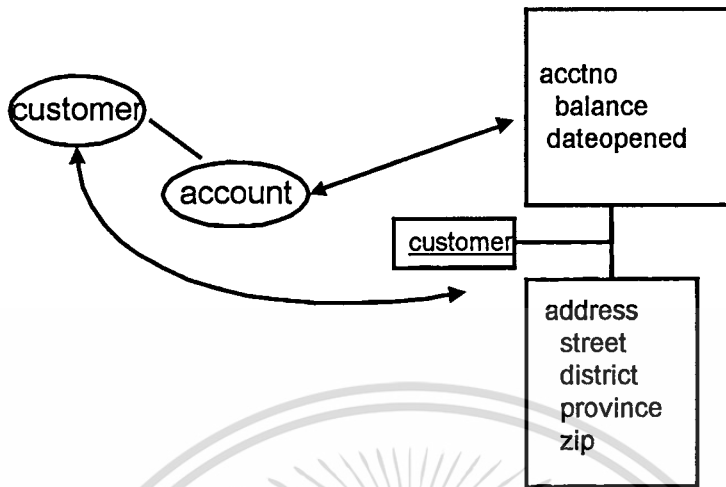
2.ระดับกลาง (Midlevel) เป็นการกำหนดรายละเอียดของข้อมูลในแต่ละEntity หรือเรียกว่ากำหนด Data item หรือ กำหนด attributed ที่จะใช้เป็น key สำหรับเชื่อมต่อกับ entity อื่น ๆ เพื่อประโยชน์ในการรวมกลุ่มข้อมูลเข้าด้วยกัน (grouping ) ซึ่งมีโครงสร้างในการกำหนดดังนี้



รูปที่ 5.5 แสดงโครงสร้างของ Data Item ใน ERD

โดยในการสร้างข้อมูลในระดับกลางนี้มีสิ่งที่จะต้องทำอยู่ 4 ขั้นตอนคือ

1. กำหนดกลุ่มข้อมูลเริ่มต้น (Primary Group ) พร้อมทั้งระบุ attributed ที่อยู่ในกลุ่มนั้น และ Attributed สำหรับเป็นค่า key
2. กำหนดกลุ่มข้อมูลกลุ่มที่สอง (Secondary Group) พร้อมทั้งระบุ attributed ที่อยู่ในกลุ่ม และหา attributed สำหรับกำหนดค่า key ของข้อมูลกลุ่มที่ 2
3. กำหนดจุดเชื่อมต่อหรือความสัมพันธ์ระหว่างกลุ่มที่ อีรี ยว่ aFore ๓ key

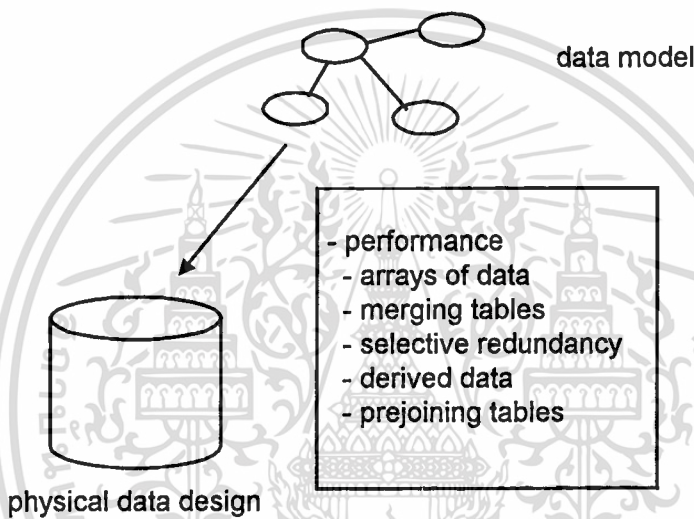


รูปที่ 5.6 กำหนด Data item และความสัมพันธ์ระหว่าง entity

แสดงความสัมพันธ์ระหว่าง customer และ account ซึ่งมีการกำหนด รายละเอียด หรือ data item ของ account และมี customer เป็น Foreign Key

4. กำหนดประเภทของกิจกรรม ซึ่งเป็นผลที่เกิดจากการกระทำระหว่างข้อมูล ที่มีความสัมพันธ์กัน ให้อยู่ในกลุ่มเดียวกัน (โดยส่วนมากจะรู้จักในลักษณะของ Transaction หรือ Fact Table ) จากนั้นพยายามจัดข้อมูลเหล่านั้นให้อยู่ในรูปแบบ Star Schema เพื่อการรวม table ให้น้อยลง โดยข้อมูลในส่วนนี้จะเรียกว่าเป็น Fact Table ในบางครั้งการออกแบบข้อมูลจะมี Fact Table มากกว่า 1 ซึ่งก็เป็นไปได้ และมี dimension table จำนวนมาก ซึ่งในกรณีของ Dimension นี้จะมีอย่างน้อย ขึ้นอยู่กับการใช้ข้อมูล ถ้าขอบเขตของการใช้ข้อมูลแคบหมายถึง ความต้องการใช้ข้อมูลส่วนมากอยู่ในรูปของ Heirarchies จำนวนของ Dimension ก็จะมีน้อย แต่ถ้าต้องการวิเคราะห์และเปรียบเทียบข้อมูล จำนวนของ dimension ก็จะมีมากแล้วแต่ความต้องการของผู้ใช้

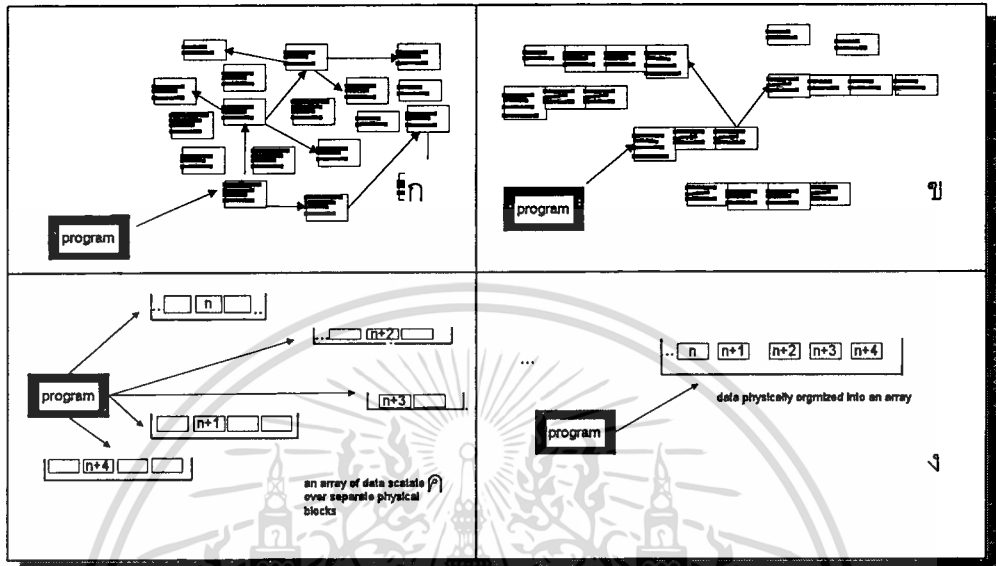
3. ระดับล่าง (Physical level) เป็นขั้นตอนของสร้างฐานข้อมูลจริง ๆ โดยสร้างจากข้อมูลในระดับกลาง ที่ได้กำหนดไว้แล้ว ซึ่งมีขั้นตอนที่สำคัญคือ รวมข้อมูลเข้าด้วยกัน โดยใช้ Key เป็นตัวเชื่อม โดยเฉพาะ Dimension Table ที่มีจำนวนมาก ๆ ซึ่งในขั้นตอนนี้เป็นเสมือนหัวใจของข้อมูลเพราะเกี่ยวข้องกับการเก็บข้อมูล และการ Access ข้อมูล ซึ่งจำเป็นจะต้องมีความสัมพันธ์กัน เพื่อประสิทธิภาพของ Data Warehouse รูปที่ 5.7 แสดงการเก็บข้อมูลเพื่อประสิทธิภาพโดยข้อมูลที่เก็บควรคำนึงถึงสิ่งต่าง ๆ ดังรูป



รูปที่ 5.7 แสดงปัจจัยที่มีผลต่อประสิทธิภาพของฐานข้อมูล  
ใน Data Warehouse

จากที่กล่าวมาแล้วว่าการเก็บข้อมูลมีผลต่อประสิทธิภาพของ warehouse ในที่นี้เน้นที่การ access ข้อมูลใน Data warehouse ถ้าข้อมูลที่เก็บในเนื้อที่หน่วยความจำที่อยู่ในบริเวณเดียวกันหรือใกล้เคียง ก็จะทำให้การ Access ข้อมูลนั้นเร็วขึ้น นั่นหมายถึง การจัดการกับข้อมูลที่มีอยู่ว่าจะมีวิธีการจัดการอย่างไรเพื่อสนับสนุนการ Access ข้อมูลให้มีประสิทธิภาพสูงสุด ซึ่งการจัดการกับข้อมูลของ data warehouse ใช้วิธีการ Denormalize ข้อมูล ซึ่งจะเป็นการลดจำนวนตารางจากฐานข้อมูลเชิงสัมพันธ์ ที่มีจำนวนมาก ๆ โดยเป็นข้อมูลที่ผ่านการ Normalize แล้ว นำมาทำการ Denormalize เพื่อเปลี่ยนเป็นฐานข้อมูลสำหรับ Data warehouse

เปรียบเทียบการเข้าถึงข้อมูลของโปรแกรมในฐานข้อมูลที่เก็บข้อมูล Normalize /Denormalize



รูปที่ 5.8 แสดงการเข้าสู่ข้อมูลที่เก็บในรูปแบบ normalize กับ denormalize

รูป ก. แสดงการประมวลผลข้อมูลที่มีหลาย ๆ ตารางที่เก็บในรูปแบบฐานข้อมูลเชิงสัมพันธ์ เป็นข้อมูลที่ผ่านการ Normalize แล้ว ซึ่งทำให้มีผลต่อประสิทธิภาพการทำงานของโปรแกรม เพราะในการทำงานจะต้องมีเข้าถึงหลาย ๆ ตาราง เพื่อหาค่าที่ต้องการ

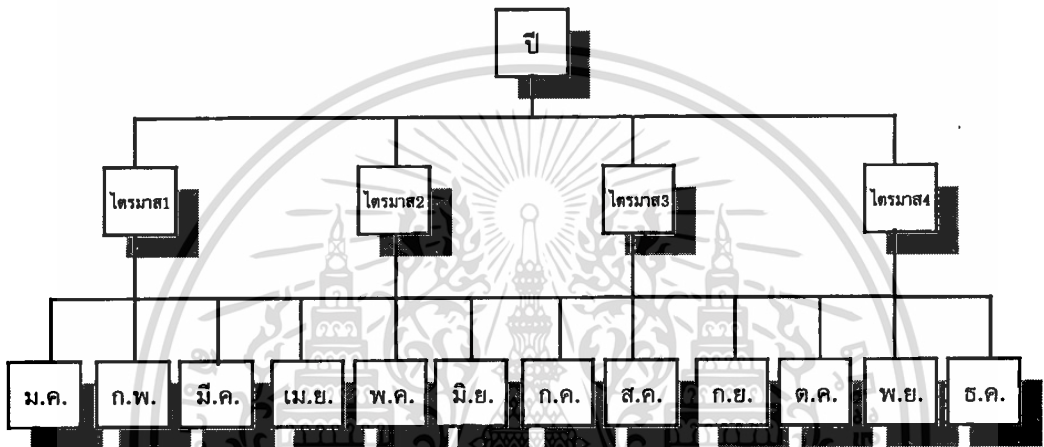
รูป ข. แสดงการประมวลผลหรือการเข้าถึงข้อมูลของโปรแกรม ที่เก็บข้อมูลในรูปแบบฐานข้อมูลเชิงสัมพันธ์ โดยเป็นข้อมูลที่อยู่ในรูป Denormalize ซึ่งมีจำนวน ตารางไม่กี่ตาราง ทำให้ประสิทธิภาพของการเข้าถึงข้อมูลของโปรแกรม ดีกว่าการเก็บข้อมูลในรูปแบบ Normalize

รูป ค และ ง. แสดงการสร้าง array สำหรับเก็บข้อมูล โดยรูป ค. มีการจองพื้นที่หน่วยความจำแบบกระจัดกระจาย เพราะฉะนั้นจึงส่งผลกระทบต่อประสิทธิภาพของการเข้าถึง ข้อมูล ในทางตรงกันข้าม array ในรูป ง. มีการจองพื้นที่ที่ติดต่อกันเป็นลำดับทำให้การเข้าถึงข้อมูลเร็วกว่า`

#### 5.4 การสร้างลำดับชั้นของข้อมูล (Hierarchies)

ข้อมูลสำหรับสนับสนุนการตัดสินใจ ควรเป็นข้อมูลที่สามารถมองขอบเขตของข้อมูลได้ หรือมองเป็นระดับจาก สูงไปต่ำหรือ ต่ำ ไปสูง เพื่อวิเคราะห์ข้อมูลเหล่านั้น. ถ้าการมองข้อมูลจากระดับสูงไปยัง ระดับต่ำเพื่อรายละเอียดจะเรียกว่า Drilling Down ส่วนการมองจากระดับเอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับกักรใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่ำไปยังระดับสูงเรียกว่า Rolling up ซึ่งทั้ง 2 คำเป็นศัพท์ที่ใช้เรียกในการมองข้อมูลใน data warehouse โดยในการมองข้อมูลของผู้ใช้จะสามารถมองได้ทีละระดับขึ้นอยู่กับความลึกของข้อมูลใน warehouse ซึ่งจะมีการออกแบบมุมมองของข้อมูลให้เป็นลำดับชั้น (Hierarchies) เพื่อการมองข้อมูลที่มีประสิทธิภาพ ตัวอย่าง ข้อมูลที่เป็นลำดับชั้น ได้แก่ ชั่วโมง ,วัน ,สัปดาห์,เดือน,ไตรมาส และปี ดังรูป ที่แสดงลำดับชั้นของ ปี ไตรมาส และเดือน

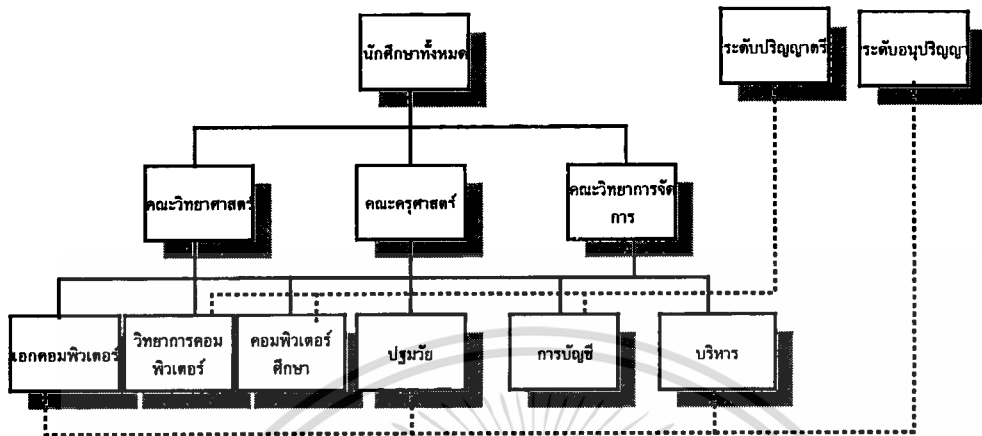


รูปที่5.9 แสดงลำดับชั้นของข้อมูล

ในธุรกิจส่วนใหญ่จำเป็นต้องดูการเคลื่อนไหวของข้อมูลอยู่ตลอดเวลาเช่นข้อมูลการขายทั้งปี โดยแยกเป็นไตรมาส หรือแยกเป็น เดือน นอกจากนี้ยังต้องการดูยอดขายทั้งประเทศ แบ่งเป็นภาคแบ่งเป็นจังหวัด เป็นอำเภอ เป็นต้น ในการออกแบบลำดับชั้นข้อมูลในธุรกิจ มักจะออกแบบจากระดับสูงไปหาระดับต่ำเสมอ และในข้อมูลชุดเดียวกันสามารถที่จะมีหลายๆ มุมมองได้ หรือเรียกว่า Multiple Hierarchies ดังรูปที่ 5.10 เพราะฉะนั้นในการออกแบบรูปแบบสำหรับมองข้อมูลจึงมีความจำเป็นอย่างยิ่งกับข้อมูลสำหรับสนับสนุนการตัดสินใจ

ในการสร้างลำดับชั้นของข้อมูลอย่างเดียวยังไม่เพียงพอสำหรับการวิเคราะห์ข้อมูลของระบบสนับสนุนการตัดสินใจ แต่ต้องมีการกำหนดความต้องการใช้ข้อมูลเพื่อมุมมองหลายมิติตัว ซึ่งเรียกว่า Dimension เพื่อเพิ่มความยืดหยุ่นในการเรียกใช้ข้อมูล เพราะฉะนั้นผู้ออกแบบจะต้องนำความต้องการเหล่านั้นมาวิเคราะห์เปรียบเทียบกับฐานข้อมูลเดิมที่มีอยู่เพื่อสร้างฐานข้อมูลสำหรับ Data Warehouse หรือที่เรียกว่า fact table และ Dimension Table ซึ่งจะต้องออกแบบ

ให้มีสัมพันธ์กันและเป็นประโยชน์สำหรับการวิเคราะห์ข้อมูลสูงสุด เพราะฉะนั้นจำเป็นต้องใช้ผู้ออกแบบฐานข้อมูลสำหรับ Data warehouse ที่มีประสบการณ์สูง



รูปที่ 5.10 แสดง Multiple Hierarchies

จากรูป การมองจำนวนนักศึกษาทั้งหมด แบ่งตาม คณะ และ เอก ส่วน อีกมุมมองหนึ่งสามารถที่จะมองนักศึกษาโดยแบ่งตามระดับ และ เอก เป็นต้น

## 5.5 แนวทางในการเรียกใช้ข้อมูล

วัตถุประสงค์หลักของการสร้าง Data Warehouse ก็เพื่อการเรียกใช้ข้อมูลที่มีประสิทธิภาพ เพราะฉะนั้นการเรียกใช้ข้อมูลใน Data Warehouse จำเป็นต้องมีสภาพแวดล้อมที่คอยสนับสนุนการเข้าถึงข้อมูลเป็นอย่างดี เป็นต้นว่า โปรแกรมหรือ Front End รวมถึงการฝึกอบรม การสนับสนุนการใช้งาน End -User และการพัฒนา DSS Application โดยเฉพาะ DSS Application ควรเป็นโปรแกรมที่สามารถกำหนดค่าต่าง ๆ เพื่อวิเคราะห์ข้อมูล และเปรียบเทียบได้ เพราะฉะนั้นจำเป็นต้องเลือก Tools ที่มีประสิทธิภาพ ซึ่ง Tools ที่มีขายทั่วไปมีคุณสมบัติที่แตกต่างกันไป บางตัวมีความสามารถสูง บางตัวมีความสามารถปานกลางทั้งนี้ขึ้นอยู่กับผู้ผลิตว่าจะให้ tools ที่ผลิตขึ้นสามารถใช้งานได้มากน้อยเพียงใด Tools ที่ใช้ในการเข้าถึงข้อมูลใน Data Warehouse สามารถแบ่งได้หลายลักษณะดังนี้

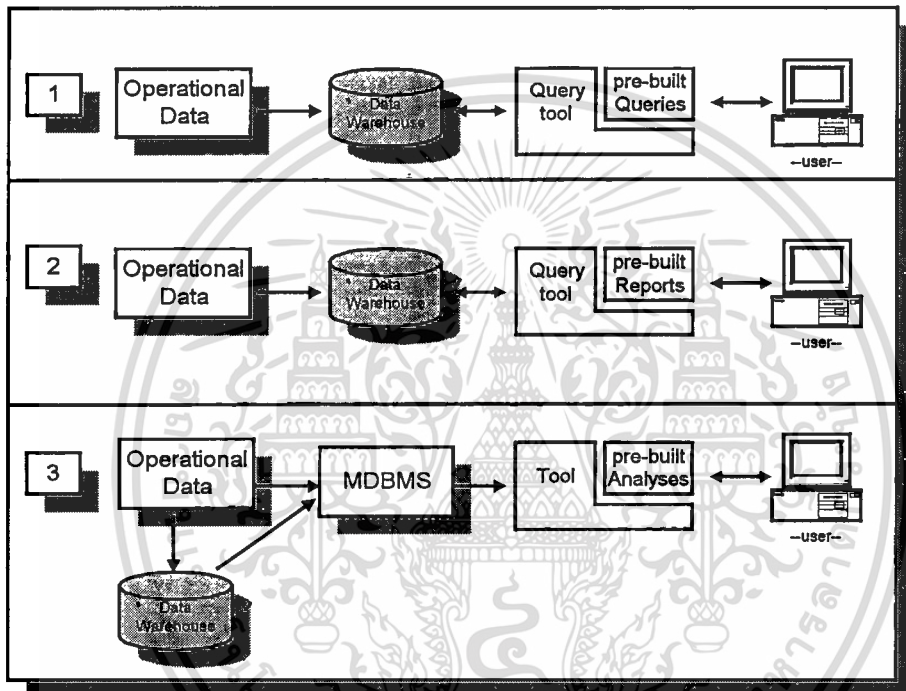
แบ่งตามลักษณะการเรียกใช้ข้อมูลได้ 3 ระดับ ดังนี้

1. Query Tools เป็น Tools ที่เข้าถึงข้อมูลใน Data Warehouse โดยตรง สามารถที่จะทำ query และวิเคราะห์ข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Report Writers เป็น Tools ที่ออกแบบมาเพื่อทำรายงานในรูปแบบต่าง ๆ โดยใช้ข้อมูลจาก Data warehouse

3. Multidimensional Database Management (MDBMS) เป็น Tools ที่ นำข้อมูลจาก Data warehouse และ Operational เข้าสู่รูปแบบฐานข้อมูลแบบหลายมิติ เพื่อเพิ่มประสิทธิภาพการใช้เรียกงาน โดยเฉพาะการวิเคราะห์ข้อมูลแบบหลายมิติ ในรูป Slice, and dice,drill-down



รูปที่ 5.11 แสดง ระดับของ tools สำหรับการเรียกใช้งาน

#### แบ่งตามการใช้งานหลาย Platform

หมายถึง tools ที่ใช้สามารถสนับสนุนการใช้งานได้หลาย Platform โดย Platform ในที่นี้หมายถึง ฮาร์ดแวร์ และซอฟต์แวร์

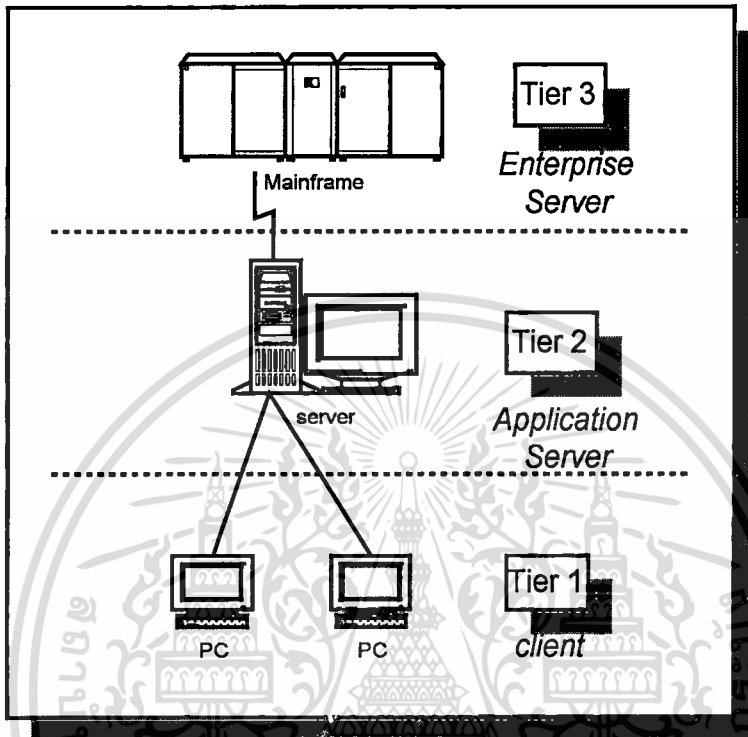
Hardware Platform จะมองในรูปแบบของการสนับสนุนการใช้งานในระบบ Client/Server หรือที่เรียกว่า Multitiered ซึ่งมีสถาปัตยกรรมดังนี้

- One - tier เป็นการทำงานที่ Client ทั้งหมด โปรแกรม ,ฐานข้อมูล, การแสดงผล และ ฟังก์ชันสำหรับใช้งาน เก็บไว้ที่เครื่องเดียวกันทั้งหมด ประมวลผลและแสดงผลที่ client

- Two - tier ทำหน้าที่เป็น Application Server โดยแยก โปรแกรมและฐานข้อมูลไว้คนละเครื่อง สามารถที่จะ Share ข้อมูล และ Share Application ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Three - tier ทำหน้าที่เป็น Enterprise Server โดยแยกโปรแกรมและฐานข้อมูลไว้คนละเครื่อง Share พื้นที่ในการเก็บข้อมูล Share ข้อมูล และ Share Application



รูปที่ 5.12 แสดงสถาปัตยกรรม Three tier Client/Server

Software พิจารณาในส่วนที่สามารถเรียกใช้ข้อมูลที่เก็บอยู่ในรูปแบบที่แตกต่างกันได้ เช่น Text, Graphic , database ,spreadsheet , ASCII ซึ่งแต่ละแบบอาจถูกจัดการโดยโปรแกรมที่แตกต่างกัน เช่น Access, Foxbase, Oracle, Informix , Sybase เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### ทฤษฎีการออกแบบที่เกี่ยวข้องกับคลังข้อมูล (Data Warehouse)

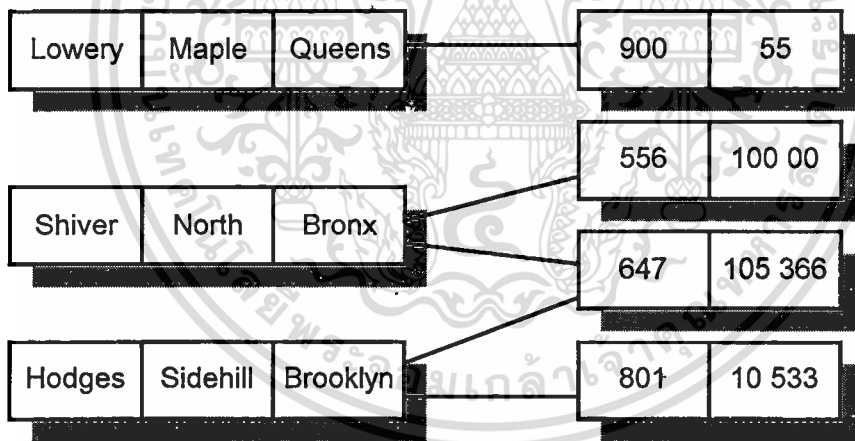
#### 6.1 ระบบฐานข้อมูล (Database System)

ฐานข้อมูล คือ แหล่งเก็บข้อมูลและความสัมพันธ์ระหว่างข้อมูลนั้น มีการออกแบบและควบคุมเป็นอย่างดีเพื่อไม่ให้ข้อมูลมีความซ้ำซ้อนหรือซ้ำซ้อนน้อยที่สุด และมีความถูกต้องสูงสุด

##### 6.1.1 ความสัมพันธ์ระหว่างข้อมูลในฐานข้อมูล

##### 1. แบบโครงข่าย (Network Model)

การจัดข้อมูลในแบบโครงข่ายนี้จะแสดงด้วยกลุ่มของเรคคอร์ด ที่มีส่วนเชื่อมต่อ หรือมีตัวชี้แสดงความสัมพันธ์ของข้อมูล โดยที่โครงสร้างความสัมพันธ์ของเรคคอร์ดที่อยู่ในฐานข้อมูลจะมีสภาพเป็นกลุ่มของความสัมพันธ์ที่ไม่มีข้อกำหนดที่แน่นอน ซึ่งมีความสัมพันธ์เป็นแบบกลุ่ม (Many to Many) ดังรูปที่ 6.1



รูปที่ 6.1 แสดงความสัมพันธ์ระหว่างข้อมูลแบบโครงข่าย

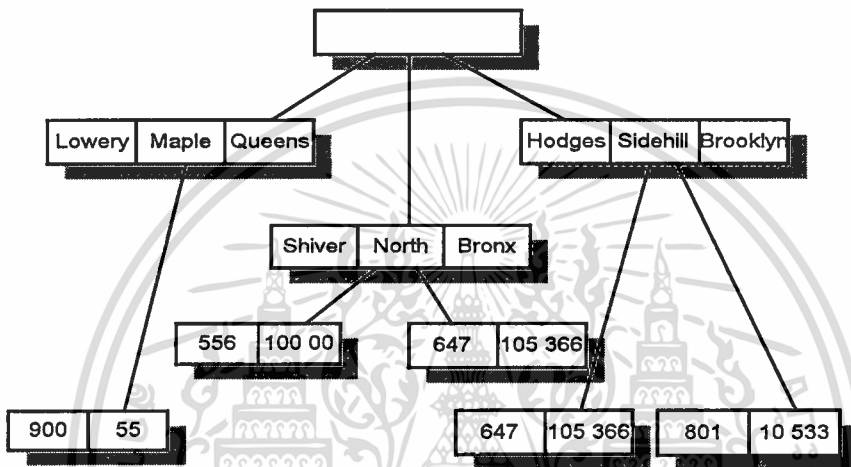
##### 2. แบบลำดับชั้น (Hierarchical Model)

ข้อมูลแบบลำดับชั้นนี้จะมีลักษณะใกล้เคียงกับของโครงข่าย แต่จะแตกต่างกันที่โครงสร้างความสัมพันธ์ของเรคคอร์ดที่อยู่ในฐานข้อมูล โดยจะมีความสัมพันธ์เป็นแบบหนึ่งกับกลุ่ม (one to many) ส่วนที่อยู่บนสุดเรียกว่าตัวแม่ (Root) และมีความสัมพันธ์กับข้อมูลให้ระดับล่างซึ่งเรียกว่าลูก (Child) แสดงความสัมพันธ์ดังรูปที่ 6.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. แบบเชิงสัมพันธ์ (Relational Model)

มีการแสดงรายละเอียดของข้อมูลและความสัมพันธ์ระหว่างข้อมูล อยู่ในรูปกลุ่มของตารางซึ่งในแต่ละตารางจะประกอบด้วยคอลัมน์ (Column) ต่าง ๆ โดยชื่อของคอลัมน์เหล่านั้นจะต้องมีชื่อไม่ซ้ำกัน และสามารถแสดงความสัมพันธ์ระหว่างข้อมูลอยู่ในรูปของตารางได้ โดยมีคีย์เป็นตัวชี้ (index) สำหรับแสดงความสัมพันธ์ระหว่างตาราง ดังรูปที่ 6.3



รูปที่ 6.2 แสดงความสัมพันธ์ระหว่างข้อมูลแบบลำดับชั้น

<i>name</i>	<i>street</i>	<i>city</i>	<i>number</i>
Lowery	Maple	Queens	900
Shiver	North	Bronx	556
Shiver	North	Bronx	647
Hodges	Sidehill	Brooklyn	801
Hodges	Sidehill	Brooklyn	647

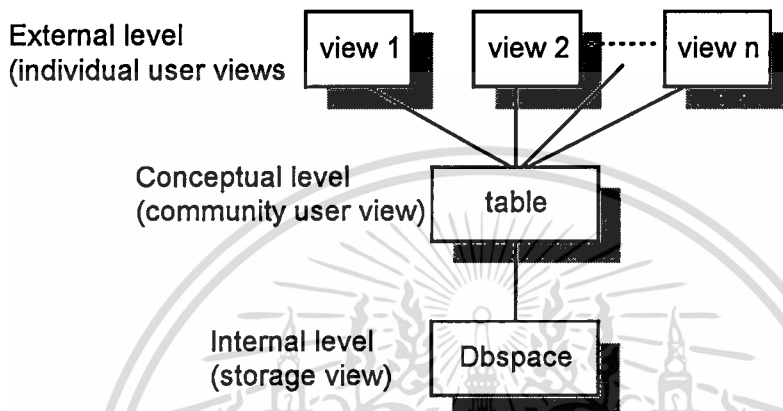
<i>number</i>	<i>balance</i>
900	55
556	10000
647	105366
801	10533

รูปที่ 6.3 แสดงความสัมพันธ์ระหว่างข้อมูลแบบเชิงสัมพันธ์.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.1.2 สถาปัตยกรรมมาตรฐานของฐานข้อมูล

สถาปัตยกรรมที่เป็นมาตรฐานของระบบฐานข้อมูลที่สร้างขึ้นเพื่อเป็นประโยชน์แก่ผู้ใช้งานฐานข้อมูล ซึ่งมีคุณสมบัติตามนิยาม 3 ชั้น ( Three level Schema Architecture) หรือ 3 ระดับดังรูปที่ 6.4 แสดงสถาปัตยกรรมของฐานข้อมูล 3 ระดับ



รูปที่ 6.4 สถาปัตยกรรมนิยามข้อมูล 3 ระดับ

1 ระดับภายใน (Internal Level) เป็นการกำหนดลักษณะโครงสร้างข้อมูลที่ถูกจัดเก็บจริงในอุปกรณ์บันทึกข้อมูล เช่น เก็บใน ฮาร์ดิสก์ ในเทป เป็นต้น การจัดเก็บข้อมูลในสื่อต่าง ๆ นั้นจะมีวิธีการจัดการที่แตกต่างกันไปขึ้นอยู่กับโปรแกรมจัดการว่าจะเก็บในรูปแบบใด เช่น แบบ B-Tree, Linklist เป็นต้น ข้อมูลในฐานข้อมูลในระดับนี้จะถูกจัดการด้วยระบบการจัดการฐานข้อมูล(DBMS) ที่แต่ละองค์กรเลือกใช้

2 ระดับแนวคิด (Conceptual Level) จะเป็นการกำหนดลักษณะรูปแบบข้อมูล ขนาดของข้อมูลและความสัมพันธ์ของข้อมูลทั้งหมดในระบบงาน เพื่อให้เห็นภาพรวมหรือเป็นการมองขอบเขตของความสัมพันธ์ของข้อมูลในระบบ เช่น ความสัมพันธ์ของข้อมูลที่อยู่ในรูปแบบเชิงสัมพันธ์ ในระดับนี้จะแสดง ระดับตาราง ชื่อคอลัมน์ ชนิดข้อมูลในแต่ละคอลัมน์ตลอดจนชื่อของคีย์ต่าง ๆ เป็นต้น

3 ระดับภายนอก (External Level) เป็นการกำหนดขอบเขตการมองข้อมูลของผู้ใช้งาน ซึ่งแต่ละคนแต่ละระดับก็จะสามารถใช้ข้อมูลที่แตกต่างกัน ทั้งนี้ขึ้นอยู่กับผู้ดูแลและผู้ควบคุมข้อมูล(DBA : Database Administration ) หรือผู้ที่มีอำนาจกำหนดสิทธิของการใช้ตาราง ๆ นั้นว่าจะให้ผู้ใช้แต่ละคนสามารถมองข้อมูลได้มากน้อยเพียงใด (View)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.1.3 ข้อดีของระบบฐานข้อมูล

#### 1. ลดความซ้ำซ้อนของข้อมูล (Redundancy)

ในฐานข้อมูลจะเก็บรวบรวมข้อมูลของระบบต่าง ๆ ไว้ ในฐานข้อมูลเดียวกันจึงมีความซ้ำซ้อนน้อยลง ทำให้ประหยัดพื้นที่ในการจัดเก็บข้อมูลด้วย และยังเพิ่มประสิทธิภาพของการเพิ่มข้อมูล (Insert) การลบข้อมูล (Delete) การเปลี่ยนแปลงแก้ไขข้อมูล (Update) ให้มีประสิทธิภาพมากที่สุด

#### 2. หลีกเลี่ยงความขัดแย้งของข้อมูลได้ระดับหนึ่ง (Inconsistency)

เมื่อข้อมูลไม่มีความซ้ำซ้อน ความขัดแย้งของข้อมูลก็ไม่มีตามไปด้วย เพราะจะมีการจัดเก็บที่เดียว เมื่อมีการเปลี่ยนแปลง เพิ่ม ลบ ก็ต้องทำที่ฐานข้อมูลทีเดียว เพราะฉะนั้นจึงไม่มีความขัดแย้งของข้อมูลในระบบฐานข้อมูล

#### 3. สามารถใช้ข้อมูลร่วมกันได้ (Share Data)

โปรแกรมที่ควบคุมดูแลระบบฐานข้อมูลก็คือโปรแกรมระบบจัดการฐานข้อมูลหรือที่เรียกว่า DBMS โดย DBMS จะจัดการให้ผู้ใช้สามารถทำงานพร้อม ๆ กันได้หลายคน ดังนั้นโปรแกรมที่พัฒนาขึ้นมาภายใต้การดูแลของระบบการจัดการฐานข้อมูลเชิงสัมพันธ์จะสามารถใช้ข้อมูลร่วมกันในฐานข้อมูลเดียวกันได้

#### 4. สามารถควบคุมมาตรฐานได้ (Standard)

ในระบบฐานข้อมูลผู้ดูแลและควบคุมฐานข้อมูล (DBA : Database Administrator) จะสามารถกำหนดมาตรฐานของขนาดข้อมูลในตัวแปรต่าง ๆ ชื่อของตัวแปรและสิทธิการใช้ข้อมูลต่าง ๆ เพื่อประโยชน์ในการพัฒนาระบบงาน การย้ายข้อมูลระหว่างระบบงานและการกำหนดรูปแบบข้อมูลให้สามารถใช้งานร่วมกันได้

#### 5. มีระบบรักษาความปลอดภัย (Security)

ผู้ดูแลและควบคุมฐานข้อมูล (DBA) จะสามารถกำหนดสิทธิและอำนาจการใช้งานข้อมูลในฐานข้อมูลให้กับผู้ใช้ต่าง ๆ ได้ ซึ่งระดับสิทธิและอำนาจการใช้งานข้อมูลอาจไม่เหมือนกันขึ้นอยู่กับความเหมาะสม สำหรับฐานข้อมูลเชิงสัมพันธ์จะมีวิว (View) ซึ่งเป็นตารางสมมติ (Virtual Table) ช่วยกำหนดสิทธิการมองเห็นและใช้งานข้อมูลในตารางที่เก็บข้อมูลจริง ๆ (Base Table)

#### 6. สามารถรักษาความถูกต้องของข้อมูลได้ (Integrity)

ในระบบการจัดการฐานข้อมูล (DBMS) สามารถรองรับการใช้งานของผู้ใช้หลายคนพร้อม ๆ กัน ดังนั้นความถูกต้องของข้อมูลจึงมีความสำคัญมากและต้องควบคุมดูแลให้ดีเนื่องจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้ต่าง ๆ สามารถใช้งานและเปลี่ยนแปลงแก้ไขข้อมูลในฐานข้อมูลได้ ซึ่งถ้ามีความผิดพลาดจะส่งผลกระทบต่อแผนงานโดยรวม

### 7. สามารถปรับความต้องการของผู้ใช้ให้เป็นในแนวเดียวกัน( Conflicting )

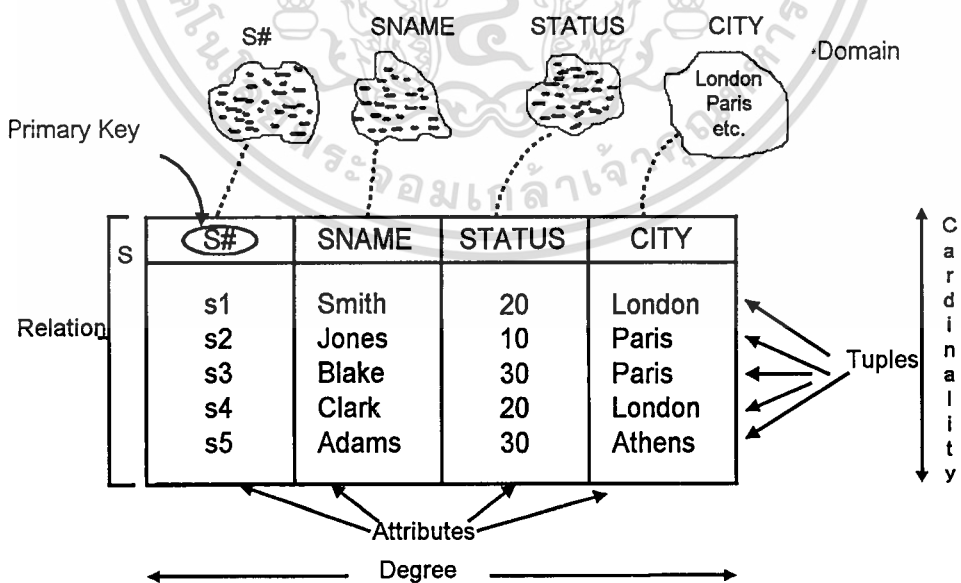
เนื่องจากผู้ดูแลและควบคุมฐานข้อมูล (DBA) เมื่อได้รับรู้ความต้องการและลำดับความสำคัญของผู้ใช้ในระบบทั้งหมด ก็จะสามารถปรับแต่งความต้องการของผู้ใช้เหล่านั้นให้ไปแนวทางเดียวกัน เพื่อก่อให้เกิดประโยชน์ของส่วนรวมให้สูงสุด เช่น จัดเตรียมอุปกรณ์ในการเก็บข้อมูลที่มีความเร็วในการเรียกใช้และเก็บข้อมูลสูงให้กับระบบงานที่มีความสำคัญต่อองค์กร

## 6.2 รูปแบบเชิงสัมพันธ์ ( Relation Model)

ทฤษฎีที่เกี่ยวข้องในหัวข้อนี้จะกล่าวถึงรายละเอียดของรูปแบบเชิงสัมพันธ์เพราะเป็นรูปแบบของข้อมูลที่เป็น Operational และข้อมูลใน data Warehouse

### 6.2.1 โครงสร้างข้อมูล (Data Structure)

โครงสร้างของข้อมูลจะต้องเป็นรีเลชัน หรือ ตาราง ที่แสดงความสัมพันธ์ของข้อมูลในรูปแบบของ แถวและคอลัมน์ กับตารางอื่น ๆ ที่เกี่ยวข้อง แสดงความสัมพันธ์ของโครงสร้างข้อมูลในฐานข้อมูลดังรูปที่ 6.5



รูปที่ 6.5 แสดงรายละเอียดของ Relation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากความสัมพันธ์สามารถที่จะเปรียบเทียบคำศัพท์ที่ใช้ในเชิงวิชาการและคำศัพท์ที่ใช้ทั่วไปได้ดังนี้

ศัพท์เชิงวิชาการ	ศัพท์เชิงธุรกิจ
Relation	Table ที่มี Row ไม่ซ้ำกัน
Tuple	Row หรือ Record
Cardinality	จำนวนของ Row ในแต่ละ Table
Attribute	Column หรือ Field
Degree	จำนวน Column ในแต่ละ table
Primary key	column ที่อาจจะเป็น 1 column หรือมากกว่า 1 column ที่เมื่อระบุค่าของข้อมูลใน column ที่เป็น primary key 1 ค่า จะได้ข้อมูลใน Table นั้นออกมาเพียง 1 row ค่าของข้อมูลที่เป็นไปได้ทั้งหมดในแต่ละ column

### 6.2.2 กฎควบคุมความถูกต้องของข้อมูล(Data Integrity Contraint)

กฎควบคุมความถูกต้องของข้อมูลเป็นการจัดการเกี่ยวกับขอบเขตค่าของโดเมนและเงื่อนไขการกระทำของคีย์หลัก(Primary key) และคีย์นอก( Foreign Key) ในระบบฐานข้อมูลเชิงสัมพันธ์มีกฎของการควบคุมความถูกต้องที่ใช้บ่อยคือ กฎควบคุมความถูกต้องของเอนทิตี(The Entity Integrity Rule) นิยามคือจะไม่มีแอททริบิวต์ที่เป็นส่วนประกอบของคีย์หลักตัวใดตัวหนึ่งในรีเลชันเป็นค่าว่าง หรือ Null ซึ่งกฎในข้อนี้จะควบคุมความถูกต้องของคีย์หลัก

### 6.2.3 การจัดการข้อมูล (Data Manipulate)

ภาษาในการจัดการข้อมูลต้องมีคุณสมบัติของการเป็น Relation อย่างสมบูรณ์ คืออย่างน้อยต้องมีความสามารถเทียบเท่าคณิตศาสตร์ของรีเลชัน (Relation Calculus) หรือพีชคณิตของ Relation Algebra ซึ่งพีชคณิตของรีเลชัน เป็นภาษาสอบถามข้อมูลแบบไพโรซีเตอร์ (Procedure) ซึ่งเกิดจากกลุ่มของการกระทำ กับรีเลชันต่าง ๆ การกระทำเหล่านั้นได้แก่ restrict ,project,union,intersaction ,join เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.3 การจัดการฐานข้อมูลเชิงสัมพันธ์ Relational Database ( RMBMS)

ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์เป็นซอฟต์แวร์ (Software) สำหรับจัดการข้อมูลและความสัมพันธ์ของข้อมูลที่เป็นเชิงสัมพันธ์ รวมทั้งมีภาษาสำหรับกำหนดโครงสร้างของข้อมูลในระบบต่าง ๆ ตามสถาปัตยกรรมของระบบฐานข้อมูล มีภาษาสำหรับการค้นหาและแก้ไขข้อมูล รวมทั้งการจัดการความปลอดภัยของข้อมูล การเข้าถึงข้อมูลและการติดต่อกับผู้ใช้และโปรแกรมอื่น ซึ่งมีประโยชน์ดังนี้

#### 6.3.1 ภาษาแอสคิวแอล (SQL : Structure Query Language)

มีภาษา SQL ช่วยจัดการการทำงานเกี่ยวกับการเข้าถึงข้อมูล สามารถช่วยใช้ในการ join ข้อมูลหรือจัดการกับข้อมูลก่อนนำเข้าสู่ Model ใน โปรแกรมจะใช้ SQL ช่วยในการค้นหาข้อมูล ซึ่งสามารถจัดการกับข้อมูลใน Model ได้อย่างมีประสิทธิภาพ เพราะการค้นหาข้อมูลใน Multidimensional จำนวนของการ join จะน้อยมาก ทำให้สามารถค้นหาข้อมูลได้อย่างรวดเร็ว ช่วยจัดการการทำงานเกี่ยวกับข้อมูลซึ่งแบ่งออกได้ 2 ส่วนคือ

1. การจัดการเกี่ยวกับการกำหนดโครงสร้างข้อมูล (Data Definition Language:DDL) มีหน้าที่ในการกำหนดโครงสร้างข้อมูลที่ใช้แต่ละคนมองเห็น โครงสร้างข้อมูลที่นักออกแบบฐานข้อมูลมองเห็น และโครงสร้างข้อมูลที่เกิดขึ้นในอุปกรณ์เก็บข้อมูลซึ่งของการแปล DDL จะเก็บในไฟล์ที่เรียกว่า พจนานุกรมของข้อมูล (Data Dictionary) ตัวอย่างคำสั่งเช่น CREATE DATABASE, CREATE TABLE เป็นต้น

2. การจัดการเกี่ยวกับข้อมูล (Data Manipulation Language : DML) มีหน้าที่ในด้านจัดการเข้าถึงข้อมูล ได้แก่ การสอบถามหรือเรียกค้นข้อมูลที่อยู่ในฐานข้อมูล การเพิ่มข้อมูลใหม่ (Insert) การลบข้อมูล(Delete) การเปลี่ยนแปลงแก้ไข ตัวอย่างคำสั่ง เช่น Select, Insert, Delete และ Update เป็นต้น

#### 6.3.2 การจัดการด้านความเป็นอิสระของข้อมูล (Data Independence)

เมื่อมีการเปลี่ยนแปลงแก้ไขโครงสร้างข้อมูล ในนิยามข้อมูลระดับแนวคิดหรือในนิยามข้อมูลระดับภายในนั้นจะไม่มีผลกระทบต่อโปรแกรมที่ให้ผู้ใช้งานในนิยามข้อมูลระดับภายนอก เนื่องจากระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ จะช่วยจัดการให้ผู้ใช้มองเห็นโครงสร้างข้อมูลในนิยามข้อมูลระดับนอกเหมือนเดิมและสามารถใช้งานได้ตามปกติ ซึ่งการจัดการด้านความเป็นอิสระของข้อมูลแบ่งออกได้ 2 ลักษณะคือ

##### 1. ความเป็นอิสระของข้อมูลในระดับตรรกะ(Logic Data Independence)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ามีการแก้ไขโครงสร้างข้อมูลในนิยามข้อมูลระดับแนวคิด จะไม่มีผลกระทบต่อโครงสร้างข้อมูลในนิยามข้อมูลระดับภายนอกที่ผู้ใช้ใช้งานอยู่ เช่น การเปลี่ยนแปลงขนาดคอลัมน์ หรือเพิ่มคอลัมน์เข้าไป

## 2. ความเป็นอิสระของข้อมูลในระดับกายภาพ (Physical Data independence)

ถ้ามีการแก้ไขโครงสร้างข้อมูลในนิยามข้อมูลระดับภายใน จะไม่มีผลกระทบต่อโครงสร้างข้อมูลในนิยามข้อมูลระดับแนวคิด เช่น มีการเปลี่ยนแปลงวิธีการเข้าถึงข้อมูลในนิยามระดับภายใน หรือการเปลี่ยนแปลงอุปกรณ์ที่ใช้จัดเก็บข้อมูล เป็นต้น

### 6.3.3 ความปลอดภัยของข้อมูล (Data Security)

ระบบการจัดการฐานข้อมูลได้มี ระบบการสอบถามรายชื่อพร้อมทั้งรหัสลับของผู้เข้ามาใช้งานในระบบเพื่อให้เข้าทำงานในส่วนที่ผู้ใช้เกี่ยวข้องเท่านั้น นอกจากนี้ ข้อมูลที่เก็บอยู่ในฐานข้อมูลเชิงสัมพันธ์นั้นจะถูกทำการเข้ารหัส (Encryption) เพื่อป้องกันการเข้าถึงข้อมูลด้วยภาษาอื่น ๆ โดยไม่ผ่านระบบการจัดการฐานข้อมูลเชิงสัมพันธ์

### 6.3.4 การจัดอำนาจในการใช้ข้อมูล (Data Authority)

ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ สามารถทำการกำหนดสิทธิอำนาจการใช้งานข้อมูลให้แก่ผู้ใช้ต่าง ๆ ได้โดยการใช้คำสั่ง GRANT หรือเพิกถอนสิทธิได้โดยใช้คำสั่ง REVOKE นอกจากนี้ยังใช้ view เพื่อช่วยกำหนดสิทธิการใช้งานข้อมูลของผู้ใช้แต่ละคนในนิยามข้อมูลระดับภายนอกได้อีกด้วย

### 6.3.5 ระบบการกู้ข้อมูลอัตโนมัติ ระบบกู้ ความเสียหาย (Recovery System)

ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ได้เตรียมการรองรับการทำการประมวลผลกลุ่มงาน(Transaction Processing) โดยกลุ่มงาน(Transaction) นี้มีความหมายถึงกลุ่มของงานในทางตรรกะ (Logical Unit of Work) ซึ่งยอมให้มีการละเมิดกฎสำหรับควบคุมความถูกต้องของข้อมูลได้ภายในกลุ่มของงานและการทำงานทั้งหมดในกลุ่มงานนี้ ถ้าทำเสร็จตามความต้องการจะต้องเสร็จด้วยกันทั้งหมด ถ้าเกิดความบกพร่องขึ้นภายในกลุ่มงานจะต้องยกเลิกทั้งหมดโดยที่ Commit คือคำสั่งยืนยันการทำงานสำเร็จ ส่วน Rollback คือคำสั่งยกเลิกการทำงานของกลุ่มงานนั้น ๆ ดังนั้นถ้า commit แล้วระบบเกิดความเสียหายขึ้นระบบการจัดการฐานข้อมูลเชิงสัมพันธ์จะรับรองว่าข้อมูลที่ commit แล้วจะไม่สูญหายไปเ็นทางกลับกันกรณีของ Rollback และการทำงานที่ยังไม่ commit นั้นระบบการจัดการฐานข้อมูลเชิงสัมพันธ์จะรับรองว่าข้อมูลเดิมก่อนการทำงานของกลุ่มงานยังไม่สูญหายไปไหน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.3.6 การดูแลผู้ใช้หลายคนให้สามารถทำงานได้พร้อม ๆ กัน (concurrency control)

ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ จะมีการรองรับการทำงานพร้อมกันของผู้ใช้หลายคนได้โดยอาศัยอาจมีปัญหาเกิดขึ้นได้ 3 ประเภทใหญ่ ๆ คือ

1. ปัญหาการสูญเสียการเปลี่ยนแปลงข้อมูล เกิดจากการอ่านข้อมูลตัวเดียวกันไปใช้งาน โดยที่มีการเปลี่ยนแปลงข้อมูลตัวนั้นในเวลาที่ไม่พร้อมกัน

2. ปัญหาที่เกิดจากการยกเลิกการเปลี่ยนแปลงข้อมูล เกิดจากการมีการนำข้อมูลไปใช้ใน ขณะที่การประมวลผลกลุ่มงาน ยังไม่เสร็จสิ้นและมีการยกเลิกการประมวลผลกลุ่มนั้น

3. ปัญหาที่เกิดจากการขัดแย้งของการทำงาน คือมีการนำข้อมูลไปประมวลผล และมีการยอมรับความเปลี่ยนแปลงนั้นโดยที่การเปลี่ยนแปลงที่เกิดขึ้นมีผลต่ออีกการทำงานหนึ่ง ซึ่งปัญหาต่าง ๆ เหล่านี้สามารถป้องกันได้โดยวิธีการจองข้อมูล (Lock) ก่อนการใช้งาน ซึ่งการจองข้อมูลนี้มีอยู่ 2 แนวทางคือ

- จองข้อมูลแบบแชร์ (Shared-mode Lock) หมายถึงถ้ากลุ่มงาน (Transaction)  $T_i$  จองแบบแชร์ (Shared-mode Lock แทนด้วย S) บนข้อมูล Q ดังนั้น  $T_i$  สามารถอ่านข้อมูลจาก Q ได้โดยไม่สามารถเขียนลง Q ได้ซึ่งกลุ่มงาน (Transaction) อื่นที่เข้ามาใช้งาน Q หลังจากนี้ก็ยังสามารถอ่านข้อมูลจาก Q ได้อย่างเดียวเท่านั้น

- จองข้อมูลแบบพิเศษ (Exclusive-mode-Lock) หมายถึงถ้ากลุ่มงาน (Transaction)  $T_i$  จองแบบพิเศษ (Exclusive-mode Lock แทนด้วย X) บนข้อมูล Q ดังนั้น  $T_i$  สามารถอ่านและเขียนข้อมูล Q โดยมีกลุ่มงาน (Transaction) อื่นที่เข้ามาใช้งาน Q หลังจากนั้นจะต้องรอให้เลิกการจอง (Unlock) ก่อนจึงจะสามารถใช้งานข้อมูล Q ได้

## 6.4 รูปแบบหลายมิติ ( Multidimension Model)

Multidimensional Model เป็นเทคนิคที่พัฒนาขึ้นมาเพื่อโฆษณาขาย Application สำหรับการใช้งานข้อมูล ซึ่ง Multidimensional Model ไม่ได้เริ่มต้นที่การออกแบบข้อมูลเบื้องต้น แต่ได้มีการนำฐานข้อมูลที่จัดเก็บในรูปของ Relational มาประยุกต์ใช้งาน Transaction ให้เกิดประโยชน์สูงสุด โดยเน้นที่การรวมข้อมูล ( data integration ) , การจัดการข้อมูล ( Manipulate ) และการวิเคราะห์ ( analysis ) ซึ่ง Multidimensional Model จะช่วยให้การใช้งาน relational database มีความสมบูรณ์ยิ่งขึ้น

### 6.4.1 ความหมายของ Multidimensional database

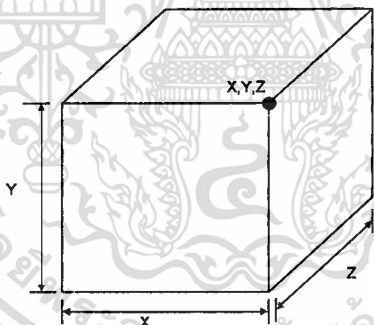
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Multidimensional database คือ กลุ่มของ Cell ข้อมูลที่จัดการเป็นมิติ เช่น ข้อมูล spreadsheet มี 2 มิติ ประกอบด้วย cell ข้อมูลที่เป็น แถว , คอลัมน์ แต่ละส่วนเรียกว่ามิติ อาจมีมากกว่า 3 มิติ สามารถมองแบบ Visualize เป็น Cube ในแต่ละ Dimension รวมทั้งเป็น Slice หรือ Dice, Drill down , Rotate เป็นต้น ในอนาคตเทคโนโลยี Multidimensional จะได้รับความนิยมเพราะช่วยกระจาย information ถึงผู้ใช้ ได้ง่ายและสะดวก และยังสนับสนุน การการตัดสินใจของผู้บริหารองค์กรได้เป็นอย่างดี

#### 6.4.2 คุณสมบัติของรูปแบบหลายมิติ (Multidimensional Model)

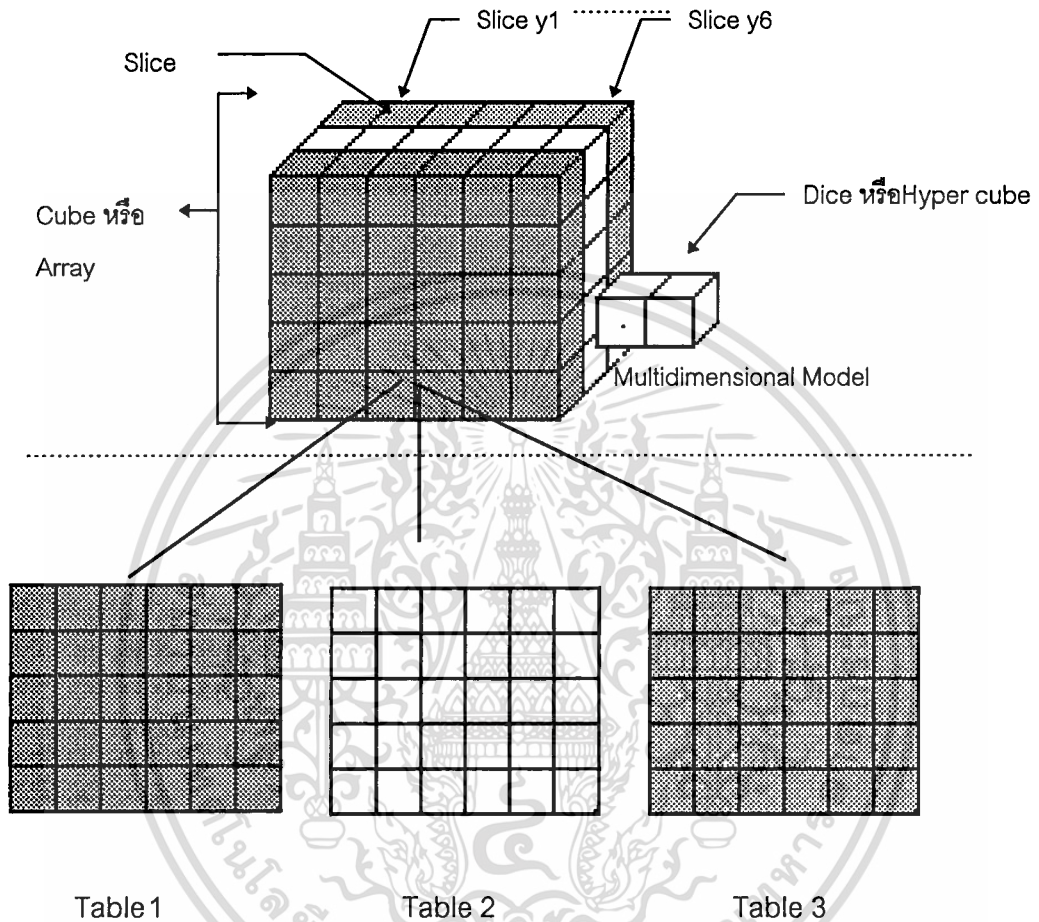
##### 6.4.2.1 โครงสร้างข้อมูล (Data Structure)

ข้อมูลในมุมมองของฐานข้อมูลแบบหลายมิติ เป็นฐานข้อมูลที่อยู่ในรูปของ Array หลายมิติ ซึ่งมีชื่อเรียกแตกต่างกันไป เช่น Cube , Cross-tabulation หรือลูกบาศก์ การกำหนดตำแหน่งถูกกำหนดเป็น Logical Intersection ซึ่งเป็นจุด Coordinate สำหรับแสดงถึง ข้อมูลแต่ละ Tuple ดังรูปที่ 6.6 โดยเรียกจุดที่ตัดกันนี้ว่า Cell หรือ Slice หรือ Element ของ Dimension ซึ่งในที่นี่จะกล่าวถึง Dimension ในความหมายเดียวกันกับ Entity ในระบบฐานข้อมูลเชิงสัมพันธ์ จากรูปที่ แสดงลักษณะข้อมูลที่อยู่ในรูปของฐานข้อมูลหลายมิติ



รูปที่ 6.6 แสดงโครงสร้างข้อมูลขนาด  $1 \times 1 \times 1$

จากรูปนี้มีจุดที่แสดงถึงตำแหน่งเพียงจุดเดียว คือ X,Y,Z เพราะลูกบาศก์มีขนาด  $1 \times 1 \times 1$  ในกรณีที่ข้อมูลมีจำนวนมากก็สามารถเพิ่มตำแหน่งหรือจุด Coordinate ได้ตามจำนวนที่ต้องการ ซึ่งสามารถแสดงโครงสร้างฐานข้อมูลในรูปแบบหลายมิติได้ดังรูปที่ 6.7



รูปที่ 6.7 แสดงโครงสร้างของ Multidimensional Model

จากโครงสร้างของ Multidimensional Model สามารถที่จะเปรียบเทียบคำศัพท์ที่ใช้ในเชิงวิชาการกับคำศัพท์ที่ใช้ทั่วไปในเชิงธุรกิจ

ศัพท์เชิงวิชาการ	รูปแบบเชิงสัมพันธ์	รูปแบบหลายมิติ
Relational	table	Table(Fact table ,Dimension table)
Tuple	row หรือ Record	Hyper cube หรือ Cell,block,dice
Attribute	column หรือ Field	Slice
cardinality	จำนวนของ row	จำนวนของ cell ทั้งหมด
primary key	column 1 หรือมากกว่า	จุด Coordinate

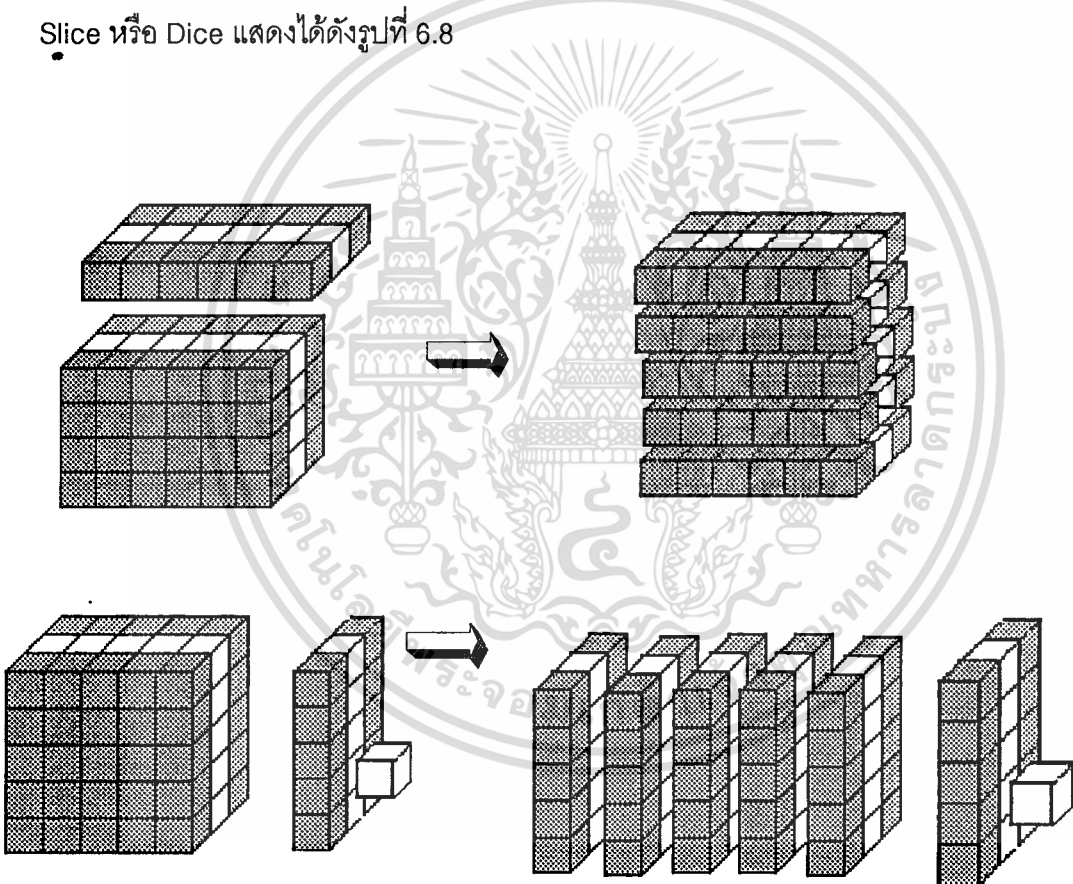
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ระบุแล้วได้ค่าออกมา ของแต่ละ Cell

1 Row

Domain ค่าของ ข้อมูลที่เป็นไปได้ จำนวน Slice ตามแนว  
ทั้งหมดในแต่ละคอลัมน์ แนวแกน x หรือ y หรือ Z

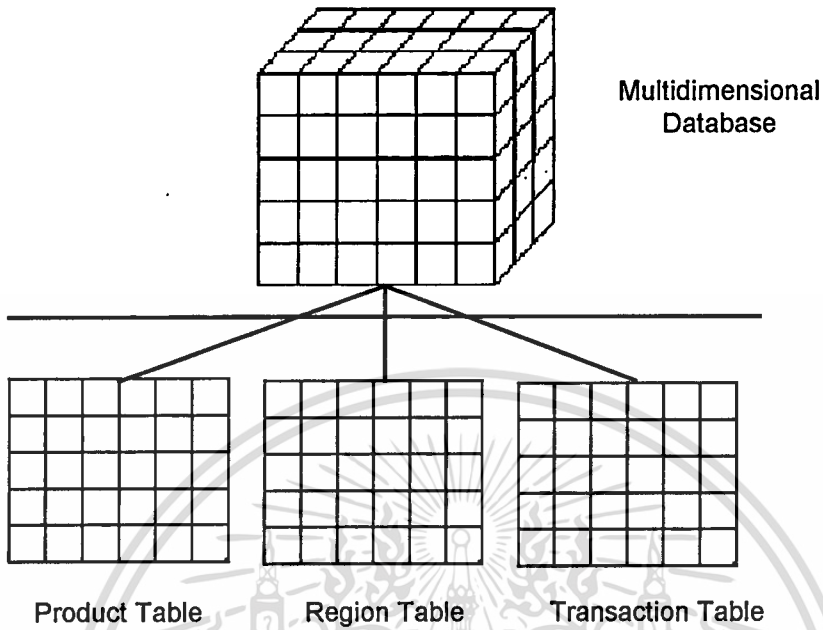
จากโครงสร้างข้อมูลของ Multidimension Model สามารถที่จะมองข้อมูลได้หลายมิตินั้น หมายถึงการเลือกข้อมูล(Attriubute) ที่อยู่ในแต่ละตารางและนำมาเปรียบเทียบหรือที่เรียกว่า การ Slice หรือ Dice แสดงได้ดังรูปที่ 6.8



รูปที่ 6.8 แสดงการ slice ข้อมูลใน cude

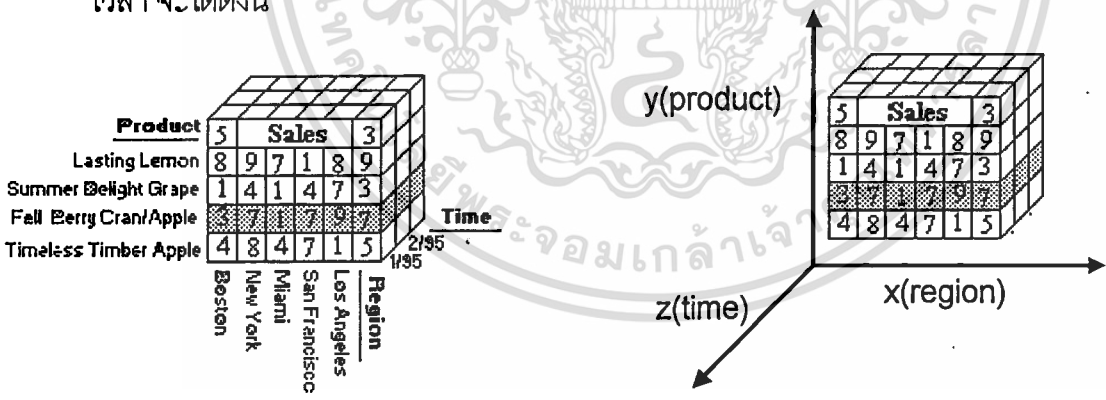
จากรูปเป็นแนวคิดในการมองข้อมูลใน Multidimensional Model โดยผู้ใช้สามารถเลือกมุมมองได้หลายมุมมอง นั้นหมายถึงการเลือกดูข้อมูลตามหัวข้อที่ต้องการ หรือ Attribute ข้อมูลหลักในรูปแบบหลายมิติคือฐานข้อมูลที่เป็น Transaction ใน Data Warehouse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.9 แสดงตารางก่อนนำเข้าสู่ Multidimensional Model

จากรูปมีจำนวนตารางที่อยู่ในรูปแบบ relational จำนวน 3 ตาราง คือ Product, Region, Transaction นำทั้ง 3 ตารางเข้าสู่รูปแบบหลายมิติเพื่อดูยอดขายโดยเพิ่มแกนเวลา จะได้ดังนี้



รูปที่ 6.10 แสดงข้อมูลใน Multidimensional Model

จาก Multidimension Model สามารถแทนด้วยกราฟได้ดังรูปด้านขวามือ คือแกน แกน x แทนด้วย Region แกน Y แทนด้วย Product แกน Z แทนด้วย Time และค่าที่อยู่ใน cell คือยอดขาย (Sale) ของ สินค้าแต่ละตัว แต่ละภูมิภาค และแต่ละปี

จากตัวอย่างการเป็นมุมมองข้อมูลในรูปแบบ 3 มิติ ซึ่งประกอบด้วย Product, Region, และ Time ซึ่งแต่ละมิติ จะมีจำนวนสมาชิกแตกต่างกัน โดย Product มีจำนวนสมาชิกเท่ากับ 5 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Region มีจำนวนสมาชิกเท่ากับ 6 และ Time มีจำนวนสมาชิก เท่ากับ 2 ซึ่งสามารถแทนด้วย Array ขนาด 5 x 6 x 2 เพราะฉะนั้นจึงเป็น Array ที่มีจำนวน Cell เท่ากับ 90 โดยค่าที่อยู่ใน cell แทนจำนวนยอดขาย

ถ้าแทนจำนวน cell ด้วย จุด Coordinate X,Y,Z สามารถที่จะแทนค่าได้ดังนี้

X (REGION)	Y (PRODUCT)	Z (TIME)
x1 (Boston)	y1(timber Apple)	z1(1/95)
x1(Boston)	y1(timber Apple)	z2 (2/95)
x1 (Boston)	y2(Berry cran Apple)	z1(1/95)
x1(Boston)	y2(Berry cran Apple)	z2 (2/95)
x1 (Boston)	y3(Delight Grape)	z1(1/95)
x1(Boston)	y3(Delight Grape)	z2 (2/95)
x1 (Boston)	..	..
x1(Boston)	y6...	z1
x1(Boston)	y6....	z2
...	....	...
...	...	....
...	...	...
...	..	...
x6	y6	z1(1/95)
x6	y6	z2 (2/95)

ตารางที่ 6.1 แสดงการแทนข้อมูล ด้วย จุด Coordinate ในรูป ตาราง

จากจำนวนจุด Coordinate ทั้งหมด 90 จุด หรือ 90 Cell หรือ 90 Tuple นั้นหมายถึงจำนวน array ที่สามารถบรรจุข้อมูลได้ทั้งหมด 90 เรคคอร์ด โดยไม่ซ้ำกันเลย(Atomic) ข้อสังเกตข้อมูลที่นำเข้าสู่ ระบบฐานข้อมูลหลายมิติเป็นข้อมูลที่อยู๋ในรูปที่ยังไม่ผ่านการทำ Normalization(1 NF) เพราะฉะนั้น จะมีข้อมูลที่เป็น redundant เป็นจำนวนมาก ทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลใน Multidimensional Model ไม่เหมาะที่จะทำการ Update แต่เหมาะสำหรับการทำ Query มากกว่า

### ลักษณะของข้อมูลใน Multidimensional Database Model

1. แนวแกน X,Y,Z เป็น Dimension หรือ Entity, หรือ Table ในระบบฐานข้อมูลเชิงสัมพันธ์
2. จุดตัดระหว่างแนวแกน X,Y,Z เรียกว่า Cell แทน 1 Tuple ในระบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งมีจำนวนค่าของ Attribute ที่เป็นไปได้ทั้งหมด คือ  $x_1-x_6, y_1-y_6, z_1-z_2$

(จากตาราง 6.2 )

3. Sparse : หมายถึง cell ที่ค่าใน Array หลาย ๆ ค่าเป็น Cell ว่าง หรือทุก ๆ ค่า เป็น Cell ว่าง หรือเรียกอีกอย่างหนึ่งว่า Nondense

4. Dense : หมายถึง Cell ที่ค่าใน Array ไม่เป็นค่าว่าง

### การแทนข้อมูลใน Array หลายมิติ ( Multidimensional Arrays )

การใช้ Array สำหรับแทนข้อมูลสามารถกำหนดได้มากกว่า 2 มิติ อาจเป็น 3,4... มิติ แล้วแต่ความต้องการของผู้ใช้ว่าต้องการแทนที่จำนวนข้อมูลกี่ Block หรือกี่ Cube การคำนวณหาจำนวน Cell คิด จำนวน แกน X , Y , Z และนำค่าที่เป็นไปได้ (Domain) ของแต่ละแกนคูณกัน เช่น ภาค มีทั้งหมด 4 ภาค แต่ละภาคมีสินค้าที่ขายภาคละ 4 รายการ และขายในปี 2539 ถึง 2540 เพราะฉะนั้นสามารถคำนวณจำนวนเซลล์ได้คือ  $4 \times 4 \times 2 = 32$  Cell ใน Array เป็นการจองพื้นที่สำหรับเก็บข้อมูล 1 Tuple ของ Transaction ใน Relational database เพราะฉะนั้น ถ้าข้อมูลมีความลึก ( Level ) หลาย ๆ ระดับก็จะต้องมีการกำหนด Array หลายมิติ สำหรับรองรับปริมาณข้อมูล แต่ละ Cell จะใช้เนื้อที่หน่วยความจำเท่าไรขึ้นอยู่กับขนาดของข้อมูลที่กำหนดตอนเริ่มต้น จะเห็นว่าถ้าข้อมูลมีจำนวนมาก ก็จำเป็นต้องจองพื้นที่สำหรับข้อมูลมากตามไปด้วย ถึงแม้ว่าข้อมูลจะมีจำนวนมากแต่ไม่เป็นปัญหาสำหรับการเข้าถึงข้อมูลทุก ๆ Element เพราะ การ Access ข้อมูลใด ๆ ที่เก็บในรูปแบบ Array จะใช้ Offset ของ Array เพื่ออ้างถึงข้อมูลนั้น ดังนั้นความยาวของ Path ที่จะไปถึงข้อมูลใด ๆ จึงเท่ากัน

#### 6.4.2.2 กฎควบคุมความถูกต้องของข้อมูล(Data Integrity Constraints)

เนื่องจากยังไม่มีมาตรฐานใดที่กำหนด กฎควบคุมความถูกต้องของข้อมูลสำหรับฐานข้อมูลแบบหลายมิติ ดังนั้นการจัดการเรื่องนี้ยังอ้างอิงถึงระบบ Relational Database กล่าวคือ

ฐานข้อมูลเบื้องต้น(Source )ไม่ว่าจะเป็นฐานข้อมูลปฏิบัติงาน (Operational) หรือฐานข้อมูลในคลังข้อมูล (Data Warehouse) ที่นำเข้าสู่ Multidimensional Model เป็นฐานข้อมูลที่อยู่ในรูปแบบเชิงสัมพันธ์ซึ่งมีกฎควบคุมความถูกต้องของข้อมูล(Data integrity Constraints) อยู่แล้ว

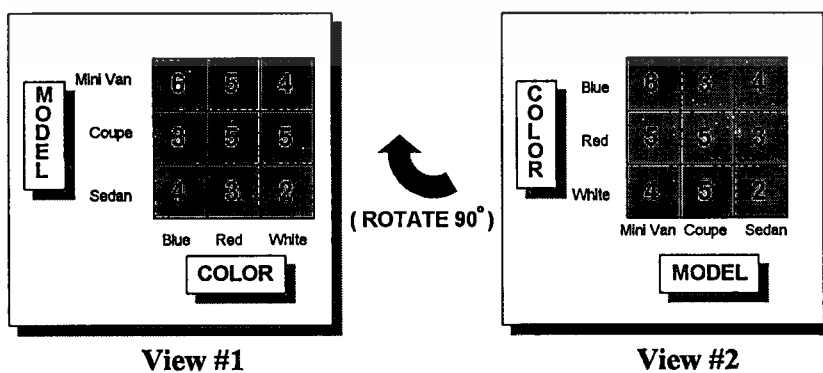
#### 6.4.2.3 การจัดการข้อมูล (Data Manipulate)

##### 1. การหมุน (Rotation)

การจัดเรียงของข้อมูลในตารางหลายมิติมีประโยชน์ ซ่อนอยู่ในตัวเองอยู่แล้ว ต่างไปจากมุมมองของตารางเชิงสัมพันธ์ มุมมองของตารางหลายมิติแสดงถึงมิติ, ตำแหน่งตามแนวของมิติ, การตัดซึ่งกันและกัน และ ข้อมูลในแต่ละช่องได้อย่างชัดเจน ที่จริงแล้วการมองตารางหลายมิติคล้ายกับการมอง spreadsheet

เพื่อให้เข้าใจถึงความมีประโยชน์ของการมองตารางหลายมิติเปรียบเทียบกับการมองตารางเชิงสัมพันธ์ ลองมาพิจารณาว่าผู้ใช้นั้นสร้างมุมมองของข้อมูลแบบ spreadsheet ป้อยแค่ไหน กับพิจารณาว่าผู้ใช้เรียกค้นข้อมูลที่อยู่ในตารางเชิงสัมพันธ์โดยตรงนานครั้งเท่านั้น ประโยชน์ของการมองข้อมูลแบบตารางหลายมิติเป็นข้อได้เปรียบพื้นฐานของฐานข้อมูลหลายมิติ พิจารณาบทนำเรื่องตารางสองมิติที่มุมมองคือปริมาณการขายจำแนกตาม MODEL และตาม COLOR ข้อมูลสามารถมองเข้าใจได้ในทันที ตารางหลายมิตียอมให้ผู้ใช้ค้นหาข้อมูลโดยตรง เหมือนกับที่ผู้ใช้ค้นหาจาก spreadsheet แต่ในตารางเชิงสัมพันธ์ การมองตามโครงสร้างดังกล่าว ทำโดยใช้การสอบถามที่ซับซ้อน นอกจากนี้หากต้องการการมองปริมาณการขายตาม COLOR และ MODEL ในฐานข้อมูลเชิงสัมพันธ์จะต้องใช้การสอบถามที่ซับซ้อนร่วมกับต้องมีการเรียงลำดับข้อมูล แต่ในฐานข้อมูลหลายมิติ มุมมองใหม่นี้จะทำโดยการหมุนจากมุมมองเดิม นั่นคือมุมมองเดิมจะถูกหมุนตามเข็มนาฬิกาไป 90 องศาโดยที่ไม่ต้องทำการจัดเรียงข้อมูลใหม่แต่อย่างใด

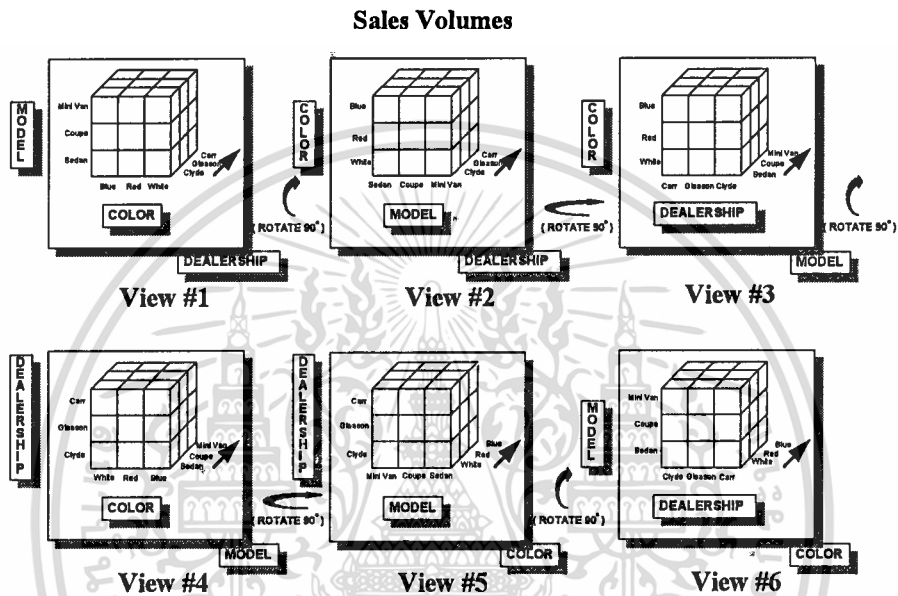
#### Sales Volumes



รูปที่ 6.11 แสดงการหมุนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางสองมิติทั้งสองตารางมีมุมมองที่เป็นไปได้สองแบบคือตาม MODEL x COLOR หรือ COLOR x MODEL ทั้งสองกรณีคือการตัดซึ่งกันและกันของมิติ 2 มิติในปริมาณการขาย ส่วนตาราง 3 มิติที่มี MODEL, COLOR และ DEALERSHIP นั้นมี 6 มุมมองที่เป็นไปได้ ที่ผู้ใช้จะสามารถเรียกดูได้ ดังจะเห็นได้ดังรูปที่ 6.12



รูปที่ 6.12 แสดงถึงมุมมองทั้ง 6 ด้านที่สามารถเข้าถึงข้อมูลโดยการหมุนหลายมิติ

ความสามารถที่ง่ายต่อการเข้าถึงมุมมองต่างๆ ของข้อมูลหลายมิตินั้น ทำให้เกิดความยืดหยุ่นต่อการที่ผู้ใช้ประเภทต่างๆ จะเลือกถามคำถามหรือต้องการข้อมูลที่หลากหลายได้มากขึ้น จากรูปมุมมองทั้ง 6 ด้าน สามารถที่จะเข้าถึงได้โดยการหมุนอย่างง่าย ๆ เพราะว่าฐานข้อมูลหลายมิติเก็บข้อมูลที่อยู่ในรูปแบบที่ง่ายต่อการหมุนโดยไม่ต้องมีการจัดเรียงข้อมูลเพิ่มเติมเลย การหมุนสามารถอ้างถึงในอีกชื่อหนึ่งว่า data slicing เพราะการหมุนแต่ละครั้งนี้จะทำให้เกิด slice ที่ต่างกัน (slice คือ ตาราง 2 มิติ)

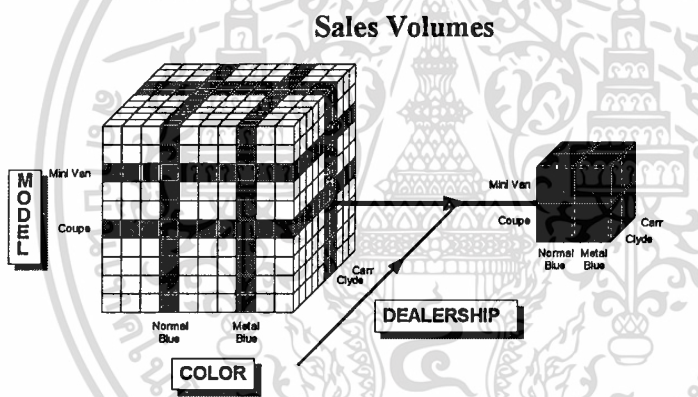
จะเห็นได้ว่า จำนวนมุมมองที่เป็นไปได้จะเพิ่มขึ้นแบบ exponential เมื่อเทียบกับจำนวนมิติ ตาราง 2 มิติจะมี 2 มุมมอง ตาราง 3 มิติจะมี 6 มุมมอง ตาราง 4 มิติ จะมี 24 มุมมอง ตาราง 5 มิติจะมี 120 มุมมอง ฐานข้อมูลหลายมิตินี้จะทำให้การเข้าถึงในแต่ละมุมมองนั้นไม่ต้องทำการเรียงข้อมูลให้แตกต่างจากรูปแบบเดิมเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 Ranging การกำหนดขอบเขต (Slice and Dice)

ฐานข้อมูลหลายมิติที่นำไปประยุกต์ใช้ได้ ๗ นั้นสามารถยอมให้ผู้ใช้เลือกดูข้อมูลบางตัว หรือดูข้อมูลกลุ่มย่อยได้ง่ายและเร็ว จากแบบจำลอง 3 มิติของข้อมูลปริมาณการขาย ผู้ใช้อาจจะต้องการเปรียบเทียบปริมาณการขายของผลิตภัณฑ์ ที่ทำด้วยสี NORMAL BLUE หรือ METAL BLUE โดยที่ผู้ใช้รู้ว่าผลิตภัณฑ์สองประเภทที่ได้รับการทำสีนี้คือ SPORTS COUPE และ MINI VAN และผู้ใช้รู้ว่ามียังสอง Dealership เท่านั้นที่มีการจำหน่ายผลิตภัณฑ์นี้คือ CARR และ CLYDE โดยการดำเนินการกำหนดขอบเขตผู้ใช้เลือกข้อมูลในแต่ละมิติดังนี้

- มิติของ MODEL ได้ SPORTS COUPE, MINI VAN
- มิติของ Dealership ได้ CARR, CLYDE
- มิติของ COLOR ได้ METAL BLUE, NORMAL BLUE



รูปที่ 6.13 การมองข้อมูลเป็นกลุ่มย่อย

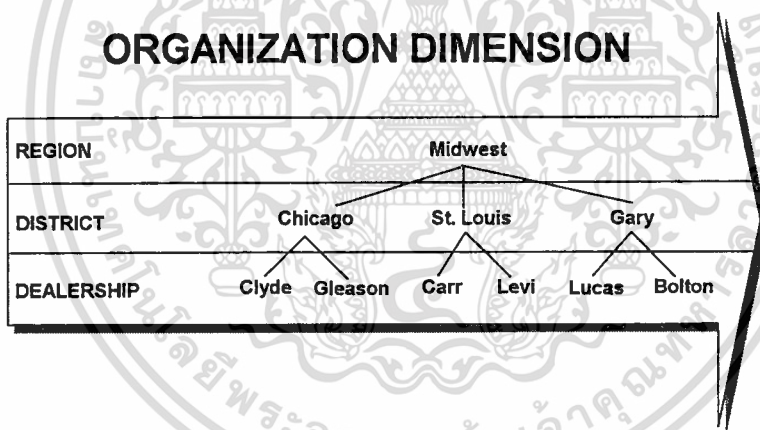
ตารางที่ลดลงนี้สามารถทำการหมุนและการคำนวณได้เหมือนตารางใหญ่ทุกประการ ในฐานข้อมูลเชิงสัมพันธ์การกำหนดขอบเขตนี้ ทำได้โดยการใช้การสอบถามที่ซับซ้อน การกำหนดขอบเขตนี้เรียกได้อีกแบบว่า Data dicing เพราะว่าข้อมูลถูกจำกัดขอบเขตเป็นกลุ่มย่อยในฐานข้อมูลหลายมิติการดำเนินการเช่นนี้มีประสิทธิภาพดีกว่าการใช้การสอบถามในแบบฐานข้อมูลเชิงสัมพันธ์

### 3.3 Hierarchies, Roll-Ups & Drill Downs

จากที่ได้กล่าวมาผู้ใช้มักจะต้องการหลวมมองของข้อมูลชุดเดียวกันอาจเป็นต่างมุมมองออกไปเลยเช่น การดูปริมาณการขายตาม MODEL กับตาม Dealership หรืออีกแบบหนึ่งก็คือเป็นมุมมองเดียวกันแต่ต่างระดับ(level) เช่น การดูปริมาณการขายจำแนกตามเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

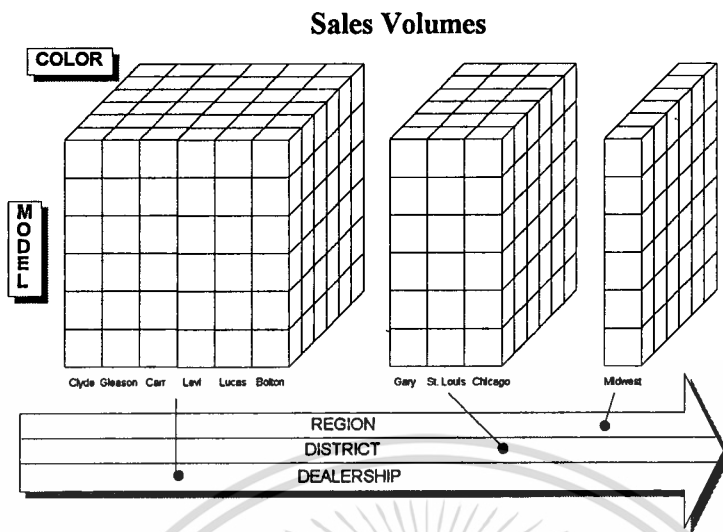
Dealership กับ จำแนกตาม District ดังตัวอย่างต่อไป นี้ สองมุมมองดังกล่าวนี้เกี่ยวข้องกันในระดับของโครงสร้างการขาย ซึ่งเป็นความสัมพันธ์ โดยธรรมชาติระหว่างระดับ District กับระดับ Dealership อยู่แล้ว ปริมาณการขายของ Dealership ทั้งหมดใน District หนึ่ง ๆ รวมกันเป็นปริมาณการขายของแต่ละ District ยังมีอีกหลายแบบที่เกิดขึ้นในโลกธุรกิจที่เป็นมุมมองที่ต่างกัน แต่มีความใกล้เคียงกันมากฐานข้อมูลหลายมิติได้ถูกออกแบบมาเป็นพิเศษเพื่อจัดการความสัมพันธ์ดังกล่าวนี้ ถึงแม้ว่ามันเป็นไปได้ที่จะสร้างมิติที่แยกและไม่ขึ้นแก่กันระหว่าง Dealership และ District แต่วิธีการที่มีประสิทธิภาพมากกว่าคือการกำหนดให้ทั้งสองความสัมพันธ์อยู่ในมิติเดียวกัน การรวมกันจะถูกมองเป็นส่วนหนึ่งของมิติเดียวกัน จากตัวอย่าง District เป็นระดับที่เหนือกว่า แต่อยู่ภายใต้มิติ

ที่เรียกว่า Organization เมื่อเรานำข้อมูลของ Dealership มาสัมพันธ์กับข้อมูล District เราจะกำหนดโครงสร้างระดับภายใต้มิติ Organization ฐานข้อมูลหลายมิติที่นำมาประยุกต์ใช้ที่ดีนั้นสามารถกำหนดระดับต่างๆภายในมิติได้เป็นระดับความสัมพันธ์ดังตัวอย่าง



รูปที่ 6.14 แสดงระดับความสัมพันธ์ของข้อมูล

ในฐานข้อมูลเชิงสัมพันธ์ จะทำการสร้างเขตข้อมูลสำหรับแต่ละระดับบน hierarchy หรือแยกเป็นตาราง 2 ตารางที่กำหนดความสัมพันธ์ดังกล่าว สำหรับผู้ใช้ความสามารถที่กำหนด hierarchy ยินยอมให้ทำการดูแลบำรุงรักษาข้อมูลและวิเคราะห์รายละเอียดของแต่ละระดับได้ง่ายและรวดเร็วการย้ายระดับใน hierarchy จะกระทำโดยวิธีการที่เรียกว่า roll-up และ drill-down



รูปที่ 6.15 การแบ่งข้อมูลเป็นกลุ่ม

## 6.5 ระบบการจัดการฐานข้อมูลหลายมิติ Multidimensional Database (MDBMS)

ระบบการจัดการฐานข้อมูลหลายมิติเป็นซอฟต์แวร์ (Software) ที่พัฒนาขึ้นมาสำหรับจัดการกับข้อมูลใน Data Warehouse หรือ ในฐานปฏิบัติงาน Operational ให้สามารถใช้ข้อมูลเหล่านั้นเกิดประโยชน์สูงสุด โดยมี Application สำหรับกำหนดโครงสร้างของข้อมูลในระบบต่างๆ ตามสถาปัตยกรรมมาตรฐานของระบบข้อมูล ระบบการจัดการฐานข้อมูลสามารถสนับสนุนการทำงานบนฐานข้อมูลได้ดังนี้

### 6.5.1 การจัดการกับปริมาณข้อมูลที่มีปริมาณมาก ๆ

ระบบการจัดการฐานข้อมูลหลายมิติ สามารถที่จะจัดการกับข้อมูลที่มีปริมาณมาก ๆ ให้มีประสิทธิภาพในการเข้าถึงได้ ซึ่งการพิจารณาหรือประเมินผลความสามารถของ MDBMS มี 2 วิธีดังนี้

1. พิจารณาจำนวนของ "Logical" cell หมายถึง ความสามารถในการกำหนด Address สำหรับนวน cell ได้สูงสุดที่ cell โดยปกติมากกว่า 100 Trillion ( 100 ล้านล้าน ) cell หรือ Tuple (record)

2. พิจารณาจำนวนของ "Actual data" หมายถึง ข้อมูลที่เก็บจริง ๆ ในสื่อรวมทั้งชุดข้อมูล (database)ที่เป็น Sparsity ด้วย ในกรณีนี้ขึ้นอยู่กับสถาปัตยกรรมของ MDBMS บางตัวสามารถรองรับได้ถึง 100 gigabytes MDBMS เป็นทางเลือกที่ดีของ fron-end สำหรับการเข้าไปวิเคราะห์ข้อมูลใน data warehouse ซึ่งเป็นข้อมูลสรุป ( Summarize) จากฐานข้อมูลปฏิบัติงาน สามารถมองแบบ "drill through" ดูรายละเอียดของข้อมูลที่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.5.2 การจัดการกับเซลล์ ข้อมูลที่ว่าง Sparse Data Handling (Storage Techniques)

การเก็บข้อมูลใน Multidimensional Model อยู่ในรูปแบบของ Cell หรืออาจเรียกว่า Mini-cube ซึ่งแต่ละ Cell ก็จะมีค่า 1 ค่า ถ้าใน Cell นั้นมีค่าข้อมูลเป็นจริง หมายถึงมีข้อมูลการเก็บก็ไม่ใช่ปัญหา แต่ถ้าค่าของข้อมูลเป็นเท็จ หมายถึงไม่มีข้อมูล หรือ cell Sparse ทำให้เกิดการสูญเสียพื้นที่ในการจัดเก็บข้อมูล (wasted space) เพราะฉะนั้นจึงมีวิธีการจัดการกับข้อมูลที่เป็น Sparse โดยนำเทคนิคของการบีบอัดข้อมูลเข้ามาช่วย หรือที่เรียกว่า (Data Compression) เพื่อให้ Block มีขนาดเล็กลงก่อนที่จะบันทึกข้อมูลลงสู่ disk เป็นการประหยัดพื้นที่ในการจัดเก็บเพื่อผู้ใช้ เรียกดูข้อมูล (Retrieved) ระบบก็จะทำงาน decompressed ข้อมูลที่บีบอัดโดยอัตโนมัติ ทั้งนี้ไม่ใช่วิธีการที่เป็นมาตรฐานขึ้นอยู่กับเทคนิคของแต่ละ product ว่าจะมีวิธีการจัดการอย่างไร

### 6.5.3 การทำดัชนี (Indexing Techniques)

การค้นหาข้อมูลในฐานข้อมูลแบบเชิงสัมพันธ์ จะมี Key Field ที่เรียกว่า Primary key สำหรับช่วยในการค้นหาข้อมูล แต่เทคนิคในการเข้าถึงข้อมูลมีหลาย ๆ เทคนิคสำหรับเข้าถึงข้อมูล ไม่ว่าจะเป็น binary searches, hash searches, และ B-tree เป็นต้น สำหรับ MDBSs ใช้เทคนิคในการเข้าถึงข้อมูลที่เรียก "address hashing" ซึ่งจะเป็นการเข้าถึงข้อมูลโดยอ้างถึงจุด Coordinate ในการค้นหา หรือในทางคณิตศาสตร์เรียกหลักการนี้ว่า linear address (linear Search) แต่อย่างไรก็ตามในบางครั้งการเก็บข้อมูลที่เป็น actual cube ที่กระจัดกระจายไปทั่ว disk ทำให้เกิดปัญหา ในการกำหนด address สำหรับ virtual cube เพราะฉะนั้นจึงมีวิธีการจัดการโดย สร้างเป็น hash table สำหรับเก็บ address ของ actual cube เพื่อเป็นเส้นทางในการหาข้อมูลของ virtual cube

### 6.5.4 ความปลอดภัยของข้อมูล (data Security)

ระบบการจัดการฐานข้อมูลได้มี ระบบการสอบตามรายชื่อพร้อมทั้งรหัสลับของผู้เข้ามาใช้งานในระบบเพื่อให้เข้าทำงานในส่วนที่ผู้ใช้เกี่ยวข้องเท่านั้น

### 6.5.5 การจัดการมิติของเวลา (Time Dimension)

ระบบฐานข้อมูลหลายมิติจะมีวิธีการจัดการกับมิติของเวลาเป็นพิเศษ โดยเวลาจะถูกแบบระดับเป็น hierarchy ของวัน สัปดาห์ เดือน ปี และช่วงเวลาพิเศษอื่น ๆ เป็นมิติที่กำหนดไว้ล่วงหน้าเพื่อช่วยลดความยุ่งยากในการสร้างทุกครั้ง ซึ่งเหมาะกับระบบงานทั่ว ๆ ไป โดยเฉพาะงานด้านธุรกิจที่ต้องการวิเคราะห์ข้อมูลตามช่วงเวลา ในส่วนนี้เป็นส่วนที่เพิ่มขึ้นมาจากการจัดการฐานข้อมูลเชิงสัมพันธ์ เพราะฐานข้อมูลเชิงสัมพันธ์จะไม่มีวิธีการพิเศษสำหรับจัดการกับเรื่องเวลาได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากความสามารถของระบบการจัดการฐานข้อมูลหลายมิติจะเห็นการจัดการกับข้อมูลจะไม่ยุ่งยากเท่ากับระบบฐานข้อมูลเชิงสัมพันธ์ คือไม่ต้องมีการควบคุมการใช้งานพร้อม ๆ กัน เพราะข้อมูลในระบบการจัดการฐานข้อมูลเป็นข้อมูลที่ไม่มีการ Update คืออ่านได้อย่างเดียว ปัญหาต่าง ๆ ที่เกิดการ Update ก็จะไม่เกิดขึ้น เป็นต้นว่า ข้อมูลสูญหาย ข้อมูลถูกเปลี่ยนแปลงแก้ไข ข้อมูลมีความขัดแย้ง การใช้ข้อมูลก็เป็นแชร์เพื่ออ่านอย่างเดียว ไม่สามารถที่จะเขียนทับได้

## 6.6 ความสัมพันธ์ ระหว่าง Multidimensional database กับ data warehouse

ทั้ง datawarehouse และ Mutidimnsional database Technogy เป็นเทคนิคที่พัฒนาขึ้นมาเพื่อโฆษณาขาย Application สำหรับการใช้งาน.

ซึ่ง warehouse มีความสัมพันธ์กับ multidimension database ในแง่ของการใช้งาน ข้อมูลที่อยู่ในคลังข้อมูล data warehouse เป็น คลังสำหรับเก็บข้อมูล ส่วน multidimunsional database เป็นเหมือนผู้คอยกระจายข้อมูล ( retailer ) ไปยังผู้ใช้ ซึ่งทั้ง Multidimensional database technology และ data warehousing สามารถเพิ่มหรือขยายคุณภาพ , ความเร็วและประสิทธิภาพในการส่งข้อมูลไปยัง end users.

แต่อย่างไรก็ตามในการทำ data warehouse ควรที่จะรู้รูปแบบของฐานข้อมูล ซึ่งเทคโนโลยี Multidimensional มีทฤษฎีสำหรับการส่งข้อมูลอย่างเหมาะสม แต่อย่างไรก็ตามเทคโนโลยีฐานข้อมูลแบบหลายมิติ ไม่มีความเหมาะสมสำหรับระบบการประมวลผล Transaction ที่เป็น Online แต่มีความเหมาะสมกับ data warehousing มากกว่า เพราะว่าฐานข้อมูลแบบหลายมิติ ( Multidimensional database ) ถูกออกแบบมาเพื่อกระจายข้อมูลไปยัง end-user ที่ต้องการโดยเฉพาะ

## 6.7 เปรียบเทียบระหว่าง Relational Database กับ Data Warehouse

Data Warehouse เมื่อมองตัวมันเองเฉิน ๆ ก็คือฐานข้อมูลชนิดหนึ่งแต่สร้างขึ้นมาจากวัตถุประสงค์ต่างออกไป สถาบันตยกรรมและข้อแม้ในการทำงานก็ไม่เหมือนกัน สิ่งที่แตกต่างอย่างเห็นได้ชัดคือข้อมูลในฐานข้อมูลคือข้อมูลดิบที่เราใส่ลงไปเรื่อย ๆ หรือที่เรียกกันว่า Operational Database เช่นรายชื่อและราคาสินค้า รายชื่อลูกค้าและสินค้าที่สั่ง แต่ Data Warehouse จะเป็นข้อมูลที่สรุป โดยนำข้อมูลมาประมวลผลใหม่เพื่อหาแนวโน้มและวิเคราะห์สิ่งที่จะเกิดขึ้นในอนาคต ซึ่งแสดงความแตกต่างระหว่างข้อมูลใน DataBase กับ Data Warehouse ได้ดังตารางที่

6.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Database กับ Data Warehouse

	Operational Database	Data Warehouse
การใช้งาน	ข้อมูล	วิเคราะห์
ลักษณะการจัดการข้อมูล	แอปพลิเคชันกับแอปพลิเคชัน (Application Oriented)	ตามหัวข้อเรื่องที่ต้องการ (Subject Oriented)
โครงสร้างข้อมูล	ซับซ้อน แล้วแต่เครื่องและการ คำนวณ แต่เป็นรูปแบบชัดเจน ประมวลผลซ้ำเรื่องเดิม	มีโครงสร้างไม่แน่นอน ประมวล ผลแบบวิเคราะห์แต่่าง่าย เหมาะ กับองค์กร
เนื้อหาและช่วงเวลา การอัปเดต	ปัจจุบัน เป็นเรื่อง ๆ ไป มีจำนวนน้อย และทำเป็นประจำ	อดีตและปัจจุบัน แล้วแต่สถานการณ์และความ ต้องการ ไม่มีการอัปเดตโดย ตรง
การเคลื่อนไหวของข้อมูล เวลาในการทำงาน	ตลอดเวลา เสี้ยววินาทีถึง 2-3 วินาที	คงที่จนกว่าจะปรับปรุงใหม่ ไม่แน่นอน ตั้งแต่หลายวินาทีจน ถึงนาที
ความแน่นอนในการใช้ ข้อมูล	แน่นอน	ไม่แน่นอน
แหล่งข้อมูล	ภายในองค์กร	ทั้งภายในและภายนอกองค์กร
ขนาดของข้อมูล	กิกะไบต์	กิกะไบต์ถึงเทราไบต์

ตารางที่ 6.2 ตารางเปรียบเทียบ Database กับ Data warehouse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สรุป

ที่ผ่านมาเทคโนโลยีสารสนเทศเลือกใช้ Relational database สำหรับข้อมูลที่เป็น report เพื่อเก็บข้อมูล Transaction ที่มีปริมาณมาก ๆ แต่ Relational database ไม่ได้ออกแบบเพื่อรองรับการวิเคราะห์ข้อมูลที่มีปริมาณมาก ๆ โดย เพราะฉะนั้นจึงเกิด Multidimensional database ขึ้นเพื่อแก้จุดอ่อนของฐานข้อมูลเชิงสัมพันธ์ในกรณีการวิเคราะห์ข้อมูล โดยออกแบบ Multidimensional Database ให้เหมาะสำหรับการวิเคราะห์ข้อมูลปริมาณมาก ๆ ซึ่ง Multidimensional database มีลักษณะที่สำคัญต่อไปนี้

1. ดึงข้อมูลจากหลายที่มาวมกันโดยเก็บไว้อย่างเป็นระบบ
2. โครงสร้างข้อมูลออกแบบสำหรับผู้ใช้งานเข้าใจได้ง่ายและเหมาะกับงานธุรกิจ
3. Datastructure Application ที่เหมาะสำหรับการวิเคราะห์ข้อมูลอย่างมีประสิทธิภาพ
4. ง่ายต่อการวิเคราะห์ข้อมูลที่มีปริมาณมาก ๆ



## บทที่ 7

### ศึกษาเครื่องมือ (Tools) ที่ช่วยในการสร้างคลังข้อมูลเพื่อการตัดสินใจ (Pilot Decision Support System)

#### 7.1 บทบาทของ Tools ในการพัฒนา Data Warehouse

ปัจจุบัน ข้อมูลที่ดี ข้อมูลที่ถูกต้อง และสมบูรณ์ นำมาสู่การตัดสินใจที่ดีในการบริหารงาน เพราะฉะนั้นทุก ๆ องค์การต่างหันมาให้ความสำคัญกับ Data Warehouse ซึ่งเป็นคลังข้อมูลขนาดใหญ่ที่ใช้เก็บข้อมูล สำหรับการตัดสินใจ ต่างก็พยายามที่จะสร้าง Data Warehouse ที่ดีโดยหา Tools ที่ดีมาช่วยจัดการกับข้อมูลที่มีจำนวนมากซึ่งเก็บไว้อย่างซับซ้อน และสะสมเพิ่มมากขึ้นทุก ๆ วัน เพื่อให้การสร้าง Data Warehouse เกิดประโยชน์สูงสุดจำเป็นต้องเลือก Tools ที่ดีมาช่วยจัดการกับข้อมูลเหล่านั้น โดย Tools ที่ใช้ควรประกอบไปด้วยลักษณะที่สำคัญดังนี้

- Query Tools เป็น Tools ที่ใช้สำหรับค้นหาข้อมูลใน data warehouse สำหรับตอบคำถาม Ad-hoc และคำถามอื่น ๆ ที่ผู้ใช้ต้องการ
- Data Loading เป็น tools ที่ใช้สำหรับจัดการกับข้อมูลที่ Load มาจาก Flat File เข้า Data Warehouse
- Data Extract เป็น Tools ที่ใช้สำหรับจัดการกับข้อมูลที่ Load มาจาก Source อื่น ๆ Migration เข้าสู่ flat File
- Data Modeling เป็น tools ที่ใช้สำหรับสร้างและบำรุงรักษาแบบข้อมูลและความสัมพันธ์ระหว่างข้อมูล
- Meta Data เป็น Tools ที่ใช้สำหรับเก็บ (Store) และจัดการกับรายละเอียดของ Data Warehouse และข้อมูลที่จะนำเข้าสู่ Data Warehouse
- Scheduling เป็น Tools ที่ทำหน้าที่อัตโนมัติ (Automatic) เพื่อดึงข้อมูล (Extracting) และอ่านข้อมูล (Loading) เข้าสู่ Data Warehouse ตามเวลาที่กำหนดไว้

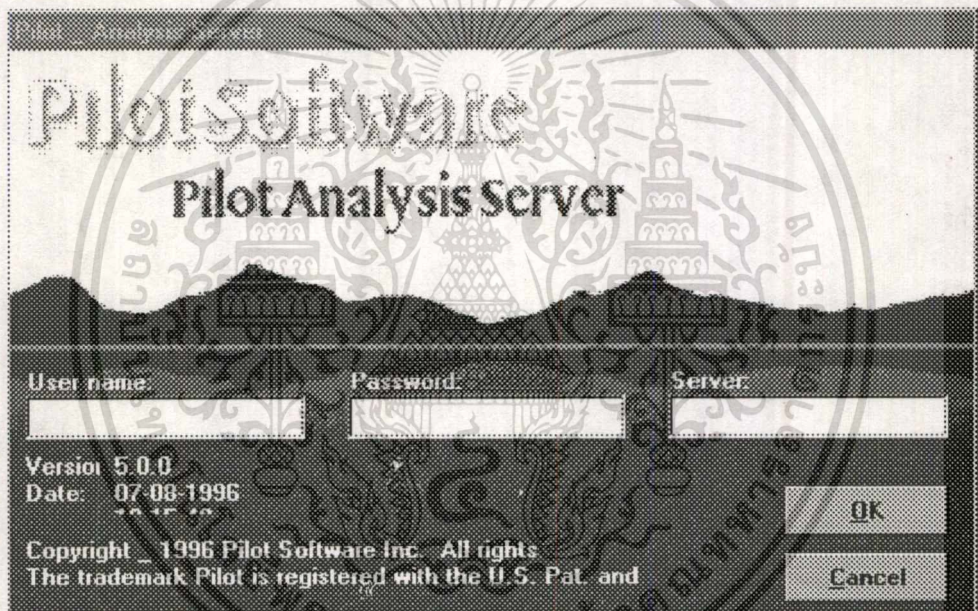
#### 7.2 ส่วนประกอบของ Pilot Decision Support

โปรแกรม Pilot Decision Support เป็นโปรแกรมที่พัฒนาขึ้นมาเพื่อสนับสนุนการใช้ข้อมูลภายในองค์กร ให้มีประสิทธิภาพ โดยเฉพาะการวิเคราะห์ เปรียบเทียบ และหาแนวโน้มของเอกสารเป็นเอกสารที่ส่งวนเวียนสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติเห็นไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลเกี่ยวกับธุรกิจ เช่น การตลาด การเงิน การขาย เกี่ยวกับลูกค้า เป็นต้น หรือเรียกว่าเป็นเครื่องมือสำหรับวิเคราะห์ข้อมูล OLAP (Online Analytical Processing) เพื่อช่วยสนับสนุนการตัดสินใจ ซึ่งสามารถมองข้อมูลได้หลายมิติอยู่ในรูป Multidimensional Model ผู้ใช้สามารถที่จะมองข้อมูลได้หลายมุมมอง โดยเฉพาะ Pilot Desktop ซึ่งออกแบบมาเพื่อการใช้งาน Ad hoc เพื่อให้สามารถตอบคำถามผู้บริหารได้อย่างสะดวก ซึ่งมีส่วนประกอบดังนี้

โปรแกรมในชุดของ Pilot Decision Support Suit ประกอบด้วย 6 ส่วนดังนี้

1. Pilot Analysis Server เป็น โปรแกรมจัดการบน Sever ที่ใช้งานทั่ว ๆ ไป



รูปที่ 7.1 แสดงการเข้าสู่ Pilot Analysis Server

2. Pilot Discovery Server เป็นโปรแกรมที่ใช้งานระบบที่ใหญ่ ๆ และมีความซับซ้อนสูง

3. Pilot Sales & Marketing Analysis Library เป็นโปรแกรมสำหรับวิเคราะห์ด้านการตลาด โดยมี หัวข้อพิเศษเฉพาะการวิเคราะห์ข้อมูลด้านการตลาด

4. Pilot Internet Publisher สำหรับเชื่อมต่อในระบบ Internet เป็นทำหน้าที่เป็น Browser

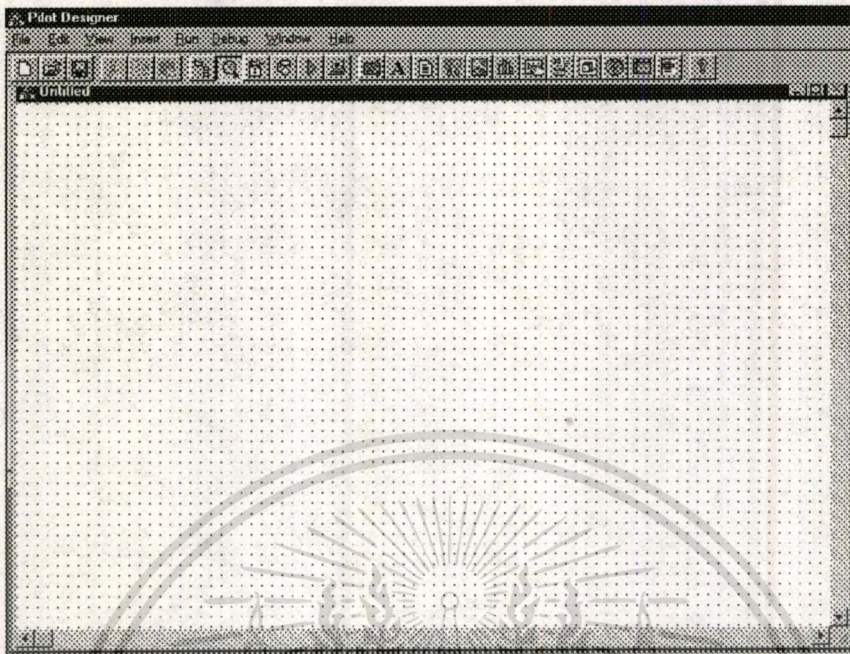
5. Pilot Designer สำหรับการออกแบบ Application ที่ต้องการ ดังรูปที่ 7.2

6. Pilot Desktop สำหรับการวิเคราะห์ข้อมูล ad hoc, Ranging, Exception ดังรูปที่ 7.3

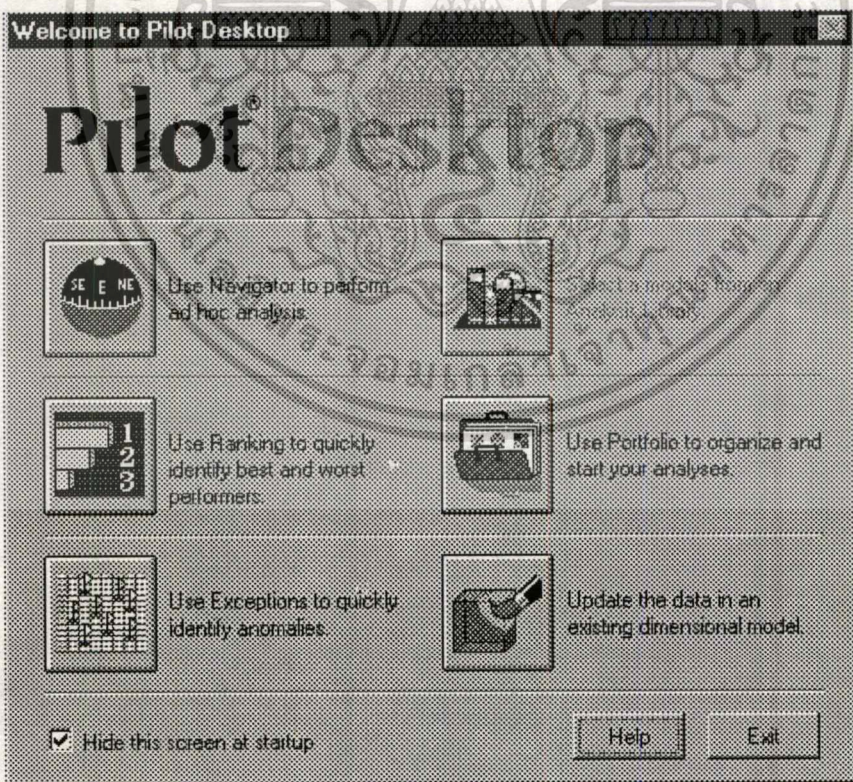
หมายเหตุ โปรแกรมที่ศึกษามีหมายเลข 1, 5, 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.2 แสดงจอภาพของ Pilot Designer

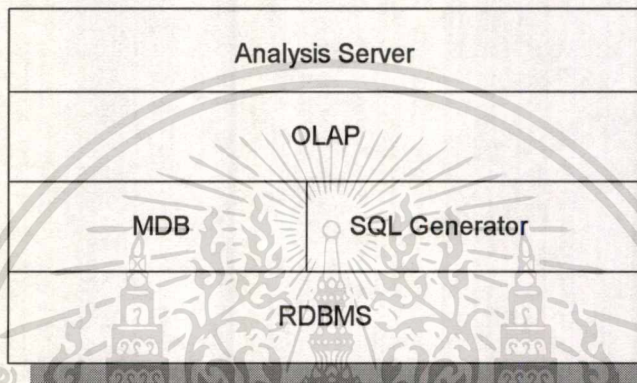


รูปที่ 7.3 แสดงเมนูของ Pilot Desktop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

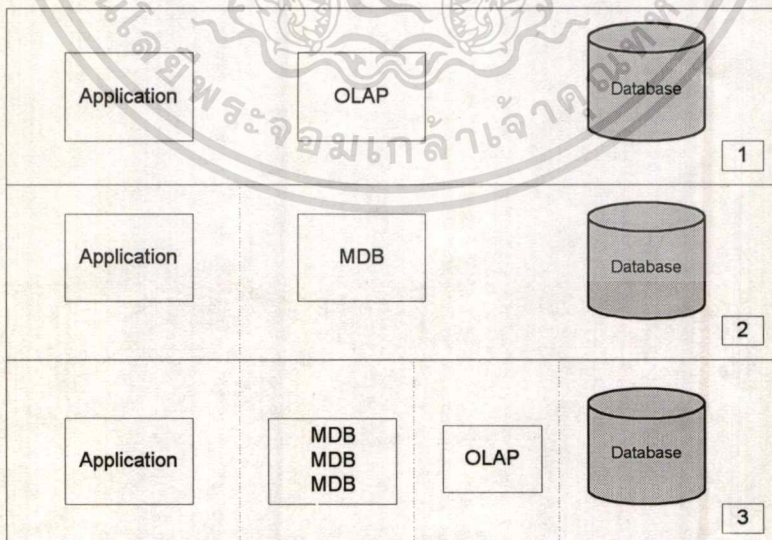
### 7.3 สถาปัตยกรรมของ Pilot Decision Support

ฐานข้อมูลที่ใช้สำหรับ pilot Decision Support อยู่ในรูปแบบฐานข้อมูลเชิงสัมพันธ์ซึ่งถูกจัดการด้วยระบบการจัดการฐานข้อมูล (RDBMS) การใช้ข้อมูลจะมี Tools สำหรับสนับสนุนการทำงานหลายลักษณะซึ่งแต่ละตัวมีความสามารถที่แตกต่างกัน Tools ดังกล่าว MDB,SQL,OLAP ซึ่งแสดงสถาปัตยกรรมของการเรียกใช้ข้อมูลได้ดังรูปที่ 7.4



รูปที่ 7.4 สถาปัตยกรรมของ Pilot Decision Support

การใช้งานโปรแกรมไม่ได้กำหนดรูปแบบที่แน่นอนตายตัวแล้วผู้ใช้งานจะนำไปประยุกต์ใช้กับระบบงานของตนเองสำหรับ Pilot Decision Support สามารถสนับสนุนการใช้งาน Client/Server ในรูปแบบ Three -Tier ดังรูปที่ 7.5



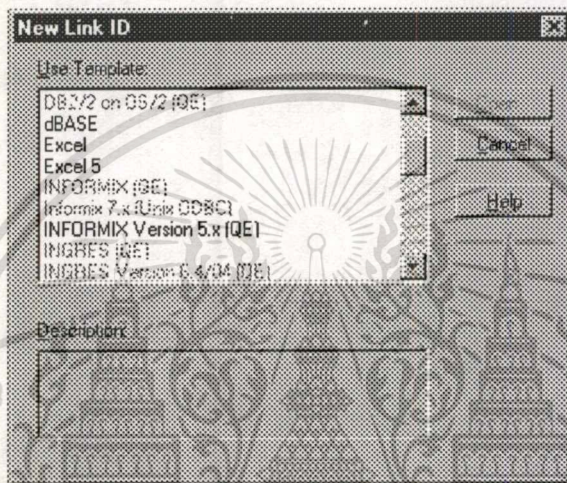
รูปที่ 7.5 แสดงสถาปัตยกรรม Three-tier ของ Pilot Decision Support

จากระดับ 1 เก็บทุกอย่างที่เครื่องเดียวกัน ระดับ 2 เก็บ database กับ Mdb เครื่องเดียว

กันแยก Application ไว้คนละเครื่อง ระดับ 3 แยก database , applicaiton ต่าง ๆคนละเครื่อง  
เอกสารนี้เป็นเอกสารทสวงนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตเห็นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

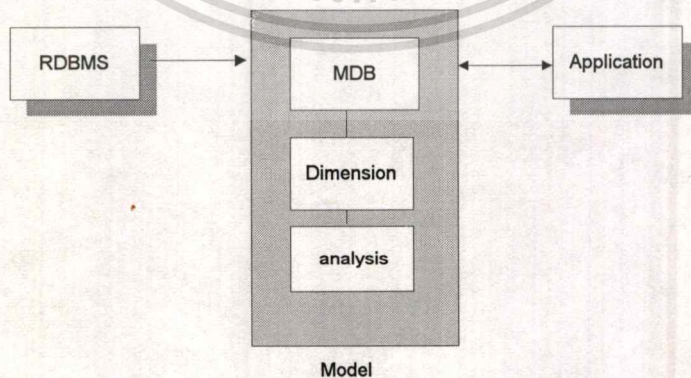
## 7. 4 การทำงานของ Pilot Decision Support

1. สร้าง Link ID เป็นการกำหนด Interface สำหรับเชื่อมต่อข้อมูลเข้าสู่ระบบ เรียกว่า DAC ( Data Access Connection ) ซึ่งมี รายการให้เลือก ถ้าต้องการใช้ข้อมูลที่เป็น dBASE ก็เลือก dBASE หรือถ้าต้องการใช้ฐานข้อมูล Informix หรือ Oracle ก็ให้เลือกตามต้องการ โปรแกรมจะให้ใส่ชื่อสำหรับการ Link และทดสอบการ Link



รูปที่ 7.6 แสดงจอภาพของการกำหนด Link Id

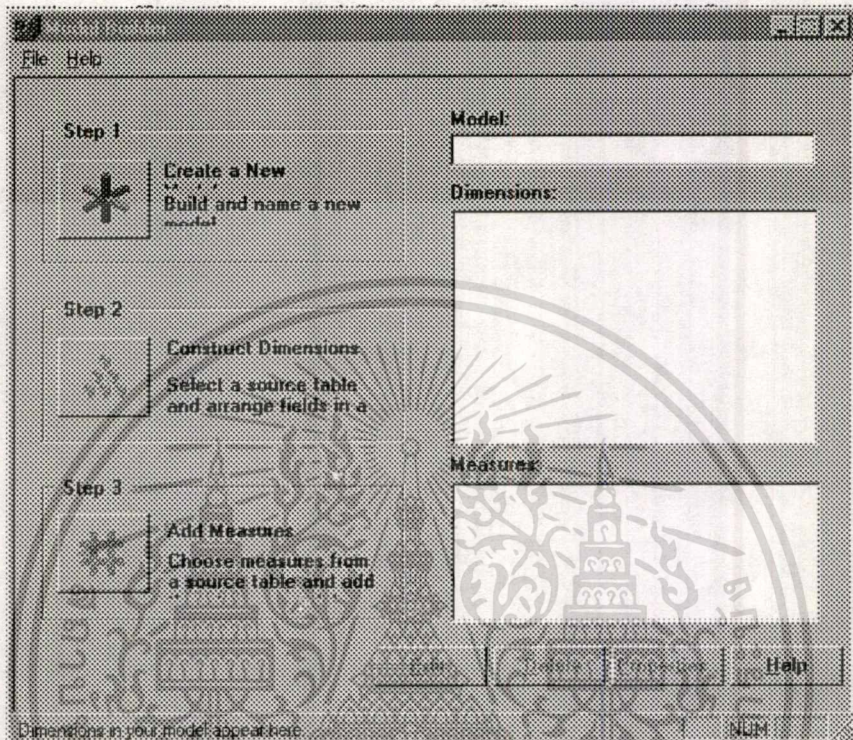
2. สร้าง Model สำหรับการใช้งาน โดยจะมีการกำหนด Dimension ตามที่ได้กำหนดไว้ เพื่อการเรียกใช้ข้อมูลจาก RDBMS เข้าสู่ Model ที่สร้างไว้ ดังรูปที่



รูปที่ 7.7 แสดงโครงสร้างของ Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ขั้นตอนการสร้างโดยเลือก Model Builder จะขึ้นจอภาพดังนี้

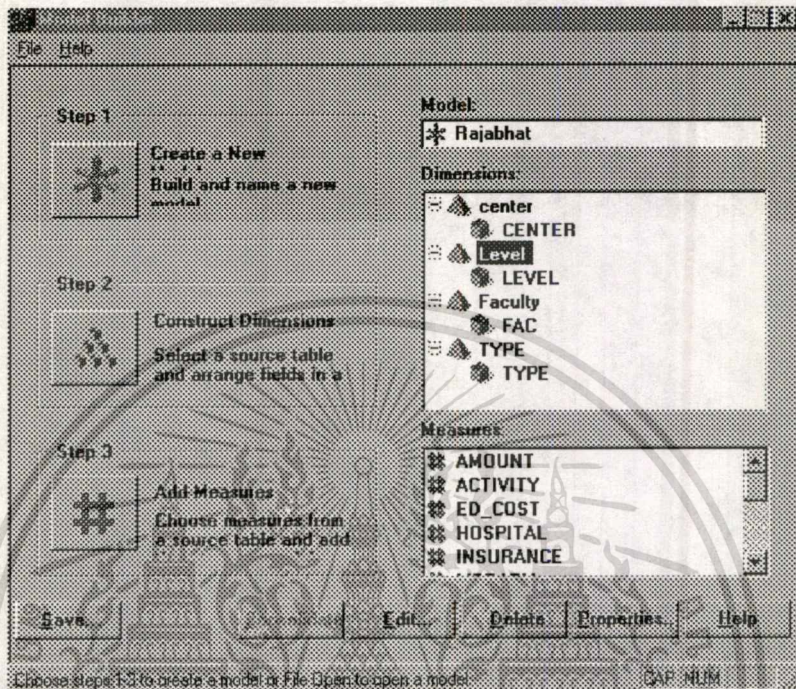


รูปที่ 7.8 แสดงจอภาพของการสร้าง Model Builder

ในการสร้าง Model ส่วนประกอบ 4 ส่วนดังนี้

1. Dimension เป็นการกำหนดลำดับชั้น หรือ มุมมองของข้อมูลโดยนำมาจาก Dimension Table
2. Measures เป็นการกำหนดวิธีการในการคำนวณ หรือกำหนดตัวแปร Measure มี 2 ลักษณะคือ เลือกจาก Field ที่เป็น Numeric หรือสร้างขึ้นใน Model
3. Time เป็นการกำหนดช่วงเวลาในการเก็บข้อมูล อาจเป็น ปี เดือน สัปดาห์ หรืออื่น ๆ ตามที่ต้องการ
4. Attribute กำหนดฟิลด์สำหรับดูรายละเอียดข้อมูล (จะมีหรือไม่มีก็ได้)

## ตัวอย่างการสร้าง Model สำหรับการมองข้อมูล



รูปที่ 7.9 แสดงการสร้าง Model

จากรูป Model ได้แก่ Rajabhat

Dimension ได้แก่ Center, Level, Faculty, Type

Measure ได้แก่ Amount, Activity, Ed\_cost, Hospital ฯลฯ

ในช่วงของการสร้างจะมีการกำหนด รหัสผ่าน ขนาดของ block จำนวน Block ที่ต้องการ กำหนดช่วงเวลาที่ต้องการ

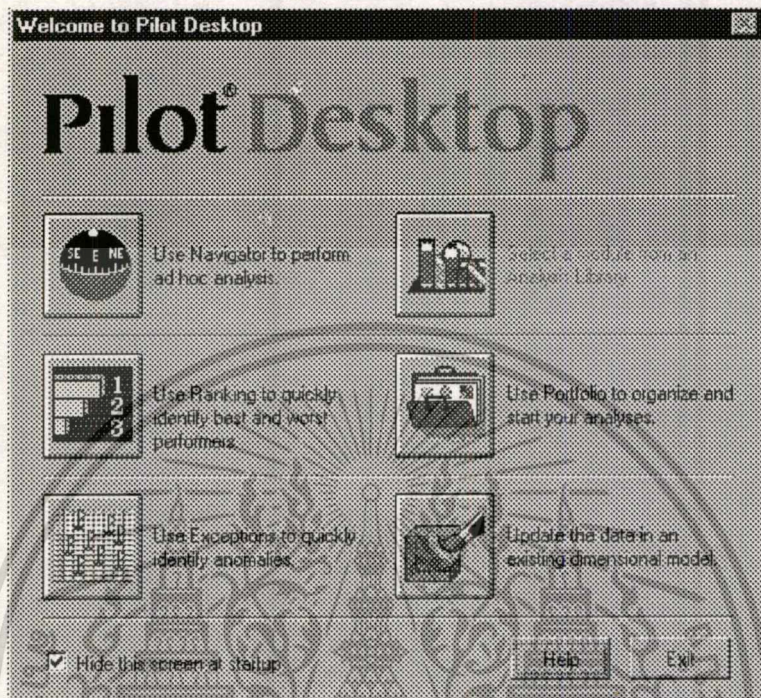
3. จัดเก็บข้อมูล Analysis Server จะเป็นตัวจัดการกับ Model ที่สร้างขึ้น ได้แก่ การบันทึกรายละเอียดทั้งหมดเกี่ยวกับ Model ทำการ Load ข้อมูลจาก Fact Table เข้าสู่ multidimensional Model ซึ่งในส่วนนี้จะมีการรวมข้อมูล กำหนดขนาดของ Block กำหนดพื้นที่ที่ต้องใช้ จัดการกับ block ข้อมูลที่เป็น Sparse เป็นต้น

4. การใช้งาน Model มี 2 วิธีคือ

1. พัฒนา Application สำหรับใช้งาน Model ที่สร้างขึ้น โดยใช้ Pilot Designer ซึ่งสามารถทำการ coding ใช้ภาษาที่เรียกว่า Basic Script ซึ่งสามารถใช้งานร่วมกับ Visual Basic ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ใช้ Pilot Desktop ซึ่งเป็น Tools ที่พัฒนาเพื่อการวิเคราะห์ข้อมูลโดยเฉพาะ



รูปที่ 7.10 เมนูของ Pilot Desktop

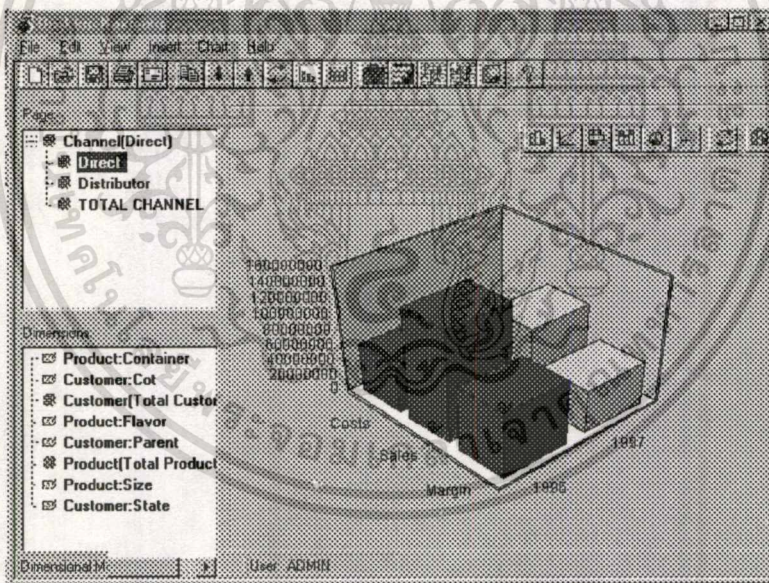
ซึ่งมี ทั้งหมด 6 ส่วนสำหรับการใช้แต่ละส่วนมีหน้าที่ดังนี้

1. Navigator เหมาะสำหรับการตอบคำถามเฉพาะกิจ (Ad hoc) โดยสามารถที่จะเลือกดูข้อมูลใน Dimension ได้ทั้งหมด วิเคราะห์และเปรียบเทียบระหว่าง Model ได้ เลือกที่จะขึ้นจอภาพของ Navigator ดังรูปที่ 7.11


การใช้งาน Navigator นี้ ผู้ใช้สามารถเปลี่ยนมุมมองได้หลายมิติตามแนวที่ต้องการ เรียกว่า Down ,หรือ Accross และปรับเปลี่ยนช่วงเวลาสำหรับวิเคราะห์ข้อมูลได้ตามต้องการ นอกจากนั้นยังสามารถที่จะแสดงเป็นรูปภาพ ได้หลายรูปแบบ เช่น กราฟแท่ง กราฟเส้น และอื่น ๆ เพื่อให้มองเห็นความแตกต่างข้อมูลได้ชัดเจนยิ่งขึ้น

Page	A	B	C	D	E	F
1		Costs				
2	Jan 96	4,900,077.00				
3	Feb 96	6,274,729.00				
4	Mar 96	5,232,139.00				
5	Apr 96	4,809,853.00				
6	May 96	5,000,003.50				
7	Jun 96	4,890,034.00				
8	Jul 96	4,669,587.50				
9	Aug 96	4,701,486.00				
10	Sep 96	5,634,515.00				
11	Oct 96	4,631,953.50				
12	Nov 96	4,803,417.00				
13	Dec 96	4,898,871.00				
14	Jan 97	6,671,811.50				
15	Feb 97	7,350,712.00				
16	Mar 97	6,862,263.00				
17	Apr 97	6,861,098.50				
18	May 97	6,602,803.50				

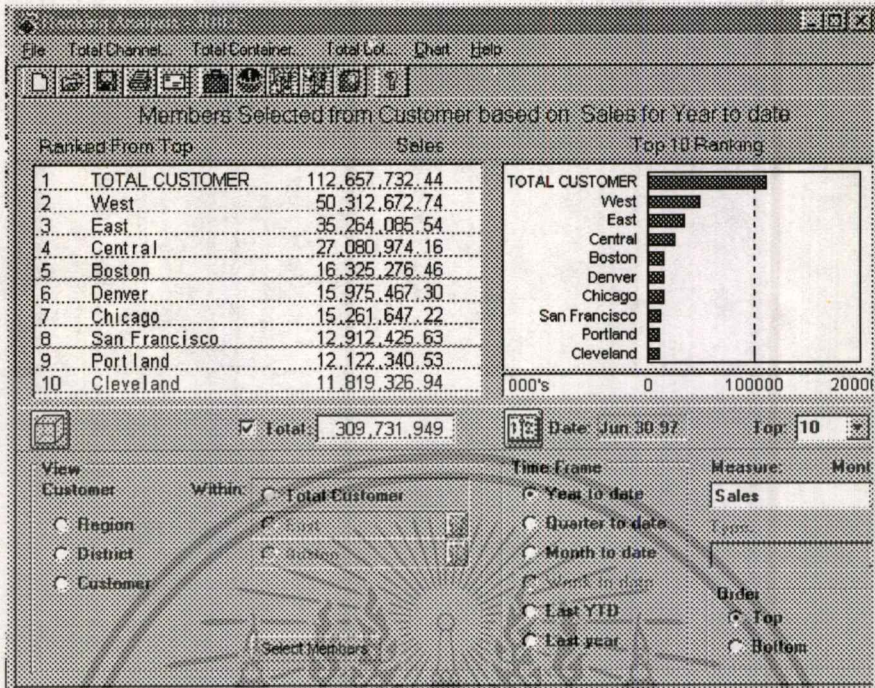
รูปที่ 7.11 แสดงการ View ข้อมูลโดยใช้ Navigator



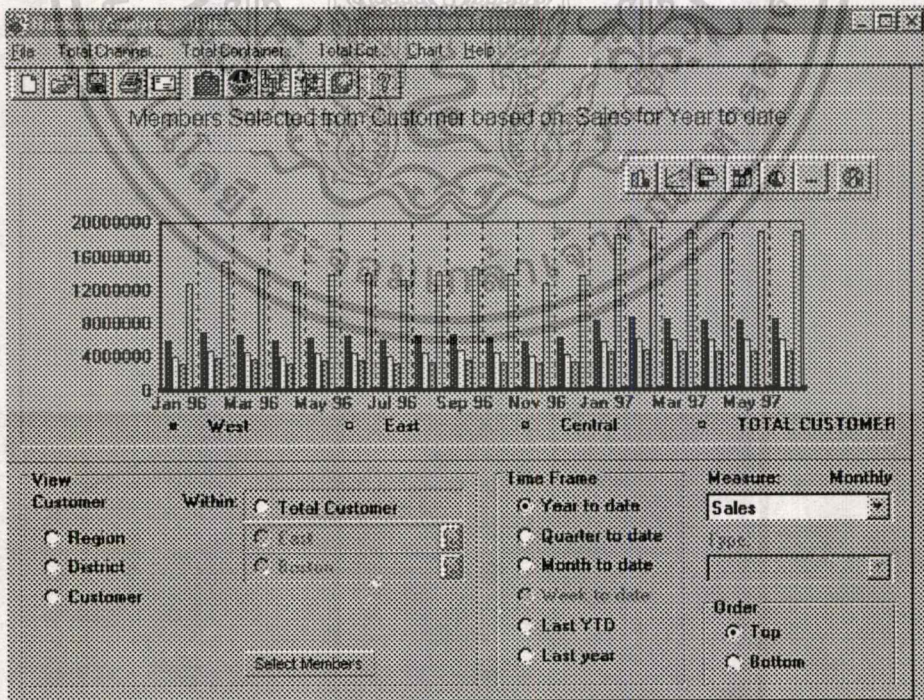
รูปที่ 7.12 แสดงการมองข้อมูลเป็นกราฟ

2. Ranging โปรแกรมในส่วนนี้มีเหมาะสำหรับการหาลำดับข้อมูล สามารถจัดลำดับตามหัวข้อต่าง ๆ ที่อยู่ใน Dimension ได้ แสดงเป็นข้อความและแสดงในรูปแบบกราฟได้ด้วย เลือกไอคอน  เข้าสู่จอภาพของ Ranging ดังรูปที่ 7.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.13 แสดงการจัดลำดับข้อมูล

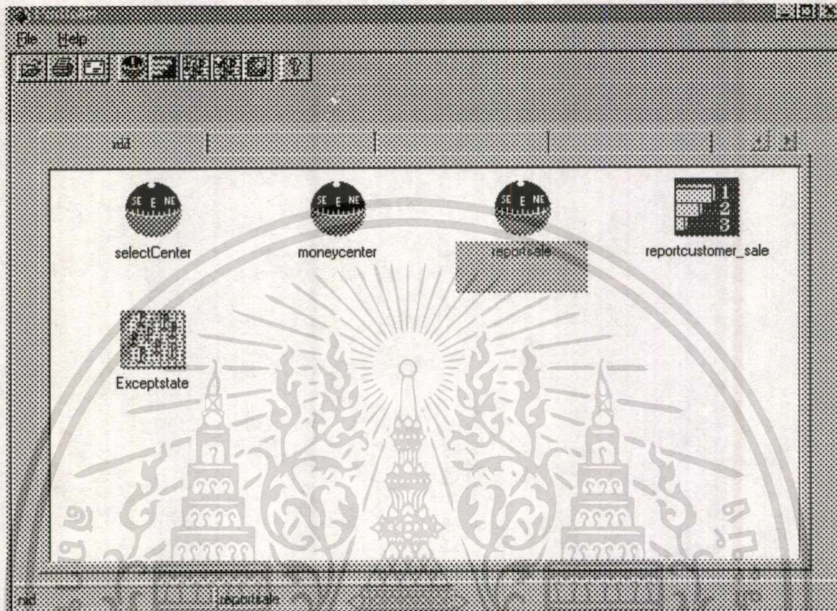


รูปที่ 7.14 กราฟแสดงยอดขายแบ่งเป็นปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



4 Portfolio เป็นเพิ่มสะสมงาน หมายความว่าเมื่อวิเคราะห์ข้อมูล หรือดูข้อมูล จากโปรแกรม Navigator,Ranging,Exception แล้วสามารถที่จะเก็บบันทึกไว้เพื่อออกรายงาน หรือเพื่อเรียกดูในครั้งต่อไปได้ โดยที่ไม่ต้องมาคำนวณใหม่ ดังรูปที่ 7.17



รูปที่ 7.17 แสดงการเก็บข้อมูลใน Portfolio

5. Analysis Library เป็นส่วนที่เรียกใช้ โปรแกรมวิเคราะห์ที่ต่าง ๆ ที่เพิ่มเข้ามา เช่นโปรแกรมวิเคราะห์ด้านการตลาด หรืออื่น ๆ เป็นต้น

6. Update Model Builder เป็นส่วนสำหรับแก้ไข Model

## บทที่ 8

### วิเคราะห์ และเปรียบเทียบ Toolsที่ใช้ในการสร้าง Data warehouse

จาก Tools ที่ช่วยจัดการข้อมูลใน Data warehouse นำมาวิเคราะห์และเปรียบเทียบกับ รูปแบบของข้อมูล (Data Model) ใน 3 หัวข้อหลักดังนี้

#### 8.1 โครงสร้างข้อมูล(Data Structure )

โครงสร้างข้อมูลที่เกิดขึ้นใน data warehouse เป็นข้อมูลที่เก็บในรูปแบบความสัมพันธ์ที่จัดการโดย Relational DBMS ซึ่งในการใช้งาน Tools จะต้องนำข้อมูลใน data warehouse ที่อยู่ในรูปแบบสรุปเข้าสู่ Multidimensional Model เพื่อการวิเคราะห์ โดยโครงสร้างข้อมูลแบบหลายมิติจะมองในรูปของ Cube หรือรูปลูกเต๋า ซึ่งเป็นการแสดงข้อมูลในรูปแบบ 3 มิติ ทำให้ผู้ใช้มีมุมมองที่กว้างขึ้นจากเดิมคือ 2 มิติ มองในแนว แถว และคอลัมน์ แต่ถ้าใช้ Multidimension Tools จะสามารถมีมุมมองเพิ่มขึ้นอีกคือในแนว Row, Column และในด้านลึก หรือเป็นรูปลูกบาศก์ ซึ่งเป็นจุดเด่นของ Multidimensional Model โดยข้อมูลทุกรายการเป็น Atomic คือ เป็นข้อมูลที่ Denormalize นำมาเสนอในระดับแนวความคิด Conceptual View ซึ่งมองโครงสร้างข้อมูลเป็น 3 มิติ สามารถแทนด้วยจุด coordinate ของกราฟ คือแกน x,y,z ซึ่งเป็นอีกมุมมองหนึ่งของการประยุกต์ใช้งานฐานข้อมูลเชิงสัมพันธ์

#### 8.2 ความถูกต้องของข้อมูล (Data Integrity )

จาก Tools ที่ทำการศึกษาการควบคุมความถูกต้องของข้อมูลใน Multidimensional Model โดยจะจัดการกับมูลที่เป็น Sparse ซึ่งจะมีวิธีการจัดการคือ มีการบีบอัดข้อมูล เพื่อประหยัดเนื้อที่ในการจัดเก็บ เพราะข้อมูลที่มองในรูปลูกบาศก์นี้เป็นการแทนด้วยจุด coordinate จึงทำให้เกิดช่องว่างขึ้น ในขณะที่ การจัดการฐานข้อมูลเชิงสัมพันธ์จะมีกฎควบคุมความถูกต้องของข้อมูล ซึ่งจะไม่ยอมให้คีย์หลักมีค่าเป็นช่องว่าง เพื่อควบคุมความถูกต้องของข้อมูลในส่วนอื่น ๆ ที่เกี่ยวข้องด้วย แต่ใน Multidimensional Model ไม่ได้ใช้คีย์สำหรับการค้นหาข้อมูลเพราะนำข้อมูลทุก ๆ เรคคอร์ด มาเก็บในที่เดียวกันไม่มีการเชื่อมระหว่างตาราง เหมือนกับ ข้อมูลเชิงสัมพันธ์ และใช้ จุด coordinate แทนการอ้างถึงข้อมูลแต่ละเรคคอร์ด หรือเรียกเทคนิคในการเข้าถึงข้อมูลนี้ว่า hash address ซึ่งมี hash table สำหรับเก็บตำแหน่งของข้อมูลแต่ละเรคคอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 8.3 การจัดการข้อมูล(Data Manipulate )

การจัดการกับข้อมูลของ ของ Tools ที่ใช้ในการศึกษา โดยมี SQL ช่วยสนับสนุนการทำงาน ซึ่งการพัฒนา Application จะใช้ SQL เป็นฐานช่วยในการจัดการกับข้อมูลทำให้มีประสิทธิภาพในการเข้าถึงข้อมูล นอกจากนี้ ยังมีระบบควบคุมความปลอดภัยของข้อมูล โดยใช้รหัสผ่านในการเข้าถึงข้อมูล การจัดการกับพื้นที่ใช้งาน ในการใช้งานเบื้องต้นจะให้ผู้เป็นผู้นำกำหนดพื้นที่ในการใช้งานแต่ถ้าไม่พอ ระบบก็จะนำพื้นที่สำรองมาแทนเพื่อไม่ให้เกิดการประมวลผลในช่วงนั้นหยุดชะงักลง ส่วนการมองข้อมูลในรูปแบบ Drill Down, Rotate ศัพท์ทางการค้าของบริษัทผู้ผลิตที่ต้องการนำเสนอในรูปแบบที่ดูแปลกใหม่ แต่ในความเป็นจริง การ Drill Down คือการเลือกดูข้อมูลเฉพาะกลุ่ม(Group) จากกลุ่ม ใหญ่ไปหากลุ่มย่อย ซึ่งใช้ความสามารถของ SQL ในการจัดการในส่วนนี้ ส่วน Rotate ก็เป็นการดูข้อมูลที่เปลี่ยนตำแหน่งในการมอง Attribute เช่น การดูข้อมูล โดยใช้ SQL ถ้ามี 3 Attribute สามารถมองได้  $= 3! = 6$  มุมมอง

ตัวอย่าง กำหนด 3 Attribute code,name,age

การ Rotate ใน Multidimension Model

มุมมองที่ 1 Select code,name,age From Student

มุมมองที่ 2 Select code,age,name From Student

มุมมองที่ 3 Select name,code,age From Student

มุมมองที่ 4 Select name,age,code From Student

มุมมองที่ 5 Select age,name,code From Student

มุมมองที่ 6 Select age,code,name From Student

เพราะฉะนั้นในการจัดการกับข้อมูลของ Multidimensional Model อาศัยความสามารถของ SQL ในการค้นหาแต่มีการนำเสนอในรูปแบบที่ง่ายเหมาะสำหรับผู้ที่ไม่มีความรู้หรือไม่มีความชำนาญในการเขียน SQL

## บทที่ 9

### กรณีศึกษา การสร้าง Data Warehouse สำหรับสนับสนุนการตัดสินใจของผู้บริหาร สถาบันราชภัฏฉะเชิงเทรา

#### 9.1 ความเป็นมาของปัญหา

สถาบันราชภัฏฉะเชิงเทราเป็นสถาบันอุดมศึกษาท้องถิ่น มีนโยบายหลักคือ ให้บริการการศึกษาแก่นักศึกษาในท้องถิ่น โดยเปิดรับนักศึกษา 2 ประเภทคือ นักศึกษาภาคปกติ ซึ่งเรียนในวันราชการ และนักศึกษาภาค กศ.บป (การศึกษานอกเวลาประจำการ) ซึ่งเป็นบุคลากรประจำการที่มีงานทำอยู่แล้ว เข้าเรียนในวันเสาร์ - อาทิตย์ โดยสถาบันจะมีศูนย์สำหรับให้บริการการศึกษาแก่นักศึกษา กศ.บป ทั้งหมด 10 ศูนย์ โดยแบ่งออกเป็น 2 จังหวัด ดังนี้

##### 1. จังหวัดฉะเชิงเทรา ประกอบด้วยศูนย์ให้การศึกษา

- สถาบันราชภัฏฉะเชิงเทรา
- โรงเรียนดัดดรุณี
- โรงเรียนเบญจมราชรังสฤษฎิ์
- โรงเรียนพุทธโสธร
- โรงเรียนเบญจมราชรังสฤษฎิ์ 2

##### 2. จังหวัดชลบุรีประกอบด้วยศูนย์ให้การศึกษา

- โรงเรียน ชลราชฎาราม
- โรงเรียนชลบุรี (สุขบท)
- โรงเรียนศรีราชา
- วิทยาลัยเทคนิคสัตหีบ

โดยเปิดสอนตั้งแต่ระดับอนุปริญญา จนถึง ปริญญาตรี มีทั้งหมด 5 คณะคือ คณะวิทยาศาสตร์ ,คณะมนุษยศาสตร์ , คณะครุศาสตร์ ,คณะวิทยาการจัดการ และคณะเทคโนโลยีอุตสาหกรรม มีโปรแกรมวิชาประมาณ 60 โปรแกรมวิชา มีจำนวนนักศึกษาประมาณ 23,000 คน

จากการรับนักศึกษาจำนวนมากทำให้ต้องมีการวางแผนเกี่ยวกับทุก ๆ เรื่องเป็นอย่างดี โดยเฉพาะงานด้านการเรียนการสอน ซึ่งต้องมีการวางแผนอย่างเป็นระบบ ตั้งแต่งานแผนการเรียน งานการจัดตารางเรียนตารางสอน งานการลงทะเบียน การเก็บค่าเงินลงทะเบียน การตรวจสอบการเรียนครบหลักสูตร ซึ่งระบบงานที่กล่าวมาทั้งหมดนั้นในปัจจุบัน สถาบันใช้คอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่วยในการประมวลผลผ่านระบบเครือข่ายระยะใกล้ Lan ซึ่งมีประสิทธิภาพการทำงานเป็นที่พอใจของผู้ปฏิบัติงานในระดับหนึ่ง โดยข้อมูลในระบบทั้งหมด จะถูกจัดเก็บอยู่ในรูปแบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งสนับสนุนการทำงานแบบ online transaction แต่ไม่เหมาะสำหรับการตอบคำถามของผู้บริหาร เพราะไม่มีข้อมูลสรุป ที่เป็นข้อมูลปัจจุบัน และข้อมูลเก่า ที่เก็บไว้ในฐานข้อมูล จึงเป็นปัญหาสำหรับผู้บริหารที่จะใช้ข้อมูลสำหรับการตัดสินใจดำเนินงานต่าง ๆ โดยเฉพาะข้อมูลที่เกี่ยวข้องจำนวนนักศึกษา และ รายได้ จากการลงทะเบียน เป็นข้อมูลที่มีความสำคัญ เพราะจะต้องมีการจัดสรรเงินเหล่านั้นไปยังส่วนต่าง ๆ เพื่อช่วยให้กระบวนการเรียนสอนสามารถดำเนินไปได้ด้วยดี และมีประสิทธิภาพยิ่งขึ้น

จากปัญหาดังกล่าวจึงได้มีการนำแนวความคิดเกี่ยวกับ Data Warehouse เข้ามาใช้เพื่อสร้างฐานข้อมูล สำหรับสนับสนุนการตัดสินใจของผู้บริหาร โดยทดลองสร้างในส่วนของข้อมูลนักศึกษา และเงินรายได้จากเงินบำรุงการศึกษา เป็นรูปแบบเบื้องต้น สำหรับให้ผู้บริหารทดลองใช้ และเป็นแนวทางสำหรับการพัฒนา data warehouse ให้ครบทุกระบบงานในสถาบัน

## 9.2 วัตถุประสงค์

- 9.2.1 เพื่อสร้างฐานข้อมูลสำหรับสนับสนุนการตัดสินใจ (จำนวนนักศึกษา และรายได้)
- 9.2.2 เพื่อเป็นแนวทางในการพัฒนา Data Warehouse
- 9.2.3 เพื่อชี้ให้เห็นประโยชน์ ของ Data warehouse

## 9.3 ขอบเขตของการศึกษา

9.3.1 รวบรวมข้อมูลจำนวนนักศึกษา แบ่งตามส่วนหัวข้อที่ต้องการคือ ประเภทนักศึกษา คณะ ระดับ และ ศูนย์การศึกษา ตั้งแต่ปี 2535-2539

9.3.2 สรุปรายได้ค่าลงทะเบียนแบ่งตาม ประเภทนักศึกษา คณะ ระดับ และศูนย์การศึกษา และแยกเป็นหมวดหมู่ ได้แก่ ค่าหน่วยกิตปฏิบัติ ทฤษฎี ค่าบำรุงการศึกษา ค่ารักษาพยาบาล ค่าห้องสมุด ค่ากิจกรรม ค่าประกันของเสียหาย ค่าบำรุงศูนย์ ค่าธรรมเนียมพิเศษ ตั้งแต่ปี 2535-2539

9.3.3 เลือกดูข้อมูลตามหัวข้อที่ต้องการ

9.3.4 วิเคราะห์ข้อมูลสรุปจำนวนนักศึกษา และ รายได้จากการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 9.4 ขั้นตอนดำเนินงาน

9.4.1 ศึกษารายละเอียดของข้อมูลในระบบงานเดิม

9.4.2 ออกแบบ Database สำหรับ Warehouse

9.4.3 ออกแบบ Model สำหรับการใช้งาน

9.4.4 สร้าง Model โดย Model Builder ของ Dss

9.4.5 วิเคราะห์ข้อมูลโดยใช้ Pilot Desktop

### 9.4.1 ศึกษารายละเอียดของข้อมูลในระบบงานเดิม

ระบบงานทะเบียนและการจ่ายเงินลงทะเบียนมีงานหลักที่ต้องทำดังนี้

1. จัดทำแผนการเรียน
2. จัดตารางสอนและตารางสอบ
3. งานการลงทะเบียน
4. งานรับและคิดค่าลงทะเบียน
5. งานทำผลการเรียนและแจ้งผลการเรียน
6. งานตรวจสอบการเรียนครบหลักสูตร
7. งานออกใบรายงานผลการศึกษา

งานทั้ง 7 งานเป็นที่มีความสัมพันธ์และเกี่ยวข้องกัน ทั้งหมดเริ่มตั้งแต่การจัดทำแผนการเรียนเพื่อให้สอดคล้องกับหลักสูตรที่กำหนด จากนั้นมีการจัดทำตารางสอนและตารางสอบ โดยนำข้อมูลจากแผนการเรียนที่กำหนดไว้ของแต่ละเอกมาจัดทำตารางเพื่อเตรียมการเรียนการสอน นักศึกษาที่ลงทะเบียนเรียนในแต่ละเทอมจะต้องลงทะเบียนตามแผนการเรียนที่กำหนดไว้ จากนั้นโปรแกรมจะคำนวณค่าลงทะเบียนของนักศึกษา โดยคิดจาก ค่าหน่วยกิตที่ลงทะเบียน ที่เป็นหน่วยกิตปฏิบัติ และบรรยาย และค่าธรรมเนียมต่าง ๆ จากนั้นออกใบเสร็จรับเงินให้กับนักศึกษาที่ลงทะเบียน ส่วนงานในข้อ 5 เป็นงานที่ทำเมื่อสิ้นสุดการเรียนการสอนในแต่ละเทอมโดยจะมีการบันทึกผลการศึกษา (เกรด) ในแฟ้มลงทะเบียน งานที่ 6 เป็นงานที่ทำเมื่อนักศึกษาเรียนจนครบหลักสูตรแล้ว เพื่อตรวจสอบความถูกต้องการที่จะจบการศึกษา งานขั้นสุดท้ายเป็นงานออกใบรายงานผลการศึกษา โดยมีการตรวจเช็คข้อมูลก่อนออกใบรายงานว่ามีนักศึกษาเรียนครบหลักสูตรจริงหรือไม่ จากการศึกษากระบวนการทั้งหมด มีแฟ้มข้อมูลการปฏิบัติงาน(Operational) จำนวนมาก โดยข้อมูลเหล่านั้นเก็บอยู่ในรูปแบบฐานข้อมูลเชิงสัมพันธ์ ใช้ระบบการจัดการฐานข้อมูล FoxBase v 2.0 Multuser ข้อมูลเก็บที่ไฟล์ File Server โดยสามารถเรียกใช้งานได้พร้อม ๆ กัน 3 ส่วนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. สำนักส่งเสริมวิชาการ : ทำหน้าที่จัดแผนการเรียน ตารางสอน และตารางสอบ
2. ฝ่ายทะเบียนและวัดผล ทำหน้าที่รับลงทะเบียน ตรวจสอบเช็ครายวิชาและผลการเรียนรวมทั้งการออกใบรับรองต่าง ๆ
3. ฝ่ายการเงิน ทำหน้าที่รับเงินค่าลงทะเบียนและอื่น ๆ

การทำงานของทั้ง 3 ส่วนมีการทำงานเป็นลำดับ โดยเริ่มจากแผนการเรียนลงทะเบียน และจ่ายเงิน ไม่สามารถที่จะทำขนานกันได้ เพราะข้อมูลมีความเกี่ยวข้องกันตลอดกระบวนการ

ในกรณีศึกษามีเพิ่มข้อมูลเก่าที่ใช้งานดังนี้

### โครงสร้างเพิ่มข้อมูล

1. เพิ่มประวัตินักศึกษา ชื่อ STUDENT.DBF รายละเอียดโครงสร้างเพิ่มมีดังนี้

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
OLDID	Character	10	รหัสประจำตัว
PNAME	Character	15	คำนำหน้าชื่อ
NAME	Character	30	ชื่อ-สกุล
E_NAME	Character	50	ชื่อ-สกุล ภาษาอังกฤษ
MOVE	Character	1	1=ย้ายมา 2=สมทบ
STTS	Character	1	1=ลงทะเบียนบัณฑิต
STATUS	Character	1	รหัสสถานภาพ
PSS	Character	3	รหัสลับนศ.
DROP	Character	40	การพักเรียน
KURU	Character	1	รหัสครุทายาท
PREL	Character	3	แสดงการลงวิชา Preliquesit
STNO	Character	5	หมายเลข Transcript
SEX	Character	1	เพศ 1=ชาย 2=หญิง
JOB	Character	15	อาชีพนักศึกษา
RELIGION	Character	10	ศาสนา
BDATE	Character	8	ว.ด.ป. เกิด

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADD1	Character	40	ที่อยู่ นศ.บรรทัดที่ 1
ADD2	Character	40	ที่อยู่ นศ.บรรทัดที่ 2
PADD1	Character	2	จว.ที่ย้ายมาเรียนสมทบ
Y_IN	Character	2	ปีที่เข้า
MINOR	Character	2	รหัสวิชาโท
OLD_ED	Character	35	วุฒิเดิมภาษาไทย
E_OLDDED	Character	35	วุฒิเดิมภาษาอังกฤษ
OLD_SC	Character	20	สถานศึกษาเดิมภาษาไทย
E_SCH	Character	2	สถานศึกษาเดิมภาษาอังกฤษ
ADMDATE	Character	8	วดป ที่เข้า
TMNDATE	Character	8	วดป ที่สำเร็จ
CENTER	Character	2	รหัสศูนย์
LEVEL	Character	1	รหัสระดับ
MJCODE	Character	3	รหัสวิชาเอก
GROUP	Character	8	กลุ่ม
TT_CR	Character	3	หน่วยกิตรวม
TRAN_CR	Numeric	3	หน่วยกิตที่โอน/ยกเว้น
TIME	Numeric	1	1 =ชำระเงินแล้ว

## 2. เพิ่มระดับการศึกษา ชื่อ Level.DBF รายละเอียดโครงสร้างเพิ่มมีดังนี้

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
Level	Character	1	รหัสระดับ
Lname	Character	10	ชื่อระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เพิ่มรหัสกลุ่มวิชา ชื่อ Sigroup.DBF รายละเอียดโครงสร้างเพิ่มมีดังนี้

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
Sjcode	Character	1	รหัสกลุ่มวิชา
Sjname	Character	15	ชื่อกลุ่ม

4. เพิ่มรายงานวิชาที่ลงทะเบียน Regis.DBF

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
ID_NO	Character	9	รหัส นศ.
SJCODE	Character	1	รหัสกลุ่มวิชา
SECTION	Character	1	Section A - Z
MJCODE	Character	8	รหัสวิชา
SJGROUP	Character	1	กลุ่มวิชา
GRADE	Character	1	เกรด
TERM	Character	4	ภาคเรียน เช่น 1/39
FLAG	Character	1	o=โอน e=ยกเว้น

5. เพิ่มรหัสวิชาเอก MJCODE.DBF โครงสร้างเพิ่มข้อมูล

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
MJCODE	Character	2	รหัสวิชาเอก
FACULTY	Character	1	รหัสคณะ
MJNAME	Character	30	ชื่อเอก
DEPART	Character	5	รหัสภาควิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6. เพิ่มรหัสรายวิชาและชื่อวิชาต่าง ๆ course.DBF

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
MJNAME	Character	20	ชื่อวิชา
CRAA	Character	6	หน่วยกิตปฏิบัติ
CRRR	Character	6	หน่วยกิตบรรยาย
HOUR	Character	2	จำนวนชั่วโมง

## 7. เพิ่มตารางประเภทนักศึกษา

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
SUBTYPE	Character	2	รหัสประเภท
SUBTNAME	Character	25	ชื่อประเภท

## 8. เพิ่มตารางภาควิชา

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
SUBPART	Character	8	รหัสภาควิชา
SUBPNAME	Character	30	ชื่อภาควิชา
TEL	Character	10	โทรศัพท์

## 9. เพิ่มศูนย์ให้การศึกษ

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
SUBPART	Character	8	รหัสภาควิชา
ADDRESS	Character	30	ที่อยู่
TEL	Character	10	โทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 10. เพิ่มอัตราการเก็บเงินค่าธรรมเนียมต่างๆ

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
Y_IN	Character	2	ปีที่เข้า
PRICR	Numeric	20	อัตราค่าลงทะเบียนบรรยาย
PRICA	Numeric	6	อัตราค่าลงทะเบียนปฏิบัติ
PRICSPORT	Numeric	3	อัตราค่าบำรุงการศึกษา
PRICB	Numeric	3	อัตราค่าบำรุงห้องสมุด
PRICESTU	Numeric	6	อัตราค่าบำรุงศูนย์

## 12. เพิ่มตารางค่าลงทะเบียน

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
OLDID	Character	10	รหัสประจำตัวคน
Y_IN	Character	2	ปีที่เข้า
TOTALR	Numeric	20	ค่าลงทะเบียนบรรยาย
TOATALA	Numeric	6	ค่าลงทะเบียนปฏิบัติ
TOTALSPORT	Numeric	3	ค่าบำรุงกีฬา
TOTALPA	Numeric	3	ค่าบำรุงห้องพยาบาล
TOTALB	Numeric	3	ค่าบำรุงห้องสมุด
TOTALSTU	Numeric	6	ค่าลงทะเบียนนักศึกษาใหม่

### 9.4.2 ออกแบบ Database สำหรับ Warehouse

มีขั้นตอนที่ต้องทำดังนี้

#### 1. กำหนด Data Model ของ ระบบงานเดิม ในระดับสูงสุด

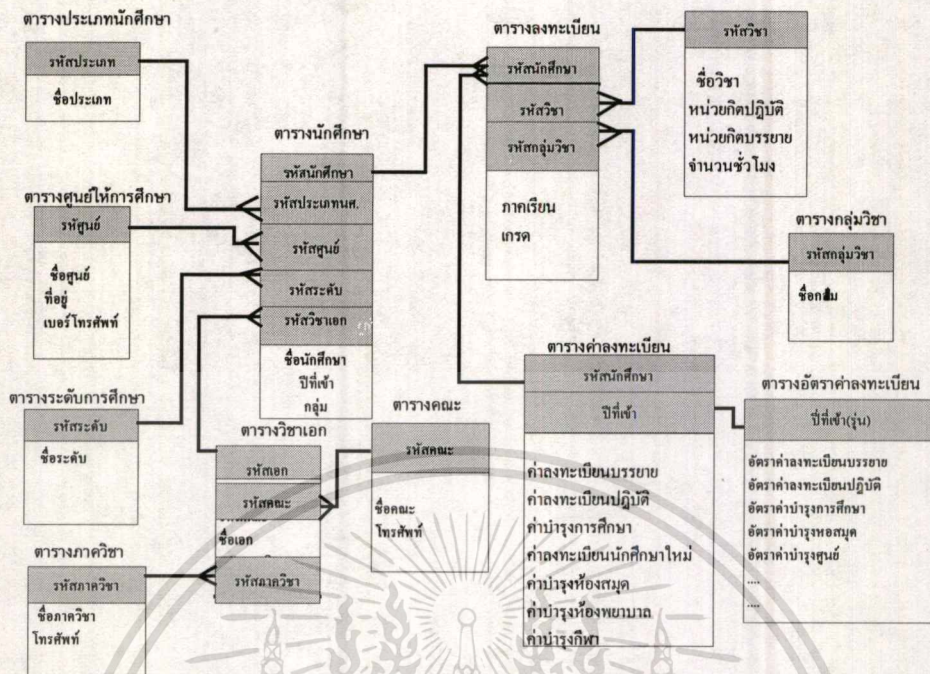
จากโครงสร้างของฐานข้อมูลและระบบงานเดิมสามารถแสดงความสัมพันธ์ของข้อมูล

ภายในระบบและกำหนดขอบเขตของข้อมูล แสดงด้วยแผนภาพ ERD ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 9.3 Star Schema ของระบบทะเบียนการเงิน

เป็นแสดงความสัมพันธ์ของข้อมูลทั้งหมดในรูปแบบ star Schema ซึ่งในกรณีศึกษาจะไม่สรุปข้อมูลเกี่ยวกับรายวิชา ในขั้นตอนนี้เป็นขั้นตอนที่สำคัญเพราะจะต้องมีการเลือกจำนวนตารางที่ต้องการ จะต้องมี join table ได้เข้าด้วยกันบ้าง จะเลือก table ไດ หรือ attribute มาเป็นหัวข้อเรื่องที่ต้องการใน data warehouse การตัดสินใจในส่วนนี้เป็นหน้าที่ของผู้ออกแบบที่จะต้อง กำหนดปัญหาหรือความต้องการก่อนว่า ระบบที่กำลังทำอยู่ต้องการอะไรบ้างเช่น

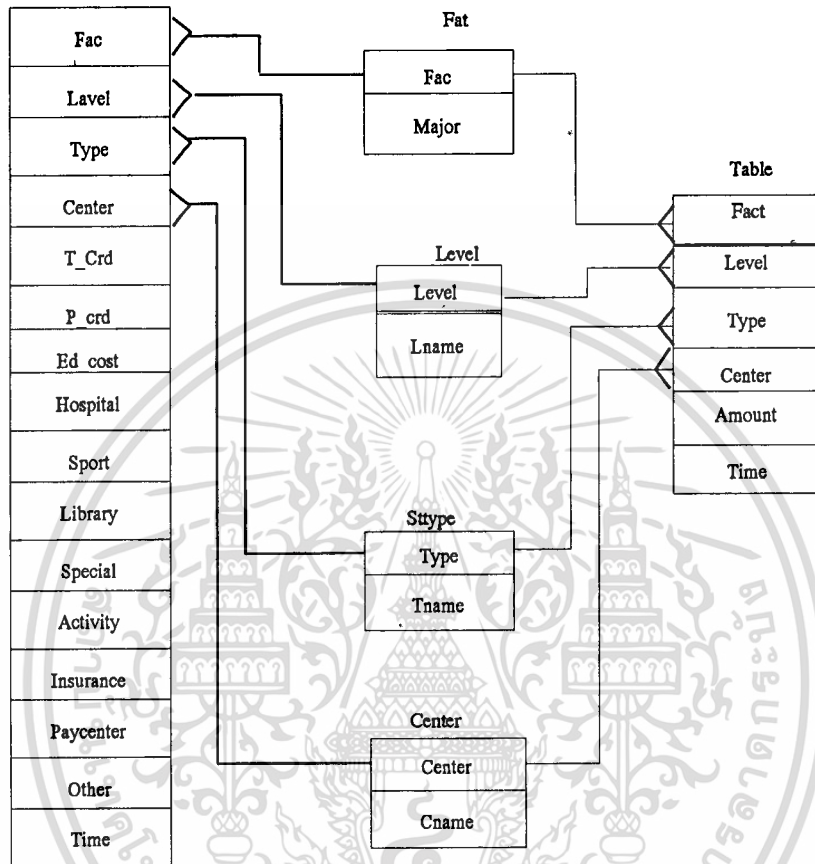
กำหนดปัญหา

1. สถาบันต้องการทราบอะไรบ้าง (Measure )
2. งานที่ทำอยู่มีการมีช่วงของการทำงานเป็นอย่างไรบ้าง
3. ผู้บริหารต้องการข้อมูลอะไรบ้าง

ตั้งคำถามเกี่ยวกับสิ่งที่ต้องการ

1. จำนวนนักศึกษา แต่ละคณะมีจำนวนเท่าไร
2. จำนวนนักศึกษา แบ่งตาม คณะ ประเภท ระดับ ศูนย์
3. จำนวนนักศึกษา แต่ละปี แต่ละ ศูนย์เป็นเท่าไร
4. จำนวนเงินรายได้จากค่าบำรุงการศึกษา แต่ละปี
5. จำนวนเงินรายได้แบ่งตามศูนย์ ตามคณะ ประเภท ระดับเป็นเท่าไร

จากปัญหาและความต้องการ พิจารณาการสร้าง Fact Table และ Dimension Table เพื่อมุมมองหลายมิติได้ดังนี้



รูปที่ 9.4 Star Schema ของระบบ

5. สร้างฐานข้อมูลของคลังข้อมูล ( Data Warehouse) โดยเลือกเฉพาะหัวข้อที่ต้องการมี และออกแบบให้มีความยืดหยุ่นในการใช้งานสูง มีโครงสร้างดังนี้

#### 1. ตารางระดับการศึกษา

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
Level	Character	1	รหัสระดับ
Lname	Character	10	ชื่อระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. ตารางศูนย์การศึกษา

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
CENTER	Character	1	รหัสศูนย์
CNAME	Character	20	ชื่อศูนย์

## 3. ตารางประเภทนักศึกษา

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
TYPE	Character	1	รหัสระดับ
TNAME	Character	10	ชื่อระดับ

## 4. ตารางคณะ

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
FAC	Character	1	รหัสคณะ
MAJOR	Character	10	ชื่อคณะ

## 5. ตารางสรุปจำนวนนักศึกษา

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
FAC	Character	1	รหัสคณะ
LEVEL	Character	1	รหัสระดับ
TYPE	Character	1	รหัสประเภท
CENTER	Character	1	รหัสศูนย์
AMOUNT	Numeric	10	จำนวนนักศึกษา
TIME	Date	8	วันเดือนปี

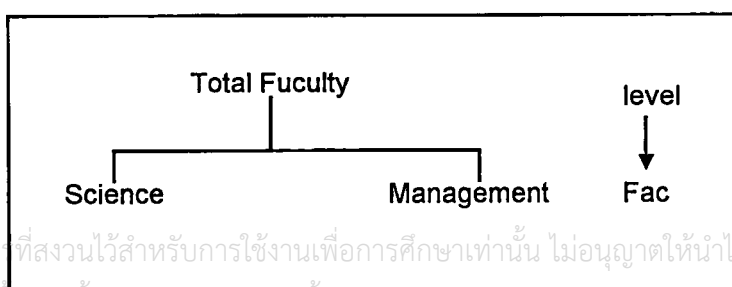
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6. ตารางสรุปจำนวนเงินค่าลงทะเบียนแยกเป็นรายการ

ชื่อฟิลด์	ชนิด	ความกว้าง	ความหมาย
FAC	Character	1	รหัสคณะ
LEVEL	Character	1	รหัสระดับ
TYPE	Character	1	รหัสประเภท
CENTER	Character	1	รหัสศูนย์
T_CR	Numeric	10	ค่าลงทะเบียนบรรยาย
P_CR	Numeric	10	ค่าลงทะเบียนปฏิบัติ
ED_COST	Numeric	10	ค่าบำรุงการศึกษา
HOSPITAL	Numeric	10	ค่าบำรุงโรงพยาบาล
SPORT	Numeric	10	ค่ากีฬา
LIBRARY	Numeric	10	ค่าบำรุงห้องสมุด
SPECIAL	Numeric	10	ค่าธรรมเนียมพิเศษ
ACTIVITY	Numeric	10	ค่ากิจกรรม
INSURANCE	Numeric	10	ค่าประกันของเสียหาย
PAYCENTER	Numeric	10	ค่าบำรุงศูนย์
OTHER	Numeric	10	อื่น ๆ
TIME	Date	8	วันเดือนปี

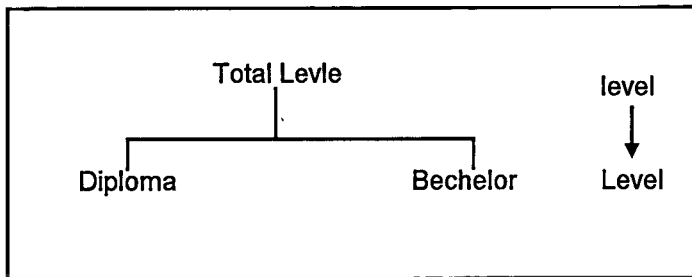
9.4.3 ออกแบบ Model โดยกำหนด Dimension เพื่อมุมมองข้อมูลได้ทั้งหมด 4 Dimension  
 ดังนี้ กำหนดในรูปแบบ hierachies ของข้อมูล

#### 1. Dimension Faculty

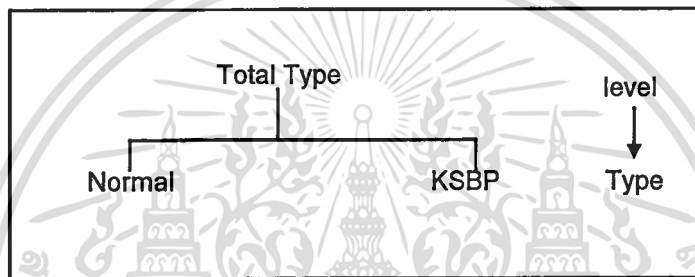


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

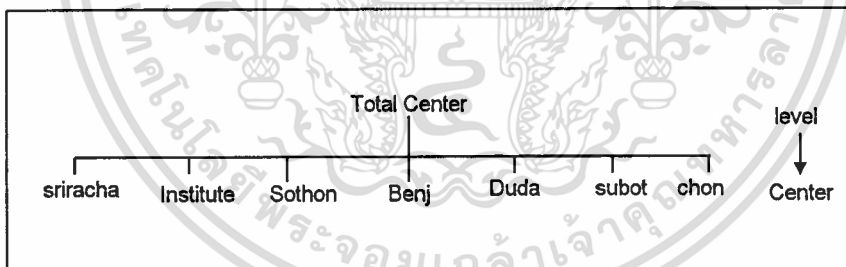
## 2. Dimension Level



## 3. Dimension Type

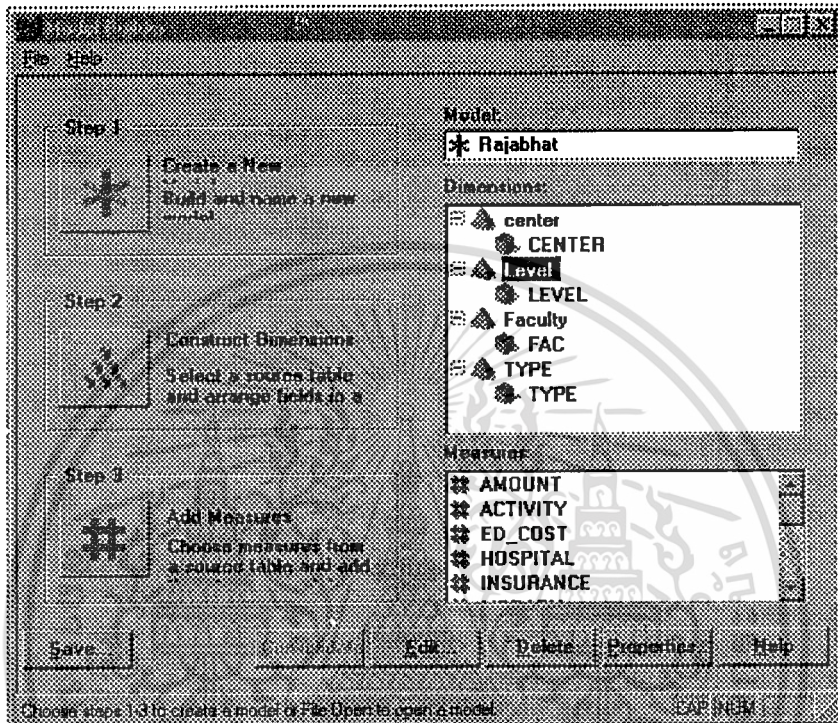


## 4. Dimension Center



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 9.4.4 สร้าง Model โดยใช้ Model Builder ของ Pilot Decision Support



รูปที่ 9.5 แสดงการสร้าง Model และ Dimension

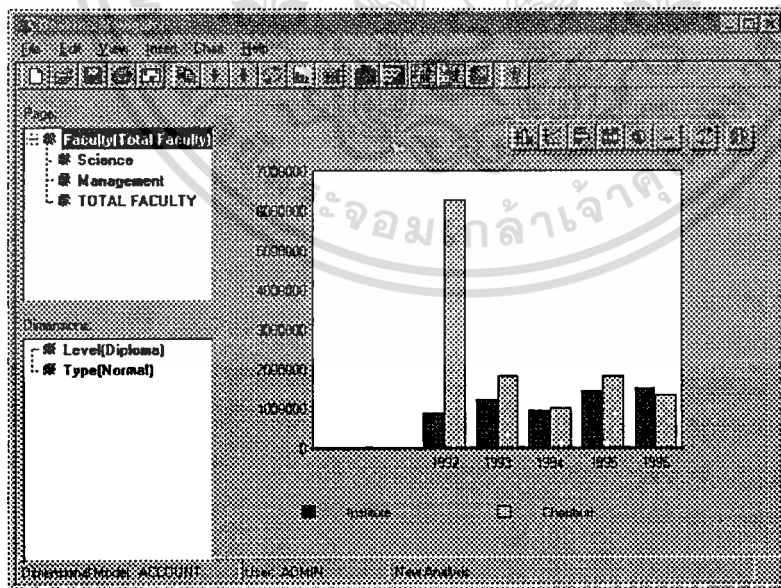
จาก Dimension ที่กำหนดไว้สามารถที่จะสร้าง Model โดยใช้ Model Builder ได้ดังรูปที่ 9.5 ซึ่งมีทั้งหมด 4 Dimension คือ Center, Level, Faculty และ Type ส่วน Measure ได้แก่ Amount, Activity และ Field อื่น ๆ ที่ต้องการ

#### 9.4.5 วิเคราะห์ข้อมูลโดยใช้ Pilot Desktop

ตัวอย่างที่ 1 ใช้ Navigator Ad hoc ขอดูจำนวนเงินค่านำร่องห้องสมุดตั้งแต่ปี 1992-1996 ของศูนย์สถาบัน และศูนย์ชลบุรี และดูคณะวิทยาศาสตร์ และ วิทยาการจัดการ

	A	B	C	D	E	F
Faculty(Total Faculty)		Library				
Science		Institute	Chonburi			
Management	1992	889,800.00	8,269,800.00			
TOTAL FACULTY	1993	1,200,000.00	1,800,000.00			
	1994	920,000.00	1,000,000.00			
	1995	1,400,000.00	1,800,000.00			
	1996	1,500,000.00	1,320,000.00			

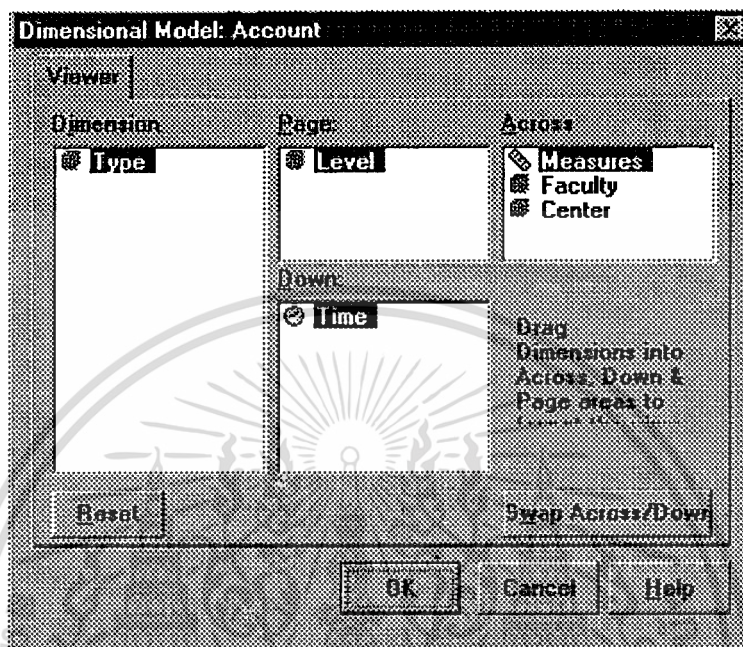
รูปที่ 9.6 แสดงการ View ข้อมูล



รูปที่ 9.7 การแสดงข้อมูลด้วยกราฟ

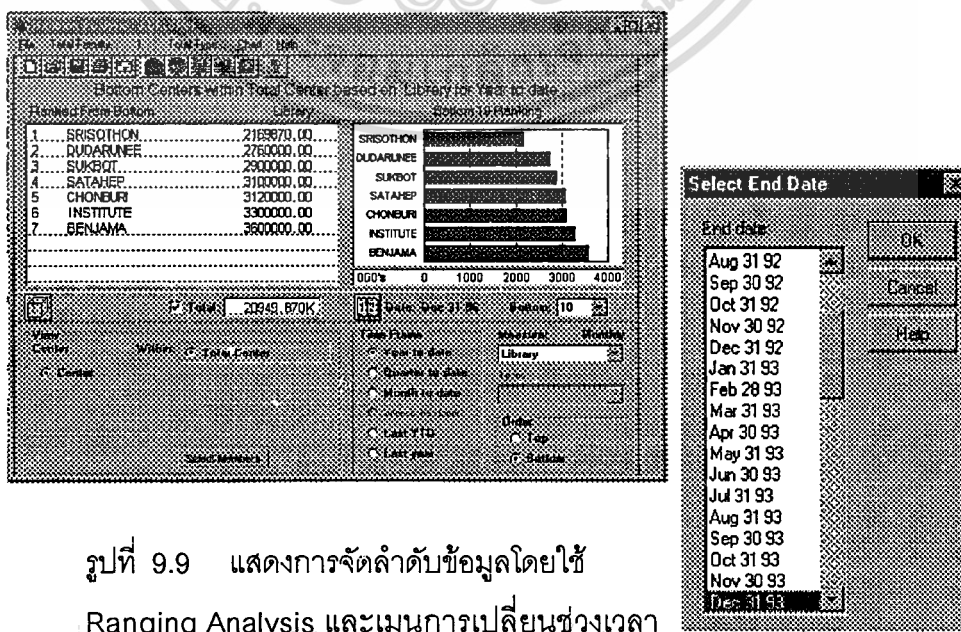
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเลือกดูสามารถหมุน (Rotate) ตามมุมมองที่ผู้ใช้ต้องการโดยกำหนดในเมนูต่อไปนี้



รูปที่ 9.8 แสดงกำหนดมุมมอง(View)

ตัวอย่างที่ 2 ใช้ Ranging ต้องการดูลำดับจำนวนเงินบำรุงห้องสมุดเรียงจากน้อยไปมากของแต่ละศูนย์การศึกษาโดยผู้ใช้สามารถเปลี่ยน measure และวันเวลา ได้ ตามที่ ๑ การ



รูปที่ 9.9 แสดงการจัดลำดับข้อมูลโดยใช้

Ranging Analysis และเมนูการเปลี่ยนช่วงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาดูเท่านั้น ไม่อนุญาตให้เผยแพร่ขึ้นตามการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 9.4.6 ข้อดีข้อเสียของการใช้งาน

การใช้งานโปรแกรม Pilot Decision Support ทำให้มีความสะดวกรวดเร็วในการวิเคราะห์ข้อมูล โดยที่ผู้ใช้ ไม่ต้องรอการเขียนโปรแกรมจากโปรแกรมเมอร์ทุกกรณี Pilot จะเตรียมโปรแกรม Desktop มาสำหรับสนับสนุนการใช้งานเฉพาะกิจ การเรียกใช้ข้อมูลสามารถทำได้รวดเร็ว ซึ่งมีข้อดีสำหรับผู้ที่มีทักษะในการใช้ SQL น้อย ไม่สามารถทำ SubQuery ได้ ในกรณีที่มีความซับซ้อนของข้อมูลมาก ๆ โปรแกรมในส่วนนี้จะช่วยสนับสนุนการใช้งานได้เต็มที่

ข้อเสียของการใช้งาน ต้องเสียเวลาเตรียมข้อมูลค่อนข้างนานเพราะเป็นข้อมูลสรุป โดยฐานข้อมูลเดิมที่เป็น Transaction จะมีจำนวนมากทำให้การรวบรวมข้อมูลเข้า จำนวนข้อมูลที่เก็บใน data Warehouse มีความซ้ำซ้อนสูงเปลืองเนื้อที่ในการจัดเก็บ และข้อมูลที่อยู่ใน Multidimensional Model ก็เป็นข้อมูลจำนวนมาก การจัดเก็บก็ต้องใช้ Disk ที่มีความจุสูง และการเรียกใช้งาน ก็ต้องใช้ hardware ที่มีประสิทธิภาพสูงตามไปด้วย

## บทที่ 10

### สรุปผลการศึกษา

#### 10.1 ผลการศึกษา

จากการศึกษาวิธีการสร้างคลังข้อมูลสำหรับสนับสนุนการตัดสินใจ หรือ สร้าง Data Warehouse โดยใช้โปรแกรม Pilot Decision Support ช่วยในการสร้าง Model และวิเคราะห์ข้อมูลในรูปแบบหลายมิติ Multidimensional Model สรุปได้ดังนี้

1. Data Warehouse สร้างมาจากฐานข้อมูลเชิงสัมพันธ์
2. การสร้างคลังข้อมูลต้องวิเคราะห์ความต้องการใช้ข้อมูลของผู้บริหาร จากนั้นเลือกข้อมูลจากฐานปฏิบัติงานที่เป็น Online Transaction และเลือกเฉพาะหัวข้อที่ต้องการนำมาสรุปแปลงข้อมูลก่อนนำเข้าเก็บใน Data Warehouse
3. การเรียกใช้ข้อมูลใน Data Warehouse สามารถเรียกใช้ได้โดยผ่าน Tools
4. การเก็บข้อมูลในรูปแบบหลายมิติเพื่อประสิทธิภาพของการเรียกใช้ข้อมูลวิเคราะห์ ซึ่งสร้างขึ้นมาเพื่อแก้ปัญหาสำหรับผู้ที่ไม่มีความรู้เกี่ยวกับภาษาเรียกค้น (SQL)
5. Pilot Decision Support เป็นโปรแกรมหรือ Tools ที่ช่วยในการวิเคราะห์ข้อมูลได้อย่างมีประสิทธิภาพทั้งนี้ขึ้นอยู่กับกรออกแบบ Model สำหรับการใช้งานด้วย
6. การวิเคราะห์ data warehouse tools ใน 3 รูปแบบ คือ Data Structure, Data Integrity, Data manipulate 3 หัวข้อของ Data warehouse อาศัยความสามารถจากการจัดการฐานข้อมูลเชิงสัมพันธ์ซึ่งพื้นฐานของข้อมูลใน data warehouse และเพิ่มเทคนิคการ indexing ที่เรียกว่า Hash Index เพื่อการค้นหาข้อมูลที่เร็วกว่า
7. กรณีศึกษาการสร้าง data warehouse สำหรับสนับสนุนการตัดสินใจของผู้บริหารในสถาบันราชภัฏยะเชิงเทรา ทำได้ในขอบเขตที่กำหนดไว้

#### 10.2 ประโยชน์ที่ได้รับ

จากการศึกษาการสร้างคลังข้อมูลสำหรับสนับสนุนการตัดสินใจ ทำให้ได้รับความรู้เกี่ยวกับความต้องการใช้ข้อมูลสำหรับสนับสนุนการตัดสินใจ และปัญหาที่เกิดขึ้นในองค์กรเกี่ยวกับการจัดการข้อมูล ที่มีปริมาณมาก ๆ แต่ไม่สามารถตอบคำถามได้ เนื่องจากการจัดการไม่ดี เมื่อมีแนวคิดเกี่ยวกับคลังข้อมูลเกิดขึ้นทำให้มีแนวทางในการจัดการกับข้อมูลเหล่านั้น เพื่อนำมาใช้ให้เกิดประโยชน์สูงสุด ในการศึกษาในครั้งนี้ทำให้เกิดแนวคิดที่จะนำหลักการของ data warehouse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปประยุกต์ใช้กับระบบงานเพื่อพัฒนาการใช้ข้อมูลขององค์กรให้มีประสิทธิภาพสูงขึ้น และเกิดทักษะในการศึกษาค้นคว้าข้อมูลเรื่องใหม่ ๆ จากเอกสาร วารสาร และตำราต่าง ๆ

### 10.3 ข้อเสนอแนะ

ในการประยุกต์ใช้ data warehouse นั้นเป็นเรื่องที่ค่อนข้างจะต้องใช้เวลามากพอสมควร และต้องใช้บุคลากรหรือผู้ออกแบบข้อมูลที่มีความเข้าใจในระบบงาน เพื่อที่จะสามารถสร้าง Model เพื่อสนับสนุนความต้องการของผู้ใช้ได้ตรงเป้าหมาย และในการข้อมูลใน data warehouse ก็จำเป็นที่ผู้ใช้จะต้องเลือกเครื่องมือ (Software Tools ) ที่ช่วยในการจัดการที่มีประสิทธิภาพ สามารถค้นหาข้อมูลได้ทุกรูปแบบ ก็จะทำให้ผู้ใช้เกิดความรู้สึกพอใจกับคำตอบ ซึ่ง tools ที่ผลิตขึ้นมาเพื่อใช้งานใน Data Warehouse มีหลาย ๆ บริษัททั้งนี้ขึ้นอยู่กับผู้ใช้งานว่าจะเลือก Tools ที่มีคุณสมบัติอย่างไรให้กับองค์กรของตนเอง

)



## บรรณานุกรม

- Appleton, Elaine L. "Use Your Data Warehouse to Compete". Datamation, may 15, 1996 : 34 - 51.
- Baum , David. "Data Warehouse". Oracle magazine, march 1996 : 34 - 43.
- Berson , Alex. Client/server architecture. New York : MC Graw - Hill, 1992.
- Bulos, Dan. "OLAP database design A New Dimension". Database Programming & Disign., June 1996 : 33 - 37.
- Celko ,Joe and Jackie McDonald. "Don't warehouse Dirty Data". Datamation, October 15 , 1995 : 42 - 44.
- Darling, Charles B. "Datamining for the masses". Datamation, february 1997 : 52- 67.
- Date , C.J. An introduction to database systems. 6nd ed. USA. : Addison - weskey Publishing Company, Inc., 1995.
- Freeman , Eva. "Dimensional Hierarchies and Formulas". Database programing & Design., June 1997 : 60 - 63.
- Gelman , Stacey J. and W. Douglas Peck "Bringing Business Information to AT & T Network Systems Through a Data Warehouse". AT & T Technical journal, march 1996 : 68 - 77.
- Greenfield , Larry. "Don't let Data Warehousing gotochas Getcha". Datamation, march 1996 : 76 - 80.
- Griffin, Jane. "Avoid data warehousing maintenance migraines". Datamation, August 1996 : 74 - 79.
- Inmon, William H. "Building the Data warehouse". QED Publishing, group, 1993.
- Inmon,Willium."Does yourdatamart Vendor care about your architecture?".Datamation, march 1997 : 105 - 107.
- Kelly, Sean. Data Warehousing. England : Johnwiley & Sons Ltd, 1996.
- Kelly, Sean. Data Warehousing. England : Johnwiley & Sons Ltd, 1996.
- Kerth ,Henry F. Database System Concepts. 2<sup>nd</sup> ed. London : Abraham Silberschatz., 1986.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Pilot Software. Introduction Pilot Designer. USA. : n.p., 1996.
- Pilot Software. Introduction Pilot Destop. USA. : n.p., 1996.
- Pilot Software. Introduction to Pilot. USA. : n.p., 1996.
- Poe, Vidette. Building a data warehouse for Decision Support. United States of America : n.p., 1996.
- Radding, Alan. "Support Decision Makers with a Data Warehouse". Datamation, march 15, 1995.
- Raden, Nell. "Modeling the data Warehouse". Multidimensional Modeling, August 1997 : 1 - 6.
- Schreiber ,Richard. "Getting Distributed Apps to Communicate : let 's Talk middleware". Datamation., July 1995 : 57 - 60.
- Schreiber, Richard. "Middleware Demystified" Datamation., April 1995 : 41 - 45.
- Strehlo ,Kevin. "Data Warehousing : Avoid Planned Obsolescence" Datamation, January 15, 1996 : 32 - 37.
- Swift , Ronald S. "Planning and Designing a Data Warehouse". Computer Associates International, July 13 - 18, 1997 : 1 - 16.
- Thomsen ,Erik . "Breaking the Three - Dimensional Barrier". Database programming & Design., May 1997 : 59 - 61.
- Thomsen ,Erik "Dimensional Modeling : An Analytical Approach". Database programming & Design., march 1997 : 29 - 35.
- Varney, Sarch E. "Datamarts : Coming to an it Mall Near You! ". Datamation., June 1996 : 44 - 46.
- Varney, Sarch E. "Datawebs! Link the web to Your Legacy Data and Apps". Datamation., April 1996 : 38 - 53.
- Weldon , Jay Louis. "Choosing Tools for Multidimensional Data". Database programming & Design.,
- Weldon, Jay Louis. "Managing Multidimensional Data : Harnessing The Power". Database programming & Design., August 1995 : 25 - 33.

Watterson , Kaven . "IBM does Data". Datamation, July 1996 : 64 - 67.

Wright, Willam . "Information Visualzation : The Fourth Dimension" Database programming & design, April 1997 : 50 - 63.

Xenakis , John J. "Multidimensional Databases". Application Development Strategies., volume VI, No.4.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน

ชื่อ	นางสาวเพ็ญศรี บั๊กกะสีนัง
วันเดือนปีเกิด	26 กันยายน 2511
สถานที่เกิด	จังหวัดมหาสารคาม
ประวัติการศึกษา	สำเร็จการศึกษาระดับปริญญาตรี วท.บ. (วิทยาการคอมพิวเตอร์) จากสถาบันราชภัฏ พระนคร บางเขน กรุงเทพฯ ปี พ.ศ. 2534
ประวัติการทำงาน	- ปี พ.ศ. 2535-2537 รับราชการตำแหน่ง อาจารย์ 1 ระดับ 3 สถาบันราชภัฏภูเก็ต -ปี พ.ศ.2537- ปัจจุบัน รับราชการตำแหน่ง อาจารย์ 1 ระดับ 5 สถาบันราชภัฏยะเซียงเทธา อ.เมือง จ.ยะเซียงเทธา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้