ฐานข้อมูลที่มีความปลอดภัยหลายระดับ

MULTILEVEL SECURE DATABASE

จันทิมา จักปราณีวิรัตน์
เชฏฐพล ตระการกิจวิชิต
อธินนท์ มานะกุลอิสสระ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

ปริญญานิพนธ์ปีการศึกษา 2553

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ฐานข้อมูลที่มีความปลอดภัยหลายระดับ

MULTILEVEL SECURE DATABASE

ผู้จัดทำ

1. นางสาวจันทิมา จักปราณีวิรัตน์ รหัสนักศึกษา 50010209
2. นายเชฏฐพล ตระการกิจวิชิต รหัสนักศึกษา 50010386
3. นายอธินนท์ มานะกุลอิสสระ รหัสนักศึกษา 50011825

อาจารย์ที่ปรึกษา

(รศ.ดร.ศุภมิตร จิตตะยโศธร)

ฐานข้อมูลที่มีความปลอดภัยหลายระดับ

นางสาวจันทิมา จักปราณีวิรัตน์ 50010209
นายเชฎฐพล ตระการกิจวิชิต 50010386
นายอธินนท์ มานะกุลอิสสระ 50011825
รศ.ดร.ศุภมิตร จิตตะยโศธร อาจารย์ที่ปรึกษา
ปีการศึกษา 2553

บทคัดย่อ

เนื่องจากในปัจจุบันนี้ฐานข้อมูลที่มีส่วนมาก จะเป็นฐานข้อมูลที่มีความปลอดภัยเพียงแค่ระดับเดียว นั่นคือถ้าผู้ใช้งานเห็นข้อมูลในตารางจะเห็นข้อมูลทั้งหมด โดยไม่อาจทราบได้เลยว่าข้อมูลที่มีอยู่นั้นเป็นของผู้ใช้งานท่านใด และมีระดับความปลอดภัยของผู้ใช้งานท่านนั้นอย่างไร การที่จะทำให้ทราบถึงข้อมูลเหล่านี้จะต้องใช้ฐานข้อมูลที่มีความปลอดภัยหลายระดับ (Multilevel Secure Database หรือ MLS) ซึ่งในปัจจุบัน หากบริษัทใดมีความต้องการที่จะใช้ฐานข้อมูลประเภทนี้จะต้องร้องขอไปยังผู้ขายผลิตภัณฑ์ให้วางระบบให้โดยเฉพาะ ด้วยเหตุนี้เองทำให้ฐานข้อมูลที่มีระดับความปลอดภัยหลายระดับนี้มีราคาสูง ยังไม่เป็นที่แพร่หลายอีกด้วย

โครงงานนี้จึงจัดทำขึ้นเพื่อทำการศึกษาการทำงานของระบบฐานข้อมูลที่มีความปลอดภัยหลายระดับ โดยใช้แนวคิดจากนักวิจัยท่านต่างๆที่ได้มีแนวคิดโมเดลความปลอดภัย โดยยึดถือหลักการความปลอดภัยของ Bell-LaPadula เป็นหลัก เพื่อนำแนวคิดเหล่านั้นมาทำการพัฒนาและแก้ไขจุดบกพร่องที่ยังคงมีอยู่ให้หมดไป แล้วออกแบบคำสั่ง MLS SQL เพื่อให้รองรับการทำงานของแนวคิด MLS ทั้งหมด จากนั้นทำการสร้างแอพพลิเคชั่นขึ้นมาเพื่อรองรับแนวคิดนี้ เพื่อติดต่อไปยังฐานข้อมูลพื้นฐาน ให้เสมือนว่าเป็นฐานข้อมูลที่มีความปลอดภัยหลายระดับ โดยจากรองรับการป้อนคำสั่ง MLS SQL จากผู้ใช้งาน เพื่อมาทำการแปลงเป็นคำสั่ง SQL ปกติ ซึ่งอาจจะมีมากกว่า 1 คำสั่ง ให้สามารถติดต่อไปยังผลิตภัณฑ์ฐานข้อมูลทั่วไปๆ ได้ และยังสามารถส่งค่าข้อมูลที่ผู้ใช้งานต้องการ กลับมายังผู้ใช้งานได้อย่างถูกต้อง อีกทั้งจะนำเอาความสามารถต่างๆที่ผลิตภัณฑ์ฐานข้อมูลที่ได้สร้างไว้เพื่อดูแลความปลอดภัยมาใช้งานร่วมกับตัวแอพพลิเคชั่นด้วย โดยโครงงานนี้มีจุดมุ่งหมายเพื่อเป็นแนวทางให้ผู้ควบคุมระบบฐานข้อมูลสามารถนำไปใช้ได้โดยไม่ต้องเสียค่าใช้จ่ายเป็นจำนวนมาก

Multilevel Secure Database

Ms. Chantima	Chakpraneewirath	50010209
Mr. Chastapont	Trakarnkijvichit	50010386
Mr. Atinon	Manakulissara	50011825
Assoc. Prof. Dr. Suphamit	Chittayasothorn	Advisor

Academic Year 2010

ABSTRACT

In the present, lots of database system use only single-level database system. So user can see all the fact in the schema if he has permission. But he can't know that which fact has been inserted by whom with what classification. By this point if this person wants to know this information he should use Multilevel Secure Database (MLS). The Dealer of product of some database system can implement MLS for the users only if the users request but in the higher cost too. Because of the cost and limit of some policy the MLS isn't much popular.

In this project we research old documents which involved MLS database by using main constraint of Bell-LaPadula and other models. And then develop the application that can get the MLS SQL from the users and translate them to standard SQL to contact with DBMS and return the correct result to the users. Our purpose is to make the idea for another developer to make MLS database system by using basic DBMS.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้อย่างดีด้วยคำแนะนำ คำปรึกษา และความกรุณาจาก รศ.ดร.ศุภมิตร จิตตะยโศธร ซึ่งเป็นอาจารย์ที่ปรึกษาปริญญานิพนธ์ ข้าพเจ้ารู้สึกซาบซึ้งในความอนุเคราะห์ และขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

ขอขอบคุณคณะวิศวกรรมศาสตร์ที่ให้ความช่วยเหลือในเรื่องต่างๆ

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำปริญญานิพนธ์ ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากปริญญานิพนธ์ ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

จันทิมา จักปราณีวิรัตน์
เชฏฐพล ตระการกิจวิชิต
อธินนท์ มานะกุลอิสสระ

สารบัญ

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญตาราง

สารบัญตาราง (ต่อ)

ตาราง **หน้า**

สารบัญรูป

สารบัญรูป

รูป *หน้า*

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

เนื่องจากในปัจจุบันนี้ฐานข้อมูลที่มีส่วนมาก จะเป็นฐานข้อมูลที่มีความปลอดภัยเพียงแค่ระดับเดียว นั่นคือถ้าผู้ใช้งานเห็นข้อมูลในตารางจะเห็นข้อมูลทั้งหมด โดยไม่อาจทราบได้เลยว่าข้อมูลที่มีอยู่นั้นเป็นของผู้ใช้งานท่านใด และมีระดับความปลอดภัยของผู้ใช้งานท่านนั้นอย่างไร การที่จะทำให้ทราบถึงข้อมูลเหล่านี้จะต้องใช้ฐานข้อมูลที่มีความปลอดภัยหลายระดับ (Multilevel Secure Database หรือ MLS) เนื่องจากเอ็มแอลเอสนั้นจะมีการเก็บข้อมูลเหล่านี้ไว้ในโครงตาราง เพื่อให้เกิดการตรวจสอบได้ ซึ่งผลิตภัณฑ์ระบบฐานข้อมูลในปัจจุบันสามารถที่จะทำได้ แต่จะต้องทำเฉพาะกับหน่วยงานที่ต้องการเท่านั้น หากบริษัทใดมีความต้องการที่จะใช้ฐานข้อมูลประเภทนี้จะต้องร้องขอไปยังผู้ขายผลิตภัณฑ์ให้วางระบบให้โดยเฉพาะ อีกทั้งเอกสารต่างๆที่อ้างถึงคุณสมบัติเหล่านี้ยังไม่เปิดเผยมากนัก และผลิตภัณฑ์ที่ใช้จริงยังห้ามนำออกจากประเทศอีกด้วย ด้วยเหตุนี้เองทำให้ฐานข้อมูลที่มีระดับความปลอดภัยหลายระดับนี้มีราคาสูง ยังไม่เป็นที่แพร่หลายอีกด้วย

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อศึกษาถึงโมเดลความปลอดภัยหลายระดับที่ได้มีการวิจัยมาแล้ว
2) เพื่อศึกษาถึงการนำแนวคิดของระบบฐานข้อมูลที่มีความปลอดภัยหลายระดับไปใช้งาน
3) เพื่อทำการออกแบบและพัฒนาส่วนติดต่อกับฐานข้อมูล และแอพพลิเคชั่น โดยเป็นการแปลงคำสั่งที่ผู้ใช้ส่งมาให้เป็นคำสั่งที่ทำงานบนดีบีเอ็มเอสได้

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1) ความรู้ความเข้าใจในโมเดลความปลอดภัยที่นักวิจัยได้พัฒนามาแล้ว
2) ได้เห็นถึงประโยชน์และการนำไปใช้จริงของระบบฐานข้อมูลที่มีความปลอดภัยหลายระดับในโลกปัจจุบัน
3) สามารถพัฒนาแอพพลิเคชั่น และส่วนติดต่อกับฐานข้อมูลที่มีความสามารถแปลง และปรับเปลี่ยนเอ็มแอลเอสเอสคิวแอลให้เป็นเอสคิวแอลปกติได้

1.4 ขอบเขตของโครงงาน

1) ออกแบบและพัฒนาภาษาสำหรับให้ผู้ใช้งานติดต่อกับฐานข้อมูลที่มีความปลอดภัยหลายระดับ (เอ็มแอลเอส)
2) ออกแบบและพัฒนาส่วนติดต่อกับฐานข้อมูล และผู้ใช้งาน เพื่อแปลงภาษาเอ็มแอลเอสเอสคิวแอลเป็นภาษาเอสคิวแอลปกติ
3) ออกแบบและพัฒนาแอพพลิเคชั่นที่ใช้งานกับฐานข้อมูลที่มีความปลอดภัยหลายระดับ

1.5 ขั้นตอนการดำเนินงาน

1) ทำการศึกษาวิจัยโมเดลโครงสร้าง และการทำงานของฐานข้อมูลที่มีความปลอดภัยหลายระดับจากผู้ที่เคยศึกษาวิจัยมาแล้ว
2) ทำการศึกษาวิจัยภาษาที่นำมาใช้ติดต่อกับฐานข้อมูลที่มีความปลอดภัยหลายระดับ
3) ศึกษาการนำแนวคิดระบบฐานข้อมูลที่มีความปลอดภัยหลายระดับไปใช้งานของดีบีเอ็มเอสในปัจจุบัน
4) ศึกษาถึงการนำไปใช้งานจริงในเชิงธุรกิจ หรืออย่างอื่นที่เกี่ยวข้องในชีวิตจริง
5) ออกแบบและพัฒนาภาษาเอ็มแอลเอสเอสคิวแอลที่จะใช้งานกับแอพพลิเคชั่นที่จะสร้างขึ้น
6) ออกแบบและพัฒนาส่วนติดต่อกับฐานข้อมูลและผู้ใช้งาน
7) ออกแบบและพัฒนาแอพพลิเคชั่นที่จะใช้ทดลองติดต่อกับฐานข้อมูลผ่านส่วนติดต่อที่ได้พัฒนาไว้แล้ว

1.6 ส่วนประกอบของรายงาน

บทที่ 2 จะกล่าวถึงความเป็นมา และนำเสนอโมเดลที่เกี่ยวข้องกับระบบฐานข้อมูลที่มีความปลอดภัยหลายระดับที่ถูกพัฒนาต่อๆกันมา

บทที่ 3 จะนำเสนอคุณสมบัติของดีบีเอ็มเอสบางตัวที่มีการนำแนวคิดของความปลอดภัยดังกล่าวไปใช้งาน

บทที่ 4 จะกล่าวถึงการนำระบบฐานข้อมูลที่มีความปลอดภัยหลายระดับไปใช้งานในเชิงธุรกิจ

บทที่ 5 จะนำเสนอภาษาเอ็มแอลเอสที่จะมีการนำมาให้ผู้ใช้งาน ใช้ติดต่อกับฐานข้อมูลผ่านทางส่วนติดต่อฐานข้อมูล

บทที่ 6 จะสรุปการทดลองและผลการทดลอง

บทที่ 7 จะสรุปและข้อเสนอแนะ

บทที่ 2
ฐานข้อมูลที่มีความปลอดภัยหลายระดับ

ถึงแม้ว่าระบบฐานข้อมูลที่มีการใช้งานในปัจจุบันสามารถรองรับรูปแบบของฐานข้อมูลในลักษณะที่เหมาะสมกับความต้องการของงานได้หลายประเภท แต่ข้อมูลที่มีความต้องการปิดเป็นความลับที่เป็นข้อมูลที่มีความอ่อนไหวต่อสถานภาพขององค์กร ข้อมูลประเภทนี้มีข้อจำกัดในการเผยแพร่แก่บุคคลอื่น หรือแม้แต่บางหน่วยงานในองค์กรเอง ฉะนั้นในช่วงสิบปีที่ผ่านมาได้มีงานวิจัยหลายงาน ได้พูดถึงการออกแบบฐานข้อมูลเพื่อให้เหมาะสมกับข้อมูลในลักษณะนี้ โดยฐานข้อมูลที่มีความปลอดภัยหลายระดับ (Multilevel Security) หรือ MLS เป็นการจัดระบบความปลอดภัยที่ได้นำมาใช้ แต่เดิมเอ็มแอลเอสได้ออกแบบมาใช้กับงานทางทหาร เนื่องจากระบบทางการทหารมีข้อมูลที่เป็นความลับ และมีความละเอียดอ่อนของการเผยแพร่ข้อมูลสูง ข้อมูลทางการทหารบางอย่างจำเป็นต้องเก็บเป็นความลับเนื่องจากอาจส่งผลแก่ความมั่นคงของประเทศชาติ และมีความต้องการของการจัดเก็บของข้อมูลที่ต้องการความปลอดภัยจากผู้ที่ไม่ได้รับอนุญาตให้สามารถเข้าถึงข้อมูลส่วนนี้ได้ การจัดการสำหรับสิทธิ์การเข้าถึงข้อมูลนั้นต้องการฐานข้อมูลที่มีการแบ่งระดับของข้อมูลและผู้ใช้ เพื่อให้ข้อมูลในแต่ระดับสามารถเข้าถึงโดยผู้ใช้ที่มีสิทธ์เท่านั้น เช่นมีการแบ่งระดับชั้นของข้อมูลที่สามารถเปิดเผยได้ในเฉพาะหน่วยงาน หรือบุคคลบางกลุ่ม ซึ่งอาจแบ่งได้ตามชั้นยศ แต่แทนที่จะแยกข้อมูลเป็นความลับที่ไม่เปิดเผยแก่ระดับอื่นเลยเท่านั้น เราต้องการให้ข้อมูลนั้นกระจายออกไปได้ในบางส่วน หรือปกปิดในบางส่วน ทั้งยังทำให้การปกปิดนั้นสามารถให้ข้อมูลที่แตกต่างตามความเป็นจริงได้เป็นความต้องการของงานฐานข้อมูลในหลายๆ หน่วยงานไม่เฉพาะเพียงการทหารเท่านั้น รูปแบบของฐานข้อมูลประเภทนี้ยังมีการใช้งานกับองค์กรอื่นๆ ด้วยไม่ว่าจะเป็นในองค์กรธุรกิจถ้าหากต้องการปกปิดยอดขายที่แท้จริง หรือในทางการแพทย์ในงานตรวจรักษาของโรงพยาบาล ในกรณีที่ผู้ป่วย ป่วยเป็นโรคที่สังคมรังเกียจ หรือไม่เป็นความประสงค์ของผู้ป่วยที่ต้องการให้ ผู้อื่นล่วงรู้อาการป่วยของตน หรือในกรณีที่มีผลต่อสถาบันความมั่นคง ดังนั้นในกรณีนี้ข้อมูลต่างๆ ของผู้ป่วยโรงพยาบาลอาจต้องเก็บข้อมูลไว้เป็นความลับ

2.1 กฎเกณฑ์พื้นฐานของฐานข้อมูลที่มีระดับความปลอดภัยหลายระดับ

สิ่งที่แตกต่างจากฐานข้อมูลปกติคือแทนที่จะมีระดับของข้อมูลเพียงระดับเดียว ผู้ใช้ที่ได้รับอนุญาตเข้าถึงข้อมูลสามารถเข้าถึงข้อมูลที่เหมือนกันหมดคือมีระดับเดียวกันทั้งหมด แต่รูปแบบฐานข้อมูลเอ็มแอลเอสมีระดับของความปลอดภัย (security level) ต่างๆ กันในมากกว่าหนึ่งระดับ การกำหนดระดับความปลอดภัยมีการให้ระดับแก่ข้อมูล (object) ตัวอย่างเช่น แถว, คอลัมน์ และ

การแบ่งระดับให้แก่ผู้ใช้ (subject) ระดับความปลอดภัยของข้อมูล (classification level) เช่น ลับสุดยอด (top secret), เป็นความลับ (secret), ปกปิด (classified), ไม่เป็นความลับ (unclassified) เช่นเดียวกันกับการกำหนดระดับของผู้ใช้ (clearance level) ก็มีลักษณะเดียวกัน ทุกระดับมีการกำหนดระดับเอาไว้ด้วย เช่น top secret > secret > classified > unclassified ซึ่งหมายถึง top secret นั้นมีระดับสูงสุดและ unclassified อยู่ในระดับต่ำสุด ทั้งหมดได้ยึดเกณฑ์การเข้าถึงข้อมูลพื้นฐานของเบล และพาลาดูลา (Bell-LaPadula) ซึ่งได้วางไว้ดังนี้

1) คุณสมบัติความปลอดภัยพื้นฐาน (Simple Security Property) โดยผู้ใช้สามารถอ่านข้อมูลได้ถ้าหากระดับความปลอดภัยนั้นตรงกันหรือสูงกว่า
2) คุณสมบัติสตาร์ (*-Property) โดยผู้ใช้สามารถทำการเข้าถึงข้อมูลแบบเขียนได้ก็เมื่อระดับความปลอดภัยนั้นตรงกันหรือต่ำกว่า

ในกฎข้อแรกมีจุดมุ่งหมายในความสามารถการเข้าถึงข้อมูลได้เมื่อผู้ใช้นั้นมีระดับความปลอดภัยที่เท่ากันหรือสูงกว่าระดับความปลอดภัยของข้อมูล กล่าวได้ว่าผู้ใช้นั้นมีสิทธิ์ในการอ่านครอบคลุมเหนือข้อมูล ส่วนกฎข้อที่สองนั้นข้อมูลสามารถถูกแก้ไขได้ก็ต่อเมื่อระดับความปลอดภัยของผู้ใช้เท่ากันหรือต่ำกว่าระดับความปลอดภัยของข้อมูล คือนอกจากสามารถแก้ไขข้อมูลที่มีระดับตรงกันแล้ว ข้อมูลที่แก้ไขนี้ส่งผลถึงข้อมูลที่ระดับสูงกว่าสามารถอ่านได้ด้วย

2.2 ความหมายของข้อมูลในฐานข้อมูลเอ็มแอลเอส

เราสามารถแบ่งระดับชั้นของฐานข้อมูลได้อย่างง่ายด้วยการเพิ่มคอลัมน์หนึ่งเข้าไป เป็นตัวกำหนดความสามารถการเข้าถึงตามกฎของเบล และพาลาดูลา คอลัมน์ที่เพิ่มมานั้นเราเรียกว่าทัปเปิลคลาส (tuple class) หรือ TC เป็นระดับความปลอดภัยในระดับของทั้งแถวข้อมูล ทำให้เราสามารถแบ่งแยกข้อมูลของแต่ละระดับชั้นได้

การนำเสนอเราขอยกตัวอย่างที่เป็นที่นิยมกันในงานวิจัยฐานข้อมูลที่มีระดับความปลอดภัยหลายระดับ นั่นคือข้อมูลของยานอวกาศจากภาพยนตร์เรื่องสตาร์เทรก เป็นตัวอย่างที่มีมานานแล้วเนื่องจากมีความเหมาะสมกับการอธิบายถึงฐานข้อมูลเอ็มแอลเอสเป็นอย่างดี ยานอวกาศที่เดินทางไปปฏิบัติภารกิจ ยังสถานที่ๆ ถูกกำหนดไว้ จึงประกอบด้วยข้อมูลสามคอลัมน์คือ ชื่อยานอวกาศ (SHIP), ภารกิจ (OBJ) และ สถานที่ (DEST)

ฐานข้อมูลของตัวอย่างตาราง SOD ซึ่งย่อมาจาก "Starship Name", "Objective" และ "Destination" ตามลำดับ แสดงปฏิบัติการของยานอวกาศ

ตาราง 2.1 SOD ในแบบจำลองเอ็มแอลเอสพื้นฐาน

SHIP	*OBJ*	*DEST*	*TC*
Enterprise	Spying	Mars	TS
Enterprise	Spying	Pluto	C
Enterprise	Shipping	Pluto	U

ถ้าเป็นข้อมูลในฐานข้อมูลแบบปกติแล้ว "SHIP" เป็นคีย์หลัก เมื่อเราพิจารณาที่ระดับเดียวแล้วจากในรูปทั้งสามแถวอยู่คนละระดับ การมองเพียงระดับเดียวเห็นได้ว่ายังคงคุณสมบัติเดิมของคีย์หลักไว้ได้ แต่เมื่อมีข้อมูลรวมอยู่ด้วยกันแล้วเราไม่ได้ทิ้งกฎเกณฑ์เดิมไปเสียทีเดียว เรียกคีย์หลักใหม่ว่า Apparent Primary Key เราสามารถเขียนได้ว่า A_k, TC $\longrightarrow A_i$ โดย A_k คือคีย์หลัก Apparent Primary Key A_i คือข้อมูลแต่ละแอททริบิว

ตัวอย่างการกำหนดระดับความปลอดภัยโดยการกำหนดเป็นสัญลักษณ์ย่อระดับความปลอดภัยดังตัวอย่างที่ผ่านมาคือ (TS = top secret, S = secret, C = classified, U = unclassified) นำมาใช้กับฐานข้อมูลภารกิจของยานอวกาศโดยการใส่ระดับความปลอดภัยให้กับทุกแถวดังตารางที่ 2.1

TC (tuple class) ใช้แสดงระดับความปลอดภัยของแต่ละแถว ผู้ใช้ในระดับ TS สามารถมองเห็นข้อมูลทั้งหมดเนื่องจากมีระดับผู้ใช้สูงที่สุด และมีสิทธิ์ในการอ่านข้อมูลได้ทั้งหมด ผู้ใช้ในระดับ *C* มองเห็นเฉพาะสองแถวล่าง และ ผู้ใช้ในระดับ *U* เห็นเฉพาะแถวสุดท้าย ถ้าหากผู้ใช้ระดับ *U* นั้นเป็นผู้ใช้ระดับต่ำสุดแล้วควรมองเห็นในลักษณะที่ไม่มีคอลัมน์ TC คือมองเห็นข้อมูลที่มีระดับเดียวดังตารางที่ 2.2 การที่มองข้อมูลเดียวกัน จากผู้ใช้ในระดับต่างกัน ที่อาจมองเห็นได้ไม่เหมือนกันนั้นเรียกว่าโพลีอินแสตนติเอชัน (Polyinstantiation)

จากตารางที่ 2.1 เรามองภาพรวมของข้อมูล หรือเป็นการมองจากระดับสูงสุดของตาราง คือ TS หรือระดับที่สูงกว่านี้ก็สามารถมองเห็นข้อมูลได้ทั้งหมด ข้อมูลในตารางฐานข้อมูลที่มีระดับความปลอดภัยหลายระดับอาจไม่เป็นความจริงในโลกความเป็นจริง ซึ่งเหมือนกับว่าอาจมีการปกปิดหลอกลวง หรือมีการทำให้เข้าใจผิดกับข้อมูลโดยเจตนาได้

ตาราง 2.2 มุมมองของผู้ใช้ระดับ U ต่อตารางที่ 2.1

SHIP	*OBJ*	*DEST*
Enterprise	Shipping	Pluto

นอกเหนือจากการกำหนดระดับความปลอดภัยให้กับทุกแถวแล้ว ยังสามารถกำหนดระดับความปลอดภัยให้กับแต่ละแอททริบิวอีกด้วย ทำให้มีความสามารถในการสร้างความสอดคล้องของข้อมูลในแต่ละระดับสำหรับแอททริบิวการกำหนดระดับความปลอดภัยนั้นก็ทำในลักษณะเดียวกันกับที่ทำกับแต่ละแถวคือเพิ่มสัญลักษณ์ย่อของระดับให้กับแต่ละแอททริบิวจากตัวอย่างข้อมูลจากตาราง SOD มีลักษณะดังตารางที่ 2.3 โดยกำหนดสัญลักษณ์ย่อกับ SC, OC, DC เป็นระดับความปลอดภัยของ "SHIP", "OBJ", และ "DEST" ตามลำดับ แสดงให้เห็นว่ายานเอนเตอร์ไพรส์ที่เป็นข้อมูลเห็นโดยผู้ใช้ระดับ TS ได้มีการปกปิดข้อมูลของภารกิจและดาวเป้าหมายกับระดับอื่น ผู้ใช้ระดับ C ไม่ทราบว่ายานเอนเตอร์ไพรส์ในความคิดของ TS เดินทางไปดาวอังคารไม่ใช่ดาวพลูโต และผู้ใช้ระดับ U ก็มีความคิดที่มีต่อยานเอนเตอร์ว่าไปทำการขนส่งซึ่งแตกต่างจาก C และ

TS-subject ในลักษณะเช่นได้เรียกว่าการปิดบังข้อมูล (cover story) ส่วนในระดับความปลอดภัยในระดับแอททริบิวนั้นบ่งบอกความเชื่อว่าเป็นความเชื่อที่มีพื้นฐานมาจากของผู้ใช้ในระดับใด หากเป็นระดับที่ต่ำกว่าคือเชื่อว่าข้อมูลนี้เป็นจริงที่ระดับล่าง และยืนยันว่าเป็นข้อมูลที่เชื่อถือได้ที่ระดับตน ในตารางที่ 2.3 ภารกิจของเอนเตอร์ไพรส์ในความคิดของผู้ใช้ระดับ TS มีความเชื่อว่าข้อมูลจากระดับ C นั้นเชื่อถือได้ และไว้วางใจให้เป็นข้อมูลที่ตนยอมรับในระดับของตนเอง

ตาราง 2.3 SOD ที่มีการมีระดับความปลอดภัยในระดับแอททริบิวและแถว

SHIP	*SC*	*OBJ*	*OC*	*DEST*	*DC*	*TC*
Enterprise	U	Spying	C	Mars	TS	TS
Enterprise	U	Spying	C	Pluto	U	C
Enterprise	U	Shipping	U	Pluto	U	U

เห็นได้ว่าหากฐานข้อมูลที่ระดับเดียว โดยไม่มีระดับความปลอดภัยเข้ามาร่วมด้วย จะมี "SHIP" เป็นคีย์หลัก แต่ในเอ็มแอลเอสแล้วไม่เพียงพอที่เป็นคีย์หลักได้แต่เรียกว่าคีย์หลักแอพพาเรนท์ โดยในเอ็มเอลเอสนั้นจะมีคุณสมบัติดังนี้คือ

1) $A_1, C_1, C_i \longrightarrow A_i$

2) $A_1, C_1, TC \longrightarrow A_i, C_i$

โดย A_1 เป็น apparent key , C_1 เป็นระดับความปลอดภัยของคีย์หลักแอพพาเรนท์ (apparent key) และ C_i เป็นระดับความปลอดภัยของ A_i ซึ่งเป็นแอททริบิวที่ไม่ได้เป็นคีย์หลักแอพพาเรนท์ในส่วนข้อมูลที่ได้รับการยืนยันจากผู้ใช้ TS คือข้อมูลที่อยู่ในระดับเดียวกันดูได้จาก TC ที่มีระดับความปลอดภัยเป็น TS และสามารถอ่านได้โดยผู้ใช้ในระดับ TS เท่านั้นหรือผู้ใช้ที่สูงกว่าถ้าหากมีได้มีงานวิจัยมากมายกล่าวถึงการกำหนดกฎเกณฑ์สำหรับการทำความเข้าใจกับข้อมูลในระดับตนและระดับต่ำกว่าเพื่อลดความสับสนเนื่องจากคุณสมบัติของโพลีอินแสตนติเอชันยอมให้มีคีย์หลักแอพพาเรนท์ตรงกันได้ ในแต่ละข้อมูลภายในแถวนั้น สามารถมามองเห็นได้ต่อเมื่อเป็นข้อมูลที่ได้รับอนุญาตให้มองเห็นได้ ในส่วนถัดไปนั้นจะกล่าวถึงโมเดลเอ็มแอลเอสต่างๆ

2.3 คุณสมบัติและข้อดีข้อเสียของเอ็มแอลเอสในโมเดลอื่น

ในช่วงสิบปีที่ผ่านมาได้มีการคิดโมเดลที่สามารถรองรับกฎพื้นฐานของเบล และลาพาดูลานำมาใช้ร่วมกับฐานข้อมูลเชิงสัมพันธ์หลายรูปแบบด้วยกัน ปัญหาหลักคือความกำกวมของข้อมูลในแต่ละระดับชั้นความปลอดภัย ได้เลือกมาเฉพาะงานวิจัยที่น่าสนใจดังต่อไปนี้

2.4 โมเดลจาโจเดีย-ซานดู

2.4.1 แนวคิดพื้นฐาน

แนวทางที่จะทำให้เกิดความปลอดภัยของดีบีเอ็มเอส (DBMS) นั่นคือการตั้งกฎข้อบังคับของความปลอดภัยให้กับดีบีเอ็มเอส ในมุมมองของการสร้างความปลอดภัยหลายระดับ การจะทำให้เกิดความปลอดภัยกับดีบีเอ็มเอสได้นั้น จะต้องใช้การตีความ (interpret) ข้อมูลแบบการควบคุมการเข้าถึงข้อมูลที่แตกต่างกันอยู่ขึ้นกับผู้มีอำนาจ (mandatory access controls) การตีความที่เป็นที่ยอมรับมากที่สุดในระบบคอมพิวเตอร์เป็นของเบล และลาพาดูลาดังที่ได้กล่าวไว้แล้วข้างต้น

จากคุณสมบัติความปลอดภัยพื้นฐานและคุณสมบัติสตาร์ของโมเดลเบล และลาพาดูลา ส่งผลกระทบทำให้ ผู้ใช้งานที่มีระดับการเข้าถึงของตัวเองแตกต่างกัน จะเห็นรูปแบบของเอ็มแอลเอสได้แตกต่างกัน เช่น จากโครงตาราง SOD ให้ Starship เป็นคีย์หลัก (primary key) และมีระดับข้อมูลบอกภายใน ให้ผู้ใช้งานมีระดับ S จะเห็นข้อมูลตามตาราง 2.4 แต่ถ้าผู้ใช้งานมีระดับ U จะเห็นข้อมูลตาราง 2.5

ตาราง 2.4 โครงตารางที่ผู้ใช้งานระดับ S เห็น

SHIP		*OBJ*		*DEST*		*TC*
Enterprise	U	Exploration	U	Talos	U	U
Voyager	U	Spying	U	Mars	U	S

ตาราง 2.5 โครงตารางที่ผู้ใช้งานระดับ U เห็น

SHIP		*OBJ*		*DEST*		*TC*
Enterprise	U	Exploration	U	Talos	U	U
Voyager	U	Null	U	Null	U	S

จะเห็นว่าระบบอาจจะไม่ปลอดภัย ถึงแม้จะทำตามกฎทั้ง 2 ข้อของเบล และลาพาดูลาแล้วก็ตาม ระบบจะต้องมีการเพิ่มส่วนที่ถูกซ่อนไว้ นั่นก็คือ ได้มีการเพิ่มตัวป้องกันข้อมูลที่เรียกว่าโคเวิร์ทแชนเนล (covert channel) ซึ่งส่งผลทางอ้อมทำให้ผู้ใช้ที่มีระดับความปลอดภัยสูงสามารถส่งผ่านข้อมูลให้กับผู้ใช้งานที่มีระดับความปลอดภัยต่ำกว่าได้ ยกตัวอย่างเช่นตาราง 2.4 ถ้าผู้ใช้งานระดับ U ต้องการอัพเดทข้อมูลในแถวที่ 2 เป็น (Voyager, Exploration, Talos) ถ้าเป็นฐานข้อมูลปกติ การอัพเดทนี้จะต้องถูกยกเลิก แต่จริงๆแล้วข้อมูลนี้ไม่สมควรที่จะถูกยกเลิก จึงทำให้เกิดการเพิ่มข้อมูลที่ถูกซ่อนไว้ ทำให้มุมมองของผู้ใช้งานระดับ S จะต้องเห็นข้อมูลดังตาราง 2.6 เรียกว่าโพลีอินแสตนติเอชัน คือการมีมากกว่า 1 แถวที่มีคีย์หลัก (primary key) ตัวเดียวกัน

ตาราง 2.6 โครงตารางที่ผู้ใช้งานระดับ S เห็น มีโพลีอินแสตนติเอชัน

SHIP		*OBJ*		*DEST*		*TC*
Enterprise	U	Exploration	U	Talos	U	U
Voyager	U	Spying	U	Mars	U	S
Voyager	U	Spying	U	Mars	U	S

2.4.2 ความสัมพันธ์ของฐานข้อมูลหลายระดับ

ความสัมพันธ์ของฐานข้อมูลหลายระดับประกอบด้วย 2 ส่วนหลักๆ

2.4.2.1 โครงสร้างของตาราง (Relation Scheme)

ความสัมพันธ์สามารถแสดงได้ดังต่อไปนี้

$$R(A_1,C_1,A_2,C_2,\ldots,A_n,C_n,TC) \tag{2.1}$$

โดยที่ A_i เป็นแอททริบิวข้อมูล (Data Attribute) บนโดเมน D_i โดยแต่ละ C_i คือ แอททริบิวระดับความปลอดภัย (Classification Attribute) ของ A_i และ TC คือ แอททริบิวทัปเปิล คลาส

2.4.2.2 กลุ่มข้อมูลของตาราง (Relation Instances)

ความสัมพันธ์สามารถแสดงได้ดังต่อไปนี้

$$R_c(A_1,C_1,A_2,C_2,\ldots,A_n,C_n,TC) \tag{2.2}$$

เมื่อมีการเข้าถึงข้อมูลด้วยระดับใดๆ กลุ่มหนึ่งๆ จะต้องเป็นมีความแต่ต่างจาก $(a_1,c_1, a_2,c_2,\ldots, a_n,c_n,tc)$ ระดับความปลอดภัยจะต้องไม่มีค่าเป็นนัล (null) เสมอ

2.4.3 คุณสมบัติโพลีอินแสตนติเอชัน (Polyinstantiation Integrity)

R จะเป็นคุณสมบัติโพลีอินแสตนติเอชัน ได้ก็ต่อเมื่อ ทุกๆ R_c ที่ A_i : AK, C_{AK}, $C_i \rightarrow A_i$ คุณสมบัติข้อนี้กำหนดให้ผู้ใช้งานทำการเลือกคีย์หลักแอพพาเรนท์ AK โดยมีการกำหนดระดับของ AK ไว้ ด้วยค่าของ C_{AK} และ C_i ที่โดยปกติแล้วจะเป็นตัวกำหนดค่าระดับความปลอดภัยของข้อมูลให้กับค่าของข้อมูล A_i คุณสมบัติโพลีอินแสตนติเอชั่นจะทำให้ความสัมพันธ์เป็นแบบตาราง 2.6 ไม่ใช่ตาราง 2.7

ตาราง 2.7 ความสัมพันธ์ที่ขาดคุณสมบัติโพลีอินแสตนติเอชัน

SHIP		*OBJ*		*DEST*		*TC*
Enterprise	U	Exploration	S	Talos	S	S
Voyager	U	Spying	S	Mars	S	S

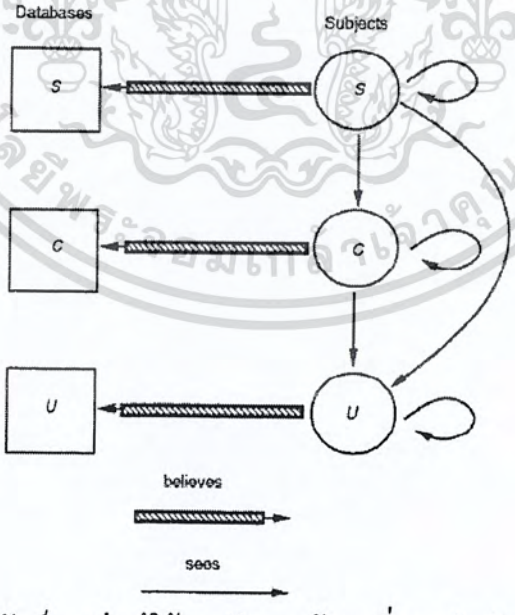
2.5 โมเดลสมิท-วินส์เลต

โมเดลสมิท-วินส์เลต (Smith-Winslett) เป็นโมเดลแรกที่เพิ่มซีแมนติค (semantic) ลงในฐานข้อมูล ซึ่งเป็นฐานข้อมูลหลายระดับที่มองเห็นเป็นเซตของฐานข้อมูลรีเลชั่น (relational) สำหรับแต่ละระดับความปลอดภัย และมีการแชร์โครงสร้างตารางในฐานข้อมูลเดียวกัน โมเดลนี้จะมีความแตกต่างกันระหว่างซีแมนติค (semantic) และซินแทคติค (syntactic) บนฐานข้อมูล MLS ดังนี้

2.5.1 ซีแมนติคสำหรับฐานข้อมูลเอ็มแอลเอสแบบรีเลชั่นนอล

2.5.1.1 การตีความโดยใช้หลักความเชื่อ (Belief-based Interpretation)

การตีความฐานข้อมูลเอ็มแอลเอส จะเป็นเซตของฐานข้อมูลรีเลชั่นนอลที่มีการแชร์โครงตารางเดียวกัน นอกจากนี้ยังมีการใช้เลเบล (label) ในแต่ละระดับอีกด้วย ซึ่งอาศัยความเชื่อของผู้ใช้ที่แตกต่างกันในแต่ละระดับ ที่อาจจะเกี่ยวข้องกับค่าของแอททริบิวในเอ็นติตี้เดียวกันก็ได้ โดยทั้งผู้ใช้งานกับฐานข้อมูลจะต้องมีความสัมพันธ์กันและถูกต้องตามกฎ ดังรูปที่ 2.1



รูป 2.1 ความสัมพันธ์ระหว่างผู้ใช้งานกับฐานข้อมูลที่แตกต่างกันในหลายๆระดับ

จากรูป จะมีผู้ใช้งานทั้ง 3 ระดับคือ U C และ S โดยที่ผู้ใช้งานจะต้องเชื่อถือข้อมูลในระดับของตัวเอง จะถูกแสดงโดยใช้ลูกศร ═══► นี้ ชี้ไปยังฐานข้อมูลที่ระดับเดียวกัน และผู้ใช้งานแต่ละระดับจะมองเห็นข้อมูลในระดับที่ต่ำกว่า จะถูกแสดงโดยใช้ลูกศร ———► นี้ ชี้ระหว่างกลุ่มของผู้ใช้งาน

ผู้ใช้งานสามารถเข้าถึงฐานข้อมูลดังนี้

1) การเข้าถึงเพิ่มอัพเดทข้อมูล (Update Access) ผู้ใช้งานสามารถเปลี่ยนแปลงข้อมูลที่อยู่ในระดับตัวเองได้
2) การเข้าถึงเพื่ออ่านข้อมูล (Read Access) ผู้ใช้งานที่ระดับ I สามารถอ่านข้อมูลได้ก็ต่อเมื่อเลเบลถูกข่มโดย I ซึ่งก็คือ มีระดับที่สูงกว่าข้อมูลนั้นๆ

2.5.1.2 เอนติตี้ในหลายๆระดับความปลอดภัย (Multilevel Secure Entities)

เมื่อพิจารณารีเลชั่นแบบดั้งเดิมดีๆจะเห็นปัญหาขึ้นมากมาย อย่างเช่น ถ้าผู้ใช้งานทำการอินเสิร์ตข้อมูลด้วยคีย์ K ผู้ใช้งานจะไม่สามารถรู้ได้เลยว่ามีข้อมูลคีย์ K อยู่แล้วในระดับที่สูงกว่า และถ้าคีย์ K นี้ถูกใช้ในระดับที่สูงกว่าที่มีเอ็นติตี้ต่างกันแล้ว หลังจากนั้นจะมีการนำเอาคีย์นี้กลับมาใช้ใหม่อีกครั้งในระดับที่ต่ำกว่าได้ ซึ่งจะก่อให้เกิดการตีความที่กำกวมและเป็นไปไม่ได้ จึงเกิดการรีเจค (reject) แล้วเปิดโคเวิร์ทแชนเนล (covert channel) ซึ่งส่งผลทางอ้อมทำให้ผู้ใช้ที่มีระดับความปลอดภัยสูงสามารถส่งผ่านข้อมูลให้กับผู้ใช้งานที่มีระดับความปลอดภัยต่ำกว่าได้ เมื่อเกิดปัญหาเช่นนี้ขึ้นจึงมีการคิดคีย์ระดับความปลอดภัย ขึ้นมาเพื่อนำมาอธิบายปัญหาของโพลีอินแสตนติเอชัน ซึ่งมีคุณสมบัติดังต่อไปนี้

คุณสมบัติข้อที่ 1 คีย์ระดับความปลอดภัย

"โครงสร้างตารางสำหรับรีเลชั่นใดๆ ที่ใช้ในการตีความ ควรจะรวมคีย์ระดับความปลอดภัย KC กับ K เข้าไว้ด้วยกัน ซึ่งจะถูกเรียกว่าตัวระบุเอ็นติตี้ (entity identifier : eid เป็นตัวพิจารณาทุกๆ แอททริบิวภายในฐานข้อมูล R ที่ระดับ I) โดยที่ KC จะแสดงระดับความปลอดภัยของข้อมูลใน K ซึ่งสามารถเขียนความสัมพันธ์ได้ดังนี้ $R(K,KC,A_1,\ldots,A_M)$"

2.5.2 ซินแทคติคของฐานข้อมูลเอ็มแอลเอสแบบรีเลชั่นนอล

ในส่วนนี้จะอธิบายซินแทคอย่างง่ายสำหรับฐานข้อมูลแบบรีเลชั่นนอล ซึ่งมีการรวมข้อมูลจากระดับที่แตกต่างกันเข้าเป็นฐานข้อมูลเดียวกันที่มีหลายระดับ และแสดงความชัดเจนของการเชื่อมกันระหว่างซินแทคกับซีแมนติค โดยสรุปคุณสมบัติข้อที่ 1 ใหม่เป็นดังนี้

คุณสมบัติข้อ 1' คีย์ระดับความปลอดภัย

"โครงสร้างสำหรับทุกๆ รีเลชั่น R จะต้องรวมคีย์ระดับความปลอดภัย (แอททริบิว KC) ที่เป็นตัวแสดงระดับความปลอดภัยของของข้อมูลใน K และระดับความปลอดภัยของทัปเปิล (แอททริบิว TC) ที่เป็นตัวแสดงระดับความปลอดภัยของทัปเปิลเข้าไว้ด้วยกัน ซึ่งสามารถแสดงความสัมพันธ์ได้ดังนี้ $R(K,KC,A_1,\ldots,A_M,TC)$ โดยที่โดเมนของ TC และ KC จะเป็นเซตของระดับความ

ปลอดภัย และค่า แอททริบิวของ TC จะต้องข่มค่าของ KC ยิ่งไปกว่านั้นแล้วการที่จะรวม K, KC และ TC (K|KC|TC) เข้าด้วยกันในฐานข้อมูลจะต้องดูค่าภายในทุกๆ ส่วนของแอททริบิวทั้งหมดด้วย โดยทั้งหมดนี้จะรวมกันเป็นรูปแบบคีย์ของซินแทคในรีเลชั่น R นั่นเอง"

พิจารณา SOD ดังตารางที่ 2.8

ตาราง 2.8 SOD ที่เป็นเอ็มแอลเอสรีเลชั่น

Starship	*KC*	*Objective*	*Destination*	*TC*
Voyager	U	Shipping	Mars	U
Enterprise	U	Exploration	Vulcan	U
Enterprise	U	Diplomacy	Romulus	C
Zardor	S	Warfare	Romulus	S

จะมีการตีความข้อมูลทั้ง 3 ระดับดังนี้

1) ฐานข้อมูลระดับ U จะประกอบด้วย 2 ทัปเปิลแรกทั้งนั้น และมีค่าของ TC เป็น U
2) ฐานข้อมูลระดับ C จะมีเพียง 1 ทัปเปิลเท่านั้น ซึ่งก็คือ แถวที่ 3 และมีค่าของ TC เป็น C
3) ในฐานข้อมูลระดับ S จะมีเพียง 1 ทัปเปิลเท่านั้น ซึ่งก็คือ แถวที่ 4 และมีค่าของ TC เป็น S

จากตัวอย่างจะแสดงให้เห็นถึงการเชื่อมกันที่ชัดเจนระหว่างซินแทคกับซีแมนติคกล่าวได้ว่า ในทุกๆ ซินแทคจะมีการตีความเพียง 1 ครั้งเท่านั้น

การตีความโดยทั่วไปแล้ว จะประกอบด้วยฐานข้อมูลเดียวเท่านั้น สำหรับแต่ละระดับโครงสร้างความปลอดภัย โดยที่ฐานข้อมูลระดับ I จะถูกเชื่อมโยงไปเป็นส่วนหนึ่งของฐานข้อมูลระดับ I' ถ้า I ข่ม I' ในขณะที่โครงสร้างตารางในแต่ละระดับจะเหมือนกัน ยกเว้นค่าของแอททริบิว TC จะถูกแยกออกทุกครั้งที่มีการตีความข้อมูล

สรุปแล้วการตีความข้อมูลจะต้องคำนึงถึงสิ่งเหล่านี้

1) จะต้องมีโครงสร้างตารางเดียวกันในทุกระดับ
2) จะไม่มีค่านัลที่แอททริบิวที่ไม่ใช่คีย์ เพราะซินแทคใน R ห้ามเป็นนัลใน K ,KC และ TC

การตีความจะใช้จะใช้คุณสมบัติข้อที่ 1 และเวลาต่อมาจะใช้คุณสมบัติข้อ 1' แทน ซึ่งการตีความก่อนอื่นเลยจะดูที่ค่า TC ก่อนและถ้าค่า TC ในทุกๆทัปเปิลเหมือนกันจะไปดูค่า K|KC แทน แล้วค่อยตีความส่วนอื่นๆของข้อมูลต่อไป ซึ่งการตีความในทุกๆครั้งจะถูกแสดงอยู่ในฐานข้อมูล

ด้วยการต่อท้ายค่าแอททริบิว TC ใหม่ในแต่ละโครงสร้างตารางของรีเลชั่นนั้นๆ และถ้าในแต่ละทัปเปิลมีค่า I อยู่แล้วใน TC มันจะทำการรวมกันแต่ละทัปเปิลที่มีความแตกต่างระดับกันของแต่ละรีเลชั่นนั้นๆเข้าไว้ด้วยกัน นอกจากนี้ยังจะมีการรวมสิ่งต่างที่ได้จากการเชื่อมกันจากซินแทคไปยังซีแมนทิคและสิ่งนี้เองจะกลับกลายมาเป็นฟังก์ชันที่เป็นตัวบ่งชี้ในฐานข้อมูลเอ็มแอลเอสนั่นเอง

ในโมเดลส่วนมากจะใช้ระดับความปลอดภัยของแอททริบิวในโครงสร้างตาราง เพื่อเพิ่มประสิทธิภาพของโมเดลให้มากที่สุด ทางหนึ่งที่จะอธิบายประสิทธิภาพที่เกิดขึ้นได้ คือ การนำซินแทคบางส่วนมาใช้แสดงซีแมนทิคบางส่วน โดยวัตถุประสงค์อื่นๆจะไม่ถูกแสดงอยู่ในซีแมนทิคของฐานข้อมูลนั้น อย่างไรก็ตามแล้วในบางซินแทค ควรจะทำให้อยู่ในรูปแบบของซีแมนทิค ซึ่งเป็นการรวมเซตระดับเดียวๆ (single level) ของฐานข้อมูล เพื่อแสดงความเชื่อในแต่ละระดับ และจัดให้อยู่ในหลักการของเอ็มแอลเอสนั้นๆ

2.6 โมเดลเอ็มแอลอาร์

แรงบันดาลใจเริ่มแรกที่ทำให้เกิดโมเดลนี้ขึ้นมา เนื่องจากโมเดลของจาโจเดีย-ซานดู มีปัญหาเกิดขึ้น อยู่ 2 สิ่งโดยที่ไม่สามารถแก้ไขปัญหานั้นได้ ปัญหาที่ว่า นั่นคือ ปัญหาซีแมนติคที่กำกวม และการทำโอเปอร์เรชั่นที่ไม่ถูกต้อง พิจารณาตัวอย่างต่อไปนี้

ตาราง 2.9 ตัวอย่างซีแมนติคที่กำกวม

SHIP		*OBJ*		*DEST*		*TC*
Enterprise	U	Mining	M_1	Talos	U	M_1
Enterprise	U	Spying	M_2	Talos	U	M_2
Enterprise	U	Enplortion	U	Talos	U	U

M_1 และ M_2 เป็นเลเบลที่ยังไม่ได้ทำการเปรียบเทียบระดับความปลอดภัยของผู้ใช้งานลงไป แต่มีข้อบังคับอยู่ว่าเลเบลทั้งสองนั้นจะต้องมีระดับที่สูงที่สุด คือ S และระดับที่ต่ำที่สุด คือ U จึงปัญหาเกิดขึ้นว่า ค่า OBJ ไหนบ้างเป็นที่ยอมรับของผู้ใช้งานระดับ S คำตอบที่ได้ อาจจะเป็น Mining หรือ Spying หรือ Exploration แต่ยังไงแล้วก็ตามผู้ใช้งานระดับ S สามารถยอมรับข้อมูลได้แค่บางระดับเท่านั้น จึงเกิดเป็นปัญหาขึ้น ซึ่งปัญหานี้ คือ ปัญหาของการมีซีแมนติคที่กำกวมนั่นเอง และต่อมาเป็นตัวอย่างที่ผู้ใช้งานระดับ S มีการทำโอเปอร์เรชั่นที่ไม่ถูกต้อง

ตาราง 2.10 ตัวอย่างการทำโอเปอร์เรชั่นที่ไม่ถูกต้อง

SHIP		*OBJ*		*DEST*		*TC*
Enterprise	U	Mining	M_1	Sirius	M_2	S

ผู้ใช้งานระดับ S มีการแบ่งแอททริบิทระดับความปลอดภัยในทัปเปิลออกเป็น 3 แอททริบิว ได้แก่ U, M_1 และ M_2 เมื่อพิจารณาจากข้อเท็จจริงแล้ว ไม่มีทางเป็นไปได้เลยที่ผู้ใช้งานระดับ S จะสามารถเพิ่มทัปเปิลนี้ลงได้ จึงเกิดเป็นปัญหาขึ้น ซึ่งปัญหาที่ว่านี้เกิดจากการทำโอเปอร์เรชั่นที่ไม่ถูกต้องนั่นเอง

ปัญหาทั้งสองอย่างข้างต้นแก้ไขได้โดยการใช้โมเดลเอ็มแอลอาร์ ซึ่งโมเดลนี้ถือว่าเป็นโมเดลที่มีประโยชน์และมีประสิทธิภาพอย่างมาก เนื่องจากสามารถจัดการตีความที่กำกวมได้และยังสามารถควบคุมการไหลของข้อมูลไปยังระดับที่สูงกว่าได้

เอ็มแอลอาร์เป็นโมเดลที่ริเริ่มกฎการยืมข้อมูลจากระดับต่ำกว่า (data-borrow) ขึ้น เป็นกฎข้อบังคับที่มีการยืมข้อมูลจากระดับที่ต่ำกว่าลงไป เพื่อสร้างความมั่นใจในการไหลของข้อมูล ซึ่งเป็นการส่งข้อมูลนั้นขึ้นไปสู่ระดับที่สูงกว่าได้แบบอัตโนมัติ โดยเป็นข้อมูลที่ระดับสูงนั้นได้ยืมมาจากระดับต่ำกว่าเมื่อมีการแก้ไขข้อมูลนั้นก็จะส่งผลถึงระดับสูงด้วย ในข้อมูลนั้นๆ สามารถมองเห็นได้ก็ต่อเมื่อเป็นข้อมูลที่ได้รับอนุญาตให้มองเห็น

2.6.1 พื้นฐานโมเดลเอ็มแอลอาร์

2.6.1.1 นิยาม

จะประกอบไปด้วย 2 นิยามดังนี้

1) โครงสร้างของตาราง (Relation schema) โดยที่โครงสร้างของรีเลชั่นหลายๆ ระดับ (Multilevel Relation) จะแสดงความสัมพันธ์ได้ดังนี้

$$R(A_1,C_1, A_2,C_2,\ldots, A_n,C_n,TC) \quad (2.3)$$

โดยที่ A_i คือ แอททริบิวข้อมูล C_i คือ แอททริบิวระดับความปลอดภัยของ A_i และ TC คือ แอททริบิวทัปเปิลคลาส

2) กลุ่มข้อมูลของตาราง (Relational instance) โดยที่กลุ่มข้อมูลของตารางจะแสดงความสัมพันธ์ได้ดังนี้

$$r(A_1,C_1, A_2,C_2,\ldots, A_n,C_n,TC) \quad (2.4)$$

2.6.1.2 การตีความข้อมูล

จะใช้หลักการของการตีความบนพื้นฐานของข้อมูลดังนี้

1) ข้อมูลที่ถูกยอมรับโดยผู้ใช้งานที่ระดับหนึ่งๆจะประกอบด้วย 2 ส่วน คือ

ข้อมูลที่สร้างด้วยตัวเองและข้อมูลที่ยืมมาจากผู้ใช้งานที่อยู่ระดับต่ำกว่า ซึ่งข้อมูลที่ยืมมาจากระดับต่ำกว่านี้สามารถที่จะเปลี่ยนแปลงได้ขึ้นกับผู้ใช้งานในระดับต่ำกว่าที่เป็นเจ้าของข้อมูลนั้น

2) ผู้ใช้งานจะเห็นข้อมูลที่ระดับตัวเองและที่ระดับต่ำกว่า

3) ถ้ามี c-tuple ประกอบไปด้วยข้อมูลที่ได้รับการยอมรับ (ทั้งที่เป็นเจ้าของและยืมมา) โดยผู้ใช้งาน c ซึ่งถ้ามองไม่เห็นข้อมูลของ c-tuple แสดงว่าเอ็นติตี้ที่มีอยู่นั้นไม่ได้รับการยอมรับจากผู้ใช้งาน c

ส่วนมากแล้วจะมีการแชร์ข้อมูลให้กับผู้ใช้งานในหลายๆระดับได้ใช้งานร่วมกัน โดยนำหลักการการตีความโดยอาศัยหลักความเชื่อมาใช้ เพื่อที่จะทำให้เกิดการตีความข้อมูลนั้นได้อย่างถูกต้องสมบูรณ์ พิจารณาจากตัวอย่างต่อไปนี้

ตาราง 2.11 การใช้หลักการการตีความบนพื้นฐานข้อมูล

SHIP		*OBJ*		*DEST*		*TC*
Enterprise	S	Spying	S	Riger	S	S
Enterprise	U	Exploration	U	Null	S	TS
Enterprise	U	Exploration	U	Talos	U	U

ตารางนี้มีทั้งหมด 2 เอ็นติตี้ คือ (Enterprise S) และ (Enterprise U) ซึ่งสร้างมาจากผู้ใช้งานระดับ S และ U ตามลำดับ และผู้ใช้งานระดับ TS มีการเพิ่ม TS ลงใน TC ซึ่งจะทำให้ทุกๆ ข้อมูลจะถูกยอมรับโดยผู้ใช้งานระดับ TS นั่นเอง โดยที่ทัปเปิล U เป็นทัปเปิลหลักที่จะสามารถลบทิ้งได้เมื่อเอ็นติตี้ทั้งหมดถูกลบออกไป และค่า OBJ ที่ทัปเปิล TS เป็นค่าที่ยืมมาจากผู้ใช้งานระดับ U ผู้ใช้งาน TS จะเปลี่ยนแปลงค่านี้ได้ก็ต่อเมื่อทัปเปิล U ถูกเปลี่ยนหรือทำลายไปโดยผู้ใช้งานระดับ U ส่วนค่านัลใน TS มีความหมายว่าผู้ใช้งานระดับ TS คาดหวังว่าจะไปยืมข้อมูล DEST จากผู้ใช้งานระดับ S แต่มันดันไม่มีข้อมูล DEST ที่เป็นของตัวเองอยู่ภายในเอ็นติตี้ นี้ พูดง่ายๆว่า มันไม่มีทัปเปิล S อยู่ภายในเอ็นติตี้เลย ทำให้การบ่งชี้ในเอ็นติตี้นี้ไม่ถูกยอมรับโดยผู้ใช้งานระดับ S นั่นเอง

2.6.2 คุณสมบัติการยืมข้อมูลจากระดับต่ำกว่า

คุณสมบัตินี้คือคุณสมบัติคีย์ของการตีความข้อมูลซึ่งอนุญาตให้มีการยืมข้อมูลจากระดับต่ำกว่า เพื่อความมั่นใจว่าโมเดลเอ็มแอลอาร์สามารถรับการไหลของข้อมูลไปยังระดับบนได้ โดยเมื่อการเปลี่ยนแปลงข้อมูลที่ระดับล่างข้อมูลที่ระดับบนก็จะถูกเปลี่ยนแปลงไปด้วยแบบอัตโนมัติ คล้ายกับการ write-up แต่จะไม่ใช้การเขียนทับข้อมูลนั้นๆในระดับข้อมูลที่สูงกว่า พิจารณาตารางดังต่อไปนี้

ตาราง 2.12 พื้นฐานของหลักการการตีความข้อมูล

SHIP	*OBJ*	*DEST*	*TC*
Enterprise U	Exploration U	Rigel S	S
Enterprise U	Exploration U	Talos U	U

จะเห็นได้ว่าทัปเปิล U เป็นทัปเปิลที่กำเนิดข้อมูล Enterprise และ Exploration เพราะถ้าขาดทัปเปิล U นี้แล้วจะทำให้ผู้ใช้ U ไม่สามารถยอมรับเอ็นติตี้ Enterprise ณ ขณะนั้นได้ ซึ่งบอกได้ว่าข้อมูลที่เป็นของผู้ใช้ U นั้นไม่มีอยู่และผู้ใช้ S ก็ไม่สามารถใช้ข้อมูลนั้นได้ สรุปได้ว่าข้อมูลที่ระดับ S นั้นเป็นข้อมูลที่ข้อมูลมาจากระดับ U นั่นเอง

2.6.3 คำสั่ง UPLEVEL

คำสั่งนี้เป็นคำสั่งที่ใช้ในการดึงข้อมูลจากระดับล่างขึ้นมาเป็นระดับของตน พิจารณาตารางดังต่อไปนี้

ตาราง 2.13 ข้อมูลผู้ป่วยในรูปแบบโมเดลเอ็มแอลอาร์

Patient Name	*PC*	*Diagnosis*	*DC*	*Age*	*AC*	*Room*	*RC*	*TC*
Jones	C	Leukemia	C	12	C	209	S	S
Jones	C	Leukemia	C	12	C	202	C	C
Adams	U	Diarrhea	U	21	U	201	U	U

จะเห็นได้ว่าผู้ใช้ในระดับ S ได้ทำการใช้คำสั่ง "UPLEVEL" ทำให้เกิดผลในแถวแรกขึ้นมาโดยข้อมูลของผู้ป่วยชื่อ Jones นี้ ได้รับการยืนยันว่าเป็นความจริงโดยผู้ใช้ในระดับ S ในส่วนของ ชื่อผู้ป่วย, กลุ่มโรค และอายุ ยกเว้นหมายเลขห้องนั้นยังไม่ได้รับการยืนยันว่าเป็นความจริงในระดับ S เป็นห้อง 209 ไม่ใช่ 202 นอกจากข้อมูลในส่วนอื่นที่ไม่ใช่หมายเลขห้องและคีย์แอพพาเรนท์แล้ว เมื่อผู้ใช้ในระดับ C ทำการแก้ไขข้อมูลก็จะส่งผลให้เกิดการแก้ไขข้อมูลตามในระดับ S ด้วย อย่างไรก็ตามยังมีจุดด้วยที่ไม่สามารถปกปิดข้อมูลในส่วนที่เป็น คีย์แอพพาเรนท์ได้เพราะเนื่องจากได้ให้คีย์แอพพาเรนท์เป็นตัวระบุเอ็นติตี้

2.7 โมเดลบีซีเอ็มแอลเอส

โมเดลบีซีเอ็มแอลเอส (BCMLS) ได้นำเอาหลักการของโมเดลสมิท-วินส์เลตที่มีอยู่มาเพิ่มกลไกโดยอ้างถึงความเชื่อเกี่ยวกับข้อมูลในระดับที่ต่ำกว่า ทำให้การตีความข้อมูลโดยผู้ใช้นั้นไม่เกิดความกำกวมและสามารถตีความได้อย่างถูกต้อง ซึ่งให้แอคเคาวน์ที่สอดคล้องกันบนวิวในทุกระดับที่ถูกข่มโดยผู้ใช้งาน ซึ่งยังไม่มีโมเดลไหนที่แก้ปัญหาการมีอยู่ของข้อมูลที่กำกวมได้เท่ากับโมเดลนี้เลย

2.7.1 คุณลักษณะโมเดลบีซีเอ็มแอลเอส

การตีความข้อมูลที่เข้าถึงได้ เพื่อเป็นการลดความกำกวมในการตีความสิ่งที่ต้องทำคือการระบุความแตกต่างของความเชื่อที่เกิดกับข้อมูลที่เก็บอยู่ในฐานข้อมูลเอ็มแอลเอส โดยอ้างโมเดลของสมิท-วินส์เลต คือ การเชื่อในระดับของตัวเองและการมองลงไปยังระดับที่ต่ำกว่า โดยใช้หลักการของจริงหรือเท็จเพื่อที่จะระบุการตีความของทัปเปิลที่ระดับต่ำกว่าจากระดับที่สูงกว่า

ในโมเดลบีซีเอ็มแอลเอส มี TC เป็นตัวกำหนดระดับความปลอดภัยของทัปเปิลและ L เป็นระดับของผู้ใช้งานในการตีความที่มีระดับเดียวกัน (TC=L) จะเชื่อตามข้อมูลในทัปเปิลนั้นเลย แต่ถ้า TC<L แล้ว ทุกๆทัปเปิลที่มีระดับต่ำกว่าจะถูกตีความโดยทุกผู้ใช้งานที่มีระดับสูงกว่า ซึ่งเป็นไปตามหนึ่งในกรณีดังต่อไปนี้

1) ทัปเปิลที่มีระดับต่ำกว่าสามารถถูกตีความโดยผู้ใช้งานที่ระดับสูงกว่า ว่าเป็นทัปเปิลที่เป็นความจริง (True) โดยผู้ใช้ที่ระดับสูงกว่าเชื่อตามทัปเปิลนั้นว่า ทุกๆ ค่า attribute ของทัปเปิลที่มีระดับต่ำกว่านั้นแทนคุณสมบัติของ real-world entity ได้ถูกต้อง ซึ่งถูกอธิบายโดยทัปเปิลนี้
2) ทัปเปิลที่มีระดับต่ำกว่าสามารถถูกตีความโดยผู้ใช้งานที่ระดับสูงกว่า ว่าเป็นทัปเปิลที่เป็นความไม่จริง (False) ซึ่งมี 2 ประเภทดังนี้
   2.2) ถ้าทัปเปิลที่มีระดับต่ำกว่า (TC<L) แทนเอ็นติตี้เดียวกันกับทัปเปิลที่มีระดับสูงกว่าอื่นแล้ว (TC=L) ทัปเปิลที่มีระดับต่ำกว่าจะถูกการตีความโดยผู้ใช้งานที่มีระดับสูงกว่าว่าเป็นทัปเปิลเท็จที่เรียกว่าการ Cover story โดยผู้ใช้งานที่มีระดับสูงกว่าจะเชื่อว่ามีบางค่าแอททริบิวของทัปเปิลที่ระดับต่ำกว่านั้นไม่ถูกต้อง
   2.2) ถ้าทัปเปิลที่มีระดับต่ำกว่าตีความโดยผู้ใช้งานที่ระดับสูงกว่า ว่าเป็นทัปเปิลที่เป็นความไม่จริง (False) และไม่เป็น Cover story tuple และทัปเปิลเท็จไม่เกี่ยวข้องกับเอ็นติตี้ real world เรียกว่า mirage tuple ทุกๆผู้ใช้งานจะเชื่อว่าทัปเปิลนี้ไม่ได้มีอยู่จริง
3) ทัปเปิลที่มีระดับต่ำกว่าสามารถถูกตีความโดยผู้ใช้งานที่ระดับสูงกว่า ว่าเป็นทัปเปิลที่ไม่มีความเกี่ยวข้องเลย โดยทุกผู้ใช้งานที่ระดับสูงกว่าเชื่อว่าทัปเปิลนี้มี ทั้งที่มีอยู่จริงหรือไม่มีและทัปเปิลนี้ไม่มีความเกี่ยวข้องกับเอ็นติตี้ real world

2.7.2 ริชเชอร์เซตของระดับความปลอดภัย

จุดเด่นที่สุดของโมเดลนี้คือ การนำเอาริชเชอร์เซต (Richer Set) มาใช้เป็นสัญลักษณ์ย่อระดับความปลอดภัยซึ่งทำให้ลดความซ้ำซ้อนของการเก็บข้อมูลหากต้องการกระจายข้อมูลในหลายระดับ และทั้งยังสามารถตีความหมายของข้อมูลกับข้อมูลที่ระดับต่ำกว่าได้อย่างง่ายดายอีกด้วย เครื่องหมายบอกระดับข้อมูลนั้นเป็นชุดของระดับความปลอดภัย

ในการตีความทัปเปิลที่มีระดับต่ำกว่าทั้งหมดนั้น จะต้องทำการขยายเซตของเลเบลความปลอดภัยที่เกี่ยวข้องกับแอททริบิวและทัปเปิลในฐานข้อมูลเอ็มแอลเอสก่อน ซึ่งจะเป็นประโยชน์กับแนวคิดของเลเบลความปลอดภัย คือ สามารถที่จะเก็บรายละเอียดของระดับได้โดยที่จุดประสงค์ของการขยายเซตของเลเบลความปลอดภัย คือ เพื่อเตรียมให้กับผู้ใช้งานกับการตีความที่ครอบคลุมทุกๆส่วนและทุกๆ ทัปเปิลที่มองเห็น

เมื่อเลเบลจากริชเชอร์ถูกใช้งาน ผู้ใช้งานสามารถสรุปถึงความเชื่อของตนเองและความเชื่อของผู้ใช้งานที่มีระดับที่ต่ำกว่าเกี่ยวกับทุกๆ ทัปเปิลที่มองเห็น

ริชเชอร์เซตของเลเบลความปลอดภัยที่จำเป็นจะมีทั้งหมด 13 เลเบลดังตารางต่อไปนี้

ตาราง 2.14 ริชเชอร์เซตของเลเบลความปลอดภัย

Label No.	Label L	U view of the label	C view of the label	S view of the label
1	U	U	U	U
2	US	U	U	US
3	U-S	U	U	U-S
4	UC	U	UC	UC
5	UCS	U	UC	UCS
6	UC-S	U	UC	UC-S
7	U-C	U	U-C	U-C
8	U-CS	U	U-C	U-CS
9	U-C+S	U	U-C	U-C+S
10	C		C	C
11	CS		C	CS
12	C-S		C	C-S
13	S			S

อักษรตัวแรก คือ ระดับความปลอดภัย (security level) ที่เพิ่มทัปเปิลเข้ามา เรียกว่า ระดับแรก (primary level) ของทัปเปิลหรือแอททริบิวนั้น ข้อมูลที่ถูกเชื่อว่าจริงโดยมีฟังก์ชันระดับแรก (primary level function pl) (L) ที่ คัดแยกระดับแรกของเลเบลความปลอดภัย L ตัวอย่างเช่น pl(UC-S)=U เป็นต้น ขณะที่ตัวอักษรตัวที่สอง คือ ระดับที่สอง (security level) ที่ข้อมูลเลเบลไม่ได้ถูกสร้างขึ้นผู้ใช้งานที่เลเบลนั้นจะเชื่อเกี่ยวกับข้อมูลเลเบล นั้น ซึ่งเรียกว่า ระดับที่สอง ถ้าตัวอักษรมีเครื่องหมาย – นำหน้าระดับที่สอง แสดงว่า ระดับที่สองนั้นเชื่อว่าข้อมูลนั้นเท็จ ถ้าไม่มีแสดงว่าเชื่อว่าข้อมูลนั้นจริง ถ้ามีเครื่องหมาย + นำหน้าแสดงว่าข้อมูลนั้นถูกเชื่อว่าจริง และข้อมูลเดียวกันนี้ในระดับที่สองที่ต่ำกว่าจะเชื่อว่าเท็จ ถ้าระดับที่สองสูงกว่าระดับแรกแล้ว ผู้ใช้งานที่อยู่ในเลเบลนั้นก็จะไม่เชื่อเกี่ยวกับข้อมูลของเลเบลนั้น

ในการจะแตกความเชื่อโดยผู้ใช้งานระดับ l ที่เกี่ยวกับข้อมูลของเลเบลโดยริชเชอร์เลเบล L จะใช้ function level's belief: lb(l , L) ดังตัวอย่าง

lb (S,U-CS) = -S แสดงว่า S เชื่อว่าเป็นข้อมูลที่เท็จ

lb(C,US) = Ø แสดงว่าผู้ใช้ที่มีระดับ C เชื่อว่าเป็นข้อมูลที่ไม่มีความสัมพันธ์กัน

lb(C,UC-S) = C แสดงว่าผู้ใช้ที่มีระดับ C เชื่อว่าเป็นข้อมูลที่จริง

ผู้ใช้งานจะเห็นได้แค่ส่วนของระดับมุมมองของเลเบลนั้น

ตัวอย่างเช่น

ถ้าทัปเปิลที่มีเลเบล U-S และ U เป็นระดับแรกของเลเบล U user จะเห็นแค่เลเบลที่เป็น U เท่านั้น

ผู้ใช้งานระดับ C เห็นเลเบลที่เป็น U เช่นกัน ส่วน S ซึ่งเป็นระดับที่ 2 ของเลเบล S ผู้ใช้งานเห็นเลเบลเป็น U-S ถ้า ระดับแรกของเลเบลสูงกว่าผู้ใช้งานแล้ว ผู้ใช้งานจะมองไม่เห็นข้อมูลของเลเบลนั้น

ตัวอย่างทัปเปิลในเอ็มแอลเอสรีเลชั่น Starship ถูกขยายโดยริชเชอร์เซต และถูกตีความโดยผู้ใช้งานจากระดับความปลอดภัยที่ต่างกัน ดังตารางต่อไปนี้

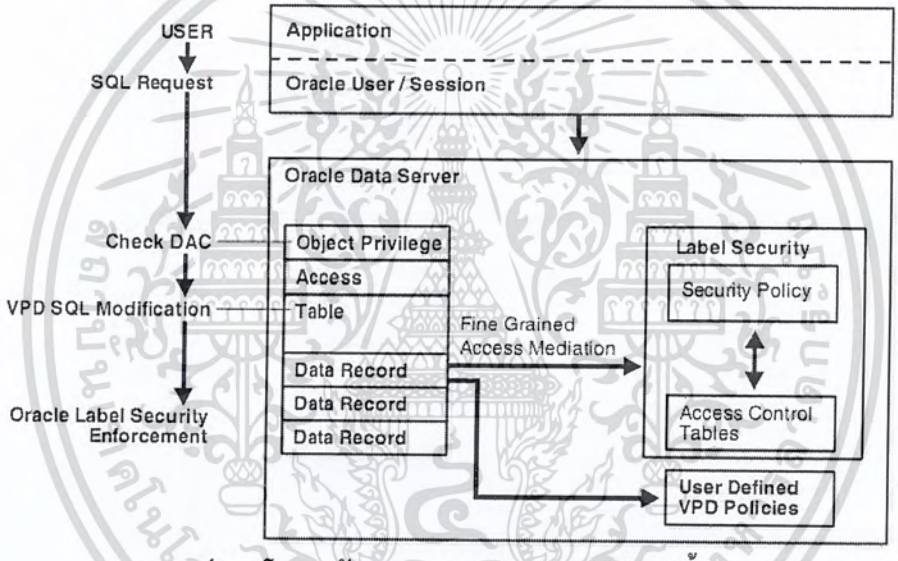
ตาราง 2.15 เอ็มแอลเอสรีเลชั่น Starship ถูกขยายโดยริชเชอร์เซต

	Vessel_Name(K)		Objective		Destination		TC
t1	Avenger	S	Shipping	S	Pluto	S	S
t2	Atlantis	UCS	Diplomacy	UCS	Vulcan	UCS	UCS
t3	Voyager	US	Spying	S	Mars	US	S
t4	Voyager	US	Training	U-S	Mars	US	U-S
t5	Falcon	U-S	Exploration	U-S	Venus	U-S	U-S
t6	Eagle	U	Patrolling	U	Degoba	U	U

บทที่ 3
การนำฐานข้อมูลที่มีความปลอดภัยหลายระดับไปใช้งานในดีบีเอ็มแอส

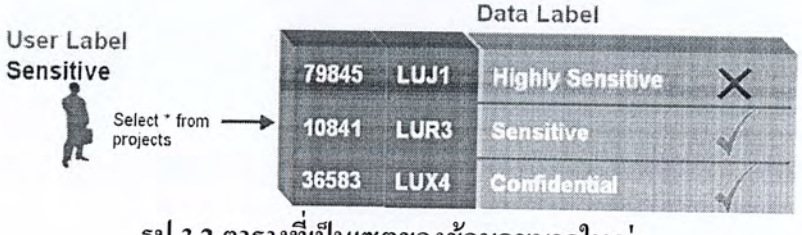
3.1 ออราเคิล

ผลิตภัณฑ์ออราเคิล (Oracle) มีลักษณะที่เกี่ยวข้องกับระดับความปลอดภัยหลายระดับที่มีชื่อว่า ออราเคิลเลเบลซีเคียวริตี้ (Oracle Label Security (OLS)) ซึ่งเป็นลักษณะที่เป็นความลับสามารถใช้เฉพาะในสหรัฐอเมริกาเท่านั้นออราเคิลเลเบลซีเคียวริตี้ มีโครงสร้างดังรูปที่ 3.1



รูป 3.1 โครงสร้างของออราเคิลเลเบลซีเคียวริตี้

ออราเคิลเลเบลซีเคียวริตี้ได้เตรียมความสามารถในการแท็กข้อมูล (tag data) ด้วยดาต้าเลเบล หรือดาต้าคลาสิฟิเคชั่น (data label หรือ data classification) ซึ่งความสามารถนี้ทำให้ฐานข้อมูลรู้ว่าข้อมูลนั้นเป็นข้อมูลที่มีความอ่อนไหว และยังให้ข้อมูลที่มีความอ่อนไหว นั้นอยู่ในตารางที่เป็นเซตของข้อมูลขนาดใหญ่ได้ ดังรูป

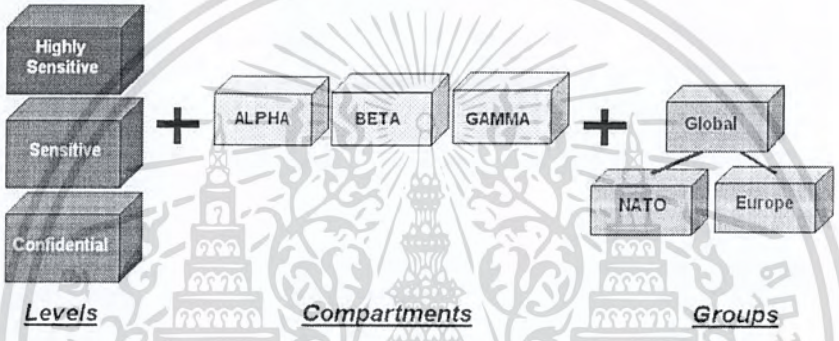


รูป 3.2 ตารางที่เป็นเซตของข้อมูลขนาดใหญ่

โดยการแอสเสจ (Access) เข้าไปยัง sensitive ซึ่งข้อมูลจะถูกควบคุมโดยการเปรียบเทียบดาต้าเลเบลกับเลเบลของผู้ใช้งานที่รีเควิส (request) เข้ามาตามระดับความปลอดภัย ระดับความปลอดภัยนี้ สามารถคิดได้ว่าเป็นต่อขยายของฐานข้อมูลเฉพาะพื้นฐาน (standard database privileges) โดยออราเคิลเลเบลซีเคียวริตี้ถูกบังคับให้ปฏิบัติตามภายในฐานข้อมูลโดยการเตรียมความปลอดภัยที่คงทนถาวรและการกำจัดความต้องการของ มุมมองแอพพลิเคชั่นที่ซับซ้อน

3.2 ดาต้าเลเบลและคอมโพเนนท์

ดาต้าเลเบลของโอแอลเอส (OLS) ประกอบด้วยระดับขั้นต่างๆที่รวมกับ Compartment และ Groups ดังรูปที่ 3.3



รูป 3.3 ดาต้าเลเบลของโอแอลเอส

โดยส่วนประกอบเลเบลเหล่านี้จะถูกใช้ในการสร้างดาต้าเลเบลและใช้เพื่อที่จะมอบหมายระดับความปลอดภัยไปยังฐานข้อมูลหรือชนิดแอพพลิเคชั่นของผู้ใช้งาน

ดาต้าเลเบลประกอบด้วย 3 ส่วน ดังนี้

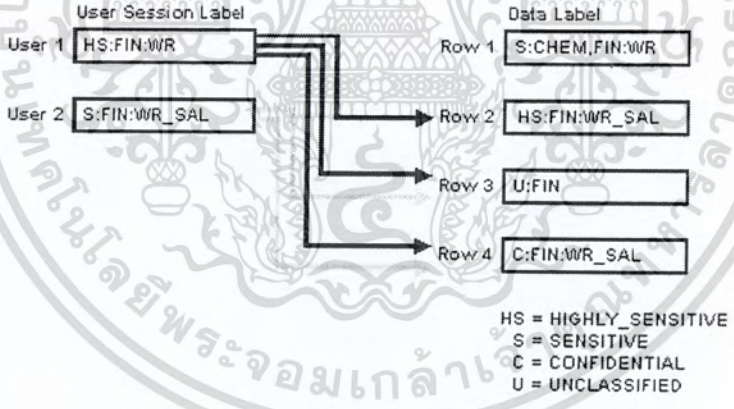
1) Level ถูกใช้สำหรับมอบหมายลำดับขั้นของ Sensitivity โดยแต่ละดาต้าเลเบล จะต้องมีระดับ ตัวอย่างเช่น องค์กรโดยทั่วไปจะกำหนดระดับเป็น Confidential, Sensitive และ Highly Sensitive
2) Compartments เป็นอ็อปชั่นนอลและบางครั้งอาจจะกล่าวได้ว่าเป็นประเภทต่างๆ (categories) ซึ่งเป็นแบบไม่เป็นลำดับขั้น โดยทั่วไป compartment เดียวหรือมากกว่าหนึ่ง compartment ถูกกำหนดเป็นข้อมูลที่เป็นส่วนๆ ในบางครั้ง compartment อาจจะถูกกำหนดเป็น ชนิดของข้อมูลโดยเฉพาะก็ได้
3) Groups เป็นอ็อปชั่นนอลและมีความคล้ายกับ Compartment มาก ต่างกันตรงที่แต่ละ group สามารถมีความสัมพันธ์แม่ลูก (parent child relationship) ได้ โดยส่วนใหญ่จะใช้ Groups ในการแบ่งแยกข้อมูลโดยองค์กร

ออราเคิลเลเบลซีเคียวริตี้ได้เตรียมความสามารถในการกำหนดเลเบลให้เหมาะกับความต้องการของธุรกิจที่มีลักษณะเฉพาะและองค์กร เช่น healthcare, law enforcement และ Human Resources ดังตารางที่ 3.1

ตาราง 3.1 กำหนดเลเบลตามความต้องการของธุรกิจและองค์กรต่างๆ

INDUSTRY	LEVEL	COMPARTMENT	GROUP
Government and Defense	Confidential Secret Top Secret	Alpha Beta	NATO Homeland Security
Law Enforcement	Level 1 Level 2 Level 3	Border Security Drug Enforcement	Local Jurisdiction FBI Justice Department
Human Resources	Confidential Sensitive Highly Sensitive	PII Data Investigation	Global United States Europe
Health Care	Confidential Public	VIP Controls	Physician Laboratory

ตัวอย่างดาต้าเลเบลและยูเซอร์เลเบลดังรูปที่ 3.4



รูป 3.4 ตัวอย่างดาต้าเลเบลและยูเซอร์เลเบล

3.3 ออราเคิลเลเบลซีเคียวริตี้โพลิซี

ออราเคิลเลเบลซีเคียวริตี้โพลิซี (Oracle Label Security Policies) เป็นชื่อของที่เก็บกลุ่มของดาต้าเลเบล, ยูเซอร์เลเบล, โพเทคอ็อบเจ็ค (Protected objects) หลายๆโพลิซี (policy) สามารถถูกกำหนดอยู่บนฐานข้อมูลเดียวกัน โดยแต่ละออราเคิลเลเบลซีเคียวริตี้โพลิซี มีค่าอัตโนมัติของเซต protective enforcement option นี้ เช่น การ READ CONTROL, WRITE CONTROL โดย default enforcement option จะถูกใช้เมื่อถูกประยุกต์ใช้ในตารางที่ถูกใช้งาน นอกจากนี้ enforcement option ยังสามารถปรับเปลี่ยนแก้ไขได้ เมื่อเรากำหนดออราเคิลเลเบลซีเคียวริตี้โพลิซี ชื่อคอลัมภ์จะต้องถูก

เตรียมไว้สำหรับเก็บ data classification label และเมื่อโพลิซีถูกประยุกต์ใช้ในตารางที่ใช้งาน คอลัมภ์ที่เพิ่มเข้าไปจะถูกซ่อนไว้ ดังนั้นแล้วเอสคิวแอลสเตจแมนท์ยังทำงานได้ต่อไปโดยไม่มีการเปลี่ยนแปลง และเมื่อผู้ใช้งานใช้คำสั่ง เช่น SELECT * หรือ DESCRIBE ดังตัวอย่าง

ตัวอย่างที่ 1

พิจารณารูปที่ 3.5

```
SQL> describe emp;
 Name                                      Null?    Type
 ----------------------------------------- -------- --------
 EMPNO                                     NOT NULL NUMBER(4)
 ENAME                                              CHAR(10)
 JOB                                                CHAR(9)
 MGR                                                NUMBER(4)
 SAL                                                NUMBER(7,2)
 DEPTNO                                    NOT NULL NUMBER(2)
 HR_LABEL                                           NUMBER(10)
```

รูป 3.5 ผู้ใช้งานใช้คำสั่ง DESCRIBE

ผู้ใช้งานใช้คำสั่ง DESCRIBE ตาราง emp จะเห็นว่าไม่ได้ ซ่อน HR_LABEL ซึ่งเป็นโพลิซีเลเบลคอลัมภ์

ตัวอย่างที่ 2

```
SQL> describe emp;
 Name                                      Null?    Type
 ----------------------------------------- -------- --------
 EMPNO                                     NOT NULL NUMBER(4)
 ENAME                                              CHAR(10)
 JOB                                                CHAR(9)
 MGR                                                NUMBER(4)
 SAL                                                NUMBER(7,2)
 DEPTNO                                    NOT NULL NUMBER(2)
```

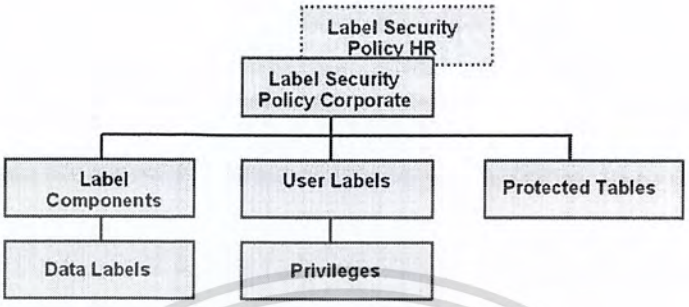
รูป 3.6 ผู้ใช้งานใช้คำสั่ง DESCRIBE ในกรณีที่ตั้งค่าอ็อปชั่นให้ซ่อนโพลิซีเลเบลคอลัมภ์เอาไว้

จากตัวอย่างที่ 2 จะเห็นว่า HR_LABEL หายไป เนื่องจากผู้ดูแลระบบได้ตั้งค่าอ็อปชั่นไว้ให้ซ่อนโพลิซีเลเบลคอลัมภ์เอาไว้ ดังนั้นผู้ใช้งานที่ใช้คำสั่ง DESCRIBE หรือ SELECT * จะมองไม่เห็นคอลัมภ์นี้

3.4 ยูเซอร์เลเบล

ยูเซอร์เลเบล (User Label) เป็นส่วนที่สำคัญส่วนหนึ่งของออราเคิลซีเคียวริตี้เลเบลและเป็นตัวที่กำหนดว่าผู้ใช้งานจะเข้าไปยังข้อมูลที่ป้องกันข้อมูลไว้ด้วยดาต้าเลเบลโดยยูเซอร์เลเบลประกอบ

ด้วยระดับต่ำสุดและระดับสูงสุด ตัวอย่างเช่น ผู้ใช้ถูกกำหนดระดับสูงสุดเป็น sensitive และระดับต่ำสุดเป็นฐานข้อมูลทั่วไป (Public) ของผู้ใช้งานนี้ จะมีเลเบลนี้เป็นค่าเริ่มต้นเมื่อเชื่อมต่อ ไปยังฐานข้อมูล



รูป 3.7 ยูเซอร์เลเบล

3.5 ซีเคียวริตี้เคลียเรนซ์คอมโพเนนซ์

ตาราง 3.2 ซีเคียวริตี้เคลียเรนซ์คอมโพเนนซ์

CLEARANCE COMPONENT:	DESCRIPTION:
Maximum Level	The maximum sensitivity level a user is authorized to access. For example this might be Sensitive or Highly Sensitive.
Minimum Level	The minimum sensitivity level a user is authorized to write data. For example, an administrator can prevent users from labeling data as Confidential by assigning a minimum level of Sensitive.
Default Level	The level used by default when a user connects to the database. For example, a user can set his or her default level to Senstive. When he or she connects to the system, the default level will be initialized to Sensitive.
Row Level	The default level used to label data inserted into the database by the user through the application or directly through a tool such as SQL*Plus.
Read Compartments	The set of compartments assigned to the user and used during READ access mediation. For example, if a user has compartments A, B and C, he could view data which has compartments A and B but not data which has compartments A, B, C and D.
Write Compartments	The set of compartments assigned to the user and used during WRITE access mediation. For example, a user could be given READ and WRITE access to compartments A and B but READ-ONLY access to compartment C. If an application record was labeled with compartments A, B and C, the user would not be allowed to update the record because he or she does not have WRITE access on compartment C.
Read Groups	The set of groups assigned to the user and used during READ access mediation. For example, if a user had the group Manager, he could view data that has the Manager group but not data that had only the Senior VP group.
Write Groups	The set of groups assigned to the user and used during WRITE access mediation. For example, a user could be given READ and WRITE access to group *Senior VP* but READ-ONLY access to group *Manager*. If an application record was labeled with a single group, *Manager*, the user would not be allowed to update the record because he or she does not have WRITE access on the *Manager* group.

3.6 ออราเคิลเลเบลซีเคียวริตี้พรีวิเลจ

ออราเคิลเลเบลซีเคียวริตี้ (Oracles Label Security) มีหลายพรีวิเลจ (privileges) ที่สามารถกำหนดให้กับผู้ใช้งาน ตัวอย่างของพรีวิเลจ คือ READ ซึ่งเป็นพรีวิเลจอย่างง่ายที่ให้ผู้ใช้งานเห็นข้อมูลทั้งหมดโดยไม่ต้องคำนึงถึงดาต้าคลาสิฟิเคชั่น (data classification) นอกจากนี้ยังมีพรีวิเลจอีกหลายชนิด ดังตารางที่ 3.3

ตาราง 3.3 ออราเคิลเลเบลซีเคียวริตี้

PRIVILEGE NAME	DESCRIPTION
READ	The READ authorization enforces no additional read access control. Access mediation is still enforced on UPDATE, INSERT and DELETE operations. Oracle Label Security makes no mediation check on SELECT
FULL	The FULL authorization turns off all Oracle Label Security access mediation. A user with the FULL authorization can perform SELECT, UPATE, INSERT and DELETE operations with no label authorizations. Note that Oracle SYSTEM and OBJECT authorizations are still enforced. For example, a user must still have SELECT on the application table. The FULL authorization turns off the access mediation check at the individual row level.
WRITEDOWN	The WRITEDOWN authorization allows a user to modify the level component of a label and lower the sensitivity of the label. For example, application data which is labeled *Highly Sensitive: Alpha, Beta* could be changed to *Sensitive: Alpha, Beta*. This authorization is only applicable to policies that use the *label update* enforcement option.
WRITEUP	The WRITEUP authorization allows a user to modify the level component of a label and raise the sensitivity of the label. For example, application data which is labeled *Sensitive: Alpha, Beta* could be changed to *Highly Sensitive: Alpha, Beta*. Note that the Maximum Level label authorization assigned to the user would limit modification. This authorization is only applicable to policies that use the *label update* enforcement option.
WRITEACROSS	The WRITEACROSS authorization allows a user to modify the compartments and groups in a label to any valid compartment and group defined in Oracle Label Security for the policy. For example, data labeled *Sensitive: Alpha* could be modified to *Sensitive: Alpha, Beta* even though the user was not authorized for the Delta compartment. This authorization is only applicable to policies that use the label update enforcement option.
PROFILEACCESS	The PROFILE ACCESS authorization allows a user to assume the Oracle Label Security authorizations of another user. For example, user Scott who has access to compartments A, B, and C could assume the profile of user Joe who has access to compartments A, B, C and D. This functionality might be useful in an environment where an application uses a single application account for all application users. Note that the PROFILEACCESS privilege cannot be granted to a stored procedure.

3.7 เอนฟอร์ซเมนท์อ็อปชั่น

เลเบลซีเคียวริตี้โพลิซี (Label Security Policy) หลายตัวที่อยู่บนฐานข้อมูลตัวเดียวกันมีเอนฟอร์ซเมนท์อ็อปชั่น (enforcement option) ที่ต่างกัน โพลิซีของเอนฟอร์ซเมนท์อ็อปชั่นสามารถปรับปรุงแก้ไขได้ เมื่อเลเบลซีเคียวริตี้โพลิซีถูกสร้างขึ้น การตั้งค่าอัตโนมัติของเอนฟอร์ซเมนท์

อ็อปชั่นก็จะถูกสร้างขึ้นด้วย ตัวอย่างของเอนฟอร์ซเมนท์อ็อปชั่น คือ READ CONTROL นอกจากนี้ยังมีเอนฟอร์ซเมนท์อ็อปชั่นอื่นๆ อีกดังตารางที่ 3.4

ตาราง 3.4 เอนฟอร์ซเมนท์อ็อปชั่น

ENFORCEMENT OPTION	DESCRIPTION
READ CONTROL	Applies policy enforcement to SELECT operations using the Oracle Label Security algorithm for read access.
INSERT CONTROL	Applies policy enforcement to INSERT operations using the Oracle Label Security algorithm for write access
UPDATE CONTROL	Applies policy enforcement to UPDATE operations using the Oracle Label Security algorithm for write access.
DELETE CONTROL	Applies policy enforcement to DELETE operations using the Oracle Label Security algorithm for write access.
WRITE CONTROL	Applies policy enforcement on INSERT, UPDATE, and DELETE operations. If this option is set, it enforces INSERT_CONTROL, UPDATE_CONTROL, and DELETE_CONTROL.
LABEL DEFAULT	If the user does not explicitly specify a label on INSERT, the user's default row label value is used. By default, the row label value is computed internally by Oracle Label Security using the user's label. The default value would be comprised of the default ROW LEVEL combined with the WRITE COMPARTMENTS and WRITE GROUPS. A user can set the row label independently, but only to: A level which is less than or equal to the level of the session label, and greater than or equal to the user's minimum level. Include a subset of the compartments and groups from the session label, for which the user is authorized to have write access.
LABEL UPDATE	Applies policy enforcement to UPDATE operations that set or change the value of a label attached to a row. The WRITEUP, WRITEDOWN, and WRITEACROSS privileges are only enforced if the LABEL_UPDATE option is set.
LABEL CHECK	Applies READ_CONTROL policy enforcement to INSERT and UPDATE statements to assure that the new row label is read-accessible by the user after and INSERT or UPDATE statement.
NO CONTROL	Applies no enforcement options. A labeling function or a SQL predicate can nonetheless be applied.

3.8 การเปิดใช้งานออราเคิลเลเบลซีเคียวริตี้

ในการติดตั้งออราเคิลเลเบลซีเคียวริตี้ (Oracle Label Security) ได้มีการสร้างค่ายูเซอร์แอดเคาวน์อัตโนมัติขึ้น (default user account) ซึ่งก็คือ LBACSYS ซึ่งจะเป็นแอดเคาวน์ (account) ที่มีสิทธิพิเศษในการบริหารจัดการโอแอลเอส (OLS) ดังตัวอย่าง ผู้ใช้งานได้สร้างออราเคิลเลเบลซีเคียวริตี้โพลิซี่ ขึ้นใช้ชื่อว่า HRSEC และก็ให้ผู้ใช้งาน HRSEC_DBA เป็นคนบริหารจัดการโพลิซีซึ่งใช้คำสั่งดังนี้

```
CONNECT LBACSYS
EXECUTE SA_SYSDBA.CREATE_POLICY ('HRSEC', 'HR_LABEL','HIDE');
```

การสร้างโอแอลเอสโพลิซี (OLS policy) มี 6 ขั้นตอน ดังนี้

1) สร้างโพลิซี
2) สร้างเลเบลคอมโพแนนท์ (Label Component) สำหรับโพลิซี
3) สร้างดาต้าเลเบลสำหรับโพลิซี
4) การให้สิทธิ์ผู้ใช้งานสำหรับโพลิซี
5) นำโพลิซีไปใช้งานในตารางของฐานข้อมูล
6) เพิ่มโพลิซีเลเบลเข้าไปในแถวของตาราง

ตาราง 3.5 ผู้ดูแลระบบกำหนดเลเบลแท็ก

Label Tag	Label String
10000	P
20000	C
21000	C:FNCL
21100	C:FNCL,OP
30000	S
31110	S:OP:WR
40000	HS
42000	HS:OP

คำสั่งที่ใช้ในการสร้างดาต้าเลเบล

```
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL ('HRSEC', '1000','C');
EXECUTE SA_COMPONENTS.CREATE_LEVEL ('HRSEC',
'3000','HS','HIGHLY_SENSITIVE');
```

คำสั่งที่ใช้ในการสร้าง COMPARTMENT และ GROUP

```
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT ('HRSEC', 100,'PII', 'Personally
Identifiable Information');
EXECUTE SA_COMPONENTS.CREATE_GROUP ('HRSEC', 500,'EU','Europe');
```

คำสั่งที่ใช้ในการสร้าง USER LABEL

```
EXECUTE SA_USER_ADMIN.SET_USER_LABELS ('HRSEC', 'TRODGERS_US', 'S');
EXECUTE SA_USER_ADMIN.SET_USER_LABELS ('HRSEC', 'JSMITH_US', 'S:PII');
```

การกำหนดเลเบลไปยังข้อมูลแถว

สำหรับข้อมูลที่มีอยู่แล้วเลเบลสามารถถูกกำหนดโดยฟังก์ชันเลเบล ดังตัวอย่างเช่น

ตัวอย่างที่ 1 CHAR_TO_LABEL เป็นฟังก์ชัน สำหรับแปลง character string เป็นเลเบล ซึ่งมีซินแทค ดังนี้

```
FUNCTION CHAR_TO_LABEL(
    Policy_name          IN VARCHAR2,
    Label_stringIN VARCHAR2)
RETURN NUMBER;
```

ตัวอย่างการใช้งานฟังก์ชั่น CHAR_TO_LABEL

```
INSERT INTO emp (empno, hr_label)
VALUES (999, CHAR_TO_LABEL ('HR', 'S:A,B:G5'));
```

โดยที่ HR เป็นชื่อของ policy, S คือ Sensitivity level, A, B คือ compartment และ G5 คือ group

ตัวอย่างที่ 2 LABEL_TO_CHAR เป็นฟังก์ชัน สำหรับแปลงเลเบลเป็น character string ซึ่งมีซินแทค ดังนี้

```
FUNCTION LABEL _TO_ CHAR (
    label        IN NUMBER)
RETURN VARCHAR2;
```

ตัวอย่างการใช้งานฟังก์ชัน LABEL_TO_CHAR

```
SELECT label_to_char (hr_label) AS label, ename from emp
WHERE ename = 'RWRIGHT';
```

Result ที่ได้ จะได้ดังนี้

```
LABEL          ENAME
-------------  ----------
S:A,B:G1       RWRIGHT
```

รูป 3.8 ผลของการใช้ฟังก์ชัน LABEL_TO_CHAR

บทที่ 4
การนำไปใช้ของระบบฐานข้อมูลที่มีความปลอดภัยหลายระดับ

4.1 บทนำ

การบริหารลูกค้าสัมพันธ์ (Customer Relationship Management หรือ CRM) เป็นเทคโนโลยีที่เชื่อมโยงความสัมพันธ์ระหว่างธุรกิจและลูกค้า โดยมีวัตถุประสงค์เพื่อสร้างความภักดีระยะยาวของลูกค้าที่มีต่อธุรกิจ โมเดลความปลอดภัยของข้อมูลหลายระดับ (Multilevel secure data model) เป็นจุดเริ่มต้นของการพัฒนาโมเดลของฐานข้อมูลสำหรับบริหารจัดการข้อมูลที่อยู่ในระบบ ด้วยการจัดระดับความปลอดภัยเป็นแบบระดับขั้น

4.2 การบริหารลูกค้าสัมพันธ์

การบริหารลูกค้าสัมพันธ์ (Customer Relationship Management หรือ CRM) เป็นกลยุทธ์การบริหารการจัดการอย่างหนึ่ง ที่ถูกออกแบบมาเพื่อช่วยให้องค์กรสามารถจัดการกระบวนการต่างๆ ภายในให้ดำเนินการได้อย่างสอดคล้องและตอบสนองต่อความต้องการของลูกค้า เพื่อให้ลูกค้าเกิดความพึงพอใจสูงสุด นำมาซึ่งรายได้ที่เพิ่มขึ้น และการทำกำไรในระยะยาว โดยจะมีการมุ่งเน้นทางด้านการสร้างความสัมพันธ์อันดีกับลูกค้าในทุกระดับ เพื่อทำให้เกิดความภักดีการใช้สินค้าหรือบริการ ทั้งนี้กลยุทธ์ในการบริหารลูกค้าสัมพันธ์จะต้องนำเอาเทคโนโลยีมาเป็นเครื่องมือเพื่อสร้างความสะดวกในการติดต่อสื่อสารในทุกๆช่องทาง รวมทั้งใช้ในการเก็บข้อมูลและความต้องการของลูกค้า ซึ่งเป็นส่วนที่สำคัญที่ทำให้องค์กรสามารถดำเนินงานเพื่อเอาอกเอาใจ สร้างความพึงพอใจให้ลูกค้าอย่างถูกต้อง มีประสิทธิภาพและรวดเร็ว

ในปัจจุบันองค์กรส่วนมากมีการดำเนินกิจกรรมต่างๆ ผ่านสื่ออิเล็กทรอนิกส์ หรือที่เรียกกันว่าธุรกิจเชิงอิเล็กทรอนิกส์ (E-business) ซึ่งเมื่อไม่นานมานี้ได้มีการผสมผสานธุรกิจเชิงอิเล็กทรอนิกส์เข้ากับเทคโนโลยีซีอาร์เอ็ม เพื่อเพิ่มศักยภาพในการบริหารธุรกิจที่ดียิ่งขึ้น โดยที่รูปแบบของซีอาร์เอ็มจะถูกนำเสนอและถูกสร้างอยู่ในส่วนต่างๆของระบบธุรกิจเชิงอิเล็กทรอนิกส์

นอกจากนี้ขบวนการวิเคราะห์ วิจัย และการวางรูปแบบของยุทธศาสตร์ในการสร้างความภักดีของลูกค้าที่มีต่อองค์กรนั้น ข้อมูลต่างๆ ที่ได้นั้น จะต้องเป็นข้อมูลที่เข้าถึงส่วนลึกของลูกค้า (Insightful) และต้องเป็นข้อมูลที่ถูกต้องจากแหล่งข้อมูลที่เชื่อถือได้ นำเอาข้อมูลเกี่ยวกับลูกค้ามาวิเคราะห์และปรับเปลี่ยนมาเป็นความรู้ หรือการเรียนรู้ในตัวลูกค้า เพื่อสร้างประโยชน์และทำกำไรสูงสุดแก่องค์กร ข้อมูลต่างๆของลูกค้าจะต้องถูกจัดเก็บลงฐานข้อมูล ซึ่งจะต้องมีความถูกต้องและปลอดภัยสูงสุด จึงมีผู้คิดค้นโมเดลขึ้นมาที่จะให้สิทธิ์ในการเข้าถึงข้อมูลในหลายๆระดับ และมีการป้องกันข้อมูลไว้อย่างปลอดภัย ซึ่งก็คือ โมเดลความปลอดภัยของข้อมูลหลายระดับที่ใช้สำหรับการ

บริหารจัดการข้อมูลในองค์กรด้วยการจัดโครงสร้างของข้อมูลเป็นระดับต่างๆตามความปลอดภัยที่กำหนดไว้นั่นเอง

4.3 วิธีการเข้าถึงฐานข้อมูล

โมเดลการเข้าถึงข้อมูล (database access model) จะถูกใช้ประโยชน์ในเครื่องมือซอร์ฟแวร์ซีอาร์เอ็มโดยธุรกิจเชิงอิเล็กทรอนิกส์ เป็นตัวพิสูจน์ปัญหาของโมเดลซึ่งจะแจ้งไปยังผู้ใช้งานและเสนอทางเลือกของวิธีต่างๆที่จะชี้ไปยังปัญหาและเพิ่มประสิทธิภาพของการแก้ปัญหาที่มีอยู่ คนส่วนใหญ่มักจะใช้ซอร์ฟแวร์แอพพลิเคชั่นเป็นตัวสร้างครอบฐานข้อมูลแบบรีเลชั่นนอล แต่วัตถุประสงค์ของ แอพพลิเคชั่นซีอาร์เอ็ม ก็คือให้ความถูกต้องของเนื้อหาข้อมูลจากฐานข้อมูลที่ใช้อยู่ ซึ่งข้อมูลที่ถูกต้องจะอยู่ในรูปแบบของความพึงพอใจของลูกค้าที่ชัดเจน โดยจะมีวิธีการในการเข้าถึงข้อมูลในฐานข้อมูลของ ซีอาร์เอ็ม (ฐานข้อมูลซีอาร์เอ็ม) ดังนี้

4.3.1 open access

เป็นวิธีพื้นฐานที่สุดที่อนุญาตให้ผู้ใช้ทุกคนเข้าดูทุกข้อมูลได้ เป็นวิธีที่ไม่ได้มีการควบคุมและป้องกันการเข้าถึงข้อมูล

4.3.2 discretionary access control

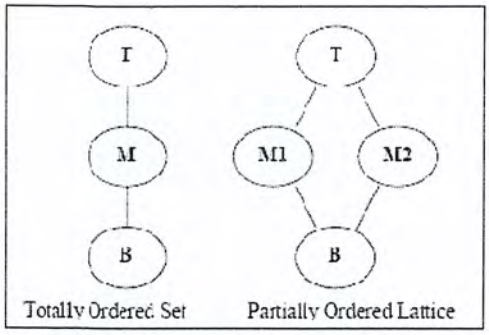
การควบคุมการเข้าถึงที่ตัดสินใจด้วยตัวเอง ซึ่งการเข้าถึงจะเป็นการให้สิทธิ์หรือถอนสิทธิ์ของแต่ละคนที่อยู่ในระบบ โดยข้อมูลเกี่ยวกับสิทธิ์ต่างๆจะถูกเก็บและรักษาในเมตริกซ์ subject/object

4.3.3 mandatory access control

การควบคุมการเข้าถึงที่ขึ้นอยู่กับผู้มีอำนาจ สร้างสิ่งต่างๆในระดับความเป็นไปได้ของผู้มีส่วนร่วมเช่นเดียวกับข้อมูล ซึ่งจะถูกจัดเป็นกลุ่มที่มีจำนวนจำกัดของระดับการเข้าถึง วิธีการนี้จะเป็นวิธีที่มีการเข้าถึงข้อมูลที่แตกต่างกันขึ้นอยู่กับระดับความปลอดภัยของผู้ใช้งาน การสร้างการควบคุมการเข้าถึงนี้จะถูกอ้างอยู่ในระบบของเอ็มแอลเอส โดยศักยภาพของมันจะพิจารณาจากมุมมองของระบบฐานข้อมูลซึ่งประกอบด้วยข้อมูลที่เปราะบางกับการเข้าถึงที่เข้มงวดและความปลอดภัยที่ต้องการ

4.4 โมเดลความปลอดภัยหลายระดับ

โมเดลความปลอดภัยหลายระดับจะขึ้นอยู่กับการจัดแบ่งระดับของสิ่งต่างๆที่อยู่ในระบบ โดยที่ระดับการเข้าถึงจะเป็นตัวที่แสดงการจัดแบ่งผู้ใช้งานแต่ละคนจะมีระดับการเข้าถึงเป็นของตัวเอง ในขณะที่ระดับการเข้าถึงข้อมูลจะถูกแสดงด้วย access label ซึ่งจะประกอบไปด้วยหนึ่งระดับหรือหลายๆระดับ ในโมเดลจะมีเซตของระดับการเข้าถึงข้อมูล 2 แบบคือ Totally Ordered Set และ Partially Ordered lattice ซึ่งแสดงได้ดังรูปที่ 4.1



รูป 4.1 เซตของระดับการเข้าถึง

จากตัวอย่างจะแสดงให้เห็น 3 ระดับ อันได้แก่ T = Top , M = Middle และ B = Bottom โดยที่ T จะเป็นระดับสูงสุดโดยสูงกว่า M และ B M เป็นระดับที่สูงกว่า B ($T \geq M \geq B$) ทั้งหมดนี้เป็นทฤษฎีของเบล และพาลาดูลา โดยจะถูกเรียกว่าคุณสมบัติความปลอดภัยพื้นฐาน (simple property) ที่กล่าวไว้ว่า "subject หรือผู้ใช้สามารถอ่าน object หรือข้อมูลได้ถ้าระดับการเข้าถึงของผู้ใช้เหนือกว่าระดับของ object" พูดง่ายๆว่าอ่านได้ในระดับเดียวกันหรือต่ำกว่าได้ แต่ห้ามอ่านสูงกว่าระดับตัวเอง

สิ่งที่สำคัญที่สุดของโมเดลเอ็มแอลเอส นี้คือการตีความหมาย โดยจะเพิ่มการตีความซึ่งขึ้นอยู่กับหลักการของความเชื่อ (Believe) ผู้ใช้งานจะมองเห็นและเชื่อถือข้อมูลในระดับของตัวเอง และมองเห็นข้อมูลของระดับที่ต่ำกว่า จากตัวอย่าง T มองเห็นข้อมูลของ T M และ B แต่จะเชื่อถือข้อมูลของ T ว่าถูกต้องเท่านั้น และจะทำการตรวจสอบข้อมูลของ M และ B ในภายหลัง

ในโมเดลพื้นฐานความเชื่อ คำว่า "belief" จะถูกใช้แทน knowledge เพราะว่าค่าของแอททริบิวเดียวกันของข้อมูลเดียวกัน สามารถมีอยู่ได้ในหลายๆระดับ จะถูกใช้ในโมเดลเอ็มแอลเอสรีเลชั่นนอลดาต้า (MLS Relational data) โดยทั่วไปแล้วจะมักจะช่วยให้บริหารสิทธิ์การเข้าถึงอย่างถูกต้องง่ายขึ้น เมื่อเปรียบเทียบกับวิธีการ discretionary access control สามารถทำให้ผู้ใช้งานเห็นถึงความแตกต่างในการแปลความ (difficult-to-interpret) ของข้อมูลจากระดับการเข้าถึงข้อมูลของตัวเองและจากระดับระดับการเข้าถึงข้อมูลที่ต่ำกว่า ความกำกวมที่เกิดจากการแปลความจะถูกขจัดออกโดยใช้โมเดลของบีซีเอ็มแอลเอส ซึ่งเป็นการวิธีสำหรับการอ้างสิทธิ์และการตีความความน่าเชื่อถือของข้อมูลที่มีระดับการเข้าถึงที่ต่ำกว่าบีซีเอ็มแอลเอสจะจดจำสิ่งที่ผู้ใช้งานระดับสูงกว่าเห็นตีความของข้อมูลที่อยู่ระดับต่ำกว่า โดยจะขึ้นอยู่กับการยืนยันและความแตกต่างของข้อมูลบนระดับนั้นๆ ซึ่งการแปลความหมายจะได้ดังตารางนี้

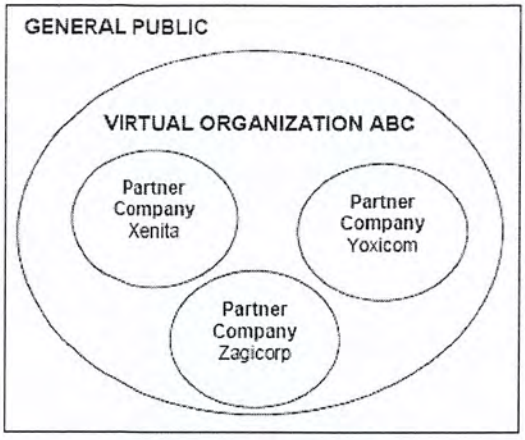
ตาราง 4.1 เซตของการแปลความที่ถูกต้องของข้อมูลระดับต่ำที่เกิดจากผู้ใช้ระดับสูง

Lower-level information is *true and complete* – i.e. consistent with a higher-level "view of the world".
Lower-level information is *partially different* - i.e. inconsistent with a higher-level view of the world where some attribute values for the same entity may appear the same to the lower level user while some do not.
Lower-level information is *completely different* - i.e. inconsistent with a higher-level view of the world where all attribute values for the same entity appear different to the lower level user
Lower-level information has been entered by a lower level user but *not yet interpreted* by a higher level.

โมเดลบีซีเอ็มแอลเอสจะสามารถแปลความกำกวมได้ทุกข้อมูลที่มองเห็น และให้การเข้าถึงข้อมูลของผู้ใช้งานไปยังความเชื่อถือของผู้ใช้งานที่ต่ำกว่าได้ โมเดลนี้จะให้ความสำเร็จที่สำคัญสำหรับการบริหารจัดการข้อมูลที่มีการเปลี่ยนแปลงตลอดเวลา เพื่อให้สามารถมั่นใจได้ว่าสามารถแสดงและแปลความบนมุมมองของข้อมูลที่มีความหลากหลายที่อนุญาตให้เกี่ยวข้องกับผู้ใช้งานอื่นๆได้อย่างถูกต้องและทันสมัย ต่อไปจะเป็นตัวอย่างของการแสดงวิธีการที่เอ็มแอลเอสใช้เก็บข้อมูลรวมทั้งความสัมพันธ์ของข้อมูลต่างๆในซีอาร์เอ็ม

ตัวอย่างที่ 1

มีองค์กรอยู่องค์กรหนึ่งชื่อ ABC (virtual organization ABC) ที่ประกอบไปด้วย 3 บริษัทที่เป็นหุ้นส่วนอยู่ (3 partner company) ที่มีชื่อว่า Xenita, Yoxicom และ Zagicorp ในแต่ละบริษัทจะมีการเก็บรวบรวมข้อมูลต่างๆไว้และมีการแสดงข้อมูลให้เห็นหลายๆมุมมอง (view) ซึ่งบริษัท Xenita มีการแสดงข้อมูลบางส่วนไปยังองค์กร ABC และบุคคลทั่วไป โดยจะให้ข้อมูลเกี่ยวกับลูกจ้างที่ชื่อว่า A.Teller แผนก R&D แค่ภายในบริษัทเท่านั้น ให้ข้อมูลเกี่ยวกับลูกจ้างที่ชื่อว่า D.Jones แผนก Information systems ไปยังองค์กร ABC (รวมทั้งบริษัทตัวเองด้วย)และให้ข้อมูลเกี่ยวกับลูกจ้างที่มีชื่อว่า M.Smith แผนก customer service ให้กับบุคคลทั่วไป (general public)



รูป 4.2 องค์กร ABC

ตาราง 4.2 ลูกจ้างของบริษัท Xenita

Name	Position	Level
M. Smith OPI	Cust. Service OPI	OPI
D. Jones PI	Inf. Systems PI	PI
A. Teller I	R&D I	I

จากตารางที่ 4.2 จะใช้เซตของการเข้าถึงข้อมูลที่ประกอบไปด้วย O (open), P (Privileged) และ I (internal) ซึ่ง I เหนือกว่า P และ P เหนือกว่า O ($I \geq P \geq O$) แต่ละแถวจะถูกจัดกลุ่มตามระดับการเข้าถึงข้อมูล โดยจะใช้คุณสมบัติความปลอดภัยพื้นฐาน เพื่อให้มั่นใจว่าผู้ใช้งานที่ระดับ I สามารถอ่านได้ 3 แถว ในขณะที่ผู้ใช้งานที่ระดับ P อนุญาตให้อ่านได้แค่ 2 แถวแรก และผู้ใช้งานที่ระดับ O อนุญาตให้อ่านได้แค่แถวแรกแถวเดียว สรุปว่าข้อมูลในระดับ O จะสามารถแสดงให้บุคคลทั่วไปเห็นได้ (general public) ส่วนข้อมูลในระดับ P จะแสดงให้เห็นแค่สมาชิกที่อยู่ในองค์กร ABC และสุดท้ายข้อมูลในระดับ I จะแสดงให้เห็นเฉพาะสมาชิกในบริษัท Xenita เท่านั้น

แอสเสซเลเบล (Access label) ประกอบด้วยมากกว่า 1 เลเบลขึ้นไป จากตัวอย่างข้างต้น เช่น OPI ระดับ O จะเท่ากับ O เท่านั้น ระดับ P จะเท่ากับ OP และระดับ I จะเท่ากับ OPI ทั้ง 3 ระดับ จะตีความ OPI (ข้อมูลของระดับ O) ว่าเป็นข้อมูลถูกต้องสมบูรณ์ ในทางตรงกันข้ามก็จะอนุญาตให้แต่ละระดับตีความได้แตกต่างกันด้วย เช่น O-PI ระดับ O จะเท่ากับ O เท่านั้น ระดับ P จะเท่ากับ O-P และระดับ I จะเท่ากับ O-PI โดยที่ระดับ O จะตีความข้อมูลว่าถูกต้องสมบูรณ์ แต่ระดับ P และ I จะตีความว่าไม่ถูกต้องและไม่น่าเชื่อถือ (เกิดจากหลายๆสาเหตุ)

จากตัวอย่างจะตีความได้ว่า บริษัท Xenita จะมีข้อมูลของลูกจ้างที่มีชื่อว่า M.Smith อยู่แผนก customer service ซึ่งข้อมูลของ M.Smith สามารถแสดงให้เห็นได้ทั้งหมดรวมทั้งบุคคลทั่วไปด้วย แต่ข้อมูลของ D.Jones ที่อยู่แผนก Information systems สามารถแสดงข้อมูลให้เห็นแค่ภายใน

องค์กร ABC และข้อมูลของ A.Teller ที่อยู่แผนก customer service สามารถแสดงให้เห็นแค่ภายในบริษัท Xenita เท่านั้น

สรุปแล้วแอสเสซเลเวลเลเบล (access level label) เป็นกุญแจสำคัญของโมเดลที่เป็นตัวเสนอถึงความเชื่อที่ถูกต้อง โดยสถานะเลเบล ที่เลเวลหรือหลายๆ เลเวล ยอมรับว่าถูกต้องสมบูรณ์ ในทำนองเดียวที่ทุกๆแอททริบิว มีเลเบลเป็นของตัวเองซึ่งใช้อ้างไปยังความถูกต้องสมบูรณ์ของค่าในแอททริบิว นั้นๆ

4.5 การนำเอาการเข้าถึงข้อมูลในโมเดลมาใช้ประโยชน์

ตัวอย่างต่อไปนี้จะทำให้เห็นภาพของการนำการวิธีการเอ็มแอลเอสมาใช้อย่างง่ายโดยประยุกต์ใช้ในธุรกิจเชิงอิเล็กทรอนิกส์ หรือการบริหารจัดการข้อมูลของซีอาร์เอ็ม

ตัวอย่างที่ 2

ระบบอินเตอร์เน็ตของร้านหนังสือที่มีชื่อว่า Andes มีโครงสร้างตารางของลูกค้า โดยจะแบ่งประเภทของลูกค้าดังต่อไปนี้

1) S (spacial) จะได้รับสิทธิพิเศษในการขนส่งสินค้าฟรีผ่านทางบริษัท USPS และ FedEx
2) V (valuable) จะได้รับสิทธิพิเศษในการขนส่งสินค้าฟรีผ่านทางบริษัท USPS แต่จะต้องเสียค่าขนส่งสินค้าเป็นจำนวน $10.99 สำหรับขนส่งสินค้าผ่านทางบริษัท FedEx
3) R (regular) จะต้องเสียค่าขนส่งสินค้าเป็นจำนวน $3.99 สำหรับขนส่งสินค้าผ่านทางบริษัท USPS และ $10.99 สำหรับขนส่งสินค้าผ่านทางบริษัท FedEx

สำหรับจากจัดโครงตารางข้างต้นจะช่วยในกระบวนการสร้างและรักษาความสัมพันธ์ของลูกค้าในระยะยาวกับลูกค้าในทุกๆระดับ ทางร้านหนังสือค้า Andes นี้ต้องการที่จะเผยแพร่ข้อมูลเกี่ยวกับรางวัลซึ่งเป็นการให้ผลประโยชน์แก่ลูกค้าที่เป็นลูกค้าระดับสูง โดยลูกค้าทั่วไปจะไม่สามารถเห็นสิทธิประโยชน์ข้างต้นได้นั่นเอง

ตารางที่ 4.3 จะแสดงตารางเอ็มแอลเอสของร้านหนังสือ Andes และค่าขนส่งสินค้าซึ่งถูกพัฒนาอยู่บนความต้องการที่ให้มาจากตัวอย่างที่ 2

ตาราง 4.3 ค่าขนส่งสินค้าของร้านหนังสือ Andes

Service Name	Price	Level
Shipping USPS RVS	$3.99 R-VS	R-VS
Shipping USPS RVS	Free VS	VS
Shipping FEDEX RVS	$10.99 RV-S	RV-S
Shipping FEDEX RVS	Free S	S

ร้านหนังสือ Andes ทำการเผยแพร่ข้อมูลที่ถูกต้องไปยังลูกค้า โดยจัดแบ่งระดับลูกค้าตามระดับการเข้าถึงข้อมูลของลูกค้าในแต่ละคน ไม่เพียงแต่ว่าลูกค้าแต่ละคนจะได้เห็นราคาค่าขนส่งที่สินค้า

ที่ถูกต้องแล้ว แต่ลูกค้าระดับสูงจะสามารถเห็นสิทธิประโยชน์และผลกำไรของสถานะของตนเองได้อีกด้วย ตัวอย่างเช่น ลูกค้าระดับ S เห็นว่าตนสามารถส่งสินค้าผ่านบริการของบริษัท USPS และ FedEx ได้ฟรี แต่ในขณะเดียวกันก็จะเห็นว่าลูกค้าที่ระดับ R ต้องทำการชำระค่าบริการขนส่งสินค้าเป็นจำนวนเงิน $3.99 สำหรับบริษัท USOS และลูกค้าที่ระดับ R และ V จะต้องชำระค่าบริการขนส่งสินค้าเป็นจำนวนเงิน $10.99 สำหรับบริษัท FedEx ในระหว่างนั้นลูกค้าในระดับต่ำกว่าจะไม่เห็นข้อมูลของลูกค้าระดับที่สูงกว่า นั่นคือไม่เห็นได้ว่าลูกค้าระดับบนได้สิทธิ์ขนส่งสินค้าได้ฟรี ซึ่งมันเป็นข้อเท็จจริงของร้านนี้เอง

4.6 ประโยชน์ของการนำเอาเอ็มแอลเอสมาใช้

ตัวอย่างที่ 3

แสดงวิธีการติดตั้งตารางค่าขนส่งสินค้าของร้านหนังสือ Andes เพื่อจะแสดงข้อมูลดังตัวอย่างที่ 2 โดยใช้วิธีการ discretionary access control

ตาราง 4.4 ค่าขนส่งสินค้าของร้านหนังสือ Andes กับการแยกเซตของสิทธิในการอ่านแต่ละแถว

Service Name		Price	*Readable by*
Shipping	USPS	$3.99	R, V, S
Shipping	USPS	Free	V, S
Shipping	FEDEX	$10.99	R, V, S
Shipping	FEDEX	Free	S

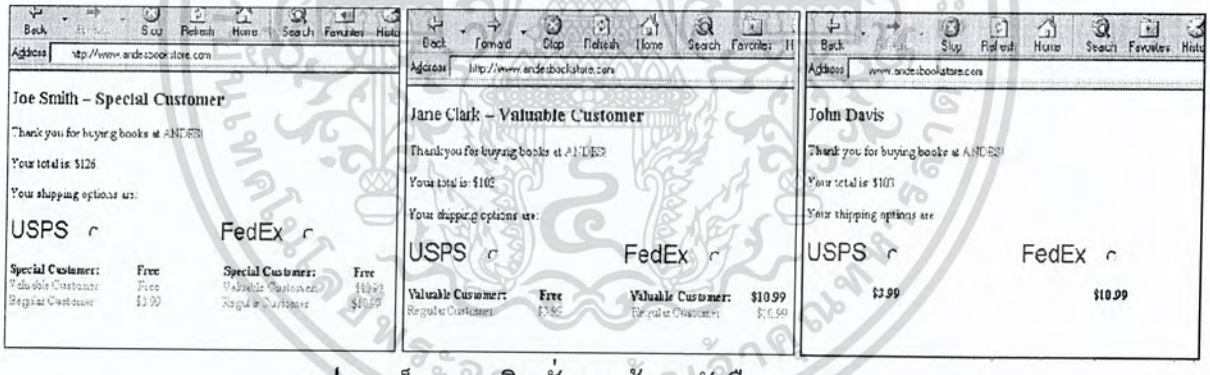
ตารางที่ 4.4 ไม่มีข้อมูลเกี่ยวกับความหมายของ 2 เรคอร์ดที่แยกกันสำหรับค่าขนส่งของบริษัท FedEx มีสถานะดังต่อไปนี้ " ลูกค้าระดับ S สามารถเห็นข้อมูลราคาที่แตกต่างกันทั้ง 2 สำหรับการขนส่งสินค้าของบริษัท FedEx คือ ฟรีและ $10.99 ซึ่ง fact สำหรับลูกค้าระดับ s คือส่งฟรี ซึ่งมีการสนับสนุน fact นี้โดยระดับ S จะเห็นว่าลูกค้าระดับ R และ V ต้องจ่ายเงินเป็นจำนวน $10.99 ที่บริการเดียวกันนี้เอง"

ในวิธีการ discretionary access control ถ้าลูกค้าหลายคนมีการเข้าถึงที่ข้อมูลเดียวกันในฐานข้อมูลนี้ เราสามารถจัดการแบ่งการเข้าถึงนี้ออกเป็นกลุ่มๆได้ โดยผู้ดูแลระบบจะเป็นคนจัดการทุกๆเวลาแต่ละกลุ่มจะมีการเพิ่มเนื้อหาใหม่ๆลงฐานข้อมูลเสมอๆ ซึ่งสิทธิในการเพิ่มข้อมูลต่างๆนี้จะจัดการโดย admin สรุปแล้วการกระทำต่างๆที่เกิดการเปลี่ยนแปลงทั้งผู้ใช้งานและเนื้อหาที่จะต้องการบำรุงรักษานี้ ซึ่งจะทำโดยผู้ดูแลระบบ ทั้งการให้สิทธิ์ (assigning revoking) หรือการถอนสิทธิ์ (reassigning privilege) ในทางตรงกันข้ามนี้เองฐานข้อมูลที่ถูกจัดระบบบนเอ็มแอลเอสในหลายๆการเปลี่ยนแปลงนี้เองจะทำโดยการเปลี่ยนแอสเสซเลเบลของผู้ใช้งานหรือแอสเสซเลเบลของเนื้อหานั่นเอง กล่าวง่ายๆว่า การบริหารจัดการของการเปลี่ยนแปลงที่เป็นไดนามิก (dynamic) ทั้งผู้ใช้งานและเนื้อหาสามารถทำให้ง่ายขึ้นอย่างมากโดยวิธีการเอ็มแอลเอสนั่นเอง

สิ่งที่คาดหวังในโมเดลของเอ็มแอลเอส นี้คือ ข้อเท็จจริงที่ถูกสร้างโดยใช้ในอาร์ดีบีเอ็มเอส (RDBMS) ทั่วไปๆ เช่น Oracle, MS SQLServer หรือ IBM DB2 จะถูกใช้ประโยชน์ในส่วนทริกเกอร์ (trigger) ซึ่งเป็นตัวดักเหตุการณ์แล้วเมื่อเหตุการณ์นั้นก็จะไปทำโพรซีเดอร์ (procedure) ที่ตั้งไว้ (trigger and stored procedure) โมเดลของเอ็มแอลเอส จะให้ platform สำหรับการสร้างตอบโจทย์ของธุรกิจเชิงอิเล็กทรอนิกส์ซีอาร์เอ็มไปยังผู้ผลิตซอร์ฟแวร์ และในการพัฒนาใช้เองในบริษัท โมเดลของเอ็มแอลเอส จะให้การตอบโจทย์ของซีอาร์เอ็มขึ้นอยู่กับการสร้างเอ็มแอลเอส ใน 1 อาร์ดีบีเอ็มเอสเท่านั้นซึ่ง เหมาะสมอย่างสมบรูณ์ และส่งไปยังเครื่องมือ (equivalent) ของการสร้างเอ็มแอลเอสบนความแตกต่างของอาร์ดีบีเอ็มเอส

บนหน้าต่างของเว็บเพจดังรูปที่ 4.3 จะใช้โมเดลของเอ็มแอลเอสจะเกี่ยวข้องกับ non-technical ของลูกค้า และลูกค้าอาจจะปรากฏความซับซ้อนได้ และเกี่ยวข้องกับระบบแอสเสจเลเบล อย่างไรก็ตามเป็นงานที่ก้าวไปรวมเอาฟังก์ชันของการจัดการแอสเสจเลเบล เข้าไปยัง front-end application ซึ่งเป็นประโยชน์ของฐานข้อมูล ที่ใช้ฐานข้อมูลรีเลชั่นนอลจะถูกทำให้สำเร็จ อย่างไรก็ตามเป็น fact ที่ผู้ใช้งานต้องการ ใช้แสดงด้วยความโปร่งใสในข้อมูล

เว็บแอพพลิเคชั่นในรูปที่ 4.3 นี้ขึ้นอยู่กับตารางที่ 4.4 เป็นลอจิกพื้นฐานสำหรับการตีความแอสเสจเลเบล



รูป 4.3 เว็บแอพพลิเคชั่นของร้านหนังสือ Andes

บทที่ 5
โครงสร้างของฐานข้อมูลและภาษาที่ใช้

5.1 โครงสร้างของฐานข้อมูล

จากที่ได้กล่าวถึงโมเดลต่างๆของระบบฐานข้อมูลที่มีความปลอดภัยหลายระดับมาแล้วหลายโมเดล โมเดลที่ใช้เป็นพื้นฐานในการพัฒนาต่อคือโมเดลของจูดิก โดยได้นำหลักการและแนวคิดต่างๆมาสร้างเป็นโครงสร้างของฐานข้อมูลได้ดังนี้

$$R(A_1,C_1,\ldots,A_n,C_n,TC) \tag{5.1}$$

โดยที่ A_i เป็นแอททริบิวข้อมูล (Data Attribute) ที่เก็บค่าของข้อมูล
C_i เป็นแอททริบิวที่เก็บค่าระดับความปลอดภัยของข้อมูล
TC เป็นค่าระดับความปลอดภัยในระดับแถว

จากที่กล่าวมาเป็นโครงสร้างที่มีอยู่แล้วในโมเดลของจูดิก แต่โครงสร้างของโมเดลนี้ไม่ได้มีการรองรับหากผู้ใช้งานระดับล่างได้ทำการลบข้อมูลที่ตนเชื่อออกไป โดยที่ข้อมูลนั้นผู้ใช้งานในระดับที่สูงกว่าก็เชื่ออยู่ด้วยเช่นกัน ดังนั้นเมื่อเกิดเหตุการณ์นี้ขึ้น ทางผู้ใช้งานระดับบนควรที่จะมีทางเลือกเกิดขึ้นว่าสมควรที่จะเชื่อต่อหรือไม่ จึงควรที่จะมีส่วนกำกับไว้ว่าข้อมูลแถวใดควรจะมีการพิจารณาอีกครั้ง โดยการเพิ่มแอททริบิวในการช่วยจดจำว่าข้อมูลใดต้องได้รับการพิจารณาอีกครั้ง โครงสร้างฐานข้อมูลใหม่ที่ได้คิดขึ้นจึงเป็นดังนี้

$$R(A_1,C_1,\ldots,A_n,C_n,TC,F\_DEL) \tag{5.2}$$

โดยที่ F_DEL จะเป็นการเก็บข้อมูลของผู้ใช้งานที่ยังไม่ได้พิจารณาหลังจากที่ได้มีการลบข้อมูลที่ตนเชื่อถือจากผู้ใช้งานระดับต่ำกว่าออกไป

โครงสร้างข้อมูลที่ได้กล่าวมา เมื่อนำมาใช้งานในดีบีเอ็มเอสจริงๆแล้วจะต้องมีการกำหนดชนิดและคีย์ โดยที่โมเดลจูดิก ได้กล่าวถึงคีย์ไว้เรียกว่า คีย์หลักแอพพาเรนท์ ซึ่งคือการรวมกลุ่มของคีย์หลักซึ่งจะคล้ายกับฐานข้อมูลทั่วไป แต่จะต่างกันตรงที่จะมีการรวมระดับของข้อมูลไว้ด้วยเช่นกัน บางครั้งจึงมีการเรียกว่า KC (Keys Classification) แต่ถ้าหากมาใช้งานจริงแล้วอาจจะมีเหตุการณ์ตามตาราง 5.1 เกิดขึ้นได้ จึงต้องทำการเพิ่มแอททริบิว TC เข้าไปเป็นคีย์อีกตัวหนึ่ง

ตาราง 5.1 การซ้ำกันของคีย์

Patient Name	Diagnosis	Age	RoomNo	TC
Alan Jones UCS	Exhaustion UCS	56 UCS	101 UCS	UCS
Diva Megastar UCS	Exhaustion U-CS	32 UC-S	201 UCS	U-CS
Diva Megastar UCS	Intoxication CS	32 UC-S	201 UCS	C-S
Diva Megastar UCS	Intoxication CS	42 S	201 UCS	S

ส่วนชนิดของข้อมูล (type of data) ที่จะใช้ในการสร้างตารางลงบนดีบีเอ็มเอสจริงๆนั้นจะต้องขึ้นกับว่าแอททริบิวข้อมูลมีชนิดของข้อมูลเป็นชนิดใด ส่วนระดับความปลอดภัยนั้นควรที่จะใช้การเก็บข้อมูลเป็น string

5.2 มุมมองการเห็นข้อมูลของผู้ใช้งานระบบ

หลังจากที่ได้โครงสร้างของฐานข้อมูลที่ใช้งานได้จริงแล้ว หากผู้ใช้งานที่มีระดับความปลอดภัยที่แตกต่างกันย่อมที่จะเห็นลักษณะของข้อมูลที่ไม่เหมือนกัน เช่น จากข้อมูลเกี่ยวกับการส่งสินค้าโดยพนักงาน ผู้ใช้งานระดับ S นั้นจะเห็นข้อมูลทั้งหมดรวมทั้ง ค่าระดับความปลอดภัยของข้อมูลด้วยว่า ผู้ใช้ในระดับที่ต่ำกว่ามีความเชื่ออย่างไรตามตาราง 5.2

ตาราง 5.2 มุมมองที่ผู้ใช้งานระดับ S เห็นจากข้อมูลการส่งสินค้า

MSG_NAME	MSG_C	PRODUCT	PRO_C	DESTINATION	DES_C
John	C	rubber	C	Bangna	C
Ronnie	UCS	metal	UCS	Minburi	UCS
Sam	U-C	gold	U-C	Silom	U-C

ผู้ใช้งานระดับ C จะเห็นข้อมูล และระดับความปลอดภัยของข้อมูลของตัวเองและผู้ใช้งานท่านอื่นที่มีระดับความปลอดภัยที่ต่ำกว่าตัวเอง โดยที่ผู้ใช้งานระดับ C นั้นจะไม่อาจทราบได้เลยว่าผู้ใช้งานที่มีระดับความน่าเชื่อถือมากกว่าตนเองนั้นจะมีการเชื่อถือข้อมูลนั้นอย่างไร ตามตาราง 5.3 ความแตกต่างจากมุมมองของผู้ใช้งานระดับ S นั้นสังเกตได้จากแอททริบิวระดับความปลอดภัยของข้อมูลที่จะไม่เห็นระดับความปลอดภัยของผู้ใช้งานที่มีระดับที่สูงกว่าเลย

ตาราง 5.3 มุมมองที่ผู้ใช้งานระดับ C เห็นจากข้อมูลการส่งสินค้า

MSG_NAME	MSG_C	PRODUCT	PRO_C	DESTINATION	DES_C
John	C	rubber	C	Bangna	C
Ronnie	UC	metal	UC	Minburi	UC
Sam	U-C	gold	U-C	Silom	U-C

ผู้ใช้งานระดับ U จะเห็นเพียงแค่ข้อมูลของตนเองเท่านั้น ไม่สามารถเห็นค่าระดับความปลอดภัยของข้อมูลได้ เนื่องจากว่าการทำงานของผู้ใช้ระดับ U นั้นจะเสมือนว่าอยู่ในโลกของระดับเดียวเท่านั้น จึงไม่จำเป็นที่จะต้องรู้ว่ามีผู้ใช้งานในระดับที่สูงกว่า ผู้ใช้งานระดับ U จะเห็นข้อมูลตามตาราง 5.4

ตาราง 5.4 มุมมองที่ผู้ใช้งานระดับ C เห็นจากข้อมูลการส่งสินค้า

MSG_NAME	PRODUCT	DESTINATION
Ronnie	metal	Minburi
Sam	gold	Silom

แต่อย่างไรก็ตามการเห็นข้อมูลของผู้ใช้งานที่ขาดความรู้ความเข้าใจเกี่ยวกับความปลอดภัยหลายระดับอาจจะไม่มีความต้องการที่จะเห็นข้อมูลเกี่ยวกับระดับความปลอดภัยแต่อย่างใดจึงอาจจะมีการใช้คำสั่งเพื่อเรียกดูข้อมูลในมุมมองที่ต่างกันได้ในระดับผู้ใช้งานเดียวกัน เช่น ผู้ใช้งานระดับ C บางคนอาจจะต้องการเห็นข้อมูลระดับความปลอดภัยด้วย แต่บางคนอาจจะไม่ต้องการเห็น จึงต้องมีการสร้างมุมมองที่แตกต่างกัน 2 แบบ คือ มุมมองแบบที่เห็นระดับความปลอดภัย กับมุมมองที่ไม่เห็นระดับความปลอดภัย ยกตัวอย่างจากตารางข้อมูลการส่งสินค้าดังที่ได้กล่าวมาแล้ว จะได้แนวการสร้างมุมมองเป็นดังนี้

1) มุมมองที่เห็นระดับความปลอดภัย

```
CREATE VIEW
VC_SHIPPING(XMSG_NAME, LB1, XPRODUCT, LB2, XDESTINATION, LB3) AS
SELECT MSG_NAME, MSG_C, PRODUCT, PRO_C, DESTINATION, DES_C
FROM SHIPPING
```

2) มุมมองแบบที่ไม่เห็นระดับความปลอดภัย

```
CREATE VIEW
V_SHIPPING(XMSG_NAME, XPRODUCT, XDESTINATION) AS
SELECT MSG_NAME, PRODUCT, DESTINATION
FROM SHIPPING
```

5.3 การตรวจสอบระดับความปลอดภัย

จะต้องมีการตรวจสอบระดับความปลอดภัยของผู้ใช้งานเมื่อผู้ใช้ได้เข้ามาในระบบ เพื่อที่จะทราบถึงค่าความปลอดภัย และนำไปใช้ได้อย่างถูกต้อง ต้องมีตารางที่ไว้ตรวจสอบว่าผู้ใช้งานท่านนี้มีระดับความปลอดภัยอะไร

ตาราง 5.5 การเก็บข้อมูลผู้ใช้งานกับค่าความปลอดภัย

USERNAME	USER_LEVEL
Sam	U
Ronnie	C
John	S

5.4 การเปลี่ยนแปลงข้อมูล

การเปลี่ยนแปลงข้อมูลจะมีการใช้คำสั่งที่เกี่ยวข้องได้แก่ Insert, Delete, Update ซึ่งเป็น 3 คำสั่งหลัก และยังมีคำสั่ง Verify เพื่อให้ผู้ใช้งานได้เลือกที่จะเชื่อข้อมูลของผู้ใช้ในระดับที่ต่ำกว่า

5.4.1 คำสั่งอินเสิร์ต (Insert)

การสร้างคำสั่งอินเสิร์ต จะต้องรับค่าคำสั่งของผู้ใช้งานเข้ามาก่อน แล้วทำการแปลงคำสั่งให้สัมพันธ์กับตารางจริงๆ เนื่องจากว่าเวลาที่ผู้ใช้งานทำการใช้คำสั่งอินเสิร์ตนั้นผู้ใช้จะไม่ทราบถึงการมีอยู่ของระดับความปลอดภัยของข้อมูลว่าจะต้องใช้การส่งค่าอะไรลงไป จะต้องทำการแปลงคำสั่งเหล่านั้นเสียก่อน เช่น ผู้ใช้งานระดับ C ต้องการเพิ่มข้อมูล

```
INSERT INTO SHIPPING
VALUES (“Max”, “aluminum”, “Bangkapi”) ;
```

คำสั่งอันนี้เราจะต้องทำการเพิ่มค่าของระดับข้อมูลต่างๆเข้าไป เพื่อให้สัมพันธ์กับตารางจริง

```
Procedure sinsert {
    get user security label ;
    transform old command to new command ;
    return the new command ;
}
```

ซึ่งคำสั่งที่ได้ออกมานั้นจะมีลักษณะดังนี้

```
INSERT INTO SHIPPING
VALUES (“Max”, 6, “aluminum”, 6, “Bangkapi”, 6, 6, 0) ;
```

5.4.2 คำสั่งเวอริฟาย (Verify)

เป็นคำสั่งที่มีไว้เปลี่ยนแปลงความเชื่อของข้อมูล โดยผู้ใช้งานท่านนั้นๆ การจะทำการเวอริฟาย นั้นจะมีเกิดขึ้นได้ 2 แบบ นั่นคือ ผู้ใช้งานเรียกใช้งานด้วยตนเอง และเกิดจากคำสั่งอื่น เช่น ดีลีท (delete) แล้วจะต้องมีการเวอริฟายตามมา เมื่อผู้ใช้งานออกคำสั่ง

VERIFY (TRUE | FALSE) [RELATION]

WHERE [CONDITION]

ยกตัวอย่างเช่นจากตาราง 5.2 เมื่อผู้ใช้งานระดับ S เรียกใช้งานคำสั่ง

VERIFY TRUE SHIPPING

WHERE MSG_NAME = "John"

ผลที่ได้จะเป็นดังตาราง 5.6

ตาราง 5.6 เมื่อผู้ใช้งานระดับ S ใช้คำสั่งเวอริฟาย

MSG_NAME	MSG_C	PRODUCT	PRO_C	DESTINATION	DES_C
John	CS	rubber	CS	Bangna	CS
Ronnie	UCS	metal	UCS	Minburi	UCS
Sam	U-C	gold	U-C	Silom	U-C

หลักการทำงานของคำสั่งนี้มีดังนี้

```
Procedure verify_table (argument) {
    get user security label;
    select the tuple;
    verify_tuple;
    change label for that tuple;
}
```

5.4.3 คำสั่งดีลีท (DELETE)

เป็นคำสั่งที่ใช้ลบข้อมูลที่ไม่ต้องการใช้อีก โดยผู้ที่จะลบได้นั้นต้องเป็นผู้ที่เพิ่มข้อมูลเข้าไปโดยสังเกตได้จากเลเบลตัวแรกทางซ้ายมือ ซึ่งหากมีผู้ใช้งานในระดับที่เหนือกว่าผู้ที่เพิ่มข้อมูลเข้าไปนั้นเชื่อข้อมูลนี้อยู่ การลบข้อมูลนั้นจริงๆจะไม่สามารถทำได้จึงต้องมีการจดจำไว้ว่าข้อมูลนี้ได้ถูกลบแล้ว และต้องให้ผู้ใช้งานระดับอื่นๆ เข้ามาเลือกว่าจะเชื่อหรือไม่เชื่อก็ได้ เนื่องจากว่าผู้ที่เป็นคนให้ข้อมูลเลือกที่จะไม่เชื่อไปแล้ว จากเหตุการณ์ดังกล่าวจึงทำให้เกิดการเวอริฟาย เมื่อมีการใช้คำสั่งดีลีทขึ้น ตัวอย่างของการลบ ถ้าผู้ใช้งานระดับ U ทำการลบข้อมูลจากตาราง 5.2 ด้วยคำสั่ง

DELETE FROM SHIPPING

WHERE MSG_NAME = “Ronnie”

จะได้รูปร่างฐานข้อมูลตามตาราง 5.7

ตาราง 5.7 เมื่อผู้ใช้งานระดับ U ลบข้อมูลออก

MSG_NAME	MSG_C	PRODUCT	PRO_C	DESTINATION	DES_C	F_DEL
John	C	rubber	C	Bangna	C	
Ronnie	CS	metal	CS	Minburi	CS	CS
Sam	U-C	gold	U-C	Silom	U-C	

จะสังเกตเห็นว่า F_DEL จะทำการจดจำไว้ว่าข้อมูลอันนี้ได้ถูกลบแล้วแต่ยังไม่ได้ถูกเวอริฟาย โดยผู้ใช้งานระดับอื่น ส่วนผู้ใช้งานระดับ U เองจะเห็นข้อมูลตามตาราง 5.7 เสมือนกับว่าข้อมูลนั้นได้ถูกลบไปแล้ว

ตาราง 5.8 ข้อมูลที่ผู้ใช้งาน U เห็นหลังจากลบข้อมูลแล้ว

MSG_NAME	PRODUCT	DESTINATION
Sam	gold	Silom

5.4.4 คำสั่งอัพเดท (UPDATE)

เป็นคำสั่งที่ใช้ในการเปลี่ยนแปลงค่าของข้อมูลแต่ต้องไม่ผิดหลักเกณฑ์ของเอ็มแอลเอสที่ได้มีมา เนื่องจากว่ามีกฎข้อบังคับต่างๆที่เกี่ยวข้องเพื่อกันความซ้ำซ้อนของข้อมูล แบ่งการอัพเดทออกเป็น 2 แบบ ได้แก่

1) การอัพเดทกับข้อมูลที่ระดับเดียวกัน ยกตัวอย่างเช่น ข้อมูลจากตาราง 5.8 เมื่อผู้ใช้งานใช้คำสั่ง

UPDATE SHIPPING

SET PRODUCT = “wheat”

WHERE PRODUCT = “rubber”

ตาราง 5.9 ข้อมูลก่อนถูกอัพเดท

MSG_NAME	MSG_C	PRODUCT	PRO_C	DESTINATION	DES_C	TC
John	UC	rubber	UC	Bangna	C	C
John	UC	gold	UC	Silom	U-C	U-C

เนื่องจากว่าข้อมูลที่ต้องการจะเปลี่ยนแปลงไม่ได้เกี่ยวข้องกับคีย์ และผู้ใช้งานระดับ C ไม่ต้องการที่จะเชื่อถือข้อมูลเก่าอีกต่อไป และผู้ใช้งานระดับ C มีสิทธิ์เข้าถึงข้อมูลได้ จึงทำให้เกิดการเปลี่ยนแปลงของข้อมูลเกิดขึ้นดังตาราง 5.9

ตาราง 5.10 ข้อมูลหลังถูกอัพเดทโดยผู้ใช้งานระดับเดียวกับข้อมูล

MSG_NAME	MSG_C	PRODUCT	PRO_C	DESTINATION	DES_C	TC
John	UC	wheat	C	Bangna	C	C
John	UC	gold	UC	Silom	U-C	U-C

2) การอัพเดทข้อมูลที่มีระดับความปลอดภัยที่ต่ำกว่า ยกตัวอย่างจากตาราง 5.8 เมื่อผู้ใช้งานระดับ S ต้องที่จะแก้ไขข้อมูลตามคำสั่งต่อไปนี้

```
UPDATE   SHIPPING
SET      PRODUCT = "metal"
WHERE    MSG_NAME = "John" and      DESTINATION = "Bangna"
```

เนื่องจากข้อมูลที่ผู้ใช้งานต้องการที่จะแก้ไขนั้นมีการเกี่ยวข้องกับคีย์ ทั้งผู้ใช้งานยังไม่มีสิทธิ์การใช้งานที่แท้จริงกับข้อมูลที่ต้องการจะแก้ไขอีกด้วย จึงจำเป็นที่จะต้องทำการเพิ่มข้อมูลขึ้นอีก 1 ชุดข้อมูล เพื่อไม่ให้เกิดการละเมิดข้อบังคับต่างๆของเอ็มแอลเอสพร้อมทั้งทำการเวอริฟายข้อมูลที่เกี่ยวข้องเพื่อแก้ไขข้อมูลความเชื่อ จะทำให้ได้ข้อมูลตามตาราง 5.11

ตาราง 5.11 ข้อมูลหลังถูกอัพเดทโดยผู้ใช้งานระดับสูงกว่า

MSG_NAME	MSG_C	PRODUCT	PRO_C	DESTINATION	DES_C	TC
John	UC	rubber	UC	Bangna	C	C-S
John	UC	gold	UC	Silom	U-C	U-C
John	UC	metal	S	Bangna	CS	S

บทที่ 6

การทดลองและผลการทดลอง

6.1 ภาษา C#

ภาษา C# เป็นภาษาใหม่ที่ทางไมโครซอฟท์ได้พัฒนาขึ้นมาพร้อมกับโปรแกรมชุด Visual Studio.NET เพื่อต้องการสร้างมาตรฐานใหม่ในการพัฒนาโปรแกรม โดยกำหนดให้เป็นหลักการของการสร้างเครื่องคอมพิวเตอร์เสมือนบนฮาร์ดแวร์เพียงชุดเดียวซึ่งมันสามารถทำงานได้เสมือนกับว่ามีคอมพิวเตอร์หลายๆเครื่อง โดยมีการแยกการทำงานของระบบต่างๆได้อย่างเป็นอิสระต่อกัน หรือที่เรียกกันว่า Virtual Machine นั่นคือ .Net Framework ซึ่งภาษาอะไรก็ตามที่จะใช้บน .Net Framework ก็จะคอมไพล์ให้เป็น IL(Immediate Language ภาษาของ .Net) เก็บไว้เป็นไฟล์ exe และเมื่อรันไฟล์ exe ตัวนั้น มันก็จะคอมไพล์ด้วย .Net Framework ให้กลายเป็นโปรแกรมจริง

วัตถุประสงค์หลักของการสร้างภาษา C# ขึ้นมาก็คือ เป็นภาษาใหม่ที่มีประสิทธิภาพการทำงานเทียบเท่าหรือเหนือกว่า C++ แต่ในขณะเดียวกันก็ไม่ต้องยุ่งยากและซับซ้อนเท่า C++ โดยสามารถใช้งานง่ายๆ เหมือนกับภาษายอดนิยมอย่าง Visual Basic และสามารถขยายขีดความสามารถของ Visual Basic ได้ให้สามารถพัฒนาแอพพลิเคชั่นได้ดียิ่งขึ้น โดยเฉพาะการเขียนรับรองเหตุการณ์ด้วยเหตุผลเหล่านั้น C# จึงได้รับการพัฒนามาจาก C++ โดยลดความซับซ้อนของตัวภาษาและปรับปรุงข้อบกพร่องต่างๆที่มีใน C# ให้หมดไปพร้อมกันนี้ก็ได้นำลักษณะความเรียบง่ายของ Visual Basic มาผสมผสานกันเข้าไปบวกกับความสามารถใหม่ๆที่เพิ่มเติมขึ้นมาอีกมากมาย ทำให้ C# นั้นเป็นภาษาที่มีความลงตัวมากที่สุดเมื่อเทียบกับภาษาอื่นๆ

ภาษา C# เน้นแนวความคิดของการเขียนโปรแกรมแบบโมเดิร์นโอโอพี (OOP) เกิดจากการที่ไมโครซอฟท์ พัฒนาคลาสต้นแบบต่างๆขึ้นมา ที่เรียกว่า Base Class library แล้วนำมาจัดหมวดหมู่ให้เป็นระเบียบ เมื่อต้องการเรียกใช้งานคลาสใดก็จะอาศัยระบบเนมสเปซ (Namespaces System) เข้ามาช่วยในการระบุคลาสต้นแบบต่างๆเพื่อให้ผู้พัฒนาสามารถนำออบเจ็กต์ต่างๆที่อยู่ในคลาสนั้นๆออกมาใช้งานได้อย่างง่ายดาย

6.2 วินโดร์เซอร์วิส

วินโดร์เซอร์วิส (Window Service) เป็นแอพพลิเคชั่นที่ทำงานได้เองโดยอัตโนมัติเมื่อระบบปฏิบัติการเริ่มบูทขึ้น ซึ่งสามารถรันได้เองโดยที่ไม่ต้องมีการล็อกอินของผู้ใช้งานเข้าไปในระบบ สามารถเลือกทำการติดตั้งวินโดร์เซอร์วิสให้รันเป็นแบบ user account หรือ system account ก็ได้ การทำงานของวินโดร์เซอร์วิสนั้นจะเป็นการทำงานแบบหลังฉาก คือ จะคอยควบคุมการทำงาน

ของระบบตามที่ได้วางเงื่อนไขการทำงานต่าง ๆ ไว้ ทำให้เสมือนแอพพลิเคชั่นนั้นมีชีวิต สามารถทำงานได้ด้วยตัวเองอยู่บนวินโดร์ แต่เป็นไปตามเงื่อนไขที่เราได้กำหนดการทำงานไว้

การจัดการและควบคุมดูแลวินโดร์เซอร์วิสสามารถทำได้ด้วย services หาได้จากการคลิก Programs → Administrative Tools หรือพิมพ์ Services.msc ใน run command ที่ start menu ภายในหน้าต่างคอนโซล services จะประกอบไปด้วยชื่อเซอร์วิส คำอธิบายฟังก์ชันเซอร์วิส สถานะเซอร์วิส ชนิดของการเริ่มต้นใช้งานเซอร์วิส นอกเหนือจากนี้ยังประกอบไปด้วยคุณสมบัติอื่นๆดังนี้

1) Start, Stop, Pause หรือ Restart เซอร์วิส
2) พารามิเตอร์เซอร์วิส
3) เปลี่ยนชนิดของการเริ่มต้นใช้งานเซอร์วิส ซึ่งประกอบไปด้วย automatic manual และ disable
   3.1) Automatic เริ่มใช้งานเซอร์วิสที่เวลาเริ่มทำงานระบบ
   3.2) Manual เริ่มใช้งานเซอร์วิสตามความต้องการหรือเมื่อถูกเรียกจากแอพพลิเคชั่น
   3.3) Disable ไม่สามารถเริ่มใช้งานเซอร์วิส
4) Automatic (Delay) เป็นชนิดของการเริ่มต้นใช้งานเซอร์วิสใหม่ที่ถูกสร้างขึ้นครั้งแรกใน window vista
   ซึ่งเป็นการเริ่มต้นการใช้งานเซอร์วิสในระยะเวลาอันสั้นหลังจากที่ระบบเสร็จสิ้นการบูทและเริ่มทำโอเปอร์เรชั่น ดังนั้นระบบ boots up จึงเร็วกว่าแบบเดิม
5) เปลี่ยน account ภายใต้การล็อกอินวินโดร์เซอร์วิส
6) ติดตั้งทางเลือกในการกู้คืนเมื่อเซอร์วิสเกิดการล้มเหลว
7) ส่งลิสของเซอร์วิสเป็นแท็กไฟล์ (text file) หรือซีเอสวีไฟล์ (CSV file)
   ใน Windows Vista และ Windows 7 นอกเหนือจากจะมีหน้าคอนโซล Services management แล้ว

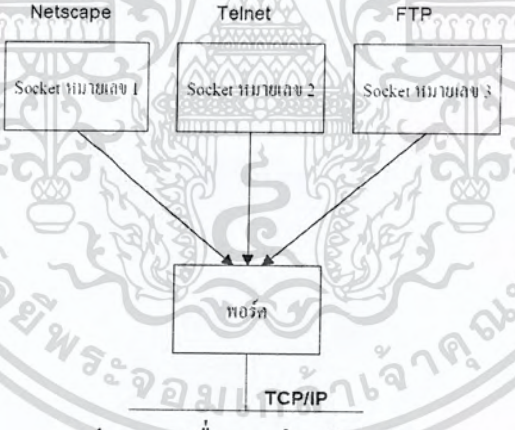
วินโดร์เซอร์วิสถูกสร้างเป็นเครื่องมือสำหรับพัฒนา เช่น Microsoft Visual Studio หรือ Embarcadero Delphi วินโดร์จะทำการเรียก Service Control Manager เพื่อสำหรับจัดการ start และ stop เซอร์วิส แอพพลิเคชั่นต้องการใช้เซอร์วิสสำหรับการเขียน handle เพื่อ start , stop และ pause แมสแสจจาก Service Control Manager

หลังจากนั้น API จะทำการเรียกชื่อของเซอร์วิสและแอททริบิวอื่นๆ ต่อไป โดยทั่วไปแล้ววินโดร์เซอร์วิสจะไม่มียูเซอร์อินเตอร์เฟสแต่นักพัฒนาก็สามารถทำการเพิ่มฟอร์มและส่วนประกอบของยูเซอร์อินเตอร์เข้าไปได้ ซึ่งจะทำให้ผู้ใช้งานสะดวกและง่ายต่อการใช้งานวินโดร์เซอร์วิสมากยิ่งขึ้น

6.3 วินซ็อก

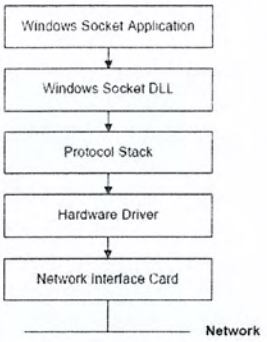
การเขียนโปรแกรมเพื่อติดต่อสื่อสารกันบนเครือข่าย เป็นการเรียกใช้ API (Application Programming Interface) จาก Winsock (Windows Socket Programming) ขึ้นมาใช้งาน ซึ่งจะเป็นการเรียกใช้จากไฟล์ WINSOCK.DLL หรือ WSOCK32.DLL ในไฟล์เหล่านี้จะมีฟังก์ชันต่างๆ ที่เป็นมาตรฐานในการติดต่อสื่อสารข้อมูลกันอย่างครบถ้วน และได้มีการกำหนดไว้ที่ไฟล์ winsock.h หรือ winsock2.h ดังนั้นก่อนเรียกใช้ฟังก์ชัน Winsock จะต้องมีการกำหนด #include <winsock.h> หรือ #include <winsock2.h> ก่อนจึงจะสามารถเรียกใช้งานได้

วินซ็อกเป็นการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์บนเครือข่าย เช่น เทลเน็ตเข้าไปในระบบยูนิกส์ ระบบจะต้องสามารถรองรับการทำงานแบบมัลติยูสเซอร์ได้ นั่นหมายความว่า ระบบจะต้องสามารถมีการติดต่อสื่อสารกับยูสเซอร์ได้พร้อมๆ กันหรือเมื่อมีการใช้งานโปรแกรมที่เกี่ยวกับอินเตอร์เน็ต เช่น เน็ตสเคป เทลเน็ต เอฟทีพี ซึ่งโปรแกรมต่างๆเหล่านี้จะต้องทำงานแยกกันโดยอิสระ ดังนั้นจึงได้เกิดวินซ็อกขึ้นมา นั่นก็คือ วินซ็อกสามารถที่จะสร้างช่องทางสื่อสารขึ้นมาได้หลายๆช่องทาง (ขึ้นอยู่กับเวอร์ชันของวินซอค์ก) และแต่ละช่องทางสื่อสารสามารถที่จะส่งข้อมูลได้โดยไม่ขึ้นกับช่องทางสื่อสารอื่นๆ ดังรูปที่ 6.2



รูป 6.1 การสื่อสารผ่านซ็อกเก็ต

วินซ็อกทำหน้าที่เป็นตัวกลางในการรับส่งข้อมูลจากแอพพลิเคชั่นกับทีซีพีไอพี จากนั้นทีซีพีไอพีจึงส่งข้อมูลลงไปบนเครือข่ายดังรูปที่ 6.2



รูป 6.2 ระดับของวินซ็อก

และการสร้างไคลเอนต์/เซิฟเวอร์โดยใช้ฟังก์ชันของวินซ็อก อธิบายได้ดังรูปที่ 6.3

รูป 6.3 การสร้าง ไคลเอนต์/เซิฟเวอร์ โดยวินซ็อก

การทำงานของวินซ็อกแบ่งออกเป็น 2 ส่วนคือ

1) การทำงานของเซิฟเวอร์ เริ่มต้นการทำงานของวินซ็อกโดยใช้ฟังก์ชัน WSAStartup() จากนั้นทำการสร้างซ็อกเก็ต (Socket) ขึ้นมาโดยใช้ฟังก์ชัน socket() เชื่อมต่อ Address เข้ากับ socket ที่สร้างขึ้นมาโดยใช้ฟังก์ชัน bind() และรอการเชื่อมต่อจากไคลเอนต์โดยใช้ฟังก์ชัน listen
2) การทำงานของไคลเอนต์ เริ่มต้นการทำงานของวินซ็อกโดยใช้ฟังก์ชัน WSAStartup() จากนั้นทำการสร้างซ็อกเก็ตขึ้นมา โดยใช้ฟังก์ชัน socket() และเชื่อมต่อไปยังเซิฟเวอร์โดยใช้ฟังก์ชัน connect() หากมีไคลเอนต์ติดต่อเข้ามาเซิฟเวอร์จะรับรู้และตอบรับกลับไปที่ไคลเอนต์และสร้างซ็อกเก็ตขึ้นมาใหม่เพื่อใช้ในการสื่อสารกับไคลเอนต์นั้นๆ ที่ติดต่อเข้า

มาโดยใช้ฟังก์ชัน accept() เมื่อถึงขั้นตอนนี้ หมายความว่า ไคลเอนต์และเซิฟเวอร์สามารถที่จะสื่อสารกันได้แล้ว โดยใช้ฟังก์ชัน send() หรือ recv

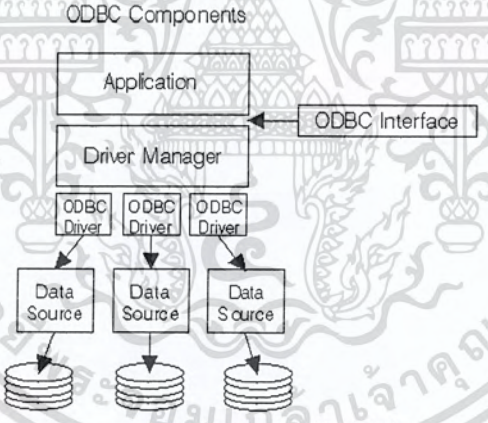
6.4 โอดีบีซี

Open Database Connectivity หรือ ODBC เป็นเทคโนโลยีของวินโดร์ที่ช่วยให้แอพพลิเคชั่นสามารถอ่านข้อมูลจากเซิร์ฟเวอร์ โดยฝั่งแอพพลิเคชั่นไม่จำเป็นต้องทราบเลยว่า ฐานข้อมูลบนเซิร์ฟเวอร์เป็นฐานข้อมูลของผลิตภัณฑ์ใด เพียงแต่ส่งเอสคิวแอลผ่านโอดีบีซี จากนั้นโอดีบีซีจะช่วยแปลงคำสั่งเอสคิวแอลให้ติดต่อกับฐานข้อมูลแบบต่างๆให้เอง สามารถจึงใช้คำสั่งเหมือนเดิมกับฐานข้อมูลต่างชนิดกันได้เหมือนกัน

โครงสร้างของโอดีบีซีประกอบไปด้วย 3 ส่วน คือ

1) ODBC Data Source
2) Driver Manager
3) Driver

โดยมีโครงสร้างดังนี้



รูป 6.4 โครงสร้างโอดีซีบี

Application --> ODBC Driver Manager --> ODBC Driver ---> ODBC Data Source--> Database

ข้อดีของโอดีบีซีมีดังนี้

1) สามารถสลับฐานข้อมูลได้ง่ายโดยเปลี่ยนแค่ตัวเชื่อมต่อหรือที่เรียกกันว่า Connection String
2) อัพเดตตัวฐานข้อมูลง่าย เพียงแค่อัพเดตข้อมูลในฐานข้อมูล เสร็จแล้วก็ตามไปอัพเดต ODBC Driver ก็จะเสร็จสมบูรณ์
3) ODBC Driver ส่วนใหญ่ฟรีไม่ต้องเสียค่าใช้จ่าย

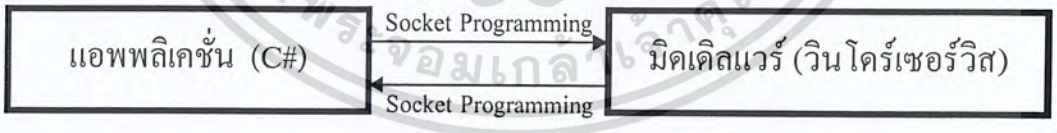
ข้อเสียของโอดีบีซีมีดังนี้

1) ช้ากว่าการเชื่อมต่อแบบ Native
2) ต้องเอา ODBC Driver ตัวที่ต้องการใช้ไปลงที่เครื่อง Client ด้วย

6.5 แนวคิดในการพัฒนาโปรแกรม

จากทฤษฎีและแนวคิดที่ได้อธิบายไปแล้วข้างต้น จะเห็นความสามารถและประโยชน์ของภาษา C# และวินโดร์เซอร์วิส ด้วยเหตุผลดังกล่าวจึงได้นำมาใช้ในการพัฒนาแอพพลิเคชั่น (Application) และมิดเดิลแวร์ (Middleware) ซึ่งข้อดีของภาษา C# ที่จะนำมาใช้ในการสร้างและพัฒนาแอพพลิเคชั่น คือ เป็นภาษาที่มีประสิทธิภาพการทำงาน สามารถใช้งานง่ายๆ และมีความสามารถใหม่ๆที่เพิ่มเติมขึ้นมาอีกมากมายเมื่อเทียบกับภาษาอื่นๆ ส่วนวินโดร์เซอร์วิสก็มีความแตกต่างจากแอพพลิเคชั่นธรรมดาทั่วไป โดยที่มีจะมีนามสกุลเป็น .cs ซึ่งปกติแล้วแอพพลิเคชั่นที่พัฒนาขึ้นทั่วไปจะมีนามสกุลเป็น .exe เมื่อต้องการที่จะใช้งาน ผู้ใช้สามารถเข้าถึงตัวเริ่มต้นโปรแกรมได้โดยตรง แต่วินโดร์เซอร์วิสจะแตกต่างออกไปนั้นคือจะไม่สามารถเข้าถึงและใช้งานได้โดยตรงตามเงื่อนไขที่เราได้กำหนดการทำงานไว้ เหตุผลดังกล่าววินโดร์เซอร์วิสจึงเหมาะแก่การนำมาสร้างและพัฒนาเป็นมิดเดิลแวร์ เพราะต้องการที่จะหลีกเลี่ยงไม่ให้ผู้ใช้งานทำการเข้าถึงส่วนที่เราใช้ติดต่อโดยตนเอง เนื่องจากเราได้กำหนดให้แอพพลิเคชั่นเป็นตัวที่สั่งให้วินโดร์เซอร์วิสทำงานแทนผู้ใช้งานอยู่แล้วนั่นเอง

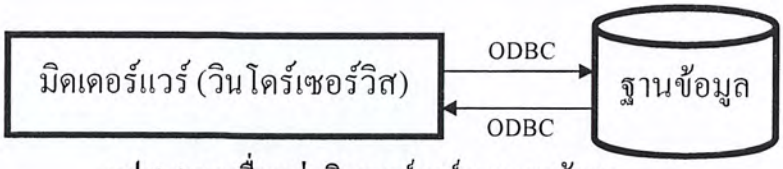
เมื่อพัฒนาแอพพลิเคชั่นและวินโดร์เซอร์วิสเสร็จแล้ว ต่อมาก็คือ ส่วนการติดต่อสื่อสารและรับส่งข้อมูลระหว่างแอพพลิเคชั่นและวินโดร์เซอร์วิส ตัวที่ทำหน้าที่เป็นตัวติดสื่อสื่อสารรับส่งข้อมูลระหว่างทั้งสองสิ่ง คือ socket programming แสดงได้ดังภาพที่ 6.5



รูป 6.5 การติดต่อสื่อสารและรับส่งข้อมูลระหว่างแอพพลิเคชั่นและมิดเดิลแวร์

เมื่อมิดเดิลแวร์เชื่อมต่อเข้ากับแอพพลิเคชั่นแล้ว ต่อมาจะเป็นส่วนของมิดเดิลแวร์เชื่อมต่อเข้ากับฐานข้อมูล โดยตัวเชื่อมต่อทั้งสองสิ่งเข้าไว้ด้วยกันคือ โอดีบีซี (ODBC) ซึ่งจะเชื่อมต่อกับฐานข้อมูลได้แทบทุกผลิตภัณฑ์ โดยเพียงแค่เปลี่ยนแปลงตัวเชื่อมต่อหรือที่เรียกกันว่า Connection String ที่โค้ดเท่านั้น โดยการเชื่อมต่อนี้มิดเดิลแวร์ไม่จำเป็นต้องทราบเลยว่าฐานข้อมูลเป็นฐานข้อมูลของผลิตภัณฑ์ใด เพียงแค่ส่งเอสคิวแอลผ่านทางโอดีบีซี จากนั้นโอดีบีซีจะช่วยแปลง

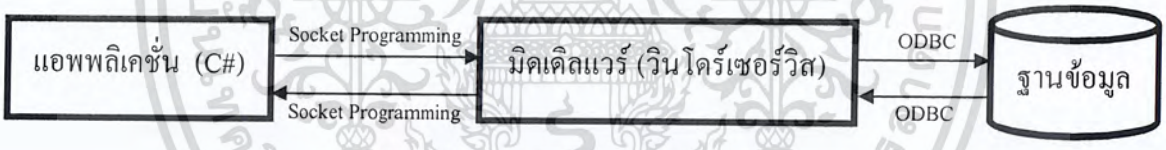
คำสั่งเอสคิวแอลให้ติดต่อกับฐานข้อมูลแบบต่างๆให้เอง ทำให้สามารถใช้คำสั่งแบบเดิมๆกับทุกฐานข้อมูลได้นั่นเอง



รูป 6.6 การเชื่อมต่อมิดเดอร์แวร์และฐานข้อมูล

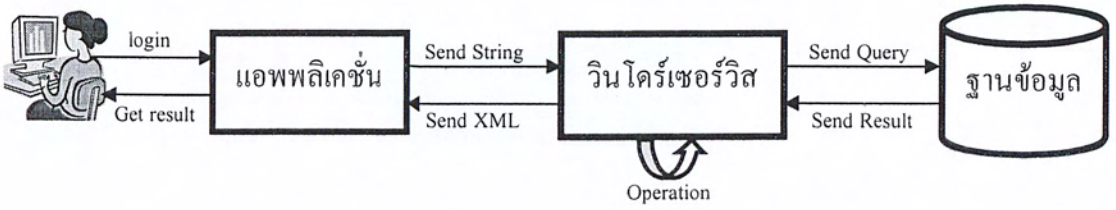
6.6 หลักการและภาพรวมของโปรแกรมทั้งหมด

โปรแกรมจะประกอบไปด้วย 3 ส่วนหลักๆ คือ แอพพลิเคชั่น มิดเดิลแวร์ และฐานข้อมูล ในส่วนของแอพพลิเคชั่นจะใช้ภาษา C# ในการสร้างและพัฒนา ในส่วนของมิดเดิลแวร์จะใช้วินโดร์เซอร์วิสที่ใช้ภาษา C# ในการสร้างและพัฒนาด้วยเช่นกัน และในส่วนของฐานข้อมูลจะใช้ทุกฐานข้อมูล การติดต่อและสื่อสารระหว่างแอพพลิเคชั่นกับวินโดร์เซอร์วิสจะใช้ Socket Programming ส่วนการเชื่อมต่อระหว่างวินโดร์เซอร์วิสกับฐานข้อมูลจะใช้ ODBC ส่วนประกอบทั้งหมดแสดงได้ดังรูปที่ 6.7



รูป 6.7 ส่วนประกอบของโปรแกรม

การทำงานของโปรแกรม เริ่มต้นจากผู้ใช้งานเข้าสู่ระบบด้วยการล็อกอิน เมื่อล็อกอินผ่านแล้วจะพบกับหน้าต่างที่ให้เลือกคำสั่งต่างๆ เช่น คำสั่ง insert เป็นต้น เมื่อผู้ใช้งานตกลงเลือกคำสั่งแล้วแอพพลิเคชั่นจะทำการส่งค่าคำสั่งไปเป็น String ให้กับวินโดร์เซอร์วิสทำการประมวลผลต่างๆ เมื่อประมวลผลเสร็จวินโดร์เซอร์วิสจะทำหน้าที่ส่ง Query ไปให้กับฐานข้อมูล จากนั้นฐานข้อมูลจะส่งผลลัพธ์กลับมาให้กับวินโดร์เซอร์วิสประมวลผลอีกครั้ง เมื่อวินโดร์เซอร์วิสประมวลผลเสร็จจะส่งผลลัพธ์เป็น XML กลับไปให้แอพพลิเคชั่น เมื่อแอพพลิเคชั่นได้รับผลลัพธ์นี้แล้วจะแสดงผลให้กับผู้ใช้งานได้เห็นที่หน้าจอคอมพิวเตอร์ การทำงานทั้งหมดจะแสดงได้ดังรูปที่ 6.8

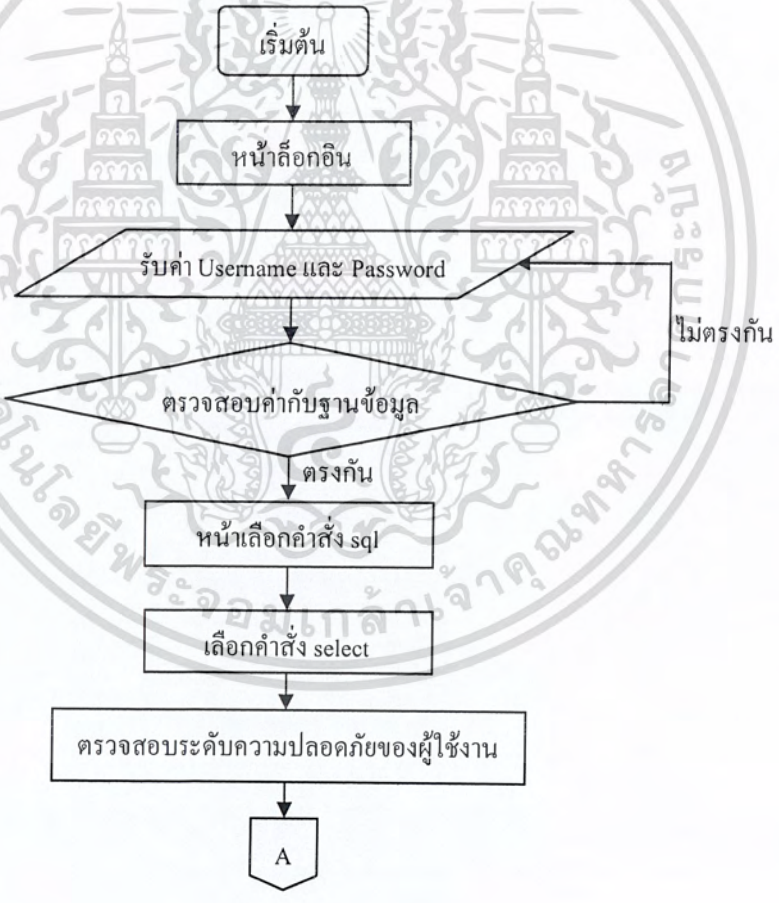


รูป 6.8 ภาพรวมการทำงานทั้งหมด

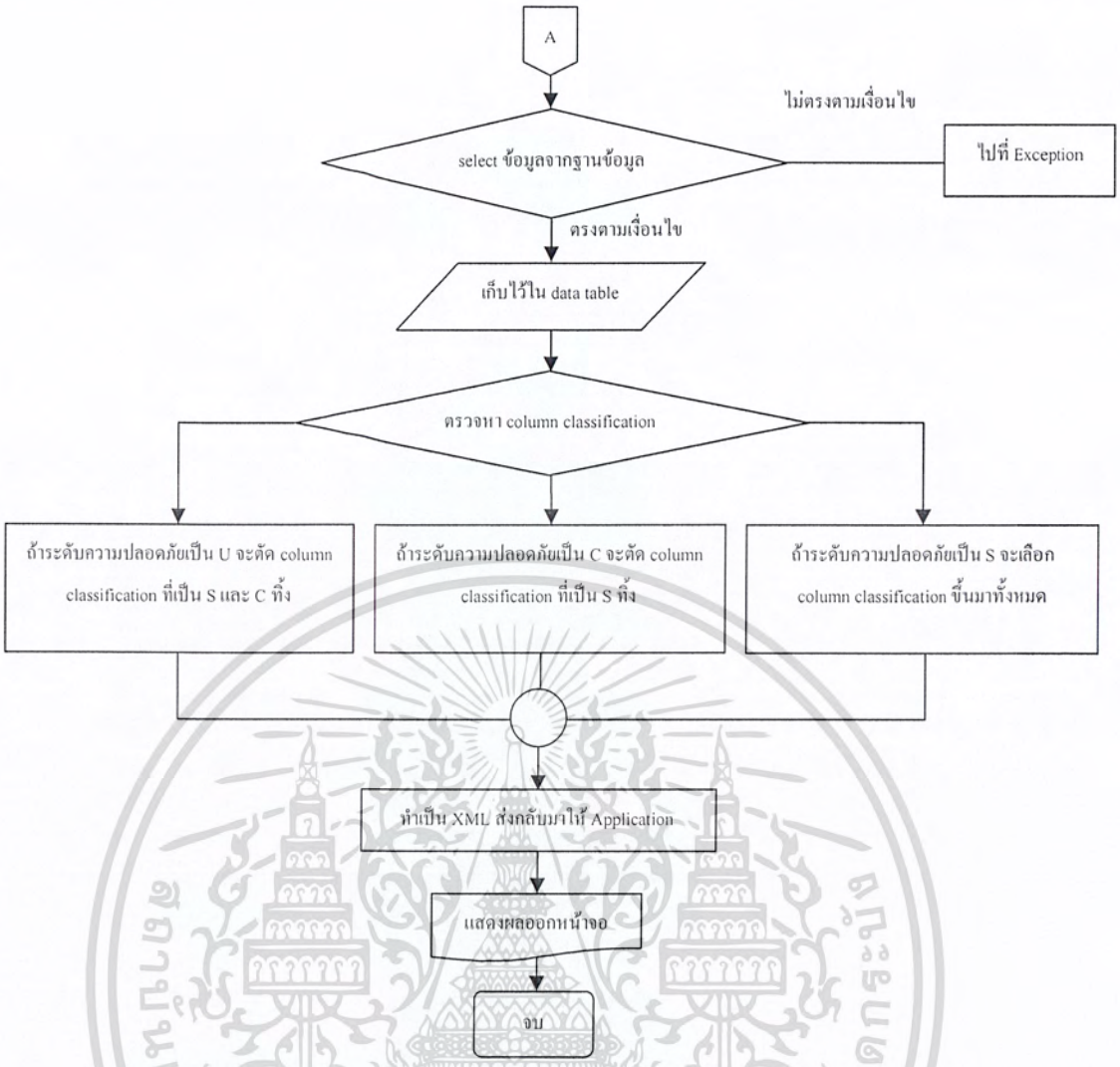
6.7 แผนผังการทำงานของโปรแกรม

แผนผังการทำงานของโปรแกรม จะประกอบด้วย 5 แผนผัง คือ แผนผัง select , แผนผัง insert , แผนผัง verify , แผนผัง delete และแผนผัง update

6.7.1 แผนผัง select

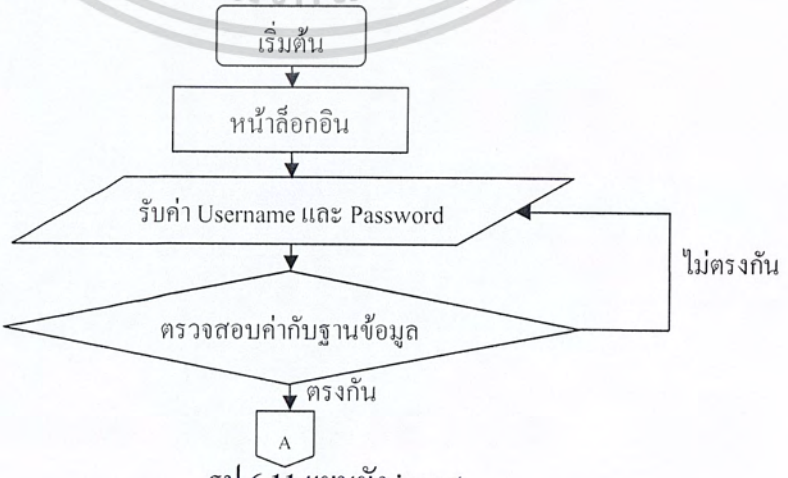


รูป 6.9 แผนผัง select

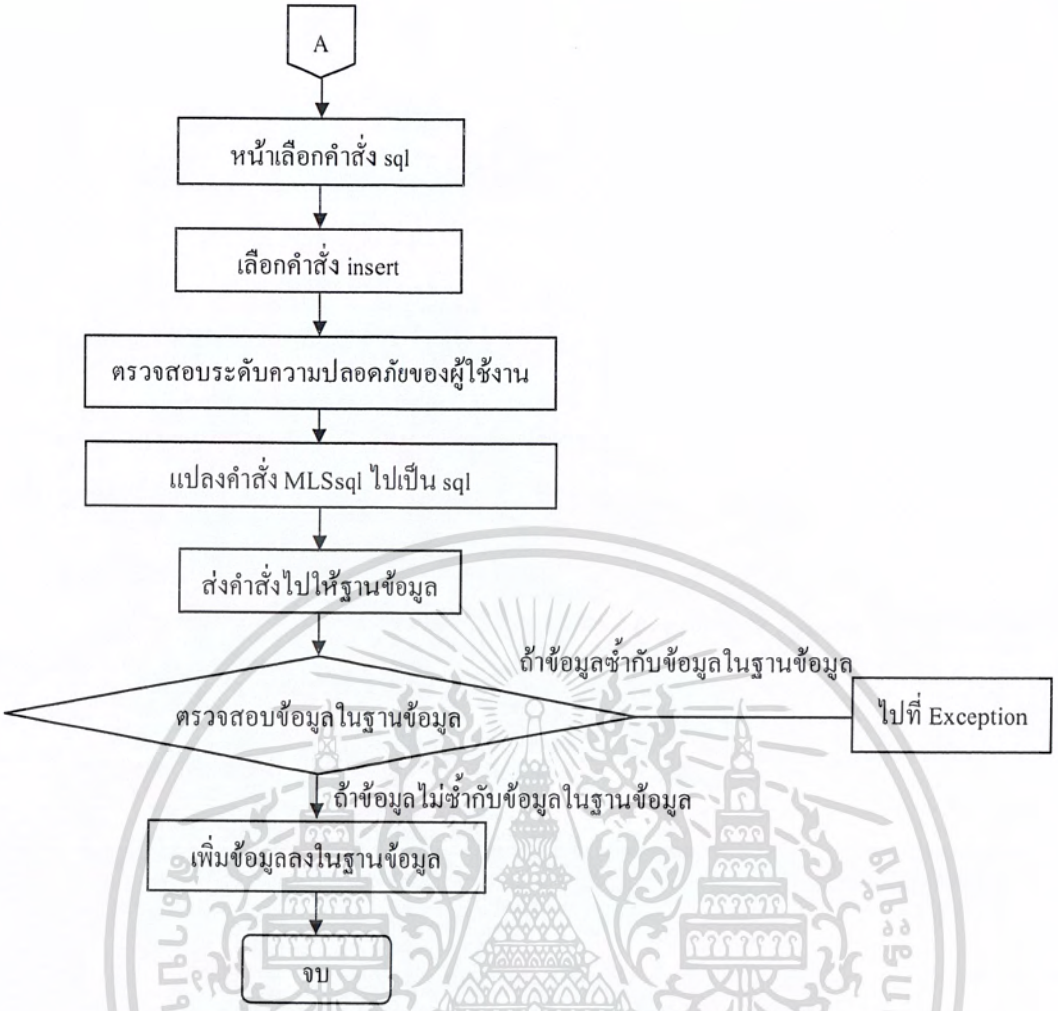


รูป 6.10 แผนผัง select (ต่อ)

6.7.2 แผนผัง insert

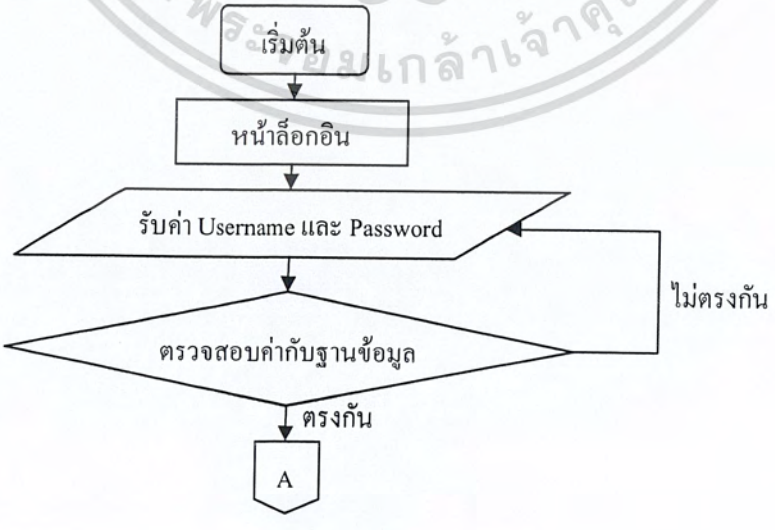


รูป 6.11 แผนผัง insert

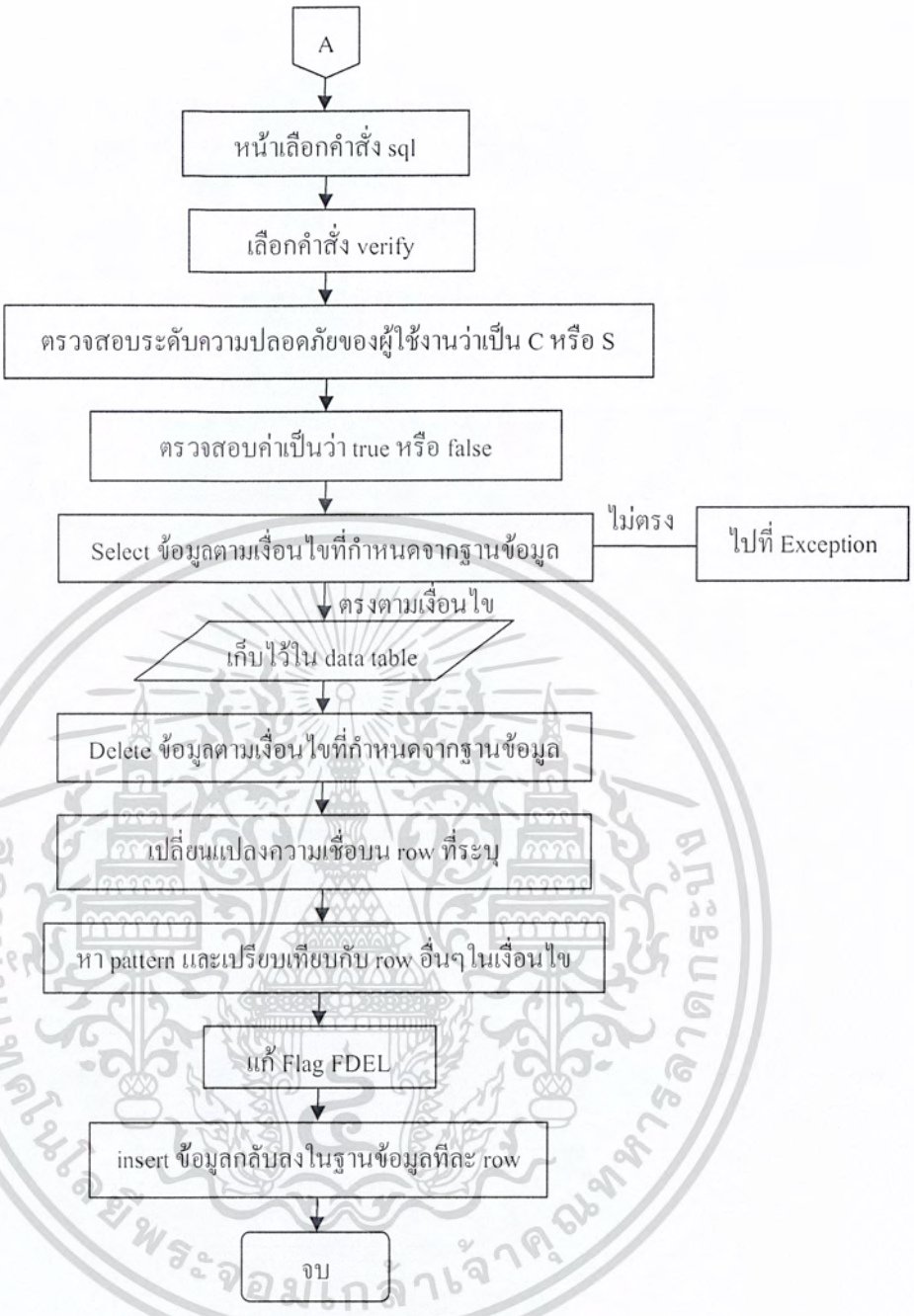


รูป 6.12 แผนผัง insert (ต่อ)

6.7.3 แผนผัง verify

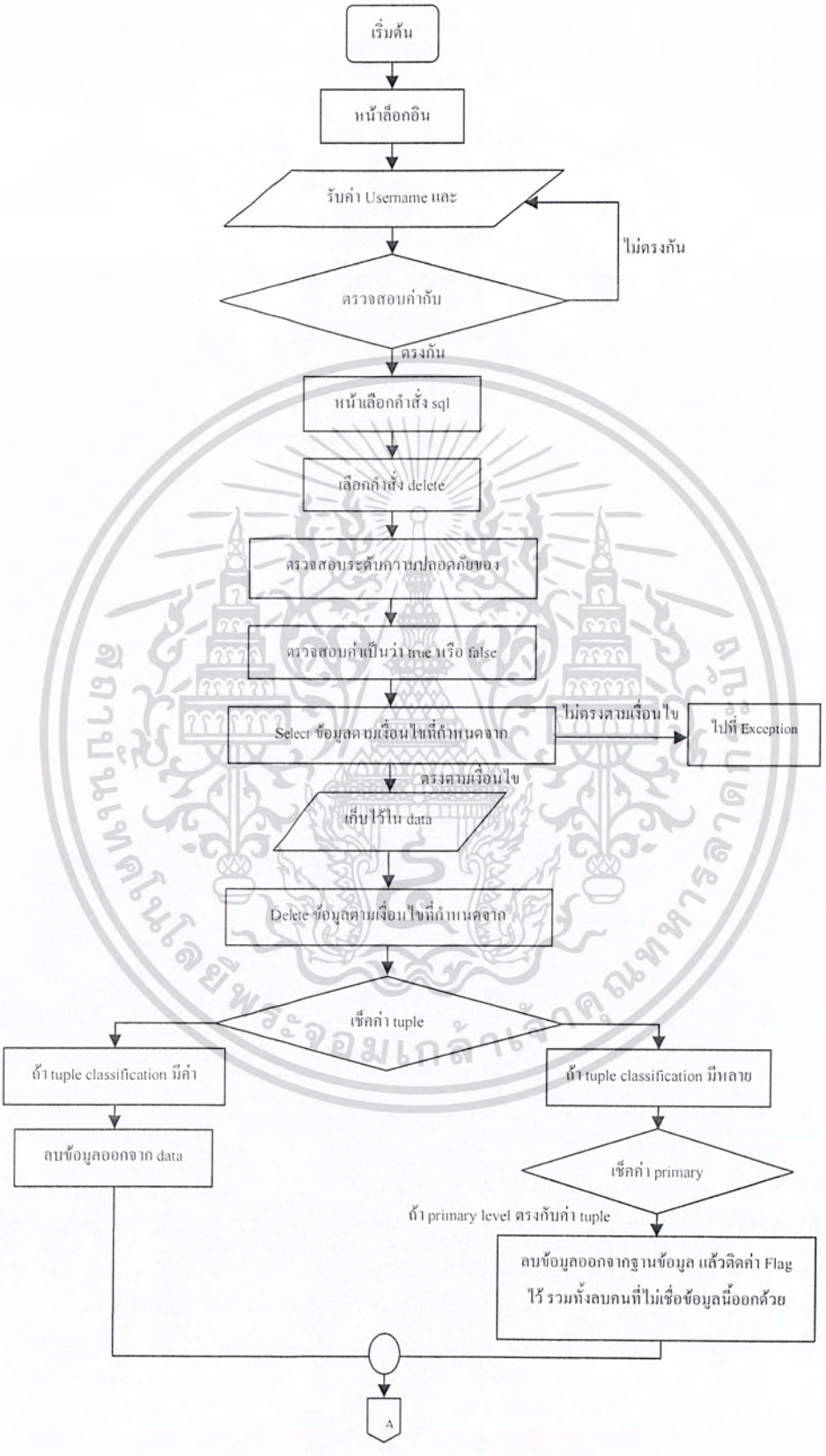


รูป 6.13 แผนผัง verify

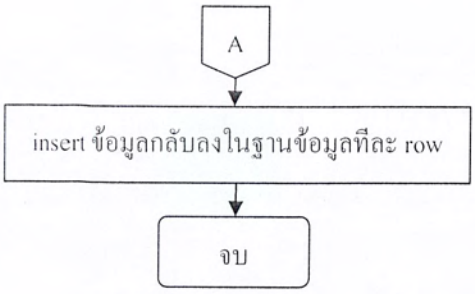


รูป 6.14 แผนผัง verify (ต่อ)

6.7.4 แผนผัง delete

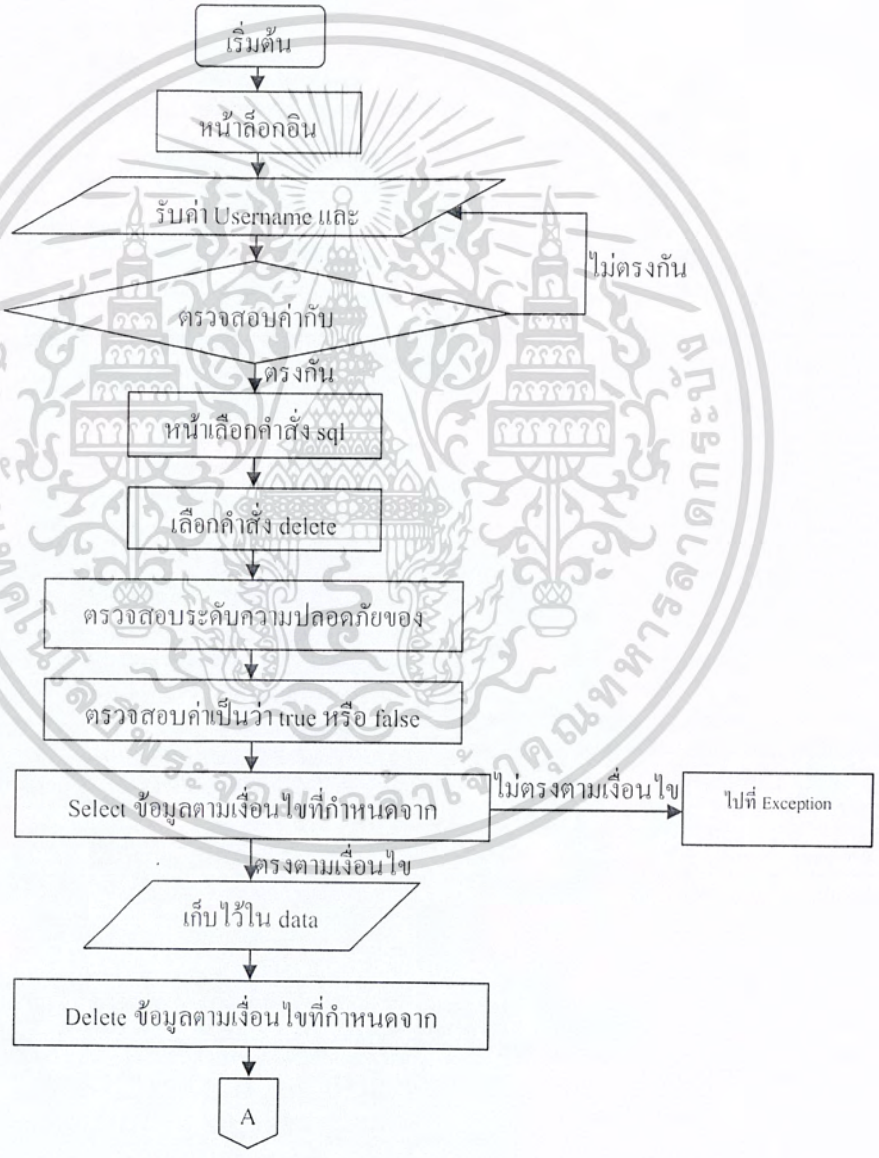


รูป 6.15 แผนผัง delete (ต่อ)

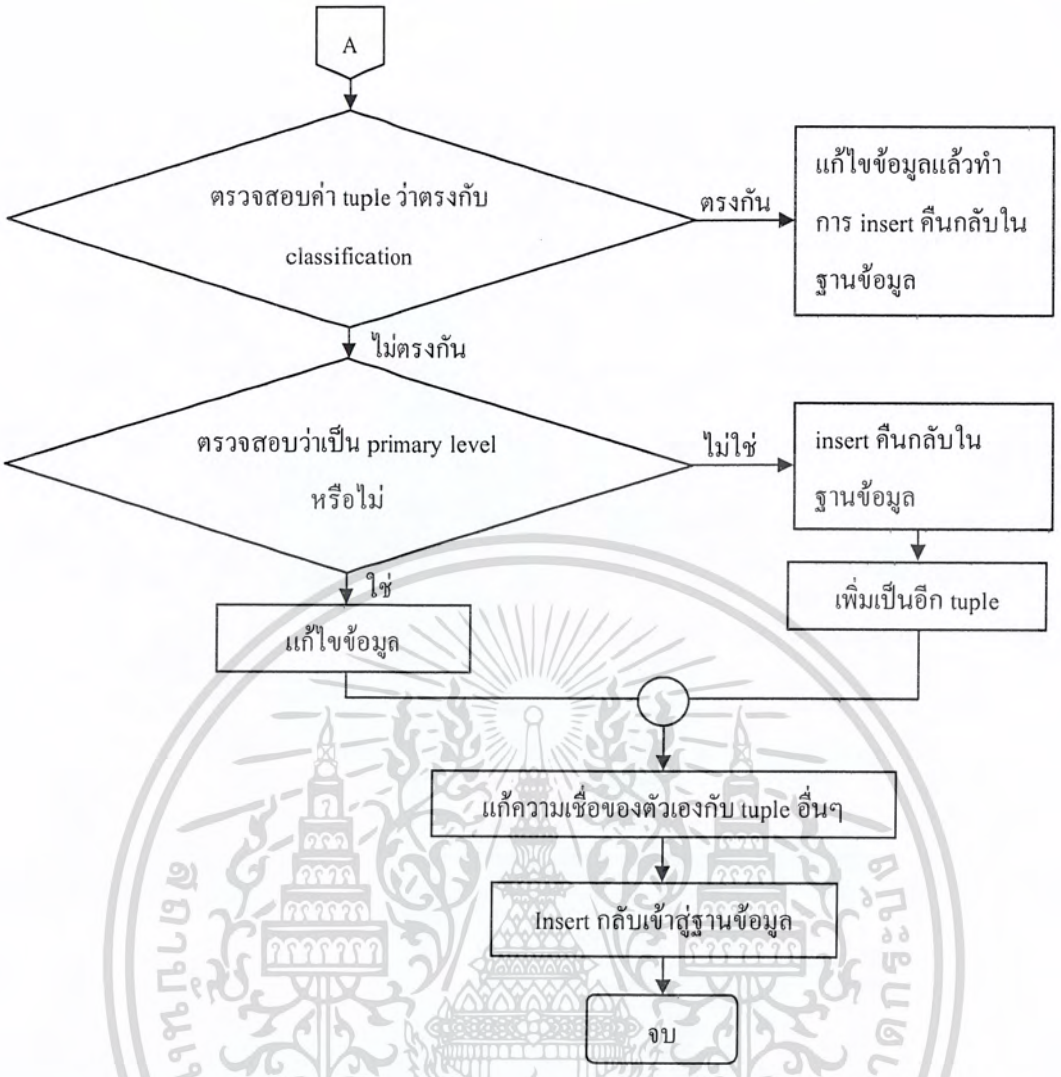


รูป 6.16 แผนผัง delete (ต่อ)

6.7.5 แผนผัง update



รูป 6.17 แผนผัง update



รูป 6.18 แผนผัง update (ต่อ)

6.8 เปรียบเทียบโปรแกรมที่สร้างขึ้นกับโมเดลของจูดิก

เมื่อทำการเปรียบเทียบโปรแกรมกับโมเดลของจูดิกจะพบว่าโปรแกรมมีความเหมือนกับโมเดลของจูดิกหลายประการ เพราะสร้างขึ้นตามทฤษฎีโมเดลของจูดิกเป็นหลัก ซึ่งในโปรแกรมก็ได้นำริชเชอร์เซต (Richer Set) มาใช้เป็นสัญลักษณ์ย่อระดับความปลอดภัย รวมทั้งหลักการและแนวคิดต่างๆก็ถูกนำมาสร้างเป็นโครงสร้างฐานข้อมูลของโปรแกรม แต่ทว่าโครงสร้างของโมเดลของจูดิกยังไม่ดีพอ จึงมีปรับปรุงโครงสร้างบางส่วนด้วยการเพิ่มแอททริบิวในการช่วยจดจำว่าข้อมูลใดต้องได้รับการพิจารณาอีกครั้ง ดังนั้นโครงสร้างฐานข้อมูลใหม่ที่ได้คิดและพัฒนาใช้กับโปรแกรมจึงเป็นดังนี้

$$R(A_1,C_1,\ldots,A_n,C_n,TC,F\_DEL) \qquad (6.1)$$

แต่ส่วนความสามารถในการเขียนนั้นจะแตกต่างกัน นั่นคือโมเดลนี้สามารถแก้ไขข้อมูลผู้ที่ใช้งานระดับล่างได้ เนื่องจากมีกรอบแนวความคิดที่แตกต่างกัน อันรวมไปถึงการ update, delete และ verify ซึ่งจะเกี่ยวเนื่องกับความเชื่อของผู้ใช้งานในระดับอื่น

6.9 เปรียบเทียบโปรแกรมที่สร้างขึ้นกับผลิตภัณฑ์ของออราเคิล

เมื่อทำการเปรียบเทียบโปรแกรมกับผลิตภัณฑ์ของออราเคิลจะพบว่า โปรแกรมที่สร้างขึ้นมานั้นมีฟังก์ชันทางความปลอดภัยหลายระดับบนฐานข้อมูลที่สามารถทำงานได้อย่างมีประสิทธิภาพไม่น้อยไปกว่าออราเคิลเลเบลซีเคียวริตีที่มีการใช้งานอยู่จริงในผลิตภัณฑ์ของออราเคิล โดยโปรแกรมนี้จะใช้หลักการเดียวกับออราเคิลเลเบลซีเคียวริตี คือ มีการใช้เลเบลเป็นตัวแบ่งระดับของยูสเซอร์ และนอกจากนี้ยังมีฟังก์ชันหลักที่สามารถทำงานได้ดังตารางต่อไปนี้

ตาราง 6.61 เปรียบเทียบฟังก์ชันการทำงานระหว่างผลิตภัณฑ์ออราเคิลกับโปรแกรมที่สร้างขึ้น

ฟังก์ชันการทำงาน	Oracle Label Security(OLS)	Application Services MLS
Read	✓	✓
Write Up	✓	✓
Write Down	✓	✓
Insert	✓	✓
Delete	✓	✓
Update	✓	✓
Label Update	✓	✓

บทที่ 7
สรุปผลและข้อเสนอแนะ

7.1 สรุปผล

จากการศึกษาวิจัยทำให้ได้ทราบถึงหลักการการทำงาน แนวคิด และวิวัฒนาการของโมเดลเอ็มแอลเอสที่มีนักวิจัยได้ทำการศึกษากันมา ได้ทราบถึงความแตกต่างของระบบฐานข้อมูลที่มีความปลอดภัยระดับเดียวว่าแตกต่างจากฐานข้อมูลที่มีความปลอดภัยหลายระดับอย่างไร อีกทั้งยังได้ทราบถึงประโยชน์ของการนำไปใช้ในงานจริงๆว่ามีการใช้ในทางใดบ้าง จึงได้ทำการออกแบบโครงสร้างของฐานข้อมูลและหลักการการใช้คำสั่งเอ็มแอลเอสเอสคิวแอล เพื่อเพิ่มขีดความสามารถและศักยภาพของระบบฐานข้อมูลที่มีความปลอดภัยระดับเดียวให้กลายเป็นฐานข้อมูลที่มีความปลอดภัยหลายระดับ

7.2 ปัญหาและอุปสรรค

เนื่องจากงานวิจัยเหล่านี้มีผู้ที่ศึกษาวิจัยเป็นจำนวนมาก แต่งานวิจัยส่วนมากจะไม่ได้ยึดหลักแนวคิดของเบล และพาลาดูลาเป็นหลัก ซึ่งทำให้ยากต่อการศึกษา อีกทั้งการค้นหาคุณสมบัติของ ดีบีเอ็มเอสนั้นจะถูกห้ามเผยแพร่ออกนอกประเทศสหรัฐอเมริกา จึงยิ่งเป็นการยากที่จะเปรียบเทียบการทำงานจริงของแนวคิดกับทฤษฎี

7.3 แนวทางการพัฒนาต่อ

หลังจากที่ได้ออกแบบคำสั่งเอ็มแอลเอสเอสคิวแอลและโครงสร้างของฐานข้อมูลไว้ รวมไปถึงการพัฒนาโปรแกรมติดต่อส่วนกลางที่เพิ่มขีดความสามารถของระบบฐานข้อมูลทั่วไปให้กลายเป็นระบบฐานข้อมูลที่มีความปลอดภัยหลายระดับ โครงงานได้พัฒนาภาษาเอ็มเอสเอสคิวแอลในส่วนของคำสั่งอินเสิร์ต อัพเดท ดีลีท และซีเล็ค ในส่วนอื่นที่ยังไม่พัฒนาให้ใช้งานได้ผ่านมิดเดิลแวร์ คือ ในส่วนของคำสั่งในการสร้างตารางและวิวต่างๆ รวมไปถึงคำสั่งในการให้สิทธิแก่ผู้ใช้งานต่างๆ ซึ่งในปัจจุบันไม่สามารถทำได้ผ่านมิดเดิลแวร์ได้ เพื่อให้เกิดภาษาเอ็มแอลเอสเอสคิวแอลที่มีขีดความสามารถในการจัดการฐานข้อมูลสมบูรณ์ยิ่งขึ้น

บรรณานุกรม

D. E. Bell and L. J. LaPadula. 1974. , "**Secure Computer Systems: Mathematical Foundations and Model.**" Technical Report, MITRE Corporation.

Jajodia S. and Sandhu R. 1991. "Toward a Multilevel Secure Relational Data Model." **In Proceedings of the IEEE Symposium on Research in Security and Privacy**. IEEE Computer Society Press, Los Alamitos, CA, 128–142.

Sandhu R., and Chen F. 1995. "The Semantics and Expressive Power of the MLR Data Model." **Proceedings: IEEE Symposium on Security and Privacy**, Oakland, Ca, May, 128-142.

Sandhu R., and Chen F. 1995. "The Multilevel Relational (MLR) DataModel." **ACM Transactions on Information and System Security**, Vol. 1, No. 1, November 1998, Pages 93–132

Smith K. and M. Winslett. 1992. "Entity modeling in the MLS relational model." **Proceedings of Eighteenth VLDB**, pp 199-210.

Jukic N, Vrbsky SV. 1997. "Asserting Beliefs in MLS Relational Models." **SIGMOD Record**, Vol. 26,No.3,pp.30-35.

Jukic N., Vrbsky S., Parrish A., Dixon B, Jukic B. 1999. "A Belief-Consistent Multilevel Secure Relational Data Model." **Information Systems**, Vol. 24, No. 5, pp. 377-402.

Jukic N., Jukic B., Meamber L., Nezlek G. "Improving E-Business Customer Relationship Management Systems with Multilevel Secure Data Models." **Hawaii International Conference on System Sciences,2002**

Pranjic M., Fertalj K., Jukic N., "Importance of Semantics in MLS Database Models" 24th Int. Conf. **Informafion Technology Interfaces IT1 2002**, June 24-2, 2002, Cavtat, Croatia.

Jukic, N., B. Jukic, L. Meamber, and G. Nezlek, "Employing a Multilevel Secure Approach in CRM Systems." **The Journal of Information Technology Theory and Application (JITTA)**. 4:2, 2002, 17-31.

Jukic N., Vrbsky S., Nestorov S., "Closing the Key Loophole in MLS Databases" **SIGMOD Record**, Vol. 32, No. 2, June 2003

Pranjic M., Jukic N., Fertalj K., "Implementing Belief-Consistent Multilevel Secure Relational Data Model: Issues and Solutions" **Journal of Computing and Information Technology - CIT 11**, 2003, 3, 225-232

S. Nestorov and N. Jukic, "Implementing SEID as a Solution for Connecting NKCS", The University Of Chicago, Computer Science Department, **Technical Report** TR-2003-03, 2003

Sumit Jeloka. 2009. "Oracle® Label Security Administrator's Guide 11g Release 2 (11.2) An Oracle White Paper" **Oracle Corporation World Headquarters USA**, Inc., 500.

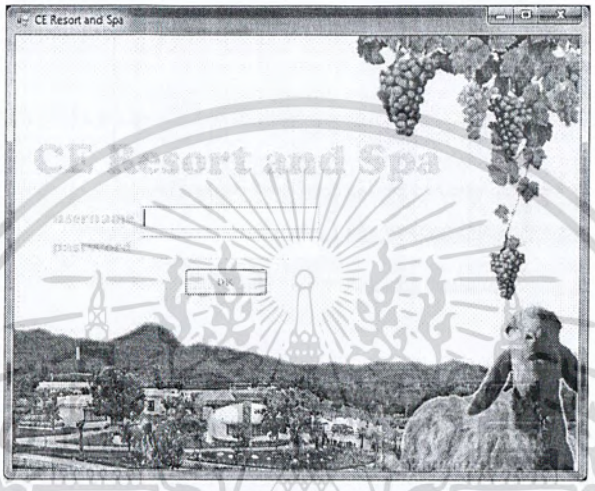
Paul Needham. 2009. "Oracle Label Security with Oracle Database 11g Release 2" **Oracle Corporation World Headquarters USA.**

ภาคผนวก ก

วิธีการใช้งานโปรแกรม

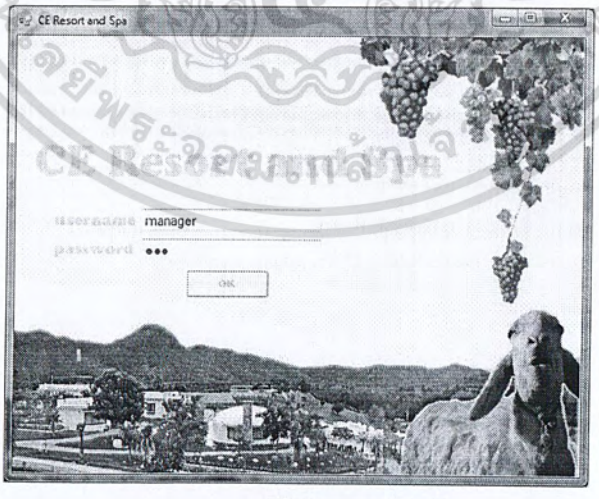
ก.1 วิธีการใช้งาน

เมื่อผู้ใช้งานเปิดโปรแกรมขึ้นมาจะพบกับหน้าต่างดังต่อไปนี้



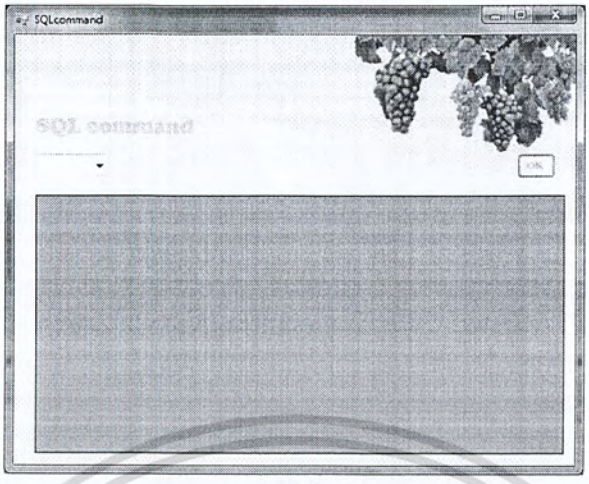
ก.1 หน้าต่างการใช้งานเริ่มต้น

ผู้ใช้งานจะต้องทำการกรอก username และ password



ก.2 หน้าต่างการล็อกอิน

หากทำการล็อกอินผ่านแล้วจะเข้าสู่ระบบด้วยหน้าต่างดังต่อไปนี้

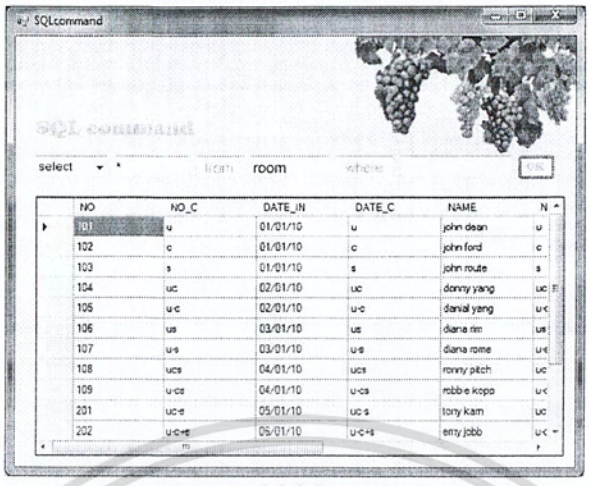


ก.3 หน้าต่างเข้าสู่ระบบ

ผู้ใช้งานจะต้องเลือกคำสั่งเอสคิวเอส ได้แก่ insert , update , delete , select และ verify (คำสั่ง verify จะใช้งานได้เฉพาะผู้ใช้งานที่อยู่ในระดับ C หรือ S เท่านั้น)

ก.4 หน้าต่างเลือกคำสั่งเอสคิวแอล

เมื่อกดปุ่ม OK แล้ว โปรแกรมจะแสดงผลของคำสั่งดังกล่าวเป็นตาราง ดังรูปต่อไปนี้



ก.5 หน้าต่างเลือกคำสั่งเอสคิวแอล

ภาคผนวก ข

วิธีการติดตั้งฐานข้อมูล

ข.1 วิธีติดตั้งฐานข้อมูล

1) สร้างฐานข้อมูล ชื่อ orcl

2) สร้างผู้ใช้งาน admin ที่มีบทบาท (role) เป็นผู้ดูแลระบบ (DBA)

3) ล็อกอินในฐานะ admin แล้วสร้างตารางชื่อ room ที่มีโครงตารางดังนี้

No , no_c , datein , date_c , name , name_c , guest , guest_c , TC , FDEL (ข.1)

4) จากนั้นก็สร้างตารางชื่อ user_login ที่มีโครงตารางดังนี้

Uid , username , password , classification (ข.2)

5) สร้าง view ได้แก่ sview, cview , uview มีเงื่อนไขดังนี้

Create view sview as select * from room; (ข.3)

Create view cview as select * from room where TC like '%u%' or TC like '%c%' (ข.4)

Create view uview as select * from room where TC like '%u%' (ข.5)

5) สร้างผู้ใช้งาน manager senior และ junior โดยให้ admin เป็นให้สิทธิ์ในการเข้าถึงข้อมูล

6) admin ให้สิทธิ์ manager เข้าถึง sview ให้สิทธิ์ senior เข้าถึง cview และให้สิทธิ์ junior เข้าถึง uview

ข.2 ตัวอย่างโครงตาราง

	NO	NO_C	DATE_IN	DATE_C	NAME	NAME_C	GUEST	GUEST_C	TC	FDEL
1	101	u	01/01/10	u	john dean	u	2	u	u	NULL
2	102	c	01/01/10	c	john ford	c	1	c	c	NULL
3	103	s	01/01/10	s	john route	s	3	s	s	NULL
4	104	uc	02/01/10	uc	donny yang	uc	2	uc	uc	NULL
5	105	u-c	02/01/10	u-c	danial yang	u-c	1	u-c	u-c	NULL
6	106	us	03/01/10	us	diana rim	us	2	us	us	NULL
7	107	u-s	03/01/10	u-s	diana rome	u-s	2	u-s	u-s	NULL
8	108	ucs	04/01/10	ucs	ronny pitch	ucs	2	ucs	ucs	NULL
9	109	u-cs	04/01/10	u-cs	robbie kopp	u-cs	2	u-cs	u-...	NULL
10	201	uc-s	05/01/10	uc-s	tony kam	uc-s	1	uc-s	u...	NULL
11	202	u-c+s	06/01/10	u-c+s	emy jobb	u-c+s	2	u-c+s	u-...	NULL
12	203	cs	06/01/10	cs	pond fang	cs	1	cs	cs	NULL
13	204	c-s	07/01/10	c-s	ann kahn	c-s	2	c-s	c-s	NULL
14	205	uc	08/01/10	uc	paul rope	c	2	uc	c	NULL
15	205	uc	08/01/10	uc	chris rope	u-c	2	uc	u-c	NULL

ข.1 ตาราง room

	UID	USERNAME	PASSWORD	CLASSIFICATION
1	1	admin	123	s
2	2	manager	123	s
3	3	senior	123	c
4	4	junior	123	u

ข.2 ตาราง user_login