บทที่ 3

แนวคิดและทฤษฎี

ฐานข้อมูล (Database)

ฐานข้อมูล หมายถึงการนำข้อมูลในองค์กรที่มีความเกี่ยวข้องกันมารวมกันไว้อย่างมีระบบในที่เดียวกันที่มีความเกี่ยวข้องกันมารวมกัน อย่างมีระบบในที่เดียวกัน โดยที่ผู้ใช้ฐานข้อมูลแต่ละคนจะมองข้อมูลนี้ในแง่มุมที่แตกต่างกันไป ตามจุดประสงค์ของการประยุกต์ใช้งาน โดยอาศัยระบบจัดการฐานข้อมูล (Database management system) เป็นตัวควบคุมวัตถุประสงค์หลักของระบบจัดการฐานข้อมูลคือการจัดหามุมมองให้แก่ผู้ใช้ โดยที่ผู้ใช้ไม่จำเป็นต้องสนใจว่าลักษณะการเก็บข้อมูลโดยแท้จริงแล้วเป็นเช่นไร โดยระบบจะซ่อนรายละเอียดต่างๆของข้อมูลเหล่านั้นว่าถูกเก็บและบำรุงรักษาอย่างไร เพื่อให้ข้อมูลสามารถถูกดำเนินการได้อย่างมีประสิทธิภาพ

ฐานข้อมูลในปัจจุบันมี 3 ประเภท (3) คือ

1 **ฐานข้อมูลแบบลำดับชั้น** (Hierarchical database) ข้อมูลถูกเก็บโดยมีโครงสร้างเป็นแบบต้นไม้ (Tree) มีความสัมพันธ์ระหว่างข้อมูลคล้ายลักษณะการแตกกิ่งก้านของต้นไม้ โดยเริ่มจากรากเดียวเป็นหลักแล้วแตกกิ่งออกมีตั้งแต่ 1 กิ่งขึ้นไปจากนั้นกิ่งย่อยแต่ละกิ่งก็อาจแตกกิ่งต่อออกไปเรื่อยๆ ข้อมูลที่อยู่ในระดับบนสุดเรียกว่าราก (Root) กิ่งที่เป็นต้นตอก่อนที่จะแตกกิ่งย่อยแต่ละจุดเรียกว่า พาเรนท์ (Parent)ซึ่งเปรียบเหมือนเป็น พ่อ แม่ และเรียกกิ่งย่อยที่แตกแขนงออกไปว่าไชล์ด (Child) ซึ่งเปรียบเหมือนลูก การออกแบบฐานข้อมูลแบบลำดับชั้นนี้ เหมาะสำหรับระบบข้อมูลที่มีโครงสร้าง ซึ่งข้อมูลลูกแต่ละชุดจะอยู่ภายใต้ข้อมูลแม่เพียงหนึ่งเท่านั้น

2 **ฐานข้อมูลแบบเครือข่าย** (Network database) โครงสร้างข้อมูลคล้ายกับโครงสร้างแบบต้นไม้แต่ความสัมพันธ์ยืดหยุ่นได้ โดยที่ข้อมูลแต่ละระดับเกิดจากข้อมูลแม่ไม่จำกัดจำนวนประเภททำให้โครงสร้างข้อมูลแบบนี้มีความซับซ้อน การเก็บข้อมูลเป็นลักษณะลิสต์เชื่อมโยง (Link list) หรือ ตัวชี้ (Pointers) จากระเบียนแม่ (Parent record) เชื่อมโยงไปหาระเบียนลูก (Children record) การออกแบบฐานข้อมูลแบบนี้ เหมาะสำหรับระบบข้อมูลที่ข้อมูลย่อยมีความมีความสัมพันธ์เกี่ยวข้องกับข้อมูลย่อยประเภทอื่นหลายประเภท และไม่มีข้อจำกัดในลักษณะของความสัมพันธ์

3 ฐานข้อมูลแบบสัมพันธ์ (Relational database) โครงสร้างข้อมูลประกอบด้วยกลุ่มข้อมูลย่อยที่มีความสัมพันธ์ในรูปตาราง (Table) แสดงความสัมพันธ์ระหว่างกลุ่มข้อมูลด้วยค่าย่อยในบางคอลัมน์ (Attributevalue) ของแต่ละความสัมพันธ์หรือตารางนั้น ๆ ฐานข้อมูลที่เป็นที่นิยมคือฐานข้อมูลแบบสัมพันธ์ (Relational database) เพราะโมเดลที่ใช้ในการออกแบบมีข้อดีดังนี้

3.1 เป็นโมเดลที่สร้างความเข้าใจได้ง่ายกว่า ในแง่การมองของผู้ใช้จะไม่มีความจะไม่มีความสลับซับซ้อนมากนัก

3.2 ระบบส่วนใหญ่ที่ใช้โมเดลนี้มีเครื่องมือที่ช่วยให้ผู้ใช้สามารถปฏิบัติการยุ่งยากกับข้อมูลด้วยคำสั่งง่าย ๆ ได้

3.3 โมเดลนี้มีเครื่องมือที่ช่วยให้ผู้ใช้สามารถค้นพบปัญหาที่เกิดขึ้น ในการออกแบบฐานข้อมูลได้โดยง่าย และช่วยในการแก้ไขการออกแบบที่ผิดพลาดนั้นด้วย

3.4 ส่วนของการจัดเก็บข้อมูลแบบกายภาพมีความแตกต่าง จากข้อมูลแบบตรรกโดยสิ้นเชิง นับว่าเป็นโมเดลที่สอดคล้องกับหลักการของฐานข้อมูลในข้อที่จะให้ผู้ใช้ไม่ต้องทราบถึงรายละเอียดของการเก็บข้อมูลจริง

ระบบจัดการฐานข้อมูล (DaTabase mangement system)

คือระบบที่มีหน้าที่ควบคุมดูแลการสร้างและเรียกใช้ฐานข้อมูล โดยที่ผู้ใช้ไม่จำเป็นต้องรู้เกี่ยวกับรายละเอียดในโครงสร้างของฐานข้อมูล ระบบจัดการฐานข้อมูลมีหน้าที่ดังนี้

1 การสร้างและแก้ไขโครงสร้างของฐานข้อมูลรวมทั้งบรรจุข้อมูลในการทำงาน

2 การเข้าถึงเนื้อหาในฐานข้อมูล เพื่อการแก้ไข หรือเรียกดูข้อมูลได้พร้อมกัน

3 กำหนดคำจำกัดความและข้อบังคับเกี่ยวกับความต้องการ ในการรักษาความปลอดภัย การให้สิทธิผู้ใช้ และการป้องกันความเสียหาย ตลอดจนการทำสำรองข้อมูล

4 รวบรวมสถิติที่เกี่ยวข้องกับการใช้ฐานข้อมูล

5 เฝ้าคุมระบบ (System monitoring) และเปลี่ยนแปลงแก้ไขระบบเพื่อให้ทำงานได้ดีขึ้น

6 มีพจนานุกรมข้อมูล (Data dictionary) เกี่ยวกับฐานข้อมูล เช่น โครงสร้าง ชนิด รูปแบบ และข้อจำกัดของข้อมูล เป็นต้น

การเลือกระบบจัดการฐานข้อมูล (4)

สิ่งที่ต้องพิจารณาในการออกแบบฐานข้อมูล นอกจากจะพิจารณาถึงชนิดของฐานข้อมูล ซึ่งได้แก่ฐานข้อมูลแบบลำดับชั้น แบบเครือข่ายและแบบความสัมพันธ์แล้ว สิ่งที่ต้องพิจารณาคือ

การเลือกระบบจัดการฐานข้อมูล ในจำนวผลิตภัณฑ์โปรแกรมจัดการฐานข้อมูล ที่มีใช้อยู่บนเครื่องคอมพิวเตอร์ส่วนบุคคลนั้น สามารถจำแนกออกได้เป็นกลุ่มใหญ่ ๆ ตามชนิดของระบบปฏิบัติการที่รองรับเช่นกลุ่มที่ทำงานบนวินโดวส์ (Windows database managers) และกลุ่มที่ทำงานบนดอส (Dos database managers) ฯลฯ ผลิตภัณฑ์โปรแกรมจัดการฐานข้อมูล ที่ทำงานบนวินโดวส์ (Windows database managers) มีรูปแบบการติดต่อกับผู้ใช้ที่ น่าดู และสะดวกต่อการใช้งาน (Friendly interface) และเป็นผลิตภัณฑ์ที่มีการออกแบบ ให้รูปแบบการติดต่อเป็นมาตราฐานเดียวกัน (Common user access : CUA) มีการออกแบบระบบช่วยเหลือแบบออนไลน์ (On-line help) ให้รายละเอียด และประสิทธิภาพในการออกแบบให้คำแนะนำแก่ผู้ใช้ได้ดีขึ้นอย่างมาก นอกจากนี้ระบบปฏิบัติการวินโดวส์ยังประกอบไปด้วยส่วนของการทำงาน ที่ช่วยเสริมสมรรถนะให้กับโปรแกรมจัดการฐานข้อมูลวินโดวส์อีกจำนวนมาก ไม่ว่าจะเป็นการทำงาน คลิปบอร์ด และดีดีอี (Dynamix data exchange : DDE) ที่ยอมให้มีการแชร์ข้อมูล (Share) ร่วมกันระหว่างโปรแกรม การทำงานโอแอลอี (Object Linking Embedding : OLE) ที่ยอมให้สร้างความสัมพันธ์ระหว่างข้อมูล หรือนำเอาข้อมูลจากโปรแกรมหนึ่ง ไปฝังตัวไว้ในอีกโปรแกรมหนึ่ง และได้มีการกำหนดมาตรฐาน สำหรับการติดต่อระหว่างโปรแกรมจัดการฐานข้อมูลขึ้นมาโดยเฉพาะเช่น มาตรฐาน โอดีบีซี (Object database connectivity : ODBC) จากบริษัทไมโครซอฟต์ และมาตรฐาน ไอดีเอพีไอ (Integrated database application programming interface : IDAPI) จากบริษัทบอร์แลนด์ ฯลฯ

คุณสมบัติของโปรแกรมจัดฐานข้อมูลบนวินโดวส์ (4)

1 ภาษาคำสั่งภายในโปรแกรม (Built in programming language)

2 รูปแบบฟอร์แมตของแฟ้ม (Native file format) ที่ง่ายต่อการใช้งาน

3 ความสามารถในการรองรับข้อมูลประเภทข้อมูลภาพ (Image data) หรือข้อมูลประเภทบีแอลโอ (Basic large Object : BLO) ที่จัดเก็บโครงสร้างหลักของข้อมูลอีกทีหนึ่ง

4 ความสามารถในการติดต่อสัมพันธ์กับเครื่องดาต้าเบสเซิร์ฟเวอร์ (SQL based database server)

ระบบแบบสัมพันธ์ (Relational system) (5)

ระบบฐานข้อมูลที่จัดเป็นระบบแบบสัมพันธ์โดยสมบูรณ์นั้นจะต้องมีคุณสมบัติ 3 ประการ คือ

1 โครงสร้างข้อมูล (Database structure) ในระบบจะต้องมีเพียงตารางเท่านั้นนั่นคือ ในระดับตรรกผู้ใช้จะมองเห็นข้อมูลเป็นตารางเท่านั้น การกระทำใดกับข้อมูล จะต้องกระทำกับตาราง

2 ระบบจะต้องมีการควบคุมความถูกต้อง (Constrain) ประกอบด้วย

2.1 **บูรณภาพของเอนทิตี** (Entity integrity) สิ่งสำคัญของบูรณภาพของเอนทิตีคือทุก ๆ บรรทัดในแต่ละตาราง จะต้องมีคอลัมน์หรือกลุ่มของคอลัมน์ใด ๆ ที่เป็นคีย์หลัก (Primary key) จะมีค่าว่าง (NULL) ไม่ได้สำหรับข้อบังคับของบูรณภาพ (Integrity constrain) ตามมาตรฐานของ ANSI (American National Standards Institute) ได้กำหนดความหมายของยูนีค (Unique) ว่าคือคอลัมน์ที่มีข้อมูลที่ยูนีคและสามารถมีค่าว่างได้มากกว่า 1 บรรทัดนอกจากนี้ยังสามารถกำหนดให้ยูนีคได้หลาย ๆ คอลัมน์ใน 1 ตาราง

2.2 **บูรณภาพของโดเมน** (Domain integrity) หมายถึงกฎการรักษาความถูกต้องของคอลัมน์ทุกคอลัมน์ในตารางรวมทั้งคีย์หลักคีย์นอก (Foreign key) และคอลัมน์ที่ไม่ใช่คีย์ซึ่งโดเมนนี้ประกอบด้วยชนิดของข้อมูล รูปแบบของข้อมูลความยาวของข้อมูลช่วงค่าของข้อมูลค่าที่กำหนดไว้มีค่าซ้ำกันหรือค่าว่างได้หรือไม่ ซึ่งส่วนใหญ่ผู้ใช้จะเป็นผู้กำหนดโดเมนขึ้นมาเองสิ่งสำคัญของบูรณภาพของโดเมนคือ ข้อมูลของแต่ละเขตข้อมูลจะต้องเป็นสมาชิกภายในโดเมนที่กำหนดไว้เท่านั้น ANSI ได้กำหนดข้อบังคับของบูรณภาพของโดเมน (Domain integrity constrain) ไว้ได้แก่ "NOT NULL" หมายถึงไม่ยอม ให้มีค่าว่างเป็นสมาชิกของโดเมนและ "CHECK" เป็นการกำหนด Constrain ที่ผู้ใช้เป็นผู้ตั้งเงื่อนไขเช่น โดเมนของข้อมูลในเขตข้อมูล FlgPicture คือ"YES" หรือ "NO"

2.3 บูรณภาพของการอ้างอิง (Referential integrity)

บูรณภาพของการอ้างอิง จะพิจารณาถึงความสัมพันธ์ระหว่างตารางหลาย ๆ ตาราง นั่นคือข้อมูลในคอลัมน์ใดคอลัมน์หนึ่ง คีย์นอกซึ่งไม่ใช่คีย์หลักจะต้องจับคู่กับข้อมูลใดข้อมูลหนึ่งของคีย์หลักของอีกตารางบูรณภาพของการอ้างอิงจะคอยควบคุม เมื่อมีการเปลี่ยนแปลงข้อมูลที่มีการอ้างอิงระหว่างตาราง เมื่อเราจะลบหรือแก้ไขคีย์หลัก โปรแกรมระบบจัดการฐานข้อมูล (DBMS) จะตรวจสอบว่ามีข้อมูลของคีย์นอก ในตารางอื่นที่มีการอ้างอิงอยู่หรือไม่ ตามมาตรฐานของ ANSI SQL-89 จะไม่ยอมให้มีการแก้ไขหรือลบคีย์หลักหากกำลังถูกอ้างอิงจากตารางอื่น ส่วนมาตรฐานของ ANSI SQL-92 ได้กำหนดลักษณะการทำงานเป็นแบบ "CASCADE" (จะลบบรรทัด หรือแก้ไขข้อมูลในคีย์นอก ที่สัมพันธ์กันให้ด้วย) "SET NULL" และ "SET DEFAULT" (จะเปลี่ยนข้อมูลในคีย์นอกให้เป็นค่าว่างหรือเป็นค่าใดค่าหนึ่งที่กำหนดไว้) ALL หรือ Nothing ANSI ได้กำหนดให้มีการ

Rollback ข้อมูลทั้งหมด ในกรณีที่คำสั่งใด ๆ ทำให้เกิดกาะละเมิดข้อบังคับของบูรณภาพ เช่น คำสั่งแก้ไขข้อมูล 10 บรรทัด จะมีผลให้เกิดการเปลี่ยนแปลงทั้ง 10 บรรทัด และจะไม่เปลี่ยนแปลงเลยแม้แต่บรรทัดเดียว หากข้อมูลในบรรทัดใดบรรทัดหนึ่ง เกิดละเมิดข้อบังคับของบูรณภาพเมื่อมีการแก้ไขข้อมูล

2.4 การตรวจสอบบูรณภาพโดยอนุโลม (Deferred integrity checking) เวลาเป็นสิ่งที่มีอิทธิพลสำคัญ ในการตรวจสอบข้อบังคับของบูรณภาพของระบบจัดการฐานข้อมูลตามมาตรฐานของ ANSI SQL-89 ได้กำหนดไว้ว่าระบบจัดการฐานข้อมูลจะรอจนกว่าการทำงานของแต่ละคำสั่งสิ้นสุดเสียก่อน จึงจะทำการตรวจสอบการข้อบังคับของบูรณภาพเพราะการทำงาน ขณะยังไม่จบคำสั่งอาจเกิดการละเมิดข้อบังคับของบูรณภาพขณะใดขณะหนึ่งได้ เช่น การแก้ไขข้อมูลของคีย์หลักในทุก ๆ บรรทัดให้มากขึ้น 1 ค่า จะเกิดเหตุการณ์ที่ข้อมูลคีย์หลักของ 2 บรรทัดมีค่าตรงกันอยู่ชั่วขณะ แต่เมื่อแก้ไขครบทุกบรรทัดแล้วข้อมูลคีย์หลักก็ยังคงยูนีคดังเดิม มาตรฐานของANSI SQL-92 ได้กำหนดเพิ่มเติมว่า ผู้ใช้สามารถกำหนดเวลาในการตรวจสอบบูรณภาพ เมื่อเสร็จสิ้นการดำเนินงานที่ต้องการได้ ซึ่งเปิดโอกาสให้นักพัฒนาโปรแกรมสามารถสร้างโปรแกรมในลักษณะต่าง ๆ ได้สะดวกขึ้น การเลื่อนเวลาในการตรวจสอบบูรณภาพไม่เพียงแต่อำนวยความสะดวกในการสร้างแอพพลิเคชั่นเท่านั้น แต่ยังมีผลต่อการทำงานของระบบด้วย เพราะไม่ต้องเสียเวลาไปกับการตรวจสอบทุก ๆ ครั้ง ในการแก้ไขแต่ละบรรทัดแต่จะตรวจสอบตารางเพียงครั้งเดียวเมื่อจบคำสั่งหรือจบการทำงาน

2.5 ข้อบังคับของบูรณภาพของทริกเกอร์ (Triggers integrity constrain) เป็นอีกวิธีการหนึ่งที่สามารถควบคุมบูรณภาพของข้อมูล นั่นคือใช้โปรแกรมตรวจสอบซึ่งโปรแกรมนั้นอาจเขียนในรูปของโพรซีเยอร์ (Procedure) โปรแกรมนี้จะต้องสนับสนุนโดยระบบจัดการฐานข้อมูล เรียกโปรแกรมเหล่านี้ว่าทริกเกอร์ (Trigger) และระบบจัดการฐานข้อมูลส่วนใหญ่ก็จะสนับสนุน ทั้งข้อบังคับของบูรณภาพและทริกเกอร์ โดยจะใช้ข้อบังคับของบูรณภาพ ในเรื่องที่เป็นข้อกำหนดมาตรฐานของฐานข้อมูล (Database) บูรณภาพของเอนทิตี บูรณภาพของโดเมน บูรณภาพของการอ้างอิง และใช้ทริกเกอร์ในการบังคับกฎเกณฑ์ที่เป็นรายละเอียดต่าง ๆ นอกเหนือตามมาตราฐาน เมื่อมีทั้งทริกเกอร์และข้อบังคับของบูรณภาพ ก็สามารถควบคุมกฎต่าง ๆ ทั้งหมด โดยรวมไว้ที่ระบบจัดการฐานข้อมูลของเซิฟเวอร์ได้ระบบจัดการฐานข้อมูลบางตัวอาจมีข้อบังคับของบูรณภาพไม่ครบตามมาตรฐาน ทำให้ต้องใช้ทริกเกอร์เกือบทั้งหมดในการทำให้เกิดบูรณภาพของข้อมูล

3 ภาษาฐานข้อมูล ในระบบดังกล่าวจะต้องมีความสามารถอย่างน้อยเทียบเท่าภาษาที่

เรียกว่าพีชคณิตแบบสัมพันธ์ (Relational algebra) หรือแคลคูลัสแบบสัมพันธ์ (Relation calculus) ภาษาใดภาษาหนึ่ง ภาษาฐานข้อมูลของโมเดลแบบสัมพันธ์ ผู้ที่คิดโมเดลแบบสัมพันธ์ได้ให้กำเนิดภาษาสำหรับโมเดลนี้ไว้ 2 แขนงด้วยกัน คือ

3.1 พีชคณิตแบบสัมพันธ์เป็นภาษาที่มีประสิทธิภาพสูง แต่มีข้อเสียคือเข้าใจและเขียนได้ยาก ทำให้นิยมใช้เป็นภาษาภายในของระบบจัดการฐานข้อมูแบบสัมพันธ์ (Relational database management system) โดยทั่วไป

3.2 แคลคูลัสแบบสัมพันธ์ ได้รับการพัฒนาต่อโดยทีมงาน 2 กลุ่ม กลุ่มแรกมีเป้าหมายที่จะพัฒนาให้ได้ภาษาที่เข้าใจ ผู้ใช้ใช้งานได้ง่ายมาก ๆ ได้พัฒนาออกมาเป็นภาษาในกลุ่มที่เรียกว่า ดีอาร์ซี (Domain oriented relational calculus : DRC) หนึ่งในภาษาในกลุ่มนี้ก็คือ คิวบีอี (Query by example : QBE) ภาษาในกลุ่มนี้ได้รับการพัฒนาออกมาได้ก่อนทีมที่ 2 มีเป้าหมายที่จะพัฒนาให้ได้ภาษาที่มีประสิทธิภาพสูง พัฒนาออกมาเป็นภาษาในกลุ่มที่เรียกว่า TRC (Tuple oriented relational calculus) หนึ่งในกลุ่มนี้ก็คือเอสคิวแอล (Structured cuery language : SQL) ซึ่งเป็นภาษาที่ได้รับการยอมรับกันมากในปัจจุบัน

ในการทำวิจัยนี้เลือกระบบจัดการฐานข้อมูล โดยพิจารณาจากความสามารถในการเก็บข้อมูลรูปภาพ และความสามารถในการเรียกค้นข้อมูล โปรแกรมจัดการฐานข้อมูลที่มีความสามารถดังกล่าวมีหลายโปรแกรมด้วยกัน และทำงานภายใต้ระบบปฏิบัติการวินโดวส์

โปรแกรมจัดการฐานข้อมูลแอคเซส (Access) แตกต่างจากโปรแกรมอื่นในด้านที่ได้รับการพัฒนาเริ่มต้นภายใต้ระบบปฏิบัติการวินโดวส์ โดยบริษัทไมโครซอฟต์เป็นผู้พัฒนา ส่วนโปรแกรมอื่น ๆ นั้น ได้รับการพัฒนามาจากโปรแกรมที่มีชื่อเสียง ซึ่งทำงานภายใต้ระบบปฏิบัติการดอสมาก่อน ทำให้เชื่อว่าโปรแกรมแอคเซส น่าจะใช้ความสามารถของวินโดวส์ได้ดีกว่าโปรแกรมอื่นๆ ดังนั้นการวิจัยนี้จึงเลือกโปรแกรมจัดการฐานข้อมูลแอคเซส

โปรแกรมแอคเซส เป็นโปรแกรมที่ได้รับการออกแบบมา เพื่อการเข้าถึงข้อมูลแบบ Unparallel access ซึ่งอนุญาตให้ผู้ใช้โปรแกรมสามารถเข้าถึงข้อมูลได้ โดยไม่จำกัดว่าฐานข้อมูลดังกล่าว นั้นมีรูปแบบอย่างไรหรือถูกเก็บไว้ในบริเวณไหนของหน่วยความจำ (Regardless of database format and location) ในส่วนฟอร์แมตของฐานข้อมูลนั้น โปรแกรมแอคเซสจะอนุญาตให้ผู้ใช้เรียกดู (View) หรือแก้ไข (Edit) ข้อมูลในฐานข้อมูลซึ่งเป็นที่นิยมในท้องตลาดได้หลายชนิดด้วยกัน เช่น โปรแกรมจัดการฐานข้อมูล Paradox เวอร์ชัน 3.5, dBASE III Plus, dBase IV,FoxBase, Btrive, Microsoft SQL Server หรือ Oracle Database เป็นต้น

ในการติดต่อกับฐานข้อมูลของโปรแกรมจัดการฐานข้อมูลชนิดอื่น ๆ โปรแกรมแอคเซส

สามารถติดต่อได้ 2 ลักษณะคือ การใช้คำสั่งอิมพอร์ต (Import) และคำสั่งแอทแทช (Attach)โปรแกรมแอคเซสได้รับการออกแบบ ให้สามารถทำงานร่วมกับวินโดวส์ได้อย่างมีประสิทธิภาพและการใช้งานที่สะดวกง่ายดาย ผู้ใช้โปรแกรมแอคเซสสามารถสร้างคำสั่งเรียกค้นข้อมูลที่เรียกค้นยาก (Complex query) ได้อย่างง่ายด้วยการทำงานแบบ GQBE (Graphical query by example) และยังมีความสามารถในเชิง OLE (Object Linking Embedding) อย่างเต็มประสิทธิภาพด้วย

ภายในแฟ้มฐานข้อมูล (*.mdb) มีขนาดความจุสูงสุด 1 กิกะไบต์ ซึ่งแฟ้มฐานข้อมูลนี้มีลักษณะเหมือนเป็นที่รวมของออบเจค (Object) หลาย ๆ ชนิด แตกต่างจากโปรแกรมจัดการฐานข้อมูลอื่น ๆ เช่น dBASE หรือ Paradox จะจัดเก็บตารางฐานข้อมูลในรูปของไฟล์สกุล.DB หรือ .DBF ต่างหากจากแฟ้มแบบฟอร์มและแฟ้มรายงาน

MDI (Multiple Document Interface)

MDI ของโปรแกรมแอคเซสประกอบด้วย 6 ออบเจค (อ) ได้แก่

1 ตาราง (Table) เป็นที่เก็บข้อมูลสามารถจะเพิ่ม แก้ไข หรือลบข้มูลในแต่ละระเบียนโดยกระทำที่ตารางโดยตรงได้ นอกจากนี้ยังสามารถสร้างหรือแก้ไขโครงสร้างตารางที่ใช้เก็บข้อมูลและกำหนดชนิดของข้อมูล (Data type) ของแต่ละเขตข้อมูลได้ ซึ่งชนิดของข้อมูลนี้อาจเป็นรูปภาพได้ และยังกำหนดเงื่อนไขต่าง ๆ ของแต่ละเขตข้อมูลได้ด้วย เช่น กำหนดให้เขตข้อมูลนี้เป็นคีย์หลักหรือไม่ มีค่าอยู่ในช่วงใดมีค่าซ้ำกันได้หรือไม่ เป็นต้น ซึ่งถ้าเขตข้อมูลนี้ถูกกำหนดให้เป็นคีย์หลักแล้ว ในการแก้ไขข้อมูลระเบียนใด ๆเขตข้อมูลนี้จะมีค่าเป็นค่าว่างไม่ได้ตามกฎการบูรณภาพของเอนทิตี โปรแกรมแอคเซสจะควบคุมให้โดยอัตโนมัติ นอกจากนี้ยังสามารถกำหนดความสัมพันธ์ระหว่างตารางในฐานข้อมูลได้ ซึ่งโปรแกรมแอคเซสจะดูแลความถูกต้องของข้อมูลตามกฎบูรณภาพของการอ้างอิงได้ เช่น ถ้ามีการสร้างความสัมพันธ์ระหว่าง 2 ตารางและมีการลบข้อมูลในตารางหลัก ซึ่งมีข้อมูลที่ถูกอ้างอิงโดยอีกตารางหนึ่ง โปรแกรมแอคเซสจะขึ้นข้อความบอกให้ผู้ใช้ทราบว่าข้อมูลนี้ถูกอ้างอิงโดยอีกตารางหนึ่ง ถ้าทำการลบข้อมูลในตารางหลักจะทำให้เกิดความผิดพลาดขึ้น เป็นต้น

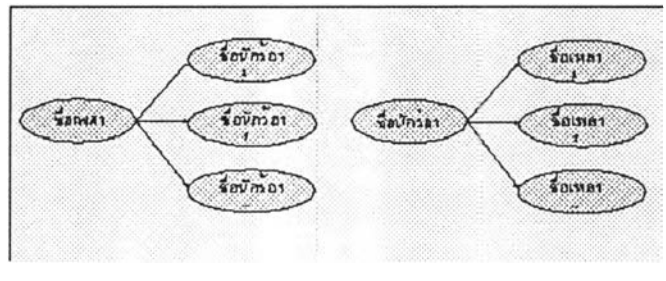
การกำหนดความสัมพันธ์ระหว่างตารางมี 2 ลักษณะ (7) คือ

1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One to One, 1:1 relationship) หมายความว่าในระหว่าง 2 ตารางนั้น เขตข้อมูลที่มีความสัมพันธ์ในแต่ละตาราง จะเป็นเขตข้อมูลที่มีความเด่นโดยเฉพาะจะไม่ซ้ำกันในแต่ละตาราง ความสัมพันธ์ที่เกิดขึ้นจะมีมากที่สุดเพียงหนึ่งเท่านั้น

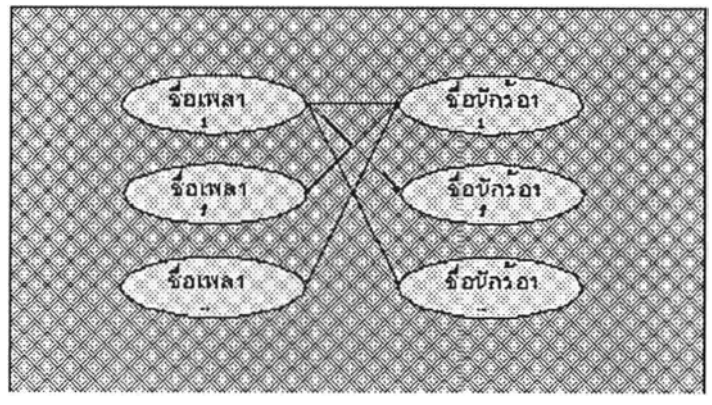
2 ความสัมพันธ์แบบหนึ่งต่อหลาย (One to Many, 1:N relationship) ความสัมพันธ์จะเกิดขึ้นในได้ตั้งแต่ไม่มีเลยจนถึงหลาย ๆ ครั้ง โดยแต่ละเอนทิตีลูก จะสัมพันธ์กับเอนทิตีแม่ได้หนึ่งค่า

เท่านั้น แต่หนึ่งค่าของเอนทิตีแม่สัมพันธ์กับเอนทิตีลูกได้หลายค่า ใช้สัญญลักษณ์ (P) –>> (C) แทน เช่น หมายเลขบริษัทและรหัสแผ่นเสียงหนึ่งมีรูปภาพจากปกประกอบได้หลายภาพ เช่น ปกหน้า ปกหลัง เป็นต้น แต่รูปภาพดังกล่าวมีหมายเลขบริษัทและรหัสแผ่นเสียงเพียงหนึ่งเท่านั้น

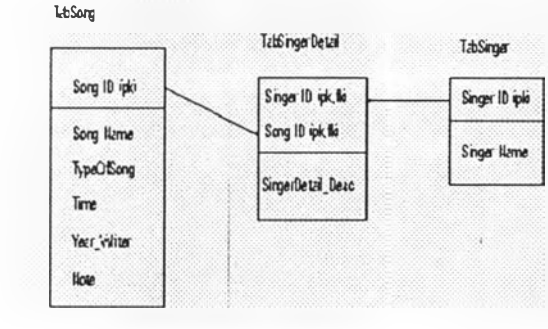
ในการออกแบบโมเดลข้อมูลเชิงตรรก จะพบว่าความสัมพันธ์ระหว่างตารางส่วนใหญ่มักจะเป็นบบหนึ่งต่อหลายและแบบหลายต่อหลาย (Many to many, M:N relationship) ซึ่งหมายถึงความสัมพันธ์จะเกิดขึ้นในได้ตั้งแต่ไม่มีเลยจนถึงหลาย ๆ ครั้งในทั้งสองทิศทางคือ แต่ละเอนทิตีลูกจะสัมพันธ์กับเอนทิตีแม่ได้หลายค่า และเอนทิตีแม่ก็สัมพันธ์กับเอนทิตีลูกได้หลายค่าเช่นกัน ใช้สัญญลักษณ์ (P) <<—–>> (C) เช่น เพลงหนึ่งเพลงมีผู้ร้องหลายท่าน และผู้ร้องแต่ละท่านก็สามารถร้องเพลงได้หลายเพลง เป็นต้น ความสัมพันธ์ในลักษณะนี้โปรแกรมแอคเซส ไม่สามารถสร้างได้ ดังนั้นผู้ออกแบบโมเดลข้อมูลจึงต้องแตกความสัมพันธ์แบบหนึ่งต่อหลายเป็น ความสัมพันธ์แบบหนึ่งต่อหลาย 2 ความสัมพันธ์ และกำหนดเอนทิตีขึ้นมาใหม่อีกเอนทิตีหนึ่งให้มีความสัมพันธ์กับเอนทิตีเดิมเป็นแบบหนึ่งต่อหลายเช่นเดียวกัน เช่น ความสัมพันธ์ระหว่างเพลงกับผู้ร้องแตกได้ โดยเพิ่มเอนทิตีรายละเอียดของผู้ร้องขึ้นมา ดังรูปที่3.3 จากนั้นให้ใส่ชื่อ ความหมาย รายละอียดของเอนทิตี และความสัมพันธ์ที่เกิดขึ้นใหม่นี้ลงในพจนานุกรมข้อมูลด้วย



รูปที่ 3.1 แสดงความสัมพันธ์แบบหลายต่อหลาย



รูปที่ 3.2 แสดงความสัมพันธ์แบบหลายต่อหลาย



รูปที่ 3.3 การแตกความสัมพันธ์แบบหลายต่อหลาย

แตกความสัมพันธ์แบบหลายต่อหลายให้เป็นความสัมพันธ์แบบหนึ่งต่อหลาย 2 ความสัมพันธ์ โดยเพิ่มเอนทิตีรายละเอียดผู้ร้อง

ในตอนล่างของจอภาพมีกล่องตรวจสอบ (Check Box) สามารถกำหนดให้โปรแกรมทำการควบคุมการอ้างอิงระหว่าง 2 ตาราง ซึ่งสามารถเลือกได้ว่าจะให้โปรแกรมทำการควบคุมหรือไม่ โดยปกติแล้วควรสั่งให้ควบคุม เพราะถ้าทำอะไรผิดพลาดกับตารางหนึ่ง โปรแกรมจะคอยกันไว้ไม่ให้ทำ โดยการขึ้นข้อความเตือนก่อนว่า ถ้าทำต่อไปความสัมพันธ์ระหว่างตารางอาจสูญเสียไปได้

2 คิวรี (Query) เป็นออบเจคที่สร้างจากตารางหนึ่งตารางหรือหลายตารางที่มีความสัมพันธ์สร้างขึ้นเพื่อใช้ในการเรียกค้นข้อมูล ภาษาฐานข้อมูลสำหรับโมเดลแบบสัมพันธ์ที่ใช้ในโปรแกรมแอคเซสมี 2 ภาษาด้วยกัน คือ

2.1 เอสคิวแอล

2.2 คิวบีอี เป็นภาษาที่มีการประสานกับผู้ใช้ได้ดี (User Interface) เนื่องจากใช้ง่าย สามารถกำหนดเงื่อนไขในการค้นหา เรียงลำดับเขตข้อมูลจากน้อยไปมาก หรือจากมากไปน้อย การหาผลรวมของเขตข้อมูล การสร้างตารางแจกแจงความถี่ได้หลายทาง (Crosstab Table) เป็นต้น และเมื่อสร้างเสร็จแล้วก็สามารถดูโปรแกรมที่เป็นเอสคิวแอลได้จากเมนู View / SQL

ในการสร้างคิวรีที่ใช้ตารางมากกว่าหนึ่งตารางขึ้นไป จะต้องสร้างความสัมพันธ์ระหว่างตารางขึ้นมาก่อน เมื่อสร้างคิวรีโปรแกรมจะสร้างเส้นโยงความสัมพันธ์ระหว่างตาราง ให้เห็นบนจอภาพอย่างชัดเจน สำหรับการเชื่อมโยงตาราง (JoinTable) นี้โดยปกติโปรแกรมจะกำหนดค่าโดยปริยาย (Default) ให้เป็น Equi Join แต่ผู้ใช้โปรแกรมสามารถกำหนดใหม่ให้เป็นแบบ Outer Join, Inner Join หรือ Self Join ได้ด้วย

3 ฟอร์ม (Form) เป็นออบเจคที่สร้างจากตาราง คิวรี หรือสร้างโดยไม่ใช้ออบเจคใดเลย สร้างขึ้นเพื่อให้ผู้ใช้งานในระดับปฏิบัติการใช้ดู แก้ไข และเพิ่มเติมข้อมูลตามที่ต้องการ ผู้ออกแบบฟอร์มสามารถจะกำหนดให้ผู้ใช้เห็นข้อมูลเฉพาะบางเขตข้อมูลได้ เพื่อดูแลในเรื่องความปลอดภัยของข้อมูลได้ง่าย และสามารถออกแบบให้การแสดงผลมีความสวยงาม และใช้งานง่าย

4 รายงาน (Report) เป็นการจัดรูปแบบการแสดงผลของตาราง หรือคิวรีที่ได้ออกแบบไว้แล้ว เพื่อพิมพ์ออกมาเป็นรายงาน รายงานนี้สามารถทำการคำนวณโดยจัดกลุ่มของระเบียนได้หลายระดับ ทั้งนี้จะคำนวณผลรวมและค่าเฉลี่ยได้ทุก ๆ ระดับ รวมทั้งสามารถวาดกราฟหรือรูปภาพในรายงานได้ด้วย

5 แมคโคร (Macro) เป็นชุดคำสั่งที่มีไว้สำหรับเขียนแอพพลิเคชันบนฐานข้อมูล ซึ่งมีคำสั่งอยู่จำนวนหนึ่ง สำหรับงานที่ไม่ซับซ้อน และต้องการความสะดวกรวดเร็ว ในการสร้าง

แอพพลิเคชัน การเขียนแมคโครทำให้ผู้ใช้พัฒนาโปรแกรมได้ง่าย สามารถที่เขียนไว้ในทริกเกอร์ของฟอร์มและรายงาน เพื่อความสะดวกไม่ต้องเขียนโปรแกรม เช่น ออกแบบฟอร์มให้มีการตรวจสอบเงื่อนไขบางอย่างก่อน ที่จะเปิดฟอร์ม (Open Form) สามารถเขียนแมคโครไว้ในพรอบเพอตี (Properties) ของฟอร์ม OnOpen แล้วตามด้วยชื่อของแมคโครดังนั้นก่อนที่จะเปิดฟอร์มนี้โปรแกรมจะทำงานตามลำดับคำสั่งของแมคโครดังกล่าว เป็นต้น

6 โมดูล(Module) เป็นออบเจคที่ใช้สำหรับสร้างแอพพลิเชันซึ่งสามารถสร้างคำสั่งที่ซับซ้อน ภาษาที่ใช้ในโมดูลเป็นภาษาของโปรแกรมแอคเซสเองเรียกว่า แอคเซสเบสิค (Access basic) ซึ่งภาษานี้คล้ายคลึงกับภาษา Visual basic เมื่อพัฒนาโปรแกรมจากภาษานี้ก็สามารถนำไปประยุกต์ใช้กับโปรแกรม Visual basic ทั่วไป รวมทั้งโปรแกรมอื่น ๆ อีกหลายโปรแกรม

นอกจากนี้โปรแกรมแอคเซสมีคำสั่งอัดแน่น (Compact) ซึ่งมีผลทำให้ประหยัดเนื้อที่ของฮาร์ดดิสก์ นั่นคือเมื่อทำการลบระเบียนออกจากตาราง ระเบียนเหล่านี้จะถูกทำเครื่องหมายว่าถูกลบแล้วแต่มิได้ถูกกำจัดออกไปจากตารางจริง การลบหรือเปลี่ยนแปลงการออกแบบ (Design)ของฟอร์ม และรายงาน ฯลฯ มักทำให้เกิดออบเจคกำพร้าในฐานข้อมูล เมื่อใช้คำสั่งอัดแน่นสามารถกำจัดออบเจคที่ไม่ต้องการออกไปจากฐานข้อมูลได้ทำให้ขนาดของแฟ้มข้อมูล (*.MDB)มีขนาดเล็กลง และทำให้การทำงานของโปรแกรมแอคเซสรวดเร็วขึ้นด้วย แต่ในการที่จะอัดแน่นฐานข้อมูลจะต้องมีพื้นที่ว่างพอ เพราะในตอนทำการอัดแน่นนั้นโปรแกรมแอคเซสจะสร้างแฟ้มข้อมูลขึ้นมาใหม่ ถึงแม้ว่าจะใช้ชื่อแฟ้มข้อมูลเดิมก็ตาม ในขณะที่ทำการอัดแน่นจะมีการสร้างแฟ้มข้อมูลขึ้นมาใหม่เสมอ เมื่อทำกระบวนการอัดแน่นเสร็จเรียบร้อยแล้ว แฟ้มข้อมูลใหม่จะไปแทนที่แฟ้มข้อมูลเก่า

ขั้นตอนการออกแบบฐานข้อมูล (Database design) (3) ประกอบด้วย

1 การออกแบบระดับสารสนเทศ (Information level desing) คือ การศึกษาและวิเคราะห์รวบรวมความต้องการของผู้ใช้ เพื่อที่จะกำหนดโครงสร้างและความต้องการสำหรับระบบสารสนเทศที่ต้องการทำขึ้น เพื่อที่จะได้ทราบรายละเอียดจุดประสงค์ในการใช้งานสามารถสนับสนุนการวางแผนและช่วยในการตัดสินใจ ดังนั้นเพื่อให้บรรลุวัตถุประสงค์ดังกล่าวต้องมีการสอบถาม รวบรวมเอกสาร รายงาน ผังองค์กร นโยบายต่าง ๆ ที่เกี่ยวข้อง นอกจากนี้ต้องสังเกตและศึกษา การดำเนินการตลอดจนกระบวนการต่าง ๆ ได้แก่ การวิเคราะห์ชนิด รายการเปลี่ยนแปลง (Transaction) และความถี่ที่เกิดขึ้น รวมทั้งกระแสการไหลของข้อมูล เป็นต้น
ข้อมูลที่ต้องมีการรวบรวมได้แก่

1 ข้อมูลที่ใช้เป็นข้อมูลในการนำเข้า

2 การค้นหาข้อมูลในทุกรูปแบบ

3 ลักษณะของรายงานทั้งหมด

4 การประมวลผลและการแก้ไขข้อมูลทั้งหมด

5 กฎเกณฑ์ข้อบังคับต่างๆ

ในการรวบรวมข้อมูลและความต้องการต่าง ๆ อาจจะได้มาในรูปแบบที่ยังไม่ดีนัก รวมทั้งภาพรวมของระบบย่อยเข้ามาเป็นโครงสร้างทั้งหมด อาจเกิดการซ้ำซ้อนหรือมีข้อขัดแย้งกันได้ ดังนั้นอาจใช้เทคนิคต่าง ๆ เพื่อแปลงให้อยู่ในรูปที่เข้าใจขึ้น ข้อมูลเหล่านี้จะถูกนำมาใช้ในการออกแบบฐานข้อมูลในระดับสารสนเทศ และเมื่อถึงเวลาของการออกแบบในระดับกายภาพต้องอาศัย ข้อมูลอื่น ๆ ที่เกี่ยวข้องกับการทำงานของระบบในระดับกายภาพอีกด้วย เช่น

1 จำนวนของแต่ละเอนทิตี

2 ความถี่ในการพิมพ์รายงาน

3 ความยาวของรายงานแต่ละรายงาน

4 กฎเกณฑ์ในการควบคุมความปลอดภัยในการใช้ข้อมูล

ผลที่ได้จากการออกแบบคือ โมเดลข้อมูลเชิงตรรกข้อมูลต่าง ๆ เหล่านี้จะถูกนำมาวิเคราะห์ พร้อมกับระบบฐานข้อมูลที่ออกแบบในระดับสารสนเทศเพื่อการออกแบบในระดับกายภาพ โดยในระดับนี้เราจะพิจารณาถึงความสามารถของระบบจัดการฐานข้อมูลเพื่อให้ผลการทำงานของระบบที่ออกแบบนี้สมบูรณ์แบบ และมีประสิทธิภาพสูงสุด

โมเดลข้อมูล (Data model)

เป็นแนวความคิดซึ่งใช้อธิบายโครงสร้างของฐานข้อมูล โดยโครงสร้างของฐานข้อมูลหมายความถึงชนิดของข้อมูล ความสัมพันธ์และข้อจำกัด ซึ่งใช้จัดการกับข้อมูลนอกจากนี้โมเดลข้อมูลจะรวมถึงการปฏิบัติการในการเรียกใช้ ค้นหา และแก้ไขบนฐานข้อมูลด้วยประเภทของโมเดลข้อมูลถูกจำแนกตามแนวความคิดของโครงสร้างของฐานข้อมูล แบ่งเป็น 2 ประเภท คือ

1 โมเดลข้อมูลเชิงมโนภาพ (Conceptual data models) หรือโมเดลข้มูลเชิงตรรก (Logical data models) หรือโมเดลระดับบน (High level data models) แนวคิดนี้จะเป็นในลักษณะที่ผู้ใช้รับรู้และเข้าถึงข้อมูล โครงสร้างของข้อมูลเชิงมโนภาพจะสะท้อนลักษณะข้อมูลของงานต่าง ๆ ในลักษณะที่ไม่ขึ้นกับฮาร์ดแวร์และซอฟท์แวร์ โมเดลระดับนี้จะใช้แนวคิดเกี่ยวกับเอนทิตีและความสัมพันธ์

2 โมเดลข้อมูลเชิงกายภาพ (Physical data models) หรือโมเดลข้อมูลระดับล่าง (Low level data models) โมเดลข้อมูลระดับนี้เป็นรายละเอียดในช่วงการเก็บข้อมูลลงสื่อบันทึกข้อมูล เช่น รูปแบบของระเบียน การเรียงลำดับระเบียนและวิถีทางในการเข้าถึงข้อมูล (Access path) การกู้ข้อมูล (Recovery) อย่างมีประสิทธิภาพ เป็นต้น

การออกแบบโมเดลข้อมูลเชิงตรรก (Logical Database Models Design)

เป็นการอธิบายความต้องการของผู้ใช้ รวมทั้งรายละเอียดของชนิดของข้อมูล ความสัมพันธ์และข้อกำหนดต่างๆ โครงสร้างจากมุมมองของผู้ใช้เป็นประโยชน์ในการเป็นข้อมูลนำเข้า (Input) ของการออกแบบฐานข้อมูล

ขั้นตอนในการออกแบบโมเดลข้อมูลเชิงตรรก (8)

LDM1 **กำหนดเอนทิตีหลัก** (Indentify major entities)

เอนทิตีเปรียบเสมือนกับเป็นคำนามได้แก่ บุคคล สถานที่ และสิ่งของซึ่งอาจเป็นสิ่งที่มีตัวตนหรือเป็นนามธรรมก็ได้ เอนทิตีของระบบฐานข้อมูลเพลงและดนตรี ได้แก่ แผ่นเสียง เพลง ผู้ร้อง ผู้แต่งทำนองเพลง เป็นต้น เมื่อรวบรวมได้เอนทิตีหลักแล้วจะต้องกำหนดชื่อและความหมายลงในพจนานุกรมข้อมูล และเขียนโมเดลข้อมูลด้วย การตั้งชื่อไม่ควรเกิน 20 ตัวอักษร

LDM2 **กำหนดความสัมพันธ์ระหว่างเอนทิตี** (Determine relationships between entities)

เป็นการกำหนดชื่อ ความหมาย ทิศทาง และขนาดอัตราส่วนที่เกิดจากความสัมพันธ์นั้นๆ และบันทึกลงพจนานุกรมข้อมูล การตั้งชื่อความสัมพันธ์ไม่ควรเกิน 20 ตัวอักษรอัตราส่วนและทิศทางของความสัมพันธ์เป็นพื้นฐาน ในการแบ่งประเภทของความสัมพันธ์ ซึ่งแบ่งได้เป็น 3 ประเภทด้วยกัน คือ

1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง

2 ความสัมพันธ์แบบหนึ่งต่อหลาย

3 ความสัมพันธ์แบบหลายต่อหลาย

LDM3 **กำหนดคีย์หลักและคีย์รอง**(Determine primary and alternate key)

คีย์หลักและคีย์รองจะเป็นแอตตริบิวแรกที่กำหนดในเอนทิตี การกำหนดคีย์หลักของทุกเอนทิตีเลือกจากแอตตริบิวที่เป็นคีย์เป็น (Candidate key) ได้ และให้ระบุคีย์รองของทุกเอนทิตีด้วย ในกรณีที่คีย์หลักและคีย์รองเป็นคีย์ประกอบ (Compound key) แอตตริบิวหนึ่งอาจเป็นส่วนของ

คีย์หลักและคีย์รองได้มากกว่าหนึ่งคีย์ สิ่งที่สำคัญอีกประการหนึ่งก็คือเอนทิตีที่เป็นซับไทป์จะต้องมีคีย์หลักเดียวกันกับเอนทิตีที่เป็นซูเปอร์ไทป์ของมัน หลังจากกำหนดแล้วให้ตั้งชื่อระบุในโมเดลข้อมูลเชิงตรรกและใส่ลงในพจนานุกรมข้อมูลด้วย

LDM4 กำหนดคีย์นอก (Determine foreign key)

สำหรับเอนทิตีที่มีความสัมพันธ์กัน จะต้องมีการกำหนดคีย์นอก คีย์นอกจะถูกกำหนดในเอนทิตีลูก และมีค่าเท่ากันกับคีย์หลักของเอนทิตีแม่ คีย์นอกมีความสำคัญคือทำให้เกิดกฎธุรกิจ (Business rules) ระหว่างเอนทิตีต่าง ๆ ทำให้ตรวจสอบได้ว่าเอนทิตีไหนเป็นเอนทิตีแม่หรือเอนทิตีลูก และทำให้ขั้นตอนการออกแบบฐานข้อมูลง่ายขึ้น พร้อมให้ตั้งชื่อและระบุในโมเดลข้อมูลเชิงตรรก และใส่ลงในพจนานุกรมข้อมูลด้วย

LDM5 กำหนดคีย์ของกฎธุรกิจ (Determine key business rules)

เป็นขั้นตอนที่ทำขึ้น เพื่อความสมบูรณ์ของข้อมูล และความถูกต้องตรงกันของค่าของข้อมูล กฎธุรกิจแบ่งได้เป็น 3 ประเภทคือ

1 กฎธุรกิจของคีย์ (Key business rules) เป็นการกำหนดเพื่อความสมบูรณ์ของความสัมพันธ์ ได้แก่ การเพิ่ม ลบ และแก้ไขข้อมูล

2 โดเมน (Domain) เป็นการกำหนดเพื่อความสมบูรณ์ของแอตตริบิว ซึ่งคือการกำหนดข้อบังคับและค่าที่เป็นไปได้ สำหรับแอตตริบิวทั้งที่เป็นคีย์และไม่ใช่คีย์

3 กฎการดำเนินการของทริกเกอร์ (Triggering operation) เป็นการกำหนดผลกระทบจากการเพิ่ม ลบ หรือดึงข้อมูลที่เกิดกับเอนทิตีอื่นภายในเอนทิตีเดียวกัน

LDM6 เพิ่มแอตตริบิวที่เหลือ (Add remaining attributes)

เป็นการกำหนดแอตตริบิวอื่น ๆ ในเอนทิตีที่ไม่ใช่คีย์เพิ่มเข้าไป โดยแอตตริบิวแต่ละตัวที่เพิ่มนั้นจะต้องขึ้นกับทั้งหมดของคีย์ในเอนทิตีนั้น ไม่ใช่ขึ้นกับบางส่วนของคีย์ และต้องขยายในเอนทิตีแม่ไม่ใช่ขยายในเอนทิตีลูกที่มีคีย์นั้นเป็นคีย์นอก นอกจากนั้นถ้าแอตตริบิวดังกล่าวขึ้นกับคีย์หลักทั้งหมดแล้วแต่มีค่ามากกว่าหนึ่งค่า (Multivalued) ให้แตกออกเป็นอีกหนึ่งเอนทิตีที่มีความสัมพันธ์กับเอนทิตีเดิมแบบหนึ่งต่อหลาย (1:N) พร้อมทั้งตั้งชื่อและระบุในโมเดลข้อมูลเชิงตรรกและใส่ลงในพจนานุกรมข้อมูลด้วย

LDM7 พิจารณาให้อยู่ในกฎนอร์มัลไลเซชัน(Validate normalization rules)

จากขั้นตอนต่าง ๆ ที่ผ่านมาเป็นการรวมแอตตริบิวเข้ามาในเอนทิตี ซึ่งการตรวจสอบว่าแอตตริบิวเหล่านั้นอยู่ในเอนทิตีที่เหมาะสมหรือไม่ ใช้เทคนิคการนอร์มัลไลเซชันโดยการวิเคราะห์แยกโครงสร้างข้อมูล ซึ่งประโยชน์ของการทำให้โมเดลที่ออกแบบอยู่ในรูปแบบนอร์มัล (Normal

form) คือ

1 ลดช่องว่างที่ต้องใช้ในการเก็บข้อมูล

2 ลดความซ้ำซ้อนของข้อมูล

3 ลดความผิดพลาด ความไม่ตรงกันของข้อมูลในฐานข้อมูล

4 ช่วยให้มีความยืดหยุ่นต่อความต้องการในการใช้งาน และทำให้สามารถออกแบบฐานข้อมูลได้กว้างขวางขึ้น

การทำนอร์มัลไลเซชันในระดับต่าง ๆ มี 5 ระดับ ซึ่งโดยปกติการออกแบบโมเดลข้อมูลให้อยู่ในรูปแบบนอร์มัลระดับที่ 3 (3NF) ก็ถือว่าเพียงพอแก่การนำไปใช้งานแล้ว วิธีการทำนอร์มัลไล-เซชัน ในระดับต่าง ๆ ตามลำดับดังนี้

1 ความสัมพันธ์นอร์มัลระดับที่ 1 (First normal form : 1NF) คือ การนำแอตตริบิวหรือกลุ่มของแอตตริบิวที่ซ้ำออกโดยการแยกแถวของข้อมูล ดังนั้นการทำนอร์มัลไลเซชันระดับที่ 1จะต้องมีการเพิ่มแอตตริบิวของคีย์เสมอ โดยสามารถกำหนดได้ว่าคีย์ตัวใหม่จะประกอบด้วยคีย์เดิมผนวกกับแอตตริบิวที่ถือเป็นคีย์หลักของกลุ่มที่ซ้ำ

2 ความสัมพันธ์นอร์มัลระดับที่ 2 (Second normal form : 2NF) จากทำให้ได้รูปแบบของความสัมพันธ์นอร์มัลระดับที่ 1 แล้วยังอาจมีปัญหาเนื่องจากเกิดความซ้ำซ้อนของข้อมูลได้ ลักษณะของปัญหาดังกล่าวทำให้เกิดปัญหาในการแก้ไข การเพิ่ม การลบและความขัดแย้งของข้อมูล ดังนั้นหลักการทำให้เป็น 2NF คือการขจัดการขึ้นต่อกันเพียงบางส่วน นั่นหมายถึงการทำให้ได้รูปแบบของความสัมพันธ์เป็น 1NF และไม่มีแอตตริบิวที่ไม่ใช่คีย์ตัวใดขึ้นกับส่วนใดส่วนหนึ่งของคีย์ การทำให้เป็น 2NF กระทำได้โดยการสร้างความสัมพันธ์ขึ้นมาใหม่

3 ความสัมพันธ์นอร์มัลระดับที่ 3 (Third normal form : 2NF) จากทำให้ได้รูปแบบของความสัมพันธ์นอร์มัลระดับที่ 2 แล้วยังอาจมีปัญหาเนื่องจากเกิดความซ้ำซ้อนของข้อมูลได้ลักษณะของปัญหาดังกล่าว ทำให้เกิดปัญหาในการแก้ไข การเพิ่ม การลบและความขัดแย้งของข้อมูล ปัญหาที่เกิดขึ้นเกิดจากมีแอตตริบิวขึ้นกับแอตตริบิวอื่นที่ไม่ใช่คีย์ ดังนั้นหลักการทำให้เป็น 3NF คือต้องไม่มีแอตตริบิวที่ไม่ขึ้นตรงกับคีย์หลัก

4 ความสัมพันธ์นอร์มัลระดับที่ 4 (Fourth normal form : 4NF) คือการทำให้ได้รูปแบบของความสัมพันธ์นอร์มัลระดับที่ 3 และไม่มีการขึ้นแก่กันแบบหลายค่าอยู่ในความสัมพันธ์

5 ความสัมพันธ์นอร์มัลระดับที่ 5 (Fifth normal form : 4NF) คือความสัมพันธ์ที่ไม่สามารถนำไปสร้างความสัมพันธ์ใหม่ ทำได้โดยการเชื่อมระหว่าง 2 ความ

สัมพันธ์ด้วยคีย์ต่างกันได้ โดยปกติมักจะเป็นปัญหาสำหรับคีย์หลักที่เป็นคีย์ประกอบ เป็นขั้นที่พิจารณาได้ยาก จะเกิดระเบียนข้อมูลใหม่ที่ไม่จริงขึ้นมาเมื่อนำเอนทิตีมารวมกัน

LDM8 กำหนดโดเมน (Determine domains)

เป็นการกำหนดกลุ่มของค่าที่เป็นไปได้สำหรับแต่ละแอตตริบิว ได้แก่

1 ชนิดของข้อมูล (Data type)

2 ความยาวของข้อมูล (Length)

3 รูปแบบของข้อมูล (Format)

4 ค่าที่อนุญาต (Allowable value)

5 ช่วงของข้อมูลหรือข้อกำหนดอื่นๆ

6 ความหมาย (Meaning) เป็นการอธิบายความหมายของแอตตริบิวว่าคืออะไร

7 ค่าความเป็นหนึ่งเดียว (Uniqueness)

8 การเป็นค่าว่าง (Null) ได้หรือไม่

9 ค่าที่กำหนด (Default value)

LDM9 กำหนดทริกเกอร์ดำเนินการ (Determine other attribute business rules triggering operation)

เป็นการกำหนดทริกเกอร์เพื่อตรวจสอบความถูกต้องเมื่อมีการเพิ่ม ลบ แก้ไขหรือดึงข้อมูล ข้อมูล ซึ่งจะพิจารณาผลรวมทั้งที่เกิดกับเอนทิตีอื่น และแอตตริบิวอื่นภายในเอนทิตีที่เรากระทำด้วย เมื่อกำหนดกฎการจัดการต่าง ๆ แล้ว ให้เก็บลงในพจนานุกรมข้อมูล โดยมีรูปแบบที่ประกอบด้วย เหตุการณ์ที่ทำ เช่น การเพิ่ม การลบ เป็นต้นเอนทิตี หรือแอตตริบิวที่เรากระทำด้วยเงื่อนไขที่กำหนดไว้ การกระทำที่จะต้องเกิดสืบเนื่องจากเหตุการณ์นั้น

LDM10 รวบรวมมุมมองของผู้ใช้ทั้งหมดเข้าด้วยกัน (Combine user views)

การรวมมุมมองของผู้ใช้เข้าด้วยกันเพื่อลดความซ้ำซ้อน ความไม่สอดคล้องของข้อมูลและการเพิ่มความสัมพันธ์ระหว่างมุมมองของผู้ใช้ ซึ่งประกอบด้วยการรวมเอนทิตีการรวมความสัมพันธ์และการรวมแอตตริบิว

LDM11 รวบรวมโมเดลที่มีอยู่แล้ว (Integrate with existing data modules)

เป็นการรวมโมเดลข้อมูลเชิงตรรกที่ได้กับของที่มีอยู่เดิม และพัฒนาโมเดลใหม่ควบคู่ไปกับการพิจารณากฎเกณฑ์ข้อบังคับของเดิม โดยอาจมีการใช้เอนทิตีหรือความสัมพันธ์ร่วมกับของเดิมและมีการกำหนดเอนทิตีขึ้นมาใหม่ด้วย

LDM12 **พิจารณาเสถียรภาพและการเติบโต** (Analyze fcr stability and growth)

เป็นการออกแบบโมเดลที่พิจารณาถึงสิ่งที่จะเกิดขึ้นในอนาคต เช่น

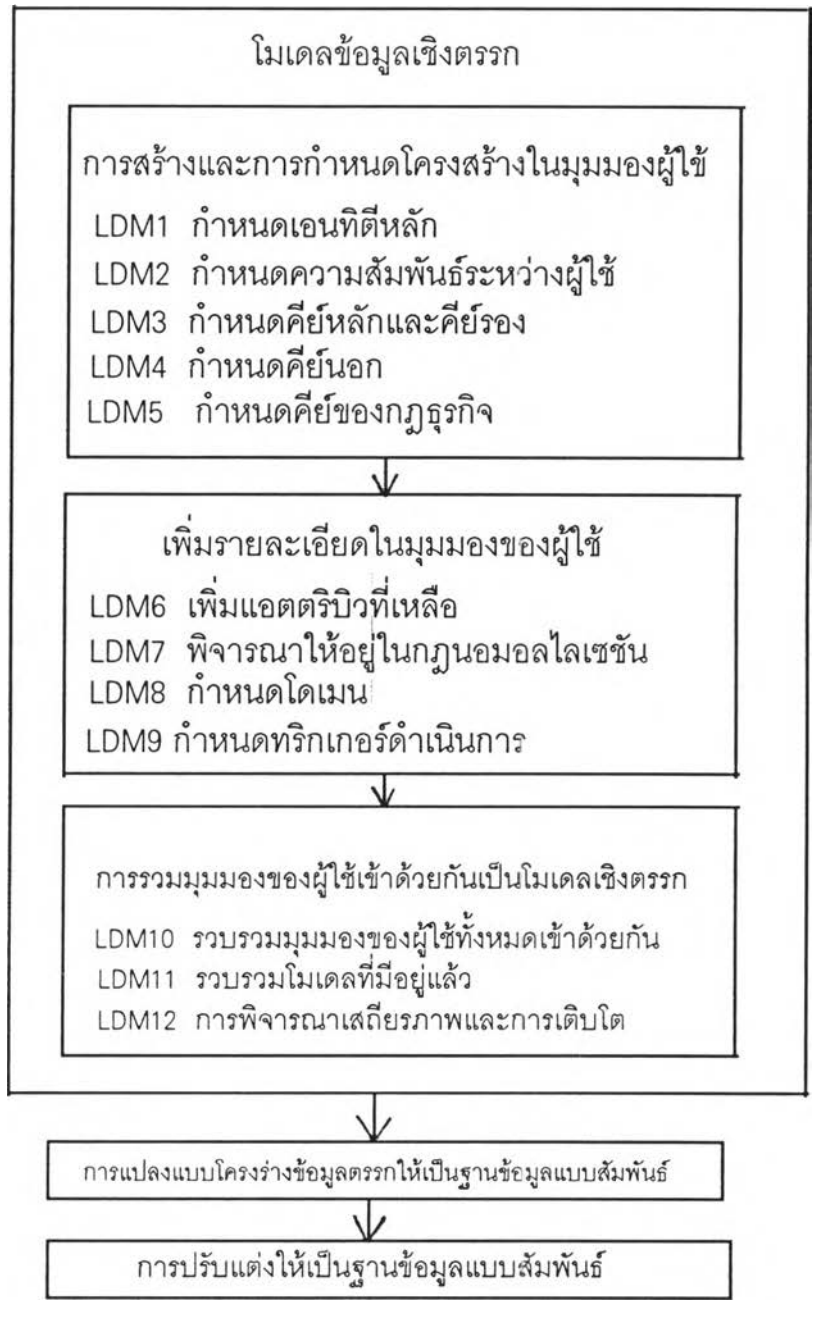
1 อาจมีเอนทิตีหรือความสัมพันธ์ใหม่ที่เกิดขึ้น ทำให้ต้องเพิ่มคีย์นอกภายในเอนทิตีของเดิม

2 ความสัมพันธ์แบบหนึ่งต่อหลาย อาจจะเปลี่ยนเป็นแบบหลายต่อหลายได้

3 คีย์หลักอาจเปลี่ยนไปเนื่องจากของเดิมไม่เป็นหนึ่งเดียวแล้ว

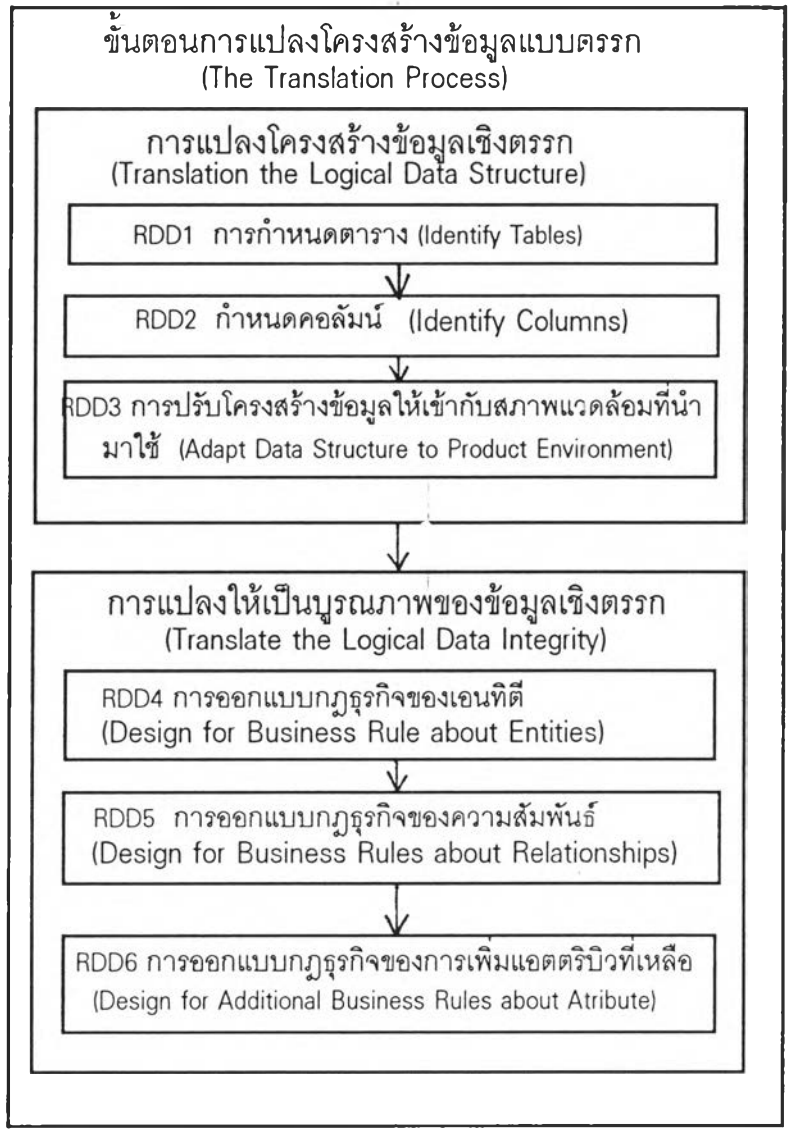
4 ฯลฯ

จากการออกแบบโมเดลเชิงตรรก ทำให้ได้แผนภาพแสดงความสัมพันธ์ระหว่างเอนทิตี ขั้นต่อมาคือ การออกแบบฐานข้อมูลแบบสัมพันธ์ ซึ่งสามารถแบ่งเป็นขั้นตอนต่าง ๆ ได้ ดังรูปที่ 3.5



รูปที่ 3.4 ขั้นตอนการออกแบบโมเดลข้อมูลเชิงตรรก

I 17135047



รูปที่ 3.5 ขั้นตอนการออกแบบฐานข้อมูลแบบสัมพันธ์

2 การออกแบบฐานข้อมูลทางกายภาพ (Physical level design) เป็นกระบวนการในการเลือกโครงสร้างในการเก็บข้อมูล และทิศทางการเข้าถึงข้อมูล สำหรับแฟ้มข้อมูลของฐานข้อมูล เพื่อให้ได้มาซึ่งฐานข้อมูลที่มีประสิทธิภาพมากที่สุด แต่ละระบบจัดการฐานข้อมูลจะมีระบบจัดการแฟ้มข้อมูล ทิศทางการเข้าถึงข้อมูล ดัชนีตัวชี้ ฯลฯ ที่ต่างกันออกไป ซึ่งจะเป็นตัวจำกัดให้เลือกสิ่งที่เหมาะสมกับแฟ้มข้อมูล

สิ่งที่พิจารณาในการออกแบบทางกายภาพ คือ

1 เวลาในการตอบสนอง (Response time) คือช่วงเวลาตั้งแต่การส่งงานเข้าไปถึงเมื่อได้รับผลลัพธ์ที่ต้องการออกมา

2 การใช้ที่ว่าง (Space utilization) จำนวนที่ว่างของหน่วยเก็บจะถูกใช้โดยแฟ้มข้อมูลของฐานข้อมูลและโครงสร้างทิศทางการเข้าถึงข้อมูล

3 งานที่ได้ออกมา (Transaction throght put) จะคิดเป็นค่าเฉลี่ยคำนวณจากจำนวนงานที่สามารถประมวลผลได้ โดยระบบจัดการฐานข้อมูลต่อหนึ่งหน่วยเวลา

ผลที่ได้จากการออกแบบขั้นตอนนี้คือ การตัดสินใจเบื้องต้นเกี่ยวกับโครงสร้างในการเก็บและวิธีการเข้าถึงข้อมูลของแฟ้มข้อมูลของแฟ้มข้อมูล