

บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้อง

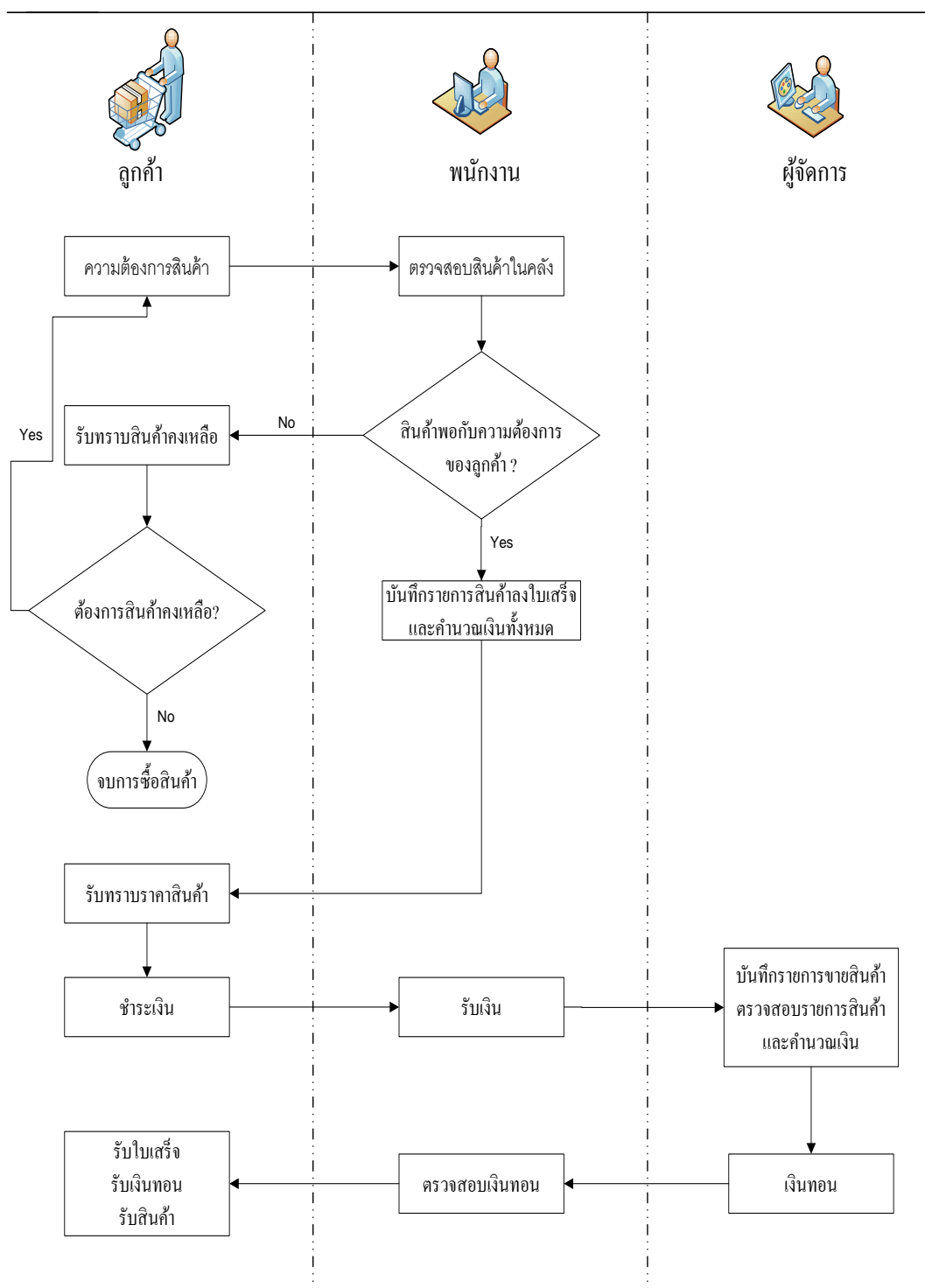
2.1 ธุรกิจร้านค้ากระเป๋าแฟชั่น กรณีศึกษาร้าน Baglisym

ร้านค้ากระเป๋าแฟชั่น Baglisym เปิดทำการมาแล้ว 4 ปี จากเดิมก่อนหน้านี้เป็นร้านค้าขนาดเล็กที่มีการทำธุรกิจในลักษณะค้าปลีกให้กับลูกค้าทั่วไป ต่อมาธุรกิจกระเป๋าของร้านค้า Baglisym เริ่มได้รับความนิยมเพิ่มขึ้นจากลูกค้าหลากหลายประเภท จึงได้เริ่มขยายขยายธุรกิจตัวเองโดยมีการค้าส่งให้กับผู้ที่สนใจนำสินค้าของทางร้านไปจำหน่ายต่อให้กับลูกค้าท่านอื่น และได้ทำการพัฒนาสินค้าของตัวเองให้มีความหลากหลายมากขึ้นให้มีความเหมาะสมกับการใช้งานในรูปแบบต่างๆ อาทิเช่น กระเป๋าเป้ กระเป๋าถือ กระเป๋าสะพายข้าง รวมถึงสินค้านำเข้าบางชนิด เป็นต้น ปัจจุบันทางร้านได้มีการดำเนินงานโดยอาศัยบุคลากรในการจัดการร้านค้า ไม่ว่าจะเป็น การขายสินค้า การสั่งซื้อสินค้า การรับสินค้าเข้า เมื่อลูกค้ามาสั่งซื้อสินค้า พนักงานต้องตรวจสอบจำนวนสินค้าให้พอกับความต้องการของลูกค้า และทำการจดบันทึกยอดขายในแต่ละวัน ซึ่งในปัจจุบันระบบงานของทางร้านไม่ได้มีการจัดการระบบงานได้ดีพอ จึงทำให้การทำงานเป็นไปได้อย่างล่าช้า และข้อมูลสินค้ามีการตกหล่น ทำให้เกิดข้อผิดพลาดในการดำเนินงานอยู่บ่อยครั้ง

การดำเนินงานของระบบจัดการบริหารร้านค้า กรณีศึกษาร้านค้ากระเป๋า
กระบวนการทำงานดังนี้

Baglisym มี

2.1.1 แผนผังการทำงานของระบบการขายสินค้าระบบปัจจุบัน



รูปที่ 2.1 แผนผังการทำงานของระบบการขายสินค้าระบบปัจจุบัน

2.2 ภาษา SQL

2.2.1 ความหมายของภาษา SQL

SQL ย่อมาจาก structured query language คือภาษาที่ใช้ในการเขียนโปรแกรมเพื่อจัดการกับฐานข้อมูลโดยเฉพาะ เป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์ และเป็นระบบเปิด (open system) หมายถึงเราสามารถใช้อคำสั่ง sql กับฐานข้อมูลชนิดใดก็ได้ และ คำสั่งงานเดียวกันเมื่อส่งงานผ่าน ระบบฐานข้อมูลที่แตกต่างกันจะได้ ผลลัพธ์เหมือนกัน ทำให้เราสามารถเลือกใช้ฐานข้อมูล ชนิดใดก็ได้โดยไม่ติดขัดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง นอกจากนี้แล้ว SQL ยังเป็นชื่อโปรแกรมฐานข้อมูล ซึ่งโปรแกรม SQL เป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้อคำสั่งเพียงไม่กี่คำสั่ง โปรแกรม SQL จึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์ และเป็นภาษาหนึ่ง ซึ่งแบ่งการทำงานได้เป็น 4 ประเภท ดังนี้

1. Select query ใช้ในการดึงข้อมูลในฐานข้อมูล จะมีการค้นหารายการจากตารางในฐานข้อมูล ตั้งแต่หนึ่งตารางขึ้นไปตามเงื่อนไขที่ตั้ง ผลลัพธ์ที่ได้จะเป็นเซตของข้อมูลที่สามารถสร้างเป็นตารางใหม่ หรือ ที่ใช้แสดงออก มาทางจอภาพเท่านั้น
2. Update query ใช้สำหรับแก้ไขข้อมูล โดยแก้ไขในคอลัมน์ที่มีตรงตามเงื่อนไข
3. Insert query ใช้สำหรับการเพิ่มข้อมูลใหม่ๆ เข้าไปในฐานข้อมูล
4. Delete query ใช้สำหรับลบข้อมูลออกไป

ปัจจุบันมีซอฟต์แวร์ระบบจัดการฐานข้อมูล (DBMS) ที่สนับสนุนการใช้อคำสั่ง SQL เช่น Oracle , DB2, MS-SQL, MS-Access นอกจากนี้ภาษา SQL ถูกนำมาใช้เขียนร่วมกับโปรแกรมภาษาต่างๆ เช่น ภาษา c/C++ , VisualBasic และ Java

2.2.2 ประโยชน์ของภาษา SQL

1. สร้างฐานข้อมูลและตาราง
2. สนับสนุนการจัดการฐานข้อมูลซึ่งประกอบด้วย การเพิ่ม การปรับปรุง และการลบข้อมูล
3. สนับสนุนการเรียกใช้หรือค้นหาข้อมูล

2.2.3 ประเภทของคำสั่ง SQL

คำสั่งในภาษา SQL ประกอบด้วยชุดคำสั่งหลักๆ 3 ประเภทด้วยกัน ได้แก่

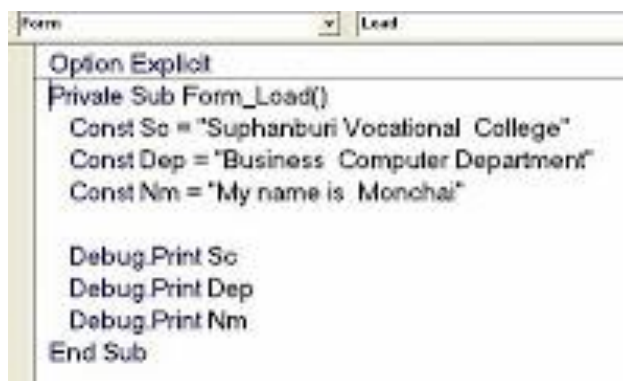
1. DDL (Data Definition Language) เป็นชุดคำสั่งที่ใช้ในการนิยาม กำหนด หรือสร้างข้อมูล เช่น table, index, view ได้แก่คำสั่ง create table, create view เป็นต้น
2. DML (Data Manipulation Language) เป็นชุดคำสั่งที่ใช้ในการประมวลผล หรือจัดการกับข้อมูลในฐานข้อมูล ได้แก่ คำสั่ง Select, insert, update เป็นต้น
3. DCL (Data Control Language) เป็นคำสั่งที่ใช้ในการควบคุมสิทธิ์ของผู้ใช้ในการใช้ข้อมูล รวมทั้งส่วนที่ใช้ควบคุมการใช้งานฐานข้อมูลจากผู้ใช้หลายๆ คนพร้อมกัน ได้แก่ คำสั่ง grant, revoke เป็นต้น

2.3 ภาษา Visual Basic.Net

2.3.1 ความหมายของภาษา Visual Basic.Net

วิชวลเบสิกคอตเน็ต (Visual Basic.NET) คือ เครื่องมือสำหรับพัฒนาโปรแกรม เป็นภาษาหนึ่งในกลุ่มไมโครซอฟท์วิชวลสตูดิโอ คอตเน็ต (Microsoft Visual Studio .NET) เป็นการโปรแกรมที่มีสภาพแวดล้อมแบบกราฟิกสำหรับระบบปฏิบัติการวินโดวส์ (Windows Operating System) โดยมีรากฐานภาษามาจากภาษาเบสิก และทำงานบน คอตเน็ตเฟรมเวิร์ค (Dotnet Framework) ถูกออกแบบให้มีความสามารถในการพัฒนาโปรแกรมเชิงวัตถุอย่างแท้จริง และรองรับการออกแบบด้วย ยูเอ็มแอล (UML = Unified Modeling Language)

วิชวลสตูดิโอคอตเน็ต (Visual Studio .NET) คือ เครื่องมือสำหรับพัฒนาโปรแกรมแบบครบวงจร เป็นการรวมเครื่องมือที่จำเป็นต่อการพัฒนาโปรแกรมอย่างครบถ้วน (IDE = Integrated Development Environment) ซึ่งรวมบริการการพัฒนาภาษาโปรแกรม บริการคลาสพื้นฐานให้นำมาใช้งานร่วมกันได้อย่างเป็นระบบ เช่น เอสคิวแอลเซิร์ฟเวอร์ (SQL Server) วิชวลเบสิกคอตเน็ต (VB.NET) วิชวลซีชาร์ป (VC#) วิชวลเจชาร์ป (VJ#) วิชวลซีพลัสพลัส (VC++) และ เอเอสพีคอตเน็ต (ASP .NET) เป็นต้น โดยทั้งหมดทำงานอยู่บนซีแอลอาร์ (CLR = Common Language Runtime) ที่รองรับการประมวลผลและเข้าใช้ทรัพยากรในเครื่องได้อย่างมีประสิทธิภาพ



```

Form                                Load
Option Explicit
Private Sub Form_Load()
    Const Sc = "Suphanburi Vocational College"
    Const Dep = "Business Computer Department"
    Const Nm = "My name is Monchai"

    Debug.Print Sc
    Debug.Print Dep
    Debug.Print Nm
End Sub

```

รูปที่ 2.2 ตัวอย่างการใช้ภาษา Visual Basic.Net

2.3.2 คุณสมบัติของภาษา Visual Basic.Net

เป็น เครื่องมือที่ช่วยในการพัฒนา Application บนระบบปฏิบัติการวินโดวส์ ทำได้โดยง่าย แม้มีใช้โปรแกรมเมอร์ก็สามารถสร้าง โปรแกรมได้ในเวลาอันรวดเร็ว จุดเด่นของ VB มีโครงสร้าง โค้ดที่เรียบง่ายมนุษย์ทำให้เรียนรู้ได้ง่าย มีเครื่องมือในการพัฒนา Application จำนวนมาก สามารถ สร้างไฟล์ .EXE ที่สามารถทำงานได้ด้วยตัวเอง ออกแบบการติดต่อกับผู้ใช้(ฟอร์ม) ได้ทันที พัฒนา Application ได้หลายแบบเช่น โปรแกรมด้านธุรกิจ ด้านอินเทอร์เน็ตและ Web Application หลักการพัฒนา Application ด้วย VB

- 1.เน้นการออกแบบส่วนการติดต่อกับผู้ใช้ แล้วค่อยเขียน โปรแกรมตามเหตุการณ์
- 2.ส่วนประกอบต่าง ๆ ใน Application ที่กำลังพัฒนา เรียกว่า วัตถุ (Object)
- 3.วัตถุ มีคุณสมบัติ ซึ่งสามารถปรับเปลี่ยนได้
- 4.Method ความสามารถที่กระทำกับวัตถุ จะกระทำโดยคำสั่งใน โปรแกรมเท่านั้น
5. การเขียนรหัสโปรแกรมฝังไว้ใน Object และจะทำงานได้ก็ต่อเมื่อเกิดเหตุการณ์ณ์กับ Object ตามที่โปรแกรมเมอร์กำหนดไว้ (Event-Driven)

ข้อดีของ Visual Basic.Net

1. รองรับ Optional argument ซึ่งสำคัญมากที่ต้องการใช้งานร่วมกับ ActiveX componen หรือการเขียนโค้ดชนกับพวก Office
2. ยอมรับการทำ late-binding ได้ ถ้าไม่กำหนด Option Strict On การเขียน โค้ดพวกนี้ใช้ กับพวก ActiveX อีกนั่นเอง
3. รองรับการทำ named indexer (การสร้าง property ที่มี argument)
4. มีคำสั่ง VB แบบเดิมๆ เช่น Left, Mid, UCase, ... ให้ใช้ สำหรับผู้ใช้ VB6 มาก่อน (การ เรียกใช้ฟังก์ชันแบบเดิมๆ นี้จะมีผลต่อประสิทธิภาพของโปรแกรม)

5. มีประโยค With..End With ให้ใช้
6. ความเรียบง่าย เช่นการสร้างประโยค Event
7. สามารถกำหนดชื่อเมธอดของการ implements interface ที่ต่างจากที่กำหนดไว้ใน interface ได้
8. มีประโยค Catch...When... ทำให้สามารถทำการ filter exception ด้วยเงื่อนไขได้นอกจากการ filter ด้วยชนิดของ exception เท่านั้น
9. Visual Studio .Net จะทำการ compile โค้ดในลักษณะ background ซึ่งช่วยเป็นข้อดีในโปรแกรมขนาดเล็ก

2.4 ระบบฐานข้อมูล (Database System)

2.4.1 ความหมายของระบบฐานข้อมูล

ฐานข้อมูล (Database) หมายถึง ชุดของข้อมูลที่มีความสัมพันธ์กันที่ถูกนำมาจัดเก็บไว้ด้วยกัน เพื่อให้สามารถใช้ข้อมูลเหล่านั้นร่วมกันได้อย่างรวดเร็วและมีประสิทธิภาพ การจัดเก็บข้อมูลจะมีประสิทธิภาพได้ก็ต่อเมื่อมีวิธีการจัดการข้อมูลที่ดี กล่าวคือ วิธีการจัดเก็บและค้นหาข้อมูลต้องเป็นไปอย่างมีประสิทธิภาพและรวดเร็ว โดยทั่วไปเมื่อข้อมูลมีขนาดใหญ่ขึ้น การสร้างฐานข้อมูลมักจะกระทำโดยใช้เครื่องหมายคอมพิวเตอร์มาช่วย เพื่อให้สามารถจัดเก็บและใช้ข้อมูลเหล่านั้นร่วมกัน ตลอดจนสามารถค้นหาได้อย่างรวดเร็ว

2.4.2 หน้าที่ของระบบจัดการฐานข้อมูล

หน้าที่ของระบบจัดการฐานข้อมูลสามารถแบ่งออกเป็นข้อๆ ได้ ดังต่อไปนี้

1. กำหนดและเก็บ โครงสร้างฐานข้อมูล (Define and Store Database Structure)
2. การเรียกใช้ข้อมูลจากฐานข้อมูล (Load Database) เมื่อมีการประมวลผลที่เกิดจากการทำงานของโปรแกรมประยุกต์ ระบบฐานข้อมูลจะทำการรับและเก็บข้อมูลที่ป้อนเข้ามาเอาไว้ในฐานข้อมูล เพื่อใช้ในการประมวลผลต่อไป
3. เก็บและดูแลข้อมูล (Store and Maintain Data) ข้อมูลในระบบฐานข้อมูลจะถูกเก็บรวบรวมไว้ด้วยกัน โดยมีระบบจัดการฐานข้อมูลเป็นผู้ดูแลรักษาข้อมูลนั้น
4. ประสานงานระหว่างระบบปฏิบัติการ (Operating System) ดังที่ได้ทราบกันอยู่แล้วว่าระบบปฏิบัติการเป็นโปรแกรมที่คอยควบคุมการทำงานของอุปกรณ์คอมพิวเตอร์

หรือโปรแกรมต่างๆ ในเครื่องคอมพิวเตอร์ และระบบการจัดการฐานข้อมูลก็จะทำหน้าที่ประสานงานกับระบบปฏิบัติการ เพื่อให้การทำงานเป็นไปอย่างถูกต้องตามที่ผู้ใช้ต้องการ ไม่ว่าจะเป็นการเรียกใช้ข้อมูล การแก้ไขข้อมูล หรือการออกรายงาน

5. ช่วยควบคุมความปลอดภัย (Security Control) ในระบบการจัดการฐานข้อมูล มีวิธีควบคุมเพื่อเป็นการป้องกันความเสียหายที่อาจเกิดขึ้น ได้กับฐานข้อมูลไม่ว่าจะเป็น การเรียกใช้หรือแก้ไขเปลี่ยนแปลงข้อมูลของผู้ใช้ในระบบ ผู้ใช้สามารถเรียกข้อมูลขึ้นมาทำการแก้ไขได้

6. การจัดทำข้อมูลสำรองและการกู้ข้อมูล (Backup and Recovery) ในระบบจัดการฐานข้อมูล จะจัดทำข้อมูลสำรองของฐานข้อมูลเอาไว้และเมื่อมีปัญหาเกี่ยวกับระบบฐานข้อมูล เช่น แฟ้มข้อมูลหาย ซึ่งอาจเกิดขึ้นเนื่องจากคิส์เสีย และการลบผิดแฟ้มข้อมูล หรือไฟไหม้ฮาร์ดดิสก์ ฯลฯ ระบบจัดการฐานข้อมูลจะใช้ระบบข้อมูลสำรองนี้ในการฟื้นฟูสภาพการทำงานของระบบให้กลับสู่ภาวะปกติได้

7. ควบคุมการใช้ข้อมูลพร้อมกัน (Concurrency Control) ในระบบคอมพิวเตอร์ที่ใช้อยู่ปัจจุบัน โปรแกรมการทำงานมักจะเป็นแบบผู้ใช้หลายคน (Multi User) จึงทำให้ผู้ใช้แต่ละคน สามารถเรียกใช้ข้อมูลได้พร้อมกัน โดยการระบบจัดการฐานข้อมูลที่มีคุณสมบัติ ควบคุมการใช้ข้อมูลพร้อมกันนี้ จะทำการควบคุมการใช้ข้อมูลพร้อมกัน ของผู้ใช้หลายคนในเวลาเดียวกันได้ โดยมีระบบการควบคุมที่ถูกต้องเหมาะสม เช่น ถ้าการแก้ไขข้อมูลนั้น ยังไม่เรียบร้อย ผู้ใช้อื่นจะไม่สามารถเรียกข้อมูลนั้นๆ ขึ้นมาทำงานใดๆ ได้ต้องรอจนกว่า การแก้ไขข้อมูล ของผู้ที่เรียกใช้ก่อนหน้านั้น เสร็จเรียบร้อยแล้วจึงจะสามารถเรียกข้อมูล นั้นไปใช้งานต่อได้ ทั้งนี้เพื่อป้องกันไม่ให้เกิดปัญหาการในเรียกใช้ข้อมูลที่ไม่ถูกต้อง

8. ควบคุมความบูรณภาพของข้อมูล (Integrity Control) ระบบจัดการฐานข้อมูลจะทำการควบคุมค่าของข้อมูลในระบบให้ถูกต้องตามที่ควรจะเป็น

9. จัดทำพจนานุกรมข้อมูล (Data Dictionary) ของระบบจัดการฐานข้อมูล มีการกำหนดโครงสร้างของกับฐานข้อมูลขึ้น เพื่อเป็นเอกสารหรือแหล่งข้อมูล เช่น ชื่อ แฟ้มข้อมูล ชื่อเขตข้อมูล เป็นต้น

2.4.3 ความสัมพันธ์ระหว่างเอนทิตี

ความสัมพันธ์ระหว่างเอนทิตี แบ่งออกเป็น 3 ประเภท คือ

1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-one Relationships) เป็นการแสดงความสัมพันธ์ของข้อมูลในเอนทิตีหนึ่งที่มีความสัมพันธ์กับข้อมูลในอีกเอนทิตีหนึ่ง ในลักษณะหนึ่งต่อหนึ่ง (1 : 1)

2. ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One-to-many Relationships) เป็นการแสดงความสัมพันธ์ของข้อมูลในเอนทิตีหนึ่ง ที่มีความสัมพันธ์กับข้อมูลหลาย ๆ ข้อมูลในอีกเอนทิตีหนึ่ง ในลักษณะ (1:m)

3. ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many-to-many Relationships) เป็นการแสดงความสัมพันธ์ของข้อมูลสองเอนทิตีในลักษณะกลุ่มต่อกลุ่ม (m:n)

ตัวอย่าง เอนทิตีใบสั่งซื้อแต่ละใบจะสามารถสั่งสินค้าได้มากกว่าหนึ่งชนิด ความสัมพันธ์ของข้อมูลจากเอนทิตีใบสั่งซื้อไปยังเอนทิตีสินค้า จึงเป็นแบบหนึ่งต่อกลุ่ม (1:m) ในขณะที่สินค้าแต่ละชนิดจะถูกสั่งอยู่ในใบสั่งซื้อหลายใบ ความสัมพันธ์ของข้อมูลจากเอนทิตีสินค้าไปยังเอนทิตีใบสั่งซื้อจึงเป็นแบบหนึ่งต่อกลุ่ม (1:n) ดังนั้นความสัมพันธ์ของเอนทิตีทั้งสอง จึงเป็นแบบกลุ่มต่อกลุ่ม (m:n)

2.4.4 การนอร์มัลไลซ์ (Normalization)

หลังจากที่ผู้ออกแบบได้ขอบเขตข้อมูลทั้งหมดที่ต้องการเก็บแล้ว ซึ่งโดยส่วนมากเกิดจากรูปแบบรายงานบ้าง รูปแบบใบเสร็จรับเงินบ้าง รูปแบบใบส่งสินค้าบ้าง โดยมากมักจะเหมารวมเอวานั้นคือ รูปแบบของตารางที่ต้องการเก็บข้อมูล ซึ่งสิ่งเหล่านั้นนำมาซึ่งความซ้ำซ้อนของข้อมูล และทำให้มีขนาดใหญ่เกินหน่วยความจำ ปัญหาของรีเลชันที่เกิดขึ้นเหล่านี้สามารถขจัดได้ด้วย “ขบวนการ Normalization” ซึ่งแนวคิดนี้ถูกคิดค้นโดย E.F.Codd ซึ่งเป็นกระบวนการที่นำเค้าร่างของ Relation มาทำให้อยู่ในรูปแบบที่เป็นบรรทัดฐาน (Normal Form) เพื่อให้แน่ใจว่าการออกแบบเค้าร่างของ Relation เป็นการออกแบบที่เหมาะสม

ให้ทำการตรวจสอบเอนทิตีต่าง ๆ ให้อยู่ในกฎนอร์มัลไลเซชันซึ่งประกอบด้วย 1NF, 2NF, 3NF, BCNF, 4NF, 5NF ซึ่งจะได้กล่าวในหัวข้อต่อไป โดยมีประโยชน์ดังนี้

1. ลดที่ว่างที่ต้องใช้ในการเก็บข้อมูล
2. ลดความผิดพลาดความไม่ตรงกันของข้อมูลในฐานข้อมูล
3. ลดการเกิดอนอร์มัลไลของการลบและแก้ไขข้อมูล
4. เพิ่มความคงทนแก่โครงสร้างฐานข้อมูล

ในทางปฏิบัติการทำ นอร์มอลไลซ์จะเริ่มจาก E-R Model ก่อน แล้วจึงทำการ Map จาก E-R Model เป็น Relation แบบ 1NF ก่อน โดยให้ Attributes ที่เกี่ยวข้องกันจะอยู่ในตารางเดียวกัน สำหรับ Application ใหญ่ ๆ มี Attributes ประมาณ 500 Attributes ใช้ E-R Model จะได้ 1NF ประมาณ 80 ตาราง เมื่อทำ ถึง 5NF จะได้ไม่เกิน 100 ตาราง ในกรณีได้ตารางเป็นนอร์มัลไลเซชันที่สมบูรณ์แล้วสิ่งที่ต้องระวังคือ ไม่แตกตารางนั้นย่อยลงไปอีก

2.4.5 ระดับนอร์มัลไลเซชัน

นอร์มัลไลเซชันเป็นกระบวนการเพื่อพัฒนาการเชื่อมต่อของข้อมูลเพื่อแก้ปัญหาของรีเลชันที่ว่ากรออกแบบฐานข้อมูลทั้งทางตรรกะและทางกายภาพที่ได้ออกมาใช้ได้หรือยัง การนอร์มัลไลเซชันแบ่งออกได้เป็นหลายระดับดังนี้

1. First Normal Form (1NF) ทุก ๆ Field ในแต่ละ Record จะเป็น Single value นั่นคือในตารางหนึ่ง ๆ จะไม่มีค่าของกลุ่มข้อมูลที่ซ้ำกัน (Repeating Group) หรือเป็นการจัดแอตทริบิวหรือกลุ่มแอตทริบิวที่ซ้ำกัน ไปอยู่ในเอนทิตีลูก เพื่อแต่ละรายการในเอนทิตี จะไม่มีค่าของแอตทริบิวหรือค่าของกลุ่มแอตทริบิวที่ซ้ำกัน

2. Second Normal Form (2NF) ต้องเป็น First Normal Form (1NF) และต้องมี Key (บางตำราอาจจะเรียกว่า Index) ที่ทุก Non-key จะต้องขึ้นอยู่กับ Key นี้ และมีเพียง Key เดียวในหนึ่งตารางซึ่งเรียกว่า Primary Key การที่ทุกตาราง (Table) ต้องมี Key ก็เพราะเราต้องการให้แน่ใจว่าทุกข้อมูลใน Record ต่าง ๆ สามารถค้นหาได้โดยใช้ Key สรุปก็คือ นอร์มัลไลเซชันระดับที่ 2 (Second normal form : 2NF) เป็นการการจัดแอตทริบิวที่ไม่ขึ้นกับทั้งส่วนของคีย์หลักออกไป เพื่อให้แอตทริบิวอื่นทั้งหมดขึ้นตรงกับส่วนที่เป็นคีย์หลักทั้งหมดเท่านั้น

3. Third Normal Form (3NF) นอร์มัลไลเซชันระดับที่ 3 (Third normal form : 3NF) คือ ขบวนการที่พยายามขจัดสภาพของ Transitive Dependency ออกไป โดยมีข้อกำหนดว่าต้องเป็น Second Normal Form (2NF) และ ไม่มี Transitive dependence หรือเป็นการจัดแอตทริบิวที่ไม่เป็นคีย์ที่ขึ้น (Transitive dependent) ตรงกับแอตทริบิวอื่นที่ไม่ใช่คีย์หลักออกไปเพื่อให้ แอตทริบิวที่ไม่ใช่คีย์หลักต้องขึ้นตรงกับทั้งส่วนที่เป็นคีย์ และไม่ขึ้นกับแอตทริบิวอื่นที่ไม่ใช่คีย์หลัก

4. BCNF (Boyce/Codd Normal Form) นอร์มัลไลเซชันระดับที่ 4 ต้องเป็น 3NF และไม่มี Attribute อื่นในรีเลชันที่สามารถระบุค่าของ Attribute ที่เป็นคีย์หลัก หรือส่วนหนึ่งของคีย์หลักในกรณีที่คีย์หลักเป็นคีย์ผสม โดยทั่วไปรูปแบบ BCNF จะอยู่ในรูปแบบ 3NF แต่ไม่จำเป็นเสมอไปที่รูปแบบ 3NF จะอยู่ในรูปแบบ BCNF ทั้งนี้เนื่องจากรูปแบบนี้เป็นการขยายขอบเขตของรูปแบบ 3NF ให้เหมาะสมยิ่งขึ้น โดยรูปแบบที่ต้องทำให้เป็น BCNF โดยจะมีคุณสมบัติ คือ เป็นรีเลชันที่มีคีย์คู่แข่งหลายคีย์ (Multiple Candidate Key) โดยที่คีย์คู่แข่งเป็นคีย์ผสม (Composite Key) และคีย์คู่แข่งนั้นมีบางส่วนซ้ำซ้อนกัน (Overlapped) หรือมี Attribute บางตัวร่วมกันอยู่

5. Forth Normal Form (4NF) ต้องอยู่ในรูปแบบ BCNF และเป็นรีเลชันที่ไม่มีความสัมพันธ์ในการระบุค่าของ Attribute แบบหลายค่าโดยที่ Attribute ที่ถูกระบุค่าเหล่านี้ ไม่มีความสัมพันธ์กัน (Independently Multivalued Dependency)

6. Fifth Normal Form (5NF) หรือเรียกว่า Project-Join Normal Form (PJ/NF) โดยจะต้องอยู่ในรูปแบบ 4NF และไม่มี Symmetric Constraint กล่าวคือหากมีการแตกรีเลชันออกเป็นรีเลชันย่อย (Projection) และเมื่อทำการเชื่อมโยงรีเลชันย่อยทั้งหมด (Joint) จะไม่ก่อนให้เกิดข้อมูลใหม่ที่ไม่เหมือนรีเลชันเดิม (Spurious Tuples) ในการแตกรีเลชันออกมาจากรูปแบบ 4NF นั้น ถ้าเชื่อมโยงรีเลชันย่อยนั้นใหม่ หากไม่มีข้อมูลที่แตกต่างไปจากรีเลชันเดิมสามารถแตกรีเลชันนั้นได้ แต่ถ้าหากแตกเป็นรีเลชันย่อยแล้วเกิดข้อมูลไม่เหมือนกับรีเลชันเดิม ก็ไม่ควรแตกรีเลชัน และให้ถือว่ารีเลชันเดิมอยู่ใน 5NF แล้ว

2.5 UML (Unified Modeling Language)

ในการพัฒนาระบบด้วยหลักการ OOAD (Object Oriented Analysis and Design) นั้น สิ่งหนึ่งที่จำเป็นอย่างหนึ่งคือการสร้างแบบจำลองของอ็อบเจกต์ คลาส และองค์ประกอบอื่นๆของระบบ ซึ่งการถ่ายทอดแบบจำลองออกมาให้แก่ผู้มีส่วนเกี่ยวข้องกับระบบได้รับทราบนั้น ทางที่จะทำให้เข้าใจได้ตรงกันมากที่สุดคือ การแสดงในรูปแบบของสัญลักษณ์ที่มองเห็นได้ เช่น รูปภาพ แผนภาพ เป็นต้น โดยเครื่องมือที่ได้รับการยอมรับและเป็นที่ยอมรับใช้มากที่สุดคือ UML (Unified Modeling Language) ซึ่ง UML จัดได้ว่าเป็นภาษาหนึ่ง เพราะมีหน่วยของภาษาคอมพิวเตอร์ กล่าวคือ มีทั้งคำศัพท์ และไวยากรณ์ที่ชัดเจน

แผนภาพของภาษา UML นั้น ถูกแบ่งออกเป็น 2 กลุ่ม รวมทั้งหมด 9 แผนภาพ โดยกลุ่มที่หนึ่งเป็นกลุ่มแผนภาพที่แสดงให้เห็นโครงสร้างเชิงสถิติของระบบ (Structural Diagram) คือ โครงสร้างในส่วนที่ไม่มีการเปลี่ยนแปลงหรือเคลื่อนไหวแม้จะมีเหตุการณ์ใดๆเกิดขึ้น ได้แก่ Class Diagram, Object Diagram, Component Diagram และ Deployment Diagram และอีกกลุ่มหนึ่งคือ กลุ่มแผนภาพที่แสดงให้เห็นภาพเชิงกิจกรรมของระบบ (Behavioral Diagram) คือแสดงให้เห็นถึงพฤติกรรมของระบบที่มีการเปลี่ยนแปลงไปเมื่อมีเหตุการณ์ใดๆเกิดขึ้น และแสดงให้เห็นถึงความสามารถของระบบที่ดำเนินการในหน้าที่บางอย่างได้ ได้แก่ Use Case Diagram, Sequence Diagram, Collaboration Diagram, State chart Diagram และ Activity Diagram

2.5.1 Use Case Diagram

โดยทั่วไปในขั้นตอนการวิเคราะห์และออกแบบระบบตามแนวคิดเชิงวัตถุ มักจะใช้ Use Case Diagram มาเป็นเครื่องมือในการจำลองหน้าที่ของระบบที่ผู้ใช้ต้องการ เนื่องจาก Use Case Diagram เป็นแผนภาพที่ใช้แสดงถึงขั้นตอนการทำงานที่สำคัญของระบบ หรือแสดงหน้าที่และงานที่ระบบจะต้องปฏิบัติ เพื่อตอบสนองต่อผู้กระทำต่อระบบ

2.5.2 Object Diagram

เป็นแผนภาพที่ใช้ในการแสดงกลุ่มของออบเจกต์และความสัมพันธ์ระหว่างออบเจกต์ที่เกิดขึ้นในคลาสต่างๆของ Class Diagram

2. 5.3 Class Diagram

เป็นแผนภาพที่แสดงให้เห็นถึงกลุ่มของคลาส โครงสร้างของคลาส และ Interface ตลอดจนแสดงความสัมพันธ์ระหว่างคลาส การเริ่มต้นสร้าง Class Diagram นั้นส่วนใหญ่จะต้องค้นหาออบเจกต์ใน Use Case Diagram ก่อนซึ่งเทคนิคที่ใช้ในการค้นหาจะแตกต่างกันออกไปตามประสบการณ์ของทีมงาน เช่น ค้นหาจากคำอธิบายรายละเอียดของ Use Case โดยชื่อคลาสหาได้จากคำนาม ส่วนคุณสมบัติ (Attribute) หาได้จากคำคุณศัพท์ และการดำเนินการ (Operation) สามารถหาได้จากคำกริยา เป็นต้น

2.5.4 Component Diagram

เป็นแผนภาพที่แสดงให้เห็นถึงโครงสร้างทางกายภาพของโปรแกรม ประกอบด้วยส่วนประกอบต่างๆ ที่เรียกว่า Component ซึ่งหมายถึง ส่วนประกอบย่อยของซอฟต์แวร์ที่ใช้ในระบบ ทำให้เห็นว่าประกอบไปด้วยไฟล์ใดบ้าง ส่วนใหญ่มักจะแสดงไฟล์ที่เป็น Source Code, Binary Code ได้แก่ .exe .dll เป็นต้น

2.5.5 Deployment Diagram

เป็นแผนภาพที่แสดงโครงสร้างทางด้านฮาร์ดแวร์ของระบบงาน ส่วนใหญ่จะใช้ร่วมกับ Component Diagram โดยการมองอุปกรณ์ฮาร์ดแวร์ทั้งหมดเป็นออบเจกต์หรือคลาสได้เช่นเดียวกัน

2.5.6 Sequence Diagram

เป็นแผนภาพที่แสดงให้เห็นถึงการปฏิสัมพันธ์ (Interaction) ระหว่างออบเจกต์ โดยเฉพาะการส่ง Message ระหว่างออบเจกต์ตามลำดับเวลาที่เหตุการณ์ขึ้น โดยจะมีสัญลักษณ์ให้เห็นลำดับการส่ง Message ตามเวลาส่งอย่างชัดเจน

2.5.7 Collaboration Diagram

เป็นแผนภาพที่แสดงให้เห็นถึงปฏิสัมพันธ์ (Interaction) ระหว่างออบเจกต์เช่นเดียวกับ Sequence Diagram แต่ต่างกันตรงที่ในส่วนของ Collaboration Diagram จะไม่มีสัญลักษณ์แสดงถึงลำดับการส่ง Message อย่างชัดเจน แต่จะเน้นส่วนของการแสดงถึงความสัมพันธ์ระหว่างออบเจกต์ตามลักษณะการทำงาน

2.5.8 State-chart Diagram

เป็นแผนภาพที่แสดงให้เห็นถึงปฏิสัมพันธ์ (Interaction) ระหว่างอ็อบเจกต์ เช่นเดียวกัน แต่ Statechart Diagram จะเน้นที่การแสดงให้เห็นถึงสถานะ (State) การเปลี่ยนสถานะ (Transition) ที่มีต่อเหตุการณ์ (Event) ที่เกิดขึ้นในช่วงชีวิตของอ็อบเจกต์ 1 ช่วง

2.5.9 Activity Diagram

เป็นแผนภาพที่แสดงให้เห็นถึงลำดับการดำเนินการกิจกรรม (Activity) จากกิจกรรมหนึ่งไปยังกิจกรรมหนึ่งซึ่งเกิดจากการทำงานของอ็อบเจกต์ภายในระบบ