

## บทที่ 2

### ปัญหา ขอบเขต และทฤษฎีที่เกี่ยวข้อง

#### 2.1 ปัญหาที่พบ

- (1) การจัดเก็บข้อมูลข้อมูลของสาขาวิชาวิศวกรรมซอฟต์แวร์อยู่ในรูปของเอกสาร
- (2) พื้นที่จัดเก็บข้อมูลอัปขึ้น ขาดการจัดวางเป็นระบบระเบียบ
- (1) เอกสารหลักฐานทำให้เกิดการเสียหาย ส่งผลให้ข้อมูลสูญหาย
- (2) เกิดความความวุ่นวายและผิดพลาด ในการจัดการข้อมูล
- (3) สืบค้นข้อมูลยุ่งยาก หลายขั้นตอน และต้องใช้บุคลากรจำนวนมาก
- (4) ใช้เวลาในการจัดการระบบนักศึกษามาก

#### 2.2 ขอบเขตของระบบ

##### 2.2.1 ความต้องการหลัก (Functional Requirement)

- (1) ระบบสามารถเพิ่มทะเบียนประวัตินักศึกษาได้
- (2) ระบบสามารถปรับปรุงแก้ไขทะเบียนประวัตินักศึกษาได้
- (3) ระบบสามารถลบทะเบียนประวัตินักศึกษาได้
- (4) ระบบสามารถแสดงรายงานข้อมูลประวัตินักศึกษาได้
- (5) ระบบสามารถแสดงรายงานใบรายชื่อของนักศึกษาตามกลุ่มเรียนได้

##### 2.2.2 ความต้องการไม่หลัก (NON-Functional Requirement)

- (1) ระบบนำเสนอเมนูคำสั่งทั้งแบบเมนูบาร์ และเมนูพูลบาร์
- (2) ระบบกำหนดช่องทางการติดต่อแจ้งปัญหา
- (3) ระบบกำหนดเมนูแสดงคู่มือการใช้งาน

#### 2.3 ทฤษฎีที่เกี่ยวข้อง

##### 2.3.1 Agile model

Agile model เป็นโมเดลที่ออกแบบให้มีความรวดเร็ว ยืดหยุ่น พร้อมทั้งจะรับกับความเปลี่ยนแปลง เพื่อลดความเสี่ยงในการพัฒนาซอฟต์แวร์ โดยการแบ่งการพัฒนาออกเป็น iteration โดยแบ่งเวลาออกเป็นช่วง ๆ แต่ละช่วงยาวนานไม่มากนัก ไม่เกิน 1 เดือน หรือ 4 สัปดาห์ การพัฒนาจะดำเนินการอย่างต่อเนื่อง ถึงแม้ว่ามีอะไรมากระทบก็ไม่สนใจ เมื่อเกิดความเปลี่ยนแปลงก็จะพัฒนาให้สามารถรองรับกับความเปลี่ยนแปลงนั้นได้ อย่างไม่มีข้อจำกัดตายตัวในการพัฒนาจะเน้นการพูดคุยกันในทีมงานและผู้ใช้มากกว่าเน้น process หรือ tools การทำงานจะยึดที่ผลผลิตหรือตัวซอฟต์แวร์

เป็นหลัก ไม่ค่อยเน้นการจัดทำเอกสาร เน้นที่ความสัมพันธ์ของทีมงานและการสื่อสารเป็นหลัก เพื่อให้ได้ความต้องการมาครบถ้วนและพร้อมที่จะยอมรับความเปลี่ยนแปลงเพิ่มเติมของความต้องการ

หัวใจของ Agile model ได้แก่ เน้นความพึงพอใจของลูกค้า โดยการส่งมอบซอฟต์แวร์ให้ลูกค้าอย่างต่อเนื่องทุก 2 สัปดาห์ ยินดียอมรับความต้องการที่เปลี่ยนแปลงเสมอ ทีมพัฒนาระบบจะดำเนินโครงการที่ไวด์ของลูกค้ามีการพบหน้ากันทุกวันจนกว่า โครงการจะเสร็จ มีการประชุมพบหน้ากันสม่ำเสมอ ทีมงานมีอำนาจในการตัดสินใจเต็มที่ วัดความก้าวหน้าของงานกันที่ตัวซอฟต์แวร์การทำงานใช้กระบวนการที่ไม่หือหาว เน้นความคุณภาพชีวิตของทีมงาน มีเทคนิคต่าง ๆ ก็นำมาแลกเปลี่ยนกัน เน้นเทคนิคการออกแบบที่ง่าย ไม่ซับซ้อน ทำให้บำรุงรักษาปรับเปลี่ยนระบบได้ง่าย

Agile model อาจเป็นส่วนขยายของกระบวนการพัฒนาซอฟต์แวร์อื่น ๆ ที่มีอยู่เดิม โดยใช้ Agile เข้าไปกำกับและเลือกเอาส่วนที่สำคัญ ๆ กิจกรรมไหนควรทำ ไม่ควรทำ แล้วนำมาจัดลำดับให้เหมาะสม สำหรับวิธีของ Agile การที่ควรรู้จัก ได้แก่ Agile UP, XP-eXtream programming, FDD-Feature Driven Development, และ Scrum

ก) ทฤษฎีของสัจยบรรณ agile ที่สำคัญบางส่วนมีดังนี้

- ลูกค้ามีความพึงพอใจอย่างรวดเร็ว และส่งมอบซอฟต์แวร์ที่ใช้ประโยชน์ได้เสมอ
- ซอฟต์แวร์ที่พัฒนาถูกส่งมอบอย่างสม่ำเสมอ(รายอาทิตย์ หรือ รายเดือน)
- ให้ทำซอฟต์แวร์ที่พัฒนาถูกวัดความคืบหน้าด้วยทฤษฎี
- มีความร่วมมือกันระหว่างผู้พัฒนาและผู้ใช้อย่างสม่ำเสมอ
- คุยกันแบบต่อหน้าเป็นการสื่อสารถึงความต้องการได้ดีที่สุด
- ความต้องการไม่ว่าจะมาช้าแค่ไหนก็ยินดีเพิ่มให้ในโปรแกรม
- มีความร่วมมือกันระหว่างผู้พัฒนาและผู้ใช้อย่างสม่ำเสมอ
- โครงการถูกสร้างโดยแรงผลักดันของคนๆ เดียว และคนนั้นต้องเชื่อถือได้
- เอาใจใส่ในเทคนิคและการดีไซน์สม่ำเสมอ
- ง่ายต่อความเข้าใจ
- ทีมที่พัฒนามีความอิสระ
- รองรับการเปลี่ยนแปลงเสมอ

การพัฒนาแบบ Agile นี้เป็นลูกผสมระหว่าง XP กับ UP มี Model มากกว่า XP และมีเอกสารน้อยกว่า UP ซึ่งอยู่ในแนวของ predictive ซึ่งจะเกิด iteration ค่อนข้างมาก (เนื่องจากความต้องการของ user จะมาเมื่อไรก็ได้) และเน้นการสื่อสารแบบเข้าไปคุยกันต่อหน้า นอกจากจะได้ความต้องการที่ตรงแล้วรายละเอียดเล็กๆ น้อยๆ ก็อาจจะได้เพิ่มด้วย รวมถึงการที่มีความสัมพันธ์ที่ดีกับลูกค้าด้วย

วิธีการพัฒนาแบบ Agile ช่วยลดความเสี่ยงในการพัฒนาซอฟต์แวร์ในระยะสั้น ในระหว่างการพัฒนาซอฟต์แวร์หนึ่งช่วงเวลาจะมีความทำงานซ้ำซึ่งมีตั้งแต่ 1 ถึง 4 สัปดาห์ การทำซ้ำแต่ละครั้ง คือ การทำหมดทั้งโครงการซอฟต์แวร์ ประกอบด้วย การวางแผน การวิเคราะห์ความต้องการ การออกแบบ การเขียนโปรแกรม การทดสอบ และการทำคู่มือ การทำซ้ำไม่สามารถรับรอง

ผลิตภัณฑ์ที่ออกสู่ตลาด แต่จุดมุ่งหมายคือการลดข้อบกพร่องเมื่อจบการทำงานซ้ำ การสิ้นสุดของการทำซ้ำแต่ละครั้ง ทีมงานต้องทำการประเมินผลโครงการอีกครั้ง

วิธีการ Agile ให้ความสำคัญกับการสื่อสารต่อกันโดยตรงมากกว่าการบันทึกลงในเอกสาร ส่วนมากที่ทีมงาน Agile ก่อตั้งสำนักงานเดี่ยว บางครั้งอ้างอิงถึง bullpen ซึ่งอย่างน้อยประกอบด้วย โปรแกรมเมอร์ และลูกค้าของพวกเขา (ลูกค้า กำหนดผลิตภัณฑ์พวกเขาอาจจะเป็นผู้จัดการผลิตภัณฑ์นักวิเคราะห์ธุรกิจ หรือ client) สำนักงานประกอบด้วย ผู้ทดสอบ นักออกแบบการทำงานร่วมกัน ผู้เขียนทางเทคนิค และผู้จัดการ

วิธีการ Agile ยังเน้นการทำงานของซอฟต์แวร์ ขณะที่วัดความก้าวหน้าพื้นฐาน ร่วมกับความพึงพอใจในการติดต่อสื่อสารต่อกันโดยตรง วิธีการ Agile ทำให้เกิดการบันทึกในเอกสารน้อยมากที่เกี่ยวกับวิธีการอื่นทำให้ผลลัพธ์ในการประเมินผลของวิธีแบบ Agile อยู่นอกเหนือความควบคุม

นิยามวิธีพัฒนาซอฟต์แวร์ แบบ agile ได้พัฒนาในกลางปี 1990 s ในส่วนที่เป็นเหตุการณ์ตอบสนองของ heavyweight method ให้เป็นแบบอย่างการควบคุมอย่างหนัก แบ่งเป็นหมวดหมู่ จัดการผลโดยใช้แบบจำลองการพัฒนาแบบ Waterfall การเริ่มพัฒนาจากการใช้แบบจำลองของ Waterfall เห็นในการราขการที่ล่าช้า ขาดความน่าเชื่อถือ และมีความขัดแย้งกับแนวทางของวิศวกรรมซอฟต์แวร์ที่มีผลต่อการทำงานที่เกิดขึ้น กรณีที่สามารถทำวิธีการพัฒนาได้ไวและทำซ้ำได้เป็นผลการปฏิบัติที่พัฒนาในประวัติศาสตร์ของการพัฒนาซอฟต์แวร์ยุคแรก ในขั้นต้นวิธีแบบ Agile เรียกกันว่า lightweight method ในปี 2001 สมาชิกที่มีชื่อเสียงของชุมชน Snowbird รัฐ Utah มีมติเปลี่ยนชื่อเป็น Agile method ต่อมาบางคนได้สร้าง Agile Alliance เป็นองค์กรไม่แสวงกำไรเพื่อประชาสัมพันธ์การพัฒนาแบบ Agile

วิธีการที่คล้ายกับ Agile ที่สร้างก่อนปี 2000 ประกอบด้วย Scrum( 1986) , Crystal Clear, Extreme Programming (1996) , Adaptive Software Development, Feature Driven Development, และ DSDM ( 1995)

#### ข) วัตถุประสงค์ของ Agile

- เน้นว่าใครกันต่ออะไร และการพูดคุยสื่อสารกัน มากกว่าการยึดติดที่เครื่องมือและกระบวนการ เช่นเปลี่ยนให้โปรแกรมเมอร์ไปคุยกับลูกค้าแทนลูกค้าบอกอะไรมาก็ทำตามนั้น

- ให้ทำงานโดยยึดที่ผลผลิตหรือซอฟต์แวร์เป็นหลัก เช่น เดิมเน้นเอกสารแต่ Agile ไม่คำนึงถึงมากนัก แต่จะให้ความสำคัญที่ว่าเรามี s/w หรือของส่งให้ลูกค้าได้ตรงตามความต้องการหรือไม่

- ให้ความสำคัญเรื่องของการติดต่อสื่อสาร เช่น เดิมมีสัญญาหรือ contact กันแต่ Agile ไม่สนใจ ให้มองที่ความสัมพันธ์ระหว่างผู้พัฒนาและลูกค้า

- ยอมรับความเปลี่ยนแปลง เช่น เดิมต้องวางแผนให้ครบเป็นอย่างดี และทำตามแผน (Gantt chart) ให้ได้ แต่ Agile ไม่ต้องทำตามแผนแต่เน้นการสนองความเปลี่ยนแปลงที่เกิดขึ้นได้

### 2.3.2.ฐานข้อมูล (Database)

ฐานข้อมูลหมายถึง กลุ่มของข้อมูลที่มีความสัมพันธ์กัน นำมาเก็บรวบรวมเข้าไว้ด้วยกัน อย่างมีระบบและข้อมูลที่ประกอบกันเป็นฐานข้อมูลนั้น ต้องตรงตามวัตถุประสงค์การใช้งานขององค์กรด้วยเช่นกัน เช่น ในสำนักงานก็รวบรวมข้อมูล ตั้งแต่หมายเลขโทรศัพท์ของผู้ที่มาติดต่อจนถึง การเก็บเอกสารทุกอย่างของสำนักงาน ซึ่งข้อมูลส่วนนี้จะมีส่วนที่สัมพันธ์กันและเป็นที่ต้องการนำออกมาใช้ประโยชน์ต่อไปภายหลัง ข้อมูลนั้นอาจจะเกี่ยวกับบุคคล สิ่งของสถานที่ หรือเหตุการณ์ใด ๆ ก็ได้ที่เราสนใจศึกษา หรืออาจได้มาจากการสังเกต การนับหรือการวัดก็เป็นได้ รวมทั้งข้อมูลที่เป็นตัวเลข ข้อความ และรูปภาพต่าง ๆ ก็สามารถนำมาจัดเก็บเป็นฐานข้อมูลได้ และที่สำคัญข้อมูลทุกอย่างต้องมีความสัมพันธ์กัน เพราะเราต้องการนำมาใช้ประโยชน์ต่อไปในอนาคต

ระบบฐานข้อมูล (Database System) หมายถึง การรวมตัวกันของฐานข้อมูลตั้งแต่ 2 ฐานข้อมูลเป็นต้นไปที่มีความสัมพันธ์กัน โดยมีวัตถุประสงค์เพื่อเป็นการลดความซ้ำซ้อนของข้อมูล และทำให้การบำรุงรักษาตัวโปรแกรมง่ายมากขึ้น โดยผ่านระบบการจัดการฐานข้อมูล หรือเรียกย่อ ๆ ว่า DBMS

รศ. ยุพิน ไทยรัตนานนท์ ( 2540 : 202 ) ได้กล่าวไว้ว่า ระบบฐานข้อมูล คือ ระบบการจัดเก็บข้อมูลด้วยคอมพิวเตอร์โดยมีวัตถุประสงค์เพื่อบำรุงรักษาข้อมูล (Maintain information) และสามารถนำข้อมูลเหล่านั้นมาใช้ได้ทุกเมื่อที่ต้องการ

รศ.ศิริลักษณ์ โรจนกิจอำนวย (2540: 12) หน้าที่ของระบบจัดการฐานข้อมูลหน้าที่ของระบบจัดการฐานข้อมูลสามารถแบ่งออกเป็นข้อย่อยๆ ได้ ดังต่อไปนี้

ก) ช่วยกำหนดและเก็บโครงสร้างฐานข้อมูล (Define and Store Database Structure)

ข) การเรียกใช้ข้อมูลจากฐานข้อมูล (Load Database) เมื่อมีการประมวลผลที่เกิดจากการทำงานของโปรแกรมประยุกต์ ระบบฐานข้อมูลจะทำการรับและเก็บข้อมูลที่ป้อนเข้ามาเอาไว้ในฐานข้อมูล เพื่อใช้ในการประมวลผลต่อไป

ค) เก็บและดูแลข้อมูล (Store and Maintain Data) ข้อมูลในระบบฐานข้อมูลจะถูกเก็บรวบรวมไว้ด้วยกัน โดยมีระบบจัดการฐานข้อมูลเป็นผู้ดูแลรักษาข้อมูลนั้น

ง) ประสานงานกับระบบปฏิบัติการ (Operating System) ดังที่ได้ทราบกันอยู่แล้วว่า ระบบปฏิบัติการเป็นโปรแกรมที่คอยควบคุมการทำงานของอุปกรณ์คอมพิวเตอร์หรือโปรแกรมต่างๆ ในเครื่องคอมพิวเตอร์ ระบบการจัดการฐานข้อมูลก็จะทำหน้าที่ประสานงานกับระบบปฏิบัติการ เพื่อให้การทำงานเป็นไปอย่างถูกต้องตามที่ผู้ใช้ต้องการ ไม่ว่าจะเป็นการเรียกใช้ข้อมูล การแก้ไขข้อมูล หรือการออกรายงาน

จ) ช่วยควบคุมความปลอดภัย (Security Control) ในระบบการจัดการฐานข้อมูล จะมีวิธีควบคุมเพื่อเป็นการป้องกันความเสียหายที่อาจจะเกิดขึ้นได้กับฐานข้อมูล ไม่ว่าจะเป็นการเรียกใช้ หรือแก้ไขเปลี่ยนแปลงข้อมูลของผู้ใช้ในระบบ ผู้ใช้สามารถเรียกข้อมูลขึ้นมาทำการแก้ไขได้แตกต่างกัน เป็นต้น

ฉ) การจัดทำข้อมูลสำรองและการกู้ (Backup and Recovery) ในระบบจัดการฐานข้อมูลจะจัดทำข้อมูลสำรองของฐานข้อมูลเอาไว้ และเมื่อมีปัญหาเกี่ยวกับระบบฐานข้อมูล เช่น

แฟ้มข้อมูลหาย ซึ่งอาจเกิดขึ้นเนื่องจากดิสก์เสีย ลบผิดแฟ้มข้อมูล หรือไฟไหม้ ฯลฯ ระบบจัดการฐานข้อมูลจะใช้ระบบข้อมูลสำรองนี้ในการฟื้นฟูสภาพการทำงานของระบบให้สู่ภาวะปกติได้

ข) ควบคุมการใช้ข้อมูลพร้อมกัน (Concurrency Control) ในระบบคอมพิวเตอร์ที่ใช้อยู่ปัจจุบัน โปรแกรมการทำงานมักจะเป็นแบบผู้ใช้หลายคน (Multi User) จึงทำให้ผู้ใช้แต่ละคนสามารถเรียกใช้ข้อมูลได้พร้อมกัน ระบบจัดการฐานข้อมูลที่มีคุณสมบัติควบคุมการใช้ข้อมูลพร้อมกันนี้จะทำการควบคุมการใช้ข้อมูลพร้อมกันของผู้ใช้หลายคนในเวลาเดียวกันได้ โดยมีระบบการควบคุมที่ถูกต้องเหมาะสม เช่น ถ้าการแก้ไขข้อมูลนั้นยังไม่เรียบร้อย ผู้ใช้อื่นๆ ที่ต้องการเรียกใช้ข้อมูลนั้นจะไม่สามารถเรียกข้อมูลนั้นๆ ขึ้นมาทำงานใดๆ ได้ ต้องรอนจนกว่าการแก้ไขข้อมูลของผู้ที่เรียกใช้ข้อมูลนั้นก่อนจะเสร็จเรียบร้อย จึงจะสามารถเรียกข้อมูลนั้นไปใช้งานต่อได้ ทั้งนี้เพื่อป้องกันไม่ให้เกิดปัญหาการใช้ข้อมูลที่ไม่ถูกต้อง

ค) ควบคุมความบูรณภาพของข้อมูล (Integrity Control) ระบบจัดการฐานข้อมูลจะทำการควบคุมค่าของข้อมูลในระบบให้ถูกต้องตามที่ควรจะเป็น

ง) จัดทำพจนานุกรมข้อมูล (Data Dictionary) ระบบจัดการฐานข้อมูลจะทำการสร้างพจนานุกรมข้อมูลขึ้นมาให้เมื่อมีการกำหนดโครงสร้างของกับฐานข้อมูลมา เพื่อเป็นเอกสารหรือแหล่งข้อมูล เช่น ชื่อ แฟ้มข้อมูล ชื่อเขตข้อมูล เป็นต้น

จ. ฌรณต กั้วกั้ววาล ( 2521 : 1 ) ได้กล่าวไว้ว่า ซอฟต์แวร์ที่ใช้ในการสร้างและจัดการฐานข้อมูล (Database) เป็นเพียงเครื่องมือใช้ทำงานเท่านั้น สิ่งที่สำคัญกว่าคือ คุณจะต้องเริ่มต้นด้วยการออกแบบระบบการใช้ข้อมูลอย่างระมัดระวัง ถ้าระบบที่ออกแบบขึ้นมาไม่ดีพอ จะทำให้การทำงานในองค์กรล่าช้าขึ้นเชื่อถือในฐานข้อมูลไม่ได้และพนักงานทุกคนในองค์กรก็จะรู้สึกอึดอัดขัดใจที่จะใช้ระบบ รูปแบบของระบบที่ดีจะมีผลทำให้ระบบนั้นคงอยู่ได้ เพราะเป็นระบบที่ง่ายต่อการใช้งานและตรงตามความต้องการขององค์กร

### 2.3.3.เอสคิวแอล (Structured Query Language:SQL)

เอสคิวแอล คือ ภาษาที่ใช้ในการเข้าถึงข้อมูลในระบบจัดการฐานข้อมูล ทั้งโดยการโต้ตอบโดยตรงหรือเขียนโปรแกรมติดต่อกับระบบจัดการฐานข้อมูล เพื่อสร้าง ค้นหา ปรับปรุงหรือลบข้อมูลในระบบจัดการ ฐานข้อมูลนั้น

ระบบจัดการฐานข้อมูลปัจจุบันมีอยู่หลากหลายระบบซึ่งเหมาะสำหรับการใช้งานต่างๆ กันไป มีทั้งระบบจัดการฐานข้อมูลขนาดใหญ่ เช่น Oracle, Informix, Microsoft SQL Sever, Sybase หรือระบบฐานข้อมูลขนาดกลางและเล็ก เช่น Interbase, Paradox, MYSQL, DBase หรือ Microsoft Access เป็นต้น ในการติดต่อกับระบบฐานข้อมูล ไม่ว่าจะเป็นการสร้างข้อมูลใหม่ ค้นหาข้อมูล ปรับปรุงหรือลบข้อมูลที่มีอยู่ ทั้งในเชิงโต้ตอบกับระบบจัดการฐานข้อมูลโดยตรง หรือเป็นการเขียนโปรแกรมติดต่อกับระบบจัดการฐานข้อมูล จะใช้ภาษาสอบถามเชิงโครงสร้างนี้ ในการ เข้าถึงข้อมูลต่างๆ ซึ่งระบบจัดการฐานข้อมูลทั้งหลายที่มีอยู่ในปัจจุบันได้มีการปรับปรุง ภาษา สอบถามเชิงโครงสร้างของตัวเองขึ้นภายใต้มาตรฐานภาษาสอบถามเชิงโครงสร้างข้อมูล ของ ANSI (American National Standards Institute) และ ISO ข้อดีของภาษาสอบถามเชิงโครงสร้างคือเป็นภาษาที่ง่ายในการเรียนรู้

ประเภทของคำสั่งของภาษาสอบถามเชิงโครงสร้าง ประเภทคำสั่งของภาษาสอบถามเชิงโครงสร้างแบ่งตามการดำเนินการกับข้อมูลใน ฐานข้อมูลนั้นเป็น 4 ประเภท ดังต่อไปนี้

ภาษาควบคุมข้อมูล หรือ Data Control Language (DCL) เป็นคำสั่งที่ใช้ในการควบคุมการเข้าถึงข้อมูล ตัวอย่างเช่น ผู้จัดการสามารถดูข้อมูลทั้งหมดของลูกค้าได้ แต่พนักงานขายสามารถดูได้แค่ชื่อบริษัท ชื่อผู้ติดต่อ เบอร์โทรศัพท์ของลูกค้าได้เท่านั้น คำสั่งนี้ผู้ใช้ทั่วไป ไม่จำเป็นต้องรู้ มีเพียงแค่ผู้ดูแลฐานข้อมูลเท่านั้นที่ใช้คำสั่งประเภทนี้ติดต่อกับระบบจัดการ ฐานข้อมูล ตัวอย่างของคำสั่งประเภทนี้เช่น คำสั่ง GRANT และ ALTER USER เป็นต้น

ภาษานิยามข้อมูล หรือ Data Definition Language (DDL) เป็นคำสั่งที่ใช้ในการกำหนดและเปลี่ยนแปลงโครงสร้างของข้อมูลที่เก็บไว้ในฐานข้อมูล คำสั่งประเภทนี้ก็เช่นเดียวกับ Data Control Language ผู้ใช้โดยทั่วไปไม่จำเป็นต้องใช้ ตัวอย่างเช่น หากต้องการสร้างตารางข้อมูลลูกค้าจะใช้ คำสั่ง CREATE TABLE ซึ่ง TABLE จะมีลักษณะคล้ายกับตารางที่เก็บข้อมูลลูกค้าไว้ การสร้างตารางจะต้องกำหนดว่ารายละเอียดของตารางมีอะไรบ้าง เช่น ชื่อบริษัท ชื่อลูกค้า ที่อยู่ เบอร์โทรศัพท์ เบอร์โทรสาร เป็นต้น หากต้องการลบ ตารางข้อมูลลูกค้าที่สร้างไว้ทั้ง สามารถทำได้โดยใช้ คำสั่ง DROP TABLE นอกจากนั้นเมื่อสร้างตารางข้อมูลลูกค้าขึ้นแล้ว หากต้องการปรับปรุงหรือเพิ่มเติมรายละเอียดของตารางก็สามารถทำได้โดยใช้คำสั่ง ALTER TABLE เป็นต้น

ภาษาจัดการข้อมูล หรือ Data Manipulation Language (DML) เป็นคำสั่งที่ใช้ในการจัดการข้อมูลที่เก็บไว้ในฐานข้อมูลทั้งหมด เช่น การเพิ่ม ค้นหา ปรับปรุงและลบข้อมูลที่มีอยู่ในฐานข้อมูล คำสั่งประเภทนี้ผู้ใช้ที่ติดต่อกับฐานข้อมูลส่วนใหญ่จำเป็นต้องรู้ตัวอย่างเช่น ผู้จัดการต้องการเพิ่มข้อมูลลูกค้าใหม่ จะใช้คำสั่ง Insert หรือต้องการค้นหา ข้อมูลสินค้าที่ลูกค้าแต่ละรายสั่งซื้อ จะใช้คำสั่ง Select ต้องการปรับปรุงข้อมูลลูกค้าที่มีอยู่ก็จะใช้คำสั่ง Update ถ้าหากต้องการลบข้อมูลลูกค้า จะใช้คำสั่ง Delete เป็นต้น

คำสั่งควบคุมรายการเปลี่ยนแปลง หรือ Transaction Control Operation (TCL) เป็นคำสั่งของภาษาสอบถามเชิงโครงสร้างที่ใช้ในการควบคุมรายการเปลี่ยนแปลงที่เกิดขึ้น (Transaction) เมื่อใช้ภาษาจัดการข้อมูลทำการเปลี่ยนแปลง ทั้งเพิ่ม ปรับปรุง หรือลบข้อมูลใน ฐานข้อมูลนั้น

ตัวอย่างเช่น หากทำการเปลี่ยนแปลงข้อมูล ในฐานข้อมูลแล้ว แต่ต้องการยกเลิกการเปลี่ยนแปลงนั้น จะใช้คำสั่ง Rollback เพื่อให้ยกเลิกการเปลี่ยนแปลงทั้งหมดที่เกิดขึ้นให้กลับไปสู่สถานะก่อนหน้าที่จะเปลี่ยนแปลง หรือหากทำการเปลี่ยนแปลงข้อมูลในฐานข้อมูลแล้ว ต้องการยืนยันการเปลี่ยนแปลงนั้นกับฐานข้อมูล จะใช้คำสั่ง Commit เพื่อดำเนินการเปลี่ยนแปลงที่เกิดขึ้นอย่างถาวรกับฐานข้อมูล กล่าวอีกนัยหนึ่งคือการเปลี่ยนแปลงสถานะของการเปลี่ยนแปลงข้อมูลนี้จากระยะการทดลองไปสู่การเปลี่ยนแปลงถาวร เป็นต้น

#### 2.3.4 เครื่องคอมพิวเตอร์ลูกข่าย (Client)

ระบบเครือข่ายคอมพิวเตอร์ ที่มีการนำอุปกรณ์ตัวลูก (Client) ที่มีคุณสมบัติเทียบเท่า PC หรืออาจจะหมายถึงเครื่อง PC และต่อพ่วงเพียงจอภาพ เมาท์ คีย์บอร์ด เท่านั้น แล้วนำไปเชื่อมต่อกันเป็นระบบเครือข่ายในลักษณะ Client Server (ทดแทนระบบ Mini Computer ซึ่งมีราคาสูงกว่ามาก) โดยเครื่อง Client นั้นไม่จำเป็นต้องติดตั้ง Harddiskการทำงานและประมวลผลจะ

ทำจากแม่ข่าย (Server) เป็นหลัก หมายความว่า ระบบการทำงาน, โปรแกรม, ข้อมูลต่างๆ จะถูกติดตั้งไว้ที่คอมพิวเตอร์ที่เป็น Server เท่านั้น อธิบายเพิ่มเติมได้ดังนี้

ก) หลักการทำงานของ Client

เครื่องแม่ข่ายจะต้องติดตั้งโปรแกรมประเภท Terminal Service ซึ่งมีอยู่มากมาย ทั้งให้ใช้ฟรีและจำหน่ายเชิงธุรกิจ ที่เครื่องแม่ข่ายจะทำการประมวลผลตามคำร้องขอของผู้ใช้งานผ่าน Client จากนั้นจะทำการบีบอัดข้อมูลที่ผู้ใช้ร้องขอส่งผ่านระบบ LAN กลับมายังเครื่อง Client เพื่อแสดงผลทางจอภาพ

ข) Client ประกอบด้วย

การตัดสินใจเพื่อเปลี่ยนมาใช้ระบบ Client นั้นง่ายมาก หากองค์กรของท่านมีระบบเครือข่ายคอมพิวเตอร์อยู่แล้วก็เพียงแค่แทนที่ PC ด้วย Client ได้เลย และหากต้องการทำให้ระบบ Client ครบสมบูรณ์ต้องประกอบด้วย

- ระบบเครือข่ายคอมพิวเตอร์ เช่น LAN, Intranet, Internet
- เครื่องแม่ข่าย ติดตั้ง OS และโปรแกรม Terminal Service
- อุปกรณ์ Client ที่ต่อพ่วงกับ จอภาพ เมาท์ คีย์บอร์ด
- อุปกรณ์เครือข่ายอื่นๆ เช่น Printer/Fax Server, VoIP ฯลฯ
- โปรแกรมต่างๆ ที่ต้องติดตั้งที่เครื่องแม่ข่าย
- ผู้ดูแลระบบ
- ผู้ใช้งานระบบ

### 2.3.5 เครื่องคอมพิวเตอร์แม่ข่าย (Server)

คอมพิวเตอร์ที่ทำหน้าที่เป็นผู้ให้บริการทรัพยากร (Resources) ต่าง ๆ และควบคุมคอมพิวเตอร์อื่น ๆ ในเครือข่ายนั้นทั้งหมด เป็นสิ่งที่เก็บโปรแกรมและข้อมูลพื้นฐาน ได้แก่ หน่วยประมวลผล หน่วยความจำ หน่วยความจำสำรอง ฐานข้อมูล และ โปรแกรมต่าง ๆ เป็นต้น ในระบบเครือข่ายท้องถิ่น (LAN) เรียกว่าคอมพิวเตอร์แม่ข่าย ในระบบเครือข่ายระยะไกล ที่ใช้เมนเฟรมคอมพิวเตอร์ หรือ มินิคอมพิวเตอร์เป็นศูนย์กลางของเครือข่าย นิยมเรียกว่า Host Computer และเรียกเครื่องที่รองรับบริการว่าลูกข่ายหรือสถานีงาน

### 2.3.6 สถาปัตยกรรมไคลเอนท์-เซิร์ฟเวอร์ (Client-Server Architecture)

สถาปัตยกรรมไคลเอนท์-เซิร์ฟเวอร์ (โครงการเครือข่ายสารสนเทศเพื่อพัฒนาการศึกษา, ทบวงมหาวิทยาลัย, 2001) เป็นระบบคอมพิวเตอร์ที่ประกอบไปด้วยเครื่องคอมพิวเตอร์จำนวนหลาย ๆ เครื่อง ทั้งคอมพิวเตอร์ส่วนบุคคล เวิร์คสเตชัน เครื่องให้บริการแฟ้มข้อมูล เครื่องให้บริการการพิมพ์ ฯลฯ โดยมีการเชื่อมต่อกันผ่านระบบเครือข่าย โดยแบ่งออกเป็นสองส่วนคือเครื่องที่ขอใช้บริการ(Client) และเครื่องให้บริการ(Server) ในเรื่องของระบบฐานข้อมูลจะแบ่งโปรแกรมระบบจัดการฐานข้อมูลออกเป็นสองระดับ คือ ไคลเอนท์ และเซิร์ฟเวอร์ บางไซต์จะรันเฉพาะโปรแกรมไคลเอนท์ และบางไซต์จะรันเฉพาะโปรแกรมเซิร์ฟเวอร์เท่านั้น ตัวอย่างเช่น ในการประมวลผลคำสั่ง SQL ดำเนินการระหว่างไคลเอนท์และเซิร์ฟเวอร์ จะมีขั้นตอนดังนี้

ก) โคลเอนท์วิเคราะห์คำสั่ง และแยกคำสั่งออกเป็นหลาย ๆ คำสั่งตามจำนวนของเซิร์ฟเวอร์ไซต์ที่จะทำการสืบค้นข้อมูล จากนั้นก็จะส่งคำสั่งไปยังแต่ละเซิร์ฟเวอร์ไซต์นั้น

ข) เซิร์ฟเวอร์แต่ละตัวจะประมวลผลคำสั่งในเครื่องตนเอง และส่งผลลัพธ์กลับไปยังโคลเอนท์

ค) โคลเอนท์รวบรวมผลลัพธ์ที่ได้รับกลับมาจากไซต์ต่าง ๆ แล้วสร้างเป็นผลลัพธ์สุดท้าย

จากแนวทางนี้เครื่องที่ทำหน้าที่ประมวลผลคำสั่งจะเรียกว่า database processor (DP) หรือเครื่อง back-end และเครื่องโคลเอนท์จะเรียกว่า application processor(AP) หรือเครื่อง front-end ในระบบจัดการฐานข้อมูลแบบกระจาย (Distributed Database Management System : DDBMS) จะแบ่งออกเป็นสามระดับคือ

ก) โปรแกรมส่วน Server รับผิดชอบเกี่ยวกับการจัดการข้อมูลที่จะเหมือนกับโปรแกรม DBMS แบบรวมศูนย์

ข) โปรแกรมส่วน Client รับผิดชอบเกี่ยวกับการเข้าถึงข้อมูลที่อยู่ใน DDBMS catalog และทำการร้องขอการใช้บริการข้อมูลไปที่ไซต์อื่น

ค) โปรแกรมส่วน Communication จะสนับสนุนการสื่อสารข้อมูลในระบบเครือข่าย ซึ่งจะถูกใช้งานโดยโปรแกรมส่วน Client เพื่อทำการส่งคำสั่งและข้อมูลไปยังไซต์ที่ต้องการ

Client จะทำหน้าที่ในการสร้างแผนการสืบค้นข้อมูลแบบกระจาย ควบคุมให้การทำงานแทรกซ้นมีคุณสมบัติ ACID อยู่เสมอ และหน้าที่ที่สำคัญอีกอย่างหนึ่งคือมีความสามารถที่จะซ่อนรายละเอียดของการกระจายของข้อมูลจากผู้ใช้ได้ นั่นคือผู้ใช้งานในไซต์ต่างๆ ก็สามารถที่จะเข้าถึงข้อมูลได้โดยไม่จำเป็นต้องรู้ว่าข้อมูลเก็บอยู่ที่ไหนเราเรียกคุณสมบัตินี้ว่า distribution Transparency

บทบาทของโคลเอนท์ / เซิร์ฟเวอร์คอมพิวเตอร์ที่เป็นโคลเอนท์จะมีบทบาทและหน้าที่แตกต่างจากคอมพิวเตอร์ ซึ่งทำหน้าที่เป็นเซิร์ฟเวอร์ดังนี้

เซิร์ฟเวอร์มีบทบาทและหน้าที่ดังนี้

- ก) ประมวลผลการแก้ไขและขอเรียกดูข้อมูล
- ข) ดำเนินการอื่นที่เกี่ยวข้องกับข้อมูล
- ค) บังคับใช้ข้อกำหนดและข้อจำกัดเกี่ยวกับฐานข้อมูล
- ง) บังคับใช้มาตรการรักษาความปลอดภัย

โคลเอนท์จะมีบทบาทและหน้าที่ดังนี้

- ก) นำเสนอข้อมูลในรูปแบบที่เข้าใจได้ง่าย ใช้งานได้ง่าย
- ข) จัดให้มีส่วนติดต่อกับผู้ใช้ในลักษณะเครื่องมือ, ข้อมูล และรายงาน
- ค) ทำหน้าที่ส่งคำร้องขอบริการไปยังเซิร์ฟเวอร์

การทำงานแบบโคลเอนท์ / เซิร์ฟเวอร์จะดีก็ต่อเมื่อแอปพลิเคชันฝั่งโคลเอนท์ไม่ได้ติดต่อกับตารางในฐานข้อมูลโดยตรง แต่ทำงานผ่านตัวกลางที่ทำหน้าที่เป็นตัวเชื่อม เช่น สฟวร์อ็อกซ์ เป็นต้น หากแอปพลิเคชันติดต่อกับตารางในฐานข้อมูลโดยตรงจะเกิดผลเสียในระยะยาว คือไม่สามารถปรับปรุงแก้ไขโครงสร้างฐานข้อมูลได้ เพราะการปรับปรุงจะทำให้แอปพลิเคชันทำงานไม่ได้