

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องจะประกอบไปด้วย

2.1 ทฤษฎีด้านระบบฐานข้อมูล

- ความรู้เบื้องต้นเกี่ยวกับฐานข้อมูล
- โครงสร้างข้อมูลและสถาปัตยกรรมของระบบฐานข้อมูล
- ฐานข้อมูลเชิงสัมพันธ์
- แผนผังความสัมพันธ์ของข้อมูล

2.2 ทฤษฎีกระบวนการผลิตซอฟต์แวร์

- กระบวนการผลิตซอฟต์แวร์

2.3 ทฤษฎีหรือเทคโนโลยีและเทคโนโลยีที่ใช้ในการพัฒนา

- ภาษาพีเอชพี (PHP)
- ภาษาเอสคิวแอล (SQL)
- ภาษาเอชทีเอ็มแอล (HTML)
- ภาษาซีเอสเอส (CSS)

2.4 ทฤษฎีของการพัฒนาระบบแบบเจ็คโอเรียนเต็ด (OOP: Object Oriented Programming)

- ความรู้เบื้องต้นเกี่ยวกับการพัฒนาระบบแบบเจ็คโอเรียนเต็ด (OOP: Object Oriented Programming)

2.5 ทฤษฎีของการพัฒนาระบบแบบเอ็มวีซี (MVC)

- ความรู้เบื้องต้นเกี่ยวกับการพัฒนาระบบแบบเจ็คโอเรียนเต็ด (OOP: Object Oriented Programming)

2.1 ทฤษฎีด้านระบบฐานข้อมูล

ความรู้เบื้องต้นเกี่ยวกับฐานข้อมูล (INTRODUCTION TO DATABASE)

ความหมายของฐานข้อมูล

ฐานข้อมูล หมายถึง การรวบรวมข้อมูลที่เกี่ยวข้องกัน โดยกำหนดรูปแบบในการจัดเก็บอย่างเป็นระบบ เพื่อให้ผู้ใช้สามารถจัดการและเรียกใช้ข้อมูลได้อย่างมีประสิทธิภาพ

ระบบการจัดการฐานข้อมูล

จะทำหน้าที่ในการควบคุมดูแลการสร้างและการเรียกใช้ข้อมูล โดยไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างฐานข้อมูล ซึ่งประกอบด้วย โครงสร้าง (Structure) และการใช้งาน (Operation)

- โครงสร้างหมายถึง โครงสร้างของระบบฐานข้อมูล
- การใช้งาน หมายถึง วิธีการที่จะให้ผู้ใช้สามารถเรียก , แก้ไขข้อมูล ในระบบฐานข้อมูลได้

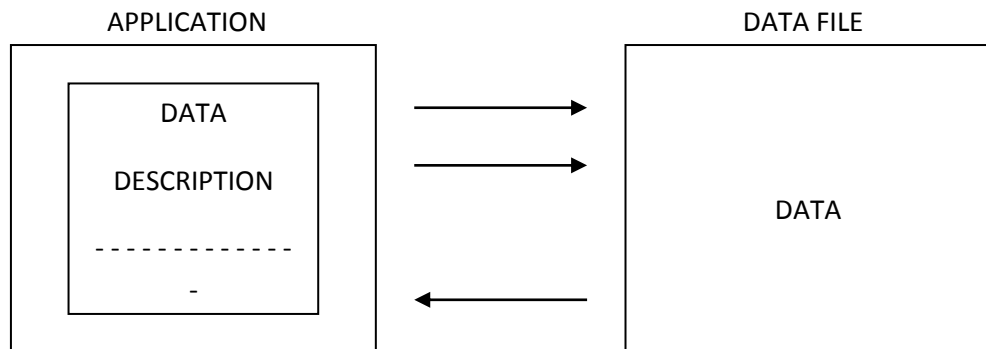
จุดมุ่งหมายของระบบจัดการฐานข้อมูล

เพื่อจัดการควบคุมฐานข้อมูล และสนับสนุนการใช้ข้อมูลร่วมกันภายในองค์กรอย่างมีระบบ รวมถึงการพัฒนาข้อมูลให้ทันสมัย ตรงกับความต้องการของผู้ใช้

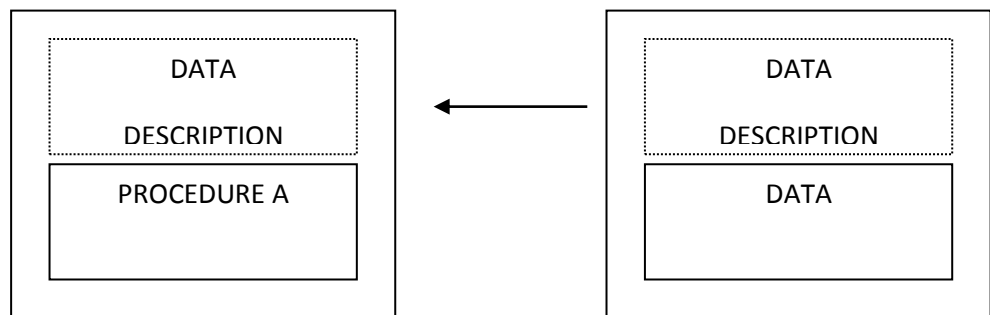
ลักษณะของการจัดการฐานข้อมูล

แบ่งได้ 2 ลักษณะใหญ่ๆคือ

1. ลักษณะข้อมูลทั่วไป จะจัดเก็บแฟ้มข้อมูลธรรมดาที่มีเฉพาะค่าของข้อมูล ส่วนรายละเอียดที่เกี่ยวข้องกับตัวข้อมูล จะถูกระบุอยู่ในแฟ้มข้อมูล โปรแกรมซึ่งเขียนขึ้นโดยผู้เขียนโปรแกรมเอง
2. ลักษณะข้อมูลแบบเมทาดาดา (Meta data) จะจัดเก็บรายละเอียดข้อมูลเกี่ยวกับตัวข้อมูลไว้ที่เดียวกัน จัดแยกเป็น 2 ส่วนคือ ข้อมูลและเมทาดาดา (Meta data) ซึ่งบางโปรแกรมเรียกว่า พจนานุกรม (Data dictionary) โดยจะมีระบบจัดการฐานข้อมูลทำหน้าที่ควบคุมจัดการ



ภาพที่ 2.1 ลักษณะข้อมูลทั่วไป ในโปรแกรมประยุกต์ใช้งานทั่วไป



ภาพที่ 2.2 ลักษณะเมทาตาตา (Meta data) ในโปรแกรมประยุกต์ใช้ฐานข้อมูล

ประโยชน์ของฐานข้อมูล

- ลดความซ้ำซ้อนของข้อมูล

การจัดเก็บข้อมูลแบบแฟ้มข้อมูลอาจจะทำให้ข้อมูลถูกเก็บไว้หลายแห่ง ทำให้เกิดความซ้ำซ้อน การนำข้อมูลรวบรวมเก็บไว้ในฐานข้อมูลเดียวกันจะช่วยลดปัญหาความซ้ำซ้อนได้

- หลีกเลี่ยงความขัดแย้งของข้อมูล

การจัดเก็บข้อมูลแบบแฟ้มข้อมูลหลายแฟ้มข้อมูล จะก่อให้เกิดความขัดแย้งของข้อมูลได้ ในฐานข้อมูล ข้อมูลจะถูกจัดเก็บรวมกัน จึงสามารถเลี่ยงการซ้ำกันของข้อมูลได้

- สามารถใช้ข้อมูลร่วมกันได้
 ผู้ใช้ข้อมูลสามารถใช้ข้อมูลจากฐานข้อมูลเดียวกันได้ ซึ่งอาจมาจากแฟ้มข้อมูลที่แตกต่างกัน โดยฐานข้อมูลจะมีลักษณะของการแชร์ข้อมูล (Share Data)
- การรักษาความถูกต้อง เชื่อถือได้ของข้อมูล
 ระบบการจัดการฐานข้อมูล สามารถที่จะระบุกฎเกณฑ์ เพื่อควบคุมความผิดพลาดที่อาจเกิดขึ้นได้ในลักษณะข้อมูลที่ถูกต้อง (Valid Data)
- กำหนดระดับความปลอดภัยของข้อมูลได้
 ผู้บริหารระบบฐานข้อมูลสามารถกำหนดสิทธิ ,ระดับการเรียกใช้ข้อมูลของผู้ใช้แต่ละคนได้
- กำหนดความเป็นมาตรฐานเดียวกัน
 ผู้บริหารระบบฐานข้อมูล สามารถกำหนดและควบคุมให้มีมาตรฐานของข้อมูล ให้เป็นรูปแบบหรือโครงสร้างเดียวกันได้
- เกิดความเป็นอิสระของข้อมูลและโปรแกรม
 ระบบฐานข้อมูล โปรแกรมประยุกต์ใช้จะทำงานอย่างมีระบบ โดยจะเชื่อมโยงฐานข้อมูลโดยที่จะสามารถใช้ข้อมูลได้อย่างอิสระ โดยวิธีการแปลงส่ง (Mapping) หรือลักษณะวิว (View)

ข้อเสีย

- มีต้นทุนสูง
- การเสี่ยงต่อการหยุดชะงักของระบบ
 เนื่องจากข้อมูลอาจถูกเก็บไว้ในลักษณะเป็นศูนย์รวม ความผิดพลาดของการทำงานบางส่วนอาจมีผลกระทบกับระบบ ทำให้ระบบหยุดชะงักได้

องค์ประกอบของระบบฐานข้อมูล

1. ฮาร์ดแวร์ (Hardware)

ระบบฐานข้อมูล จะต้องมีส่วนฮาร์ดแวร์ที่มีประสิทธิภาพ ในเรื่องของหน่วยประมวลผลกลางที่ต้องการความเร็วสูง มีขนาดหน่วยความจำหลักที่มากพอ และหน่วยความจำสำรองที่จะจัดเก็บข้อมูลเพียงพอกับความต้องการของระบบ รวมถึงอุปกรณ์นำเข้าข้อมูลเข้าและออกรายงานตามคุณสมบัติที่ระบบต้องการ เพื่อให้ระบบการประมวลผลมีประสิทธิภาพ

2. ซอฟต์แวร์ (Software)

ระบบฐานข้อมูล จะต้องมีส่วนโปรแกรมที่ทำหน้าที่ควบคุมดูแลการสร้าง การสืบค้นหาข้อมูล การปรับเปลี่ยน การแก้ไขโครงสร้าง การจัดทำรายงาน โดยเป็นสื่อกลางระหว่างผู้ใช้กับระบบงาน และโปรแกรมประยุกต์ต่างๆที่มีอยู่ รวมถึงการสื่อสารข้อมูลระหว่างเครื่องด้วย

คุณสมบัติและหน้าที่ของโปรแกรมการจัดการฐานข้อมูล

- สามารถกำหนดและเก็บโครงสร้างฐานข้อมูล
- สามารถรับและจัดเก็บข้อมูลไว้ในฐานข้อมูล
- สามารถดูแลรักษาข้อมูลในฐานข้อมูล
- สามารถประสานงานกับระบบปฏิบัติการ
- สามารถจัดทำข้อมูลสำรองและการกู้ข้อมูล
- สามารถควบคุมความปลอดภัยในระบบฐานข้อมูล
- สามารถควบคุมการใช้งานของผู้ใช้ระบบในเวลาเดียวกัน
- สามารถควบคุมความบูรณาภาพของข้อมูล
- สามารถจัดทำพจนานุกรมข้อมูล
- สามารถติดต่อเชื่อมโยงในลักษณะของระบบเครือข่าย

1. ข้อมูล (Data)

ฐานข้อมูลเป็นการจัดเก็บข้อมูลที่เป็นข้อเท็จจริงขั้นต้น

2. บุคลากร (People)

- ผู้ใช้ทั่วไป (user) เป็นบุคลากรที่ใช้ข้อมูลจากระบบฐานข้อมูล
- พนักงานปฏิบัติการ (operator) เป็นผู้ป้อนข้อมูลเข้าเครื่องคอมพิวเตอร์
- นักวิเคราะห์และออกแบบระบบ (system analysis) เป็นผู้ทำหน้าที่วิเคราะห์ระบบฐานข้อมูล
- ผู้เขียนโปรแกรมประยุกต์ใช้งาน (programmer) เป็นผู้ทำหน้าที่เขียนโปรแกรมประยุกต์ใช้งานต่างๆ ตามความต้องการของผู้ใช้
- ผู้บริหารข้อมูล (database administration : DBA) เป็นผู้ทำหน้าที่บริหาร และควบคุมระบบฐานข้อมูลทั้งหมด รวมถึงเป็นผู้ตัดสินใจในการเก็บ กำหนดโครงสร้าง วิธีการเรียกใช้ข้อมูล เพื่อบริหารฐานข้อมูลอย่างมีประสิทธิภาพ

3. ขั้นตอนการทำงานและหน้าที่ต่างๆในระบบฐานข้อมูล

- ผู้บริหารฐานข้อมูล (DBA)
คือผู้ทำหน้าที่ควบคุมการบริหารงานของระบบฐานข้อมูลทั้งหมด
- ตัดสินใจว่าจะรวบรวมงานของระบบฐานข้อมูลทั้งหมด
- วิเคราะห์และทำการตัดสินใจว่าจะเก็บข้อมูลด้วยวิธีใด และจะใช้เทคนิคใดในการเรียกใช้ข้อมูล
- ประสานงานกับผู้ใช้
- กำหนดแบบความปลอดภัยและความคงสภาพของข้อมูล
- กำหนดแบบแผนในการสร้างระบบข้อมูลสำรองและการฟื้นฟูสภาพ
- คอยสำรวจผลการทำงานและตรวจตราความต้องการของผู้ใช้

โครงสร้างข้อมูลและสถาปัตยกรรมของระบบฐานข้อมูล (DATA STRUCTURE AND DATABASE SYSTEM ARCHITECTURE)

ลักษณะของโครงสร้างข้อมูล สามารถแบ่งเป็นหน่วยข้อมูลได้ดังนี้

- บิต (bit) หมายถึง หน่วยของข้อมูลที่มีขนาดเล็กที่สุดในระบบแฟ้มข้อมูล
- ไบท์ (byte) หมายถึง หน่วยของข้อมูลที่ประกอบด้วยบิตหลายๆบิตมารวมกันโดยมีขอบเขตที่แทนความหมาย หมายถึงสิ่งหนึ่ง เช่น รหัสประจำตัว เป็นตัวแทนค่าตัวเลขที่มีความหมายตามขอบเขต

ตัวอย่าง ID_NUMBER มีจำนวน 8 หลักเป็นต้น

- เรคคอร์ด (record) หมายถึง หน่วยของข้อมูลที่ประกอบด้วยหลายๆฟิลด์มารวมกัน ที่มีความสัมพันธ์กัน เช่น รหัสประจำตัว ชื่อ ที่อยู่ เป็นต้น
- แฟ้มข้อมูล (file) หมายถึง หน่วยของข้อมูลที่ประกอบด้วยหลายๆเรคคอร์ดมารวมกัน

หน่วยการจัดเก็บของฐานข้อมูลเชิงสัมพันธ์ สามารถเปรียบเทียบกับแฟ้มข้อมูล โดยอาศัยศัพท์ดังนี้

| ศัพท์แฟ้มข้อมูล | ศัพท์ฐานข้อมูล | ศัพท์เทคนิค |
|--------------------------------------|------------------------------|--------------------------------|
| แฟ้มข้อมูล (file) | ตาราง (table) | รีเลชัน (relation) |
| เรคคอร์ด (record) | แถว (row) | ทูเพิล (tuple) |
| ฟิลด์ (field) | คอลัมน์ (column) | แอททริบิวต์ (attribute) |
| จำนวนเรคคอร์ด (number of records) | จำนวนแถว (number of rows) | คาร์ดินาลิตี้ (cardinality) |
| กลุ่มข้อมูล (data type) | กลุ่มข้อมูล (data type) | โดเมน (domain) |

ตารางที่ 2.1 หน่วยการจัดเก็บของฐานข้อมูลเชิงสัมพันธ์

- รีเลชั่น หมายถึง การแทนข้อมูลของตารางในฐานะข้อมูลเชิงสัมพันธ์ ซึ่งอาจเกี่ยวกับคน สถานที่ สิ่งของ เช่น รีเลชั่นของข้อมูลสินค้า เป็นต้น
- ทูเพิล หมายถึง การแทนค่าข้อมูลในแต่ละแถว หรือ อาจเรียกกันว่า ระเบียน (record)
- แอททริบิวต์ หมายถึง การแทนค่าขอบเขตของข้อมูลในแต่ละคอลัมน์
- คาร์ดินาลิตี้ หมายถึง จำนวนแถวของข้อมูลในแต่ละตาราง
- โดเมน หมายถึง ขอบเขตหรือค่าที่ควรจะเป็นของข้อมูลในแอททริบิวต์หนึ่ง

ฐานข้อมูลเชิงสัมพันธ์ เป็นการจัดเก็บรวบรวมข้อมูลในลักษณะ 2 มิติ คือ แถวและคอลัมน์ โดยมีการจัดเก็บอย่างมีระบบและกฎเกณฑ์

| รหัสวิชา | ชื่อวิชา | หน่วยกิต | จำนวนชั่วโมง | |
|-----------|--------------------|----------|--------------|-------|
| 4122201 | ฐานข้อมูลเบื้องต้น | 3 | (2-2) | Tuple |
| 4122202 | การสื่อสารข้อมูล | 3 | (2-2) | |
| 4122203 | คอมพิวเตอร์กราฟฟิก | 3 | (2-2) | |
| | | | | |
| Attribute | | | | |

CARDINALITY

ภาพที่ 2.3 ตารางตัวอย่างของการแสดงข้อมูลของรายวิชา

สถาปัตยกรรมของฐานข้อมูล

ระบบฐานข้อมูล เป็นโครงสร้างข้อมูลที่ใช้หลายกลุ่ม ต้องการเรียกใช้ข้อมูลบางส่วนหรือทั้งหมด โดยจะแตกต่างกันไปตามงานที่ปฏิบัติ ลักษณะของผู้ใช้ก็มีหลากหลายโดยเฉพาะผู้ใช้ทั่วไป (end user) เป็นผู้ใช้ที่ไม่มีความจำเป็นต้องรู้รายละเอียดของการจัดเก็บข้อมูล ดังนั้น เพื่อให้การทำงานเป็นไปตามระดับของผู้ใช้จึงสามารถแบ่งระดับของข้อมูลได้ 3 ระดับ คือ

1. ระดับภายนอก (External Level)

เป็นระดับของข้อมูลที่ใช้ใช้แต่ละคนมองเก็บข้อมูล ตามระบบงานที่ผู้ใช้ได้รับผิดชอบงานนั้น ผู้ใช้ส่วนใหญ่เป็น operator หรือ โปรแกรมเมอร์

2. ระดับแนวคิด (Conceptual Level)

เป็นโครงร่างที่อธิบายถึงฐานข้อมูล โครงสร้างข้อมูล ความสัมพันธ์ของข้อมูล กฎเกณฑ์และข้อจำกัดที่มีต่อฐานข้อมูล โดยจะผ่านการวิเคราะห์และออกแบบจากผู้บริหารฐานข้อมูล คือ DBA

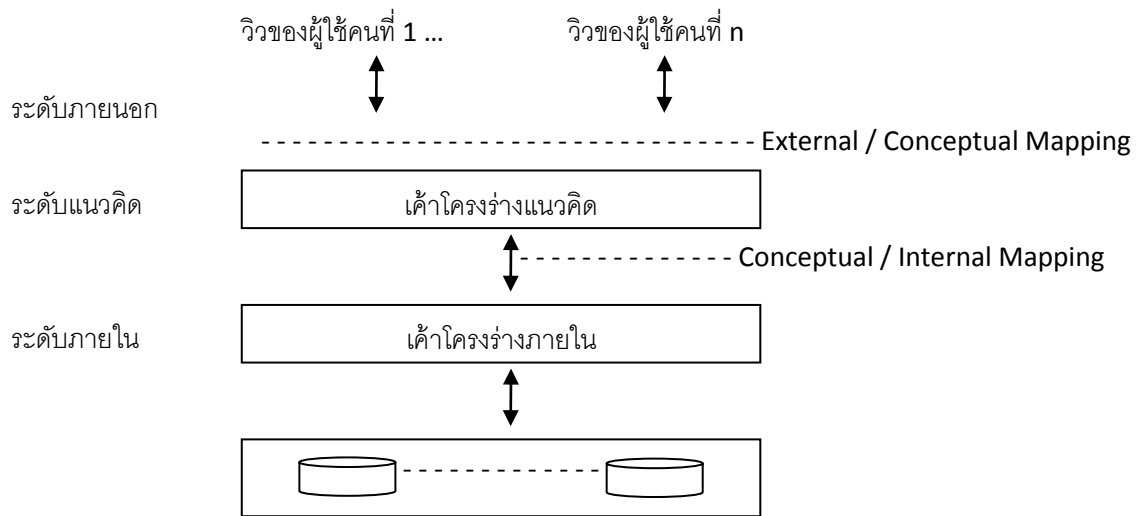
3. ระดับภายใน (Internal หรือ Physical Level)

เป็นโครงร่างที่อธิบายถึงการจัดเก็บข้อมูลจริง จัดเก็บในรูปแบบใดวิธีการเข้าถึงข้อมูลวิธีใด เช่น การแฮชซิง (Hashing Function) , การอินเด็กซ์ (Indexing) เป็นต้น

ข้อมูลแต่ละระดับจะมีตัวเชื่อมโยงโดยจะถูกจัดการด้วยระบบการจัดการฐานข้อมูล โดยจะแปลงความหมายข้อมูลจากระดับหนึ่ง ไปยังอีกระดับหนึ่งที่เรียกกันว่า การแปลงส่ง (Mapping)

การแปลงส่ง (Mapping) แบ่งออกเป็น 2 ประเภท

1. การแปลงส่งระหว่างระดับภายนอกกับระดับแนวคิด (External / Conceptual Mapping) ได้แก่ การใช้ภาษาสำหรับนิยามข้อมูล Data Definition Language (DDL)
2. การแปลงส่งระหว่างระดับแนวคิดกับระดับภายใน (Conceptual / Internal Mapping)



ภาพที่ 2.4 ฐานข้อมูล

การแบ่งระดับของข้อมูลดังกล่าวทำให้เกิดความเป็นอิสระของข้อมูล โดยสามารถแบ่งออกเป็น 2 อย่างคือ

1. ความเป็นอิสระของข้อมูลเชิงตรรกะ (Logical Data Independence) คือ ความเป็นอิสระของข้อมูลในระดับภายนอกกับระดับแนวคิด โดยการเปลี่ยนแปลงข้อมูลที่เกิดขึ้นในระดับแนวคิด จะไม่มีผลกระทบต่อระดับภายนอก เช่น การเพิ่มแอตทริบิวต์จะไม่มีผลต่อโปรแกรมประยุกต์ที่เขียนขึ้น
2. ความเป็นอิสระของข้อมูลในเชิงกายภาพ (Physical Data Independence) คือ ความเป็นอิสระของข้อมูลในระดับแนวคิดกับระดับภายใน โดยการเปลี่ยนแปลงที่เกิดขึ้นในระดับภายใน จะไม่มีผลกระทบต่อระดับแนวคิด เช่น การเพิ่มประสิทธิภาพการเข้าถึงข้อมูลให้เร็วขึ้น โดยการปรับปรุงเค้าโครงร่างภายใน

ภาษาที่ใช้ในระบบฐานข้อมูล

แบ่งออกเป็น 3 ประเภท คือ

1. ภาษาที่ใช้ในการนิยามข้อมูล (Data Definition Language , DDL)
เป็นภาษาคำสั่งที่ใช้กำหนดโครงสร้างข้อมูลของฐานข้อมูล

2. ภาษาที่ใช้ในการดำเนินการจัดการข้อมูล (Data Manipulation Language , DML)

เป็นภาษาคำสั่งที่ใช้ในการเรียกใช้ข้อมูล การเปลี่ยนแปลงแก้ไขข้อมูล การเพิ่มหรือลดข้อมูล ภาษาที่ใช้เรียกกันว่า SQL

3. ภาษาที่ใช้ในการควบคุมข้อมูล (Data Control Language , DCL)

เป็นภาษาคำสั่งที่ใช้ในการควบคุม หรือ ป้องกันข้อมูล ที่เกิดจากการใช้ข้อมูลจากผู้ใช้หลายคน

ฐานข้อมูลเชิงสัมพันธ์ (RELATIONAL DATABASE)

ฐานข้อมูลเชิงสัมพันธ์

เป็นการจัดเก็บรวบรวมข้อมูลในลักษณะตาราง 2 มิติ คือ แถวและ คอลัมน์ โดยมีการจัดเก็บอย่างเป็นระบบและกฎเกณฑ์ ซึ่งได้ผ่านกระบวนการทำ รีเลชั่นให้เป็นบรรทัดฐาน (Normalized) เพื่อลดปัญหาที่จะเกิดขึ้นในการจัดเก็บและค้นหาข้อมูล

การจัดเก็บฐานข้อมูลเชิงสัมพันธ์ จะทำให้ข้อมูลเป็นกลุ่มตาราง ผู้ใช้จะเรียนรู้และเข้าใจง่าย โดยไม่จำเป็นต้องรู้ว่า ข้อมูลจริง ๆ มีการจัดเก็บอย่างไร รวมถึงวิธีการเรียกใช้ข้อมูล ภาษาที่ใช้คือเอชคิวแอล (SQL) ซึ่งเป็นภาษาที่มีลักษณะคล้ายภาษาอังกฤษ ง่ายต่อการเขียนคำสั่ง จึงเหมาะสมกับการใช้เป็นคำสั่งแบบสอบถาม

คุณสมบัติของตาราง

- ข้อมูลในแต่ละแถวจะไม่ซ้ำกัน โดยระบบจัดการฐานข้อมูลจะควบคุมไม่ให้เกิดความซ้ำซ้อนกันขึ้น
- ข้อมูลตารางในแต่ละแถวไม่มีความสำคัญในการเรียงลำดับ
- ในแต่ละคอลัมน์ของแถวหนึ่งๆจะบรรจุข้อมูลได้เพียงค่าเดียว
- ค่าของข้อมูลในแต่ละคอลัมน์ จะบรรจุข้อมูลประเภทเดียวกัน
- คอลัมน์ ทางเทคนิค จะใช้ Attribute
- แถว ทางเทคนิค จะใช้ Tuple


ความหมายของคีย์ (Key)

ข้อมูลเชิงสัมพันธ์ สามารถเชื่อมโยงข้อมูลด้วยการใช้คีย์ ซึ่งความหมายของคีย์ ก็จะใช้คุณสมบัติในการจัดแบ่ง โดยแบ่งออกเป็นประเภทใหญ่ๆ ได้ 2 ประเภท คือ


1. คีย์หลัก (Primary Key) เป็นคีย์ที่เลือกจากแอททริบิวต์ที่มีคุณสมบัติ คือ มีค่าที่ไม่ซ้ำซ้อนกัน ในแต่ละแถว (Tuple) นั้นเอง
2. คีย์นอก (Foreign Key) เป็นคีย์ที่ใช้อ้างอิง แอททริบิวต์เดียวกันในอีกตาราง (Relation) หนึ่ง ซึ่งคีย์นอกนี้ จะมีคุณสมบัติเป็นคีย์หลักในตารางอ้างอิง

ตารางตัวอย่างของการแสดงข้อมูลพนักงาน


| รหัสพนักงาน | ชื่อพนักงาน | นามสกุล | รหัสแผนก | เงินเดือน |
|-------------|-------------|----------|----------|-----------|
| 4001001 | นายขจร | ชาติดี | 01 | 7,500 |
| 4001002 | นายขจร | รุ่งอรุณ | 01 | 8,000 |
| 4002003 | นายใหญ่ | ใจดี | 01 | 9,000 |
| 4002004 | นางเล็ก | ใจใหญ่ | 02 | 6,000 |
| 4002005 | นางดำ | ดีใจ | 02 | 7,000 |



Primary Key



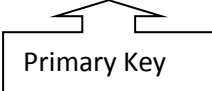
Alternate Key



Foreign Key

ภาพที่ 2.5 ตารางบุคลากร

| รหัสแผนก | ชื่อแผนก | จำนวน |
|----------|-------------|-------|
| 01 | คอมพิวเตอร์ | 2 |
| 02 | การเงิน | 3 |
| 03 | บัญชี | 2 |



Primary Key

ภาพที่ 2.6 ตารางแผนก

การสร้างคีย์สามารถทำได้ในแต่ละตาราง ซึ่งแอททริบิวต์ใดไม่สามารถเป็นคีย์หลักได้ ก็จะมีการรวมแอททริบิวต์ที่เป็นคีย์รอง (ALTERNATE KEY) มารวมกัน เพื่อให้เป็นคีย์หลัก ซึ่งมีค่าเป็นค่าที่ไม่ซ้ำกัน (UNIQUE) โดยมีวิธีทำ 3 แบบคือ

1. แบบคีย์ผสม (COMPOSITE) เป็นการนำคีย์รอง (ALTERNATE KEY) ที่มากกว่า 2 คีย์ขึ้นไป นำมารวมกัน ทำให้เป็นคีย์หลัก (PRIMARY KEY)
2. แบบคีย์คู่แข่ง (CANDIDATE) เป็นการนำคีย์หลัก (PRIMARY KEY) และคีย์รอง (ALTERNATE KEY) มารวมกันและเมื่อตัดเอาคีย์รอง (ALTERNATE KEY) ออก ค่าก็ยังเป็นค่าที่ไม่ซ้ำกัน (UNIQUE) การนำมารวมกันเพื่อเป็นประโยชน์ในการค้นหา
3. แบบคีย์ประกอบ (COMBINED) เป็นการนำคีย์รอง (ALTERNATE KEY) ที่มีมากกว่า 2 คีย์ขึ้นไปมารวมกัน เพื่อเป็นค่าที่ไม่ซ้ำกัน (UNIQUE) ทำเป็นคีย์หลัก (PRIMARY KEY) โดยหากขาดคีย์ตัวใดไปแล้วค่าจะไม่เป็นค่าที่ไม่ซ้ำกัน (UNIQUE)

ฐานข้อมูลเชิงสัมพันธ์กับกฎของคีย์

- กฎความบูรณภาพของเอนทิตี (THE ENTITY INTEGRITY RULE) คอลัมน์ใดที่เป็นคีย์หลัก จะมีค่าว่าง (NULL) ไม่ได้ เพราะจะไม่ทราบค่าที่แน่นอน
- กฎความบูรณภาพของการอ้างอิง (THE REFERENTIAL INTEGRITY RULE) การอ้างอิงของคีย์นอก จะต้องสามารถอ้างอิงให้ตรงกับค่าของคีย์หลัก เพื่อเชื่อมโยงข้อมูลระหว่างตารางได้

หลักการแก้ไขและลบข้อมูล

- การแก้ไขหรือลบข้อมูลแบบมีข้อจำกัด (RESTRICT)
การแก้ไขหรือลบข้อมูลได้ก็ต่อเมื่อ ไม่มีข้อมูลที่ถูกอ้างอิงโดยคีย์นอกของอีกตารางหนึ่ง เช่น รหัสวิชาในตารางวิชาจะถูกแก้ไขหรือลบได้ก็ต่อเมื่อไม่มีวิชาใดที่เปิดสอน เป็นต้น

- การแก้ไขหรือลบข้อมูลแบบต่อเรียง (CASCADE)
การแก้ไขหรือลบข้อมูลแบบต่อเรียงของคีย์หลัก ระบบจะทำการลบข้อมูลของคีย์นอก ในตารางที่มีคีย์นอกอ้างอิงอยู่ เช่น ในกรณีลบวิชาในตารางวิชา ข้อมูลของตารางอื่นที่เปิดสอนจะต้องถูกลบด้วย เป็นต้น
- การแก้ไขหรือลบข้อมูลโดยเปลี่ยนเป็นว่าง (NULLIFY)
การแก้ไขหรือลบข้อมูลจะทำได้ก็ต่อเมื่อ มีการเปลี่ยนแปลงค่าคีย์นอกในข้อมูลที่ถูกอ้างอิงเป็นว่างเสียก่อน เช่น รายวิชาที่เปิดสอนจะถูกเปลี่ยนเป็นค่าว่างเสียก่อน แล้วจึงลบข้อมูลของตารางวิชาที่ต้องการลบได้ เป็นต้น
- การแก้ไขหรือลบข้อมูลแบบใช้ค่าโดยปริยาย (DEFAULT)
จะเป็นการปรับค่าคีย์นอกที่กำหนดขึ้น ในกรณียกเลิกวิชาที่เปิดสอน เป็นปรับเปลี่ยนค่าเป็น “0” สำหรับค่าที่ไม่มีการเปิดวิชาเรียน เป็นต้น

ประเภทของตาราง

- ตารางของข้อมูลเชิงสัมพันธ์ จะมีตารางหลักที่ถูกกำหนดขึ้น เพื่อเก็บข้อมูลจริงจะทำการเรียกใช้ข้อมูลในภายหลัง
- ตารางของข้อมูลเชิงสัมพันธ์แบบวิว (VIEW) เป็นตารางที่ถูกสร้างขึ้นตามความต้องการของผู้ใช้แต่ละคนตามงานที่ผู้ใช้กำหนดขึ้น หรือเราเรียกกันว่า ตารางสมมติ

แผนผังความสัมพันธ์ของข้อมูล (E-R DIAGRAM)

จะใช้เป็นตัวเขียนโครงร่างในการออกแบบฐานข้อมูล ซึ่งแบ่งเป็น 2 ส่วนคือ

1. เอนทิตี (ENTITY)

เป็นอ็อบเจกทีฟ (OBJECTIVE) อะไรก็ได้ที่เราสนใจที่จะนำมาเป็นหลักในการทำแฟ้มข้อมูลของเรื่องนั้น ซึ่งสามารถแบ่งออกได้ว่าเป็นเอนทิตี (ENTITY) เช่น โต๊ะ จะมีแอททริบิวต์เป็นหมายเลขครุภัณฑ์ ,หนังสือ จะมีแอททริบิวต์เป็นหมายเลขหมวดหมู่

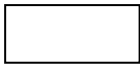
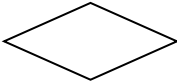
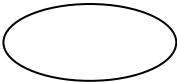
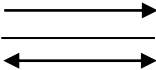
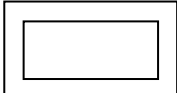
2. กลุ่มของเอนทิตี (ENTITY SET)

เป็นกลุ่ม (SET) ของเอนทิตี (ENTITY) ที่เป็นชนิดเดียวกัน เช่น กลุ่มของเอนทิตี (ENTITY SET) ของหนังสือ เป็นต้น

- แอททริบิวต์ (ATTRIBUTE) เป็นตัวบอกรายละเอียดของเอนทิตี (ENTITY) เช่น หนังสือ มีแอททริบิวต์เป็น เลขที่ ,หมวดหมู่ ,ชื่อผู้แต่ง , ชนิด เป็นต้น
- ความสัมพันธ์ (RELATIONSHIP) เป็นการแสดงความสัมพันธ์ระหว่างเอนทิตี (ENTITY) หรือกลุ่มของเอนทิตี (ENTITY SET) ว่าเกี่ยวข้องซึ่งกันและกันเป็นแบบใด
- ความสัมพันธ์แบบกลุ่ม (RELATIONAL SET) เป็นการแสดงความสัมพันธ์ของกลุ่ม (SET) ที่เป็นชนิดเดียวกัน

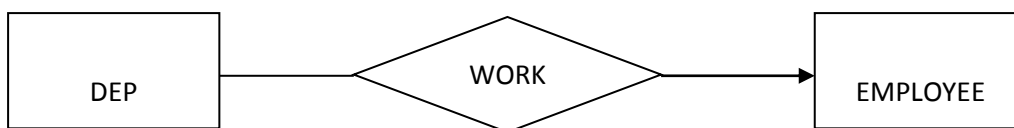
วิธีเขียนแผนผังความสัมพันธ์ของข้อมูล (E-R DIAGRAM)

สัญลักษณ์ที่ใช้เขียนมี 6 แบบคือ

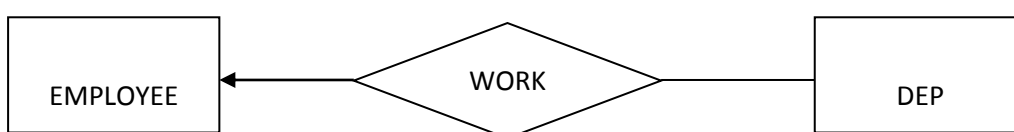
1.  แทนกลุ่มของเอนทิตี (ENTITY SET)
2.  แทนความสัมพันธ์ (RELATION SHIP)
3.  แทนแอททริบิวต์ (ATTRIBUTE)
4.  แทนเส้น (LINE)
5.  แทนเอนทิตีอ่อนแอ (WEAK ENTITY)

เอนทิตีอ่อนแอ (WEAK ENTITY) เป็นเอนทิตี (ENTITY) ที่เกิดขึ้นใหม่ ในภายหลังโดยอาศัยจากเอนทิตี (ENTITY) อื่น ๆ และจะต้องมีความสัมพันธ์อยู่ภายใต้ของอิทธิพลของเอนทิตี (ENTITY) นั้น

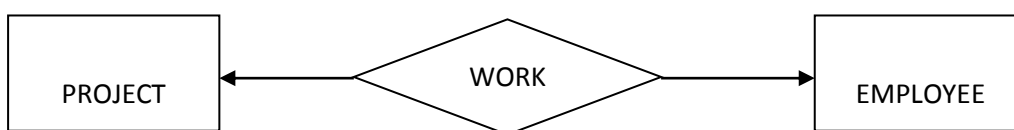
ตัวอย่าง รูปแบบการเขียนแผนผังความสัมพันธ์ของข้อมูล (E-R DIAGRAM)



ภาพที่ 2.7 ความสัมพันธ์การทำงานสังกัดแผนกใด ซึ่งเป็นความสัมพันธ์แบบ 1:N



ภาพที่ 2.8 ความสัมพันธ์บุคลากรแต่ละคนสังกัดแผนกเดียว ซึ่งเป็นความสัมพันธ์แบบ N:1



ภาพที่ 2.9 ความสัมพันธ์บุคลากรกับงาน ซึ่งเป็นความสัมพันธ์แบบ N:N

ตัวอย่าง รูปแบบการเขียนโครงร่างเอนทิตี (ENTITY)

EMPLOYEE (CODE ,NAME ,SURNAME ,DEP ,SALARY)

DEP (DEPCODE ,DNAME ,AMOUNT)

การเขียนรูปแบบความสัมพันธ์

การเขียนรูปไดอะแกรม (DIAGRAM) แบบความสัมพันธ์ โดยทั่วไปสามารถเขียนได้ 2 แบบใหญ่ คือ

1. ความสัมพันธ์แบบทั่วไป (GENERALIZATION)
2. ความสัมพันธ์แบบการประยุกต์ (APPLICATION)

ความสัมพันธ์แบบทั่วไป (GENERALIZATION) เป็นความสัมพันธ์ของเอนทิตี (ENTITY) ที่สามารถแยกได้เป็นหลาย (ENTITY)

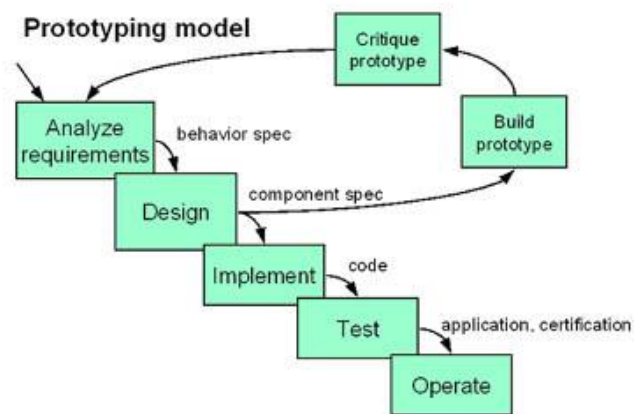
2.2 ทฤษฎีกระบวนการผลิตซอฟต์แวร์

กระบวนการผลิตซอฟต์แวร์ (Software Process)

โมเดลแบบ Pototype (Pototype Model)

Prototype model คือกระบวนการพัฒนาระบบซอฟต์แวร์โดยสร้างตัวต้นแบบขึ้นมาเพื่อที่จะให้ผู้ใช้งานระบบได้เห็นถึงรูปร่างลักษณะของระบบที่ตนเองต้องการ ทำให้ความต้องการของผู้ใช้งานชัดเจน ลดข้อผิดพลาดที่อาจจะเกิดจากความต้องการไม่ชัดเจน

ภาพที่ 2.10 รูปแสดงหลักการทำงานของระบบโมเดลแบบโปโตไทป์
(Pototype model)



ขั้นตอนการทำงานซอฟต์แวร์แบบโปรโตไทป์

Prototype model แบ่งออกเป็น 2 อย่างดังนี้

1. Throw Away prototype การสร้างตัวต้นแบบ แบบใช้งานเสร็จแล้วทิ้ง สร้างมาเพื่อให้เห็นรูปแบบของ interface ตัวโปรแกรมต่างๆ ส่วนมากเป็นการเขียนในกระดาษ

2. Evolutionary Prototype การสร้างตัวต้นแบบที่สามารถใช้งานได้จริงๆ อาจจะค่อยๆ ทำในทีละส่วนปรับปรุงแก้ไขโปรแกรมจนกระทั่งเป็นโปรแกรมจริงที่สามารถใช้งานได้ตามความต้องการ

เริ่มเก็บความต้องการจากผู้ใช้งาน หลังจากนั้นก็เริ่มวาดการพัฒนาตัวต้นแบบเริ่มต้นจากการออกแบบ (Design) อาจจะเป็นการออกแบบ Interface หลังจากนั้นนำตัวต้นแบบมาให้ผู้ใช้งานได้ทำการลองใช้งานหรือให้ผู้ใช้งานได้เห็นได้ประเมินการใช้งาน หากผู้ใช้งานยอมรับหรือว่าตัวต้นแบบในการพัฒนาเป็นไปตามความต้องการของผู้ใช้งาน ทีมพัฒนาก็จะดำเนินการพัฒนาระบบต่อซึ่งขั้นตอนต่อไป แต่หากผู้ใช้งานต้องการให้แก้ไขงานเพิ่มเติมหรือไม่ตรงความต้องการ ก็จะกลับไปทำการออกแบบอีกครั้งเพื่อให้ระบบซอฟต์แวร์ตรงตามความต้องการที่สุด

ข้อดีของซอฟต์แวร์แบบโปรโตไทป์

การพัฒนาระบบโดยใช้โปรโตไทป์ (Prototype Model) มีการทำสร้างต้นแบบ interface ให้ผู้ใช้งานดูก่อนเริ่มพัฒนาระบบ ทำให้ผู้ใช้งานได้ทำการลองใช้งานหรือให้ผู้ใช้งานได้เห็นได้ประเมินการใช้งาน ทำให้มีจุดแก้ไขน้อยจุด

ข้อด้อยของซอฟต์แวร์แบบโปรโตไทป์

เนื่องจากต้องทำต้นแบบซ้ำแล้วซ้ำอีกหลายๆครั้งทำให้กระบวนการนี้สิ้นเปลืองงบประมาณเป็นอย่างมาก ไม่เหมาะกับโครงการที่มีขนาดใหญ่

2.3 ทฤษฎีหรือเทคโนโลยีและเทคโนโลยีที่ใช้ในการพัฒนา

ภาษาพีเอชพี (PHP)

ภาษาพีเอชพี (PHP) ย่อมาจาก " Hypertext Preprocessor " เป็นภาษาสคริปต์ฝั่งเซิร์ฟเวอร์ (Server-Side Script) อีกภาษาหนึ่งเช่นเดียวกับเอเอสพี (ASP) ที่มีการทำงานที่เครื่องคอมพิวเตอร์ฝั่งเซิร์ฟเวอร์ (Server) ซึ่งรูปแบบในการเขียนคำสั่งการทำงานนั้นจะมีลักษณะคล้ายกับภาษาเพิร์ล (Perl) หรือภาษาซี (C) และสามารถใช้ร่วมงานกันกับภาษาเอชทีเอ็มแอล (HTML) ได้อย่างมีประสิทธิภาพ

หมายเหตุ ในการสร้างเว็บจะใช้สคริปต์ (Script) อยู่ 2 รูปแบบคือ

สคริปต์ฝั่งเซิร์ฟเวอร์ (Server-Side Script) เป็นลักษณะการทำงานบนเครื่องเซิร์ฟเวอร์ (Server) และแปลงออกมาเป็นภาษาเอชทีเอ็มแอล (HTML) เช่น ภาษาเอเอสพี (ASP), ภาษาซีจีไอ (CGI)

สคริปต์ฝั่งเครื่องผู้ใช้ (Client-Side Script) เป็นลักษณะการทำงานบนเครื่องผู้ใช้ (Client) เช่น จาวาสคริปต์ (JavaScript), วีบีสคริปต์ (VBScript)

ภาษาพีเอชพี (PHP) สามารถทำอะไรได้บ้าง

ความสามารถของภาษาพีเอชพี (PHP) นั้นสามารถที่จะทำงานเกี่ยวกับไดนามิกเว็บ (Dynamic Web) ได้ทุกรูปแบบ เหมือนกับภาษาซีจีไอ (CGI) หรือภาษาเอเอสพี (ASP) ไม่ว่าจะเป็นการจัดการดูแลระบบฐานข้อมูล ระบบรักษาความปลอดภัย การรับ – ส่ง คุกกี้ (Cookies)

โดยที่ภาษาพีเอชพี (PHP) นั้นสามารถที่จะติดต่อกับโปรแกรมจัดการฐานข้อมูลที่มีอยู่มากมาย ดังนี้

| | | |
|----------|-----------------|------------------|
| Adabas D | InterBase Solid | Microsoft Access |
| DBase | mSQL | Sybase |
| Empress | MySQL | Velocis |
| FilePro | Oracle | Unix dbm |
| Informin | PostgreSQL | MS SQL Server |

แต่ความสามารถที่พิเศษกว่านี้ก็คือ ภาษาพีเอชพี (PHP) สามารถที่จะติดต่อกับบริการต่างๆผ่านทางโพรโตคอล (Protocol) เช่น IMAP, SNMP, NNTP, POP3, HTTP และยังสามารถติดต่อกับเด้ารับ (Socket) ได้อีกด้วย

ทำไมถึงต้องเลือกภาษาพีเอชพี (PHP)

จากที่กล่าวไปข้างต้นแล้วว่าภาษาพีเอชพี (PHP) ก็เป็นภาษาสคริปต์ฝั่งเซิร์ฟเวอร์ (Server-Side Script) อีกภาษาหนึ่งเช่นเดียวกับภาษาเอเอสพี (ASP) แต่คุณสมบัติที่มากกว่าก็คือ

1. ภาษาพีเอชพี (PHP) นั้นสามารถรันบนระบบปฏิบัติการได้มากมายเช่น Windows, Unix, Linux และอื่นๆ
2. ภาษาพีเอชพี (PHP) นั้นรองรับกับการใช้งาน โปรแกรมเซิร์ฟเวอร์ (Server) จำลองมากมายเช่น Apache, IIS และอื่นๆ
3. ภาษาพีเอชพี (PHP) นั้นเป็นของฟรีที่สามารถไปหาดาวน์โหลด (Download) มาใช้งานได้ฟรีโดยไม่ต้องเสียค่าลิขสิทธิ์เหมือนภาษาเอเอสพี (ASP) ที่เว็บอย่างเป็นทางการของภาษาพีเอชพี (PHP) ที่ <http://www.php.net/>

แล้วจะเริ่มใช้งานภาษาพีเอชพี (PHP) ได้อย่างไร

1. ทำการติดตั้งตัวแปรภาษาพีเอชพี (PHP) ที่ได้ดาวน์โหลด (Download) มา
2. ทำการติดตั้งโปรแกรม Server จำลองบนเครื่องของคุณเพื่อรันภาษาพีเอชพี (PHP)
3. ทำการติดตั้งโปรแกรมจัดการฐานข้อมูลมายเอสคิวแอล (MySQL) ลงในเครื่องของคุณ

จากที่กล่าวไปข้างต้นว่าภาษาพีเอชพี (PHP) นั้นรองรับกับโปรแกรมการจัดการฐานข้อมูลหลายโปรแกรมแต่เนื่องจากโปรแกรมที่นิยมใช้มากที่สุดคือมายเอสคิวแอล (MySQL) ดังนั้นผมจึงจะสอนการใช้งานภาษาพีเอชพี (PHP) กับมายเอสคิวแอล (MySQL) และแอคเซส (Access)

องค์ประกอบของภาษาพีเอชพี (PHP)

ในส่วนนี้ผมจะกล่าวถึงโครงสร้างภาษาพีเอชพี (PHP) ซึ่งเราต้องเรียนรู้กันก่อนที่จะเข้าไปถึงในส่วนของการเขียนคำสั่ง โดยเฉพาะในส่วนของการกำหนดตัวแปร การเลือกเงื่อนไข คำสั่งทำซ้ำ คำสั่งวนรอบ คำสั่งคำนวณ การเลือกเงื่อนไข ซึ่งเป็นสิ่งที่เราต้องใช้อยู่ประจำ

โครงสร้างพื้นฐานของภาษาพีเอชพี (Basic Syntax)

อย่างที่กล่าวไปแล้วว่า ภาษาพีเอชพี (PHP) นั้นสามารถใช้งานร่วมกันกับภาษาเอชทีเอ็มแอล (HTML) ดังนั้นในการเขียนโค้ดจะต้องแยกส่วนกันให้ชัดเจน ว่าส่วนใดคือภาษาพีเอชพี (PHP) ส่วนใดคือภาษาเอชทีเอ็มแอล (HTML) ซึ่งสัญลักษณ์ที่นำมาใช้แยกโค้ด ภาษาพีเอชพี (PHP) มีดังนี้

แบบที่ 1.

```
<?
echo (" ASPThai.Net tutorial PHP ")
?>
```

เปิดคำสั่งด้วยแท็ก <? และปิดด้วย ?> ซึ่งก็คล้ายกับเอเอสพี (ASP)
ภายใต้แท็ก <? . . . ?> คือคำสั่งในรูปแบบของภาษาพีเอชพี (PHP)

แบบที่ 2.

```
<?php
echo (" ASPThai.Net tutorial PHP ")
?>
```

เปิดคำสั่งด้วยแท็ก <? ตามด้วยคำว่า php และปิดด้วย ?> ภายใต้แท็ก
<?php . . . ?> คือคำสั่งในรูปแบบของภาษาพีเอชพี (PHP)

แบบที่ 3.

```
<script language="php">
echo (" ASPThai.Net tutorial PHP ")
</script>
```

เปิดคำสั่งด้วยแท็ก <script language="php"> และปิดด้วย </script>
ภายใต้สคริป (script) คือคำสั่งในรูปแบบของภาษาพีเอชพี (PHP)

แบบที่ 4.

```
<%
echo (" ASPThai.Net tutorial PHP ")
ใครที่เคยเขียนภาษาเอเอสพี (ASP) คงคุ้นกับแท็กแบบนี้ดีคือเปิดคำสั่งด้วย
แท็ก <% และปิดด้วย %> แต่ถ้ามีการอ้างอิงถึงตัวแปรใดก็จะใช้ <%= $
ตัวแปร%> แต่แบบที่ 4 นี้จะใช้ได้กับภาษาพีเอชพี (PHP) เวอร์ชัน 3.0.4
```

ขึ้นไปเท่านั้นซึ่งต้องมีการแก้ไขไฟล์ php3.ini ที่เราเก็บไว้ที่ C:\Windows
ในบรรทัดดังต่อไปนี้

```
.....  
~~~~~
```

```
; Language Options ;
```

```
.....  
~~~~~
```

```
asp_tag = Off ; allow ASP-style <% %> tags
```

ให้เราทำการเปลี่ยนจาก Off เป็น On เพียงเท่านี้เราก็สามารถใช้งานได้
แล้ว

เอสคิวแอล (SQL :Structure Query Language)

เอสคิวแอล (SQL)

เอสคิวแอล (SQL) คือ ภาษาสอบถามข้อมูล หรือภาษาจัดการข้อมูลอย่างมีโครงสร้าง มีการพัฒนาภาษาคอมพิวเตอร์ และโปรแกรมฐานข้อมูลที่รองรับมากมาย เพราะจัดการข้อมูลได้ง่าย เช่น มาย เอสคิวแอล (MySQL), เอ็มเอส เอสคิวแอล (MsSQL), PostgreSQL หรือเอ็มเอส แอซเซส (MS Access) เป็นต้น สำหรับโปรแกรมฐานข้อมูลที่ได้รับความนิยมคือ มาย เอสคิวแอล (MySQL) เป็นโอเพนซอร์ส (Open Source) ที่ใช้งานได้ทั้งในลินุกซ์ (Linux) และวินโดวส์ (Windows)

เอสคิวแอล (SQL) เป็นภาษาที่ใช้ในการเขียนโปรแกรม เพื่อจัดการกับฐานข้อมูลโดยเฉพาะ เราสามารถแบ่งการทำงานได้เป็น 4 ประเภท ดังนี้

1. การเลือกข้อมูล (Select query) ใช้สำหรับดึงข้อมูลที่ต้องการ
2. การแก้ไขข้อมูล (Update query) ใช้สำหรับแก้ไขข้อมูล
3. การเพิ่มข้อมูล (Insert query) ใช้สำหรับการเพิ่มข้อมูล
4. การลบข้อมูล (Delete query) ใช้สำหรับลบข้อมูลออกไป

การเลือกข้อมูล (Select query)

ใช้ในการดึงข้อมูลในฐานข้อมูล จะมีการค้นหารายการจากตารางในฐานข้อมูล ตั้งแต่หนึ่งตารางขึ้นไป ตามเงื่อนไขที่สั่ง ผลลัพธ์ที่ได้จะเป็นเซตของข้อมูลที่สามารถสร้าง เป็นตารางใหม่ หรือใช้แสดงออกมาทางจอภาพเท่านั้น โดยมีรูปแบบดังนี้

Select รายละเอียดที่เลือก from ตารางแหล่งที่มา where กำหนดเงื่อนไขฐานข้อมูลที่เลือก Group by ชื่อคอลัมน์
ข้อสังเกต

1. ประโยคย่อย WHERE เราสามารถระบุเงื่อนไขได้โดยใช้โอเปอเรเตอร์ทั่วไป เช่น NOT < > = กรณีที่คอลัมน์เป็นตัวเลข เราก็สามารถระบุเงื่อนไขที่เป็นการคำนวณได้เช่น +, -, *, /

2. คำว่า Like ใช้กับค่าในคอลัมน์ประเภทตัวอักษรว่าตรงกับประโยคที่ต้องการหรือไม่ เราสามารถใช้เครื่องหมาย wildcard เช่น *,?%,% ในประโยคได้ ตามตัวอย่างข้างต้น
3. ในการคำนวณนั้นมีฟังก์ชัน COUNT,SUM,AVG.MIN,MAX ซึ่งสามารถนำมาใช้ได้ เช่น

```
Select Count(id) From stdinfo
```

หมายถึง ให้แสดงจำนวนรายการทั้งหมดในตาราง

4. ในกรณีที่ตารางสองตารางมีความสัมพันธ์กัน เราก็สามารถดูข้อมูลทั้งสองตารางพร้อมกันได้ เช่น ตารางที่ 1 ข้อมูลนักศึกษาชื่อ stdinfo กำหนดให้มีคอลัมน์รหัสประจำตัว (id) ,ชื่อ (fname), นามสกุล (lname) ตารางที่ 2 ข้อมูลเกี่ยวกับวิชาที่เรียน ชื่อ substd กำหนดให้มีคอลัมน์ชื่อวิชา (subject), รหัสประจำตัวอ้างอิง (rid) ,อาจารย์ผู้สอน (teacher) เราต้องการดูข้อมูลรหัสประจำตัว ชื่อ นามสกุล และชื่อวิชาที่เรียน เราจะใช้คำสั่งดังนี้

```
Select stdinfo.id, stdinfo.fname,stdinfo.lname,  
substd.subject
```

```
From stdinfo,substd Where stdinfo.id=substd.rid
```

การแก้ไขข้อมูล (Update query)

ใช้สำหรับการแก้ไขข้อมูลในตาราง โดยแก้ไขในคอลัมน์ที่มีค่าตรงตามเงื่อนไข มีรูปแบบดังนี้

```
Update ชื่อตาราง Set [ชื่อคอลัมน์=ค่าที่จะใส่เข้าไปในคอลัมน์นั้น ๆ]  
Where เงื่อนไข
```

เช่น จากตารางแสดงรายชื่อนักศึกษากรณีที่นักศึกษาชื่อ สมบัติ มั๊กน้อย ย้ายโปรแกรมวิชา จาก สังคมศึกษา ไปเป็นภาษาไทย เราใช้คำสั่งดังนี้

```
Select stdinfo Set program='ภาษาไทย' Where Fname='สมบัติ' and Lname='มั๊กน้อย'
```

การเพิ่มข้อมูล (Insert query)

ใช้ในการเพิ่มเติมข้อมูลใหม่ ๆ เข้าไปในฐานข้อมูล มีรูปแบบดังนี้
 Insert Into ชื่อตาราง [=ชื่อคอลัมน์1,2...] Values [ค่าที่จะใส่ลงในคอลัมน์ 1,2...]
 เช่น ต้องการเพิ่มรายชื่อนักศึกษา ที่มีรหัสประจำตัวเป็น 007 ชื่อ กมลวรรณ ศิริกุล
 โปรแกรมวิชา วิทยาศาสตร์ เราสามารถใช้คำสั่งดังนี้

```
Insert into stdinfo (id,fname,lname,programe) Values
('007','กมลวรรณ','ศิริกุล','วิทยาศาสตร์')
```

การลบข้อมูล (Delete query)

ใช้ลบข้อมูลออกจากตาราง มีรูปแบบดังนี้

```
Delete From ชื่อตาราง Where เงื่อนไข
```

เช่น ต้องการลบรหัสประจำตัวนักศึกษา 005 ออกจากฐานข้อมูล เราใช้
 คำสั่งดังนี้

```
Delete From stdinfo Where id='005'
```

มายเอสคิวแอล (MySQL)

มายเอสคิวแอล (MySQL) เป็นระบบจัดการฐานข้อมูลโดยใช้ภาษาเอสคิวแอล (SQL) แม้ว่า มายเอสคิวแอล (MySQL) เป็นซอฟต์แวร์โอเพนซอร์ส แตกต่างจากซอฟต์แวร์โอเพนซอร์สทั่วไป โดยมีการพัฒนาภายใต้บริษัทมายเอสคิวแอลเอบี (MySQL AB) ในประเทศสวีเดน โดยจัดการมายเอสคิวแอล (MySQL) ทั้งในแบบที่ให้ใช้ฟรี และแบบที่ใช้ในเชิงธุรกิจ

มายเอสคิวแอล (MySQL) สร้างขึ้นโดยชาวสวีเดน 2 คน และชาวฟินแลนด์ ชื่อ David Axmark, Allan Larsson และ Michael “Monty” Widenius.

ปัจจุบันบริษัทซันไมโครซิสเต็มส์ (Sun Microsystems, Inc.) เข้าซื้อกิจการของมายเอสคิวแอลเอบี (MySQL AB) เรียบร้อยแล้ว ฉะนั้นผลิตภัณฑ์ภายใต้มายเอสคิวแอลเอบี (MySQL AB) ทั้งหมดจะตกเป็นของซันไมโครซิสเต็มส์ (Sun Microsystems, Inc.)

ชื่อ “MySQL” อ่านออกเสียงว่า “มายเอสคิวเอล” หรือ “มายเอสคิวแอล” (ในการอ่านอักษร L ในภาษาไทย) ซึ่งทางซอฟต์แวร์ไม่ได้อ่าน มายซีควอล หรือ มายซีควอล เหมือนกับซอฟต์แวร์จัดการฐานข้อมูลตัวอื่น

คำสั่งภาษาเอสคิวแอล (SQL Statement)

- ดึงข้อมูล (Data Retrieval)

SELECT ใช้ดึงข้อมูลจาก Database

- ภาษาจัดการข้อมูล (DML : Data Manipulation Language)

INSERT Enter new row(s) into table

UPDATE Changes existing row(s) in table

DELETE Removes unwanted row(s) from table

- ภาษานิยามข้อมูล (DDL : Data Definition Language)

CREATE ใช้สร้าง object ใหม่ใน database เช่น สร้าง table ใหม่ เป็นต้น

ALTER ใช้เปลี่ยนแปลงโครงสร้างของ object ที่มีอยู่ใน Database เช่น เพิ่ม column ใน table เป็นต้น

DROP ใช้ลบ object ที่ไม่ต้องการออกจาก database

RENAME ใช้เปลี่ยนชื่อ object

TRUNCATE ใช้ลบ object ออกจาก database

- ภาษาควบคุมข้อมูล (DCL : Data Control Language)

GRANT ใช้กำหนดสิทธิการทำงานต่าง ๆ ให้กับผู้ใช้ฐานข้อมูล

REVOKE ใช้ถอดถอนสิทธิต่าง ๆ ของผู้ใช้ในฐานข้อมูล

- ควบคุมการทำธุรกรรม (Transaction Control)

COMMIT คำสั่งให้มีการบันทึกความเปลี่ยนแปลงจากคำสั่ง DML ที่ทำไว้ให้เก็บเป็นการถาวร

ROLLBACK คำสั่งให้คืนสภาพข้อมูลให้กลับสู่สภาพก่อนที่จะถูกเปลี่ยนแปลงจากคำสั่ง DML ที่ทำไว้

SAVEPOINT คำสั่ง ให้เก็บสถานะข้อมูลในปัจจุบันที่ได้เปลี่ยนแปลงไว้ก่อนเพื่อให้สามารถสั่ง ROLLBACK กลับมายังจุดนั้นได้

เอชทีเอ็มแอล (HTML : Hypertext Markup Language)

ความหมายของภาษาเอชทีเอ็มแอล

HTML คือ ภาษาหลักที่ใช้ในการเขียนเว็บเพจ โดยใช้ Tag ในการกำหนดการแสดงผล HTML ย่อมาจากคำว่า Hypertext Markup Language โดย Hypertext หมายถึง ข้อความที่เชื่อมต่อกันผ่านลิง (Hyperlink) Markup language หมายถึงภาษาที่ใช้ Tag ในการกำหนดการแสดงผลสิ่งต่างๆที่แสดงอยู่บนเว็บเพจ ดังนั้น HTML จึงหมายถึง ภาษาที่ใช้ Tag ในการกำหนดการแสดงผลเว็บเพจที่ต่างก็เชื่อมถึงกันใน Hyperspace ผ่าน Hyperlink

ประวัติความเป็นมาของภาษาเอชทีเอ็มแอล

ปี ค.ศ. 1961 อินเทอร์เน็ตได้เกิดและเติบโตขึ้น พร้อมกับภาษาคอมพิวเตอร์และโปรโตคอล (Protocol) จำนวนมาก เพื่รองรับกับ การเติบโตอย่างรวดเร็วของอินเทอร์เน็ต หนึ่งในภาษาคอมพิวเตอร์และโปรโตคอลนั้นคือ ภาษา HTML และ TCP/IP (Transmission Control Protocol/ Internet Protocol) ทั้งนี้เพราะ World Wide Web แม้จะเป็นเพียงส่วนหนึ่งของอินเทอร์เน็ต แต่ได้รับความนิยมอย่างสูง และรวดเร็ว โปรโตคอล HTTP จึงได้รับการพัฒนาเพื่อสนับสนุนภาษา HTML ซึ่ง ใช้ในการจัดเก็บเอกสารบน World Wide Web

ภาษา HTML ในยุคแรกเริ่ม ถูกใช้เพื่อจัดทำรูปแบบตัวอักษรบนเอกสารประเภท Hypertext ความสามารถจึงยังคงจำกัดอยู่มาก

ในอดีต สิ่งที่ทำให้ผู้เขียนภาษา HTML ประสบกับปัญหามากมาย คือ การที่โปรแกรม browser หลักๆ ยังคงมีความแตกต่างทาง ด้านเทคโนโลยีกันอย่างมาก และต่างก็ได้พัฒนา ภาษา HTML ของตนเพิ่มเติมจากมาตรฐานในเวอร์ชัน 3.2 กันเอง โดยไม่ได้มีการตกลงระหว่างกัน เป็นผลให้ผู้เขียนภาษา HTML โดยถือตามมาตรฐาน หรืออ้างอิงค่ายใดค่ายหนึ่งไม่สามารถจะแสดงรูปแบบของเอกสาร ให้เป็นอย่างที่ต้องการได้ ในปัจจุบัน ปัญหาดังกล่าวกลับไม่ลดน้อยลงไป แม้ว่าจะได้มีการจัดตั้งคณะกรรมการ เพื่อกำหนดมาตรฐาน ให้โปรแกรม browser ในเวอร์ชันใหม่ๆ นำเอา HTML เวอร์ชัน 4.0 เป็นมาตรฐานก็ตาม แต่ปัญหายังคงมีอยู่ต่อไป ด้วยเหตุที่คณะกรรมการชุดดังกล่าวไม่สามารถผลักดันให้โปรแกรม browser รับเอามาตรฐานดังกล่าวไปใช้อย่างรวดเร็วเพียงพอ

ข้อดีของภาษาเอชทีเอ็มแอล

ถูกพัฒนาขึ้นเพื่อใช้งานกับเว็บโดยเฉพาะ และใช้รหัสข้อมูลแบบธรรมดา ทำให้ไฟล์ HTML สามารถใช้ได้กับทุกๆ โปรแกรม ภาษามาตรฐานเปิดไฟล์ที่สามารถอ่านเข้าใจ ใช้งานระบบ Hypertext ได้ สามารถทำงานกับมัลติมีเดีย

ข้อเสียของภาษาเอชทีเอ็มแอล

การเพิ่มเติมออบเจกต์หรือส่วนประกอบต่างๆ ใน HTML มีข้อจำกัด และมีข้อจำกัดในการใช้คำสั่ง เนื่องจากไม่สามารถสร้างคำสั่งใหม่ๆ ขึ้นมาใช้เองได้ จึงต้องใช้คำสั่งเก่าที่มีอยู่เท่านั้น

ซีเอสเอส (CSS)

ความหมายของภาษาพีเอชพี

CSS ย่อมาจากคำว่า Cascading Style Sheets เป็นวิธีการกำหนดการแสดงผลของสิ่งต่างบนเว็บไซต์ เช่น ลักษณะอักษร ขนาด สี พื้นหลัง หน้าตา ระยะห่าง ฯลฯ อย่างที่ต้องการ ด้วยการกำหนดคุณสมบัติให้กับ Element ต่างๆ ของ HTML เช่น <body>, <p>, <h1> เป็นต้น

ข้อดีของภาษาพีเอชพี

การใช้ CSS ในการจัดรูปแบบการแสดงผล จะช่วยลดการใช้ภาษา HTML ในการตกแต่งเอกสารเว็บเพจ ทำให้ code ภายในเอกสาร HTML เหลือเพียงส่วนเนื้อหา ทำให้เข้าใจง่ายขึ้น การแก้ไขเอกสารทำได้ง่ายและรวดเร็ว เมื่อ code ภายในเอกสาร HTML ลดลง ทำให้ขนาดไฟล์เล็กลง จึงดาวน์โหลดได้เร็ว สามารถกำหนดรูปแบบการแสดงผลจากคำสั่ง style sheet ชุดเดียวกัน ให้มีผลกับเอกสาร HTML ทั้งหมด หรือทุกหน้าได้ ทำให้เวลาแก้ไขหรือปรับปรุงทำได้ง่าย ไม่ต้องไล่ตามแก้ที่ HTML tag ต่างๆ ทั้งทั้งเอกสาร สามารถควบคุมการแสดงผลให้เหมือนกัน หรือใกล้เคียงกันได้ ในหลาย Web Browser สามารถกำหนดการแสดงผลในรูปแบบที่เหมาะสมกับสื่อชนิดต่างๆ ไม่ว่าจะเป็นการแสดงผลบนหน้าจอ, บนกระดาษ เมื่อสั่งพิมพ์, บนมือถือ โดยที่เป็นเนื้อหาเดียวกัน ทำให้เป็นเว็บไซต์ที่มีมาตรฐาน ปัจจุบันการใช้ attribute ของ HTML ตกแต่งเอกสารเว็บเพจ

ข้อเสียของภาษาพีเอชพี

ออกแบบและเรียนรู้การใช้งานได้ยาก อาจจะไม่สามารถใช้ได้กับโปรแกรมที่ใช้ดูหน้าเว็บหรือบราวเซอร์เวอร์ชันเก่าๆ Learning curve สูง เพราะว่ามันเป็นเรื่องใหม่ของใครหลายคน จากเดิม รู้แค่ html จัดตารางก็ได้แล้ว ต้องมานั่งศึกษา CSS เพิ่มอีก ไม่สามารถใช้กราฟฟิคหรือรูปภาพ หรือรูปแบบเฉพาะบางอย่างในหน้าเว็บได้ มีข้อจำกัดในการออกแบบ การเขียน Layout อยู่ในขั้นยากกว่าปกติ ไม่เหมาะสำหรับเว็บที่นำเสนอเนื้อหาเยอะๆ เพราะเมื่อผู้ใช้งานเปลี่ยนขนาดตัวหนังสือ เลย์เอาท์หน้าเว็บจะไม่สามารถขยายตามได้ ทำให้ตัวหนังสือซ้อนทับเลย์เอาท์

2.4 ทฤษฎีของการพัฒนาระบบแบบเจ็ทโอเรียนเต็ต (OOP: Object Oriented Programming)

ความหมายของการพัฒนาระบบแบบเจ็ทโอเรียนเต็ต

เป็นวิธีการเขียนโปรแกรมซึ่งจัดดำเนินการกับกลุ่มของ ออบเจ็กต์ (Object) ที่มีอยู่ในโปรแกรม ออบเจ็กต์ เป็นชนิดของข้อมูลซึ่งประกอบไปด้วยกลุ่มของข้อมูล (data) และกลุ่มของฟังก์ชัน (Function) โดยการใช้ข้อมูลและฟังก์ชันเหล่านี้ แต่ละออบเจ็กต์จะทำงาน 1 งานได้สมบูรณ์ (ทั้งนี้เนื่องจากตัวออบเจ็กต์เองประกอบไปด้วยข้อมูลและฟังก์ชัน) ออบเจ็กต์ เป็นสมาชิกของ ตัวแปรคลาส (class variable) มีลักษณะเป็นโมดูล (modularity) ซึ่งประกอบไปด้วย ตัวแปร ชนิดต่าง ๆ ที่สัมพันธ์กัน และประกอบด้วย ฟังก์ชัน ต่าง ๆ โดยที่ คลาส (class) จะห่อหุ้มข้อมูลและฟังก์ชันรวมไว้ด้วยกันมีลักษณะที่เรียกว่า encapsulation ดังนั้นจึงมีความสะดวกในการใช้งาน สามารถป้องกันส่วนอื่น ๆ ของโปรแกรมไม่ให้เข้าถึงตัวแปรชนิดโลคอลภายในคลาสได้อย่างดีเยี่ยม ดังนั้น การเขียนโปรแกรมแบบ OOP คือ การสร้างและ การเรียกใช้ออบเจ็กต์ให้ทำงานตามที่เราต้องการ ในการเรียกใช้ออบเจ็กต์นั้น จะสนใจเฉพาะการทำงานของออบเจ็กต์เท่านั้น ไม่จำเป็นต้องสนใจรายละเอียดภายในของออบเจ็กต์ว่าเป็นอย่างไรการใช้ออบเจ็กต์ของโปรแกรมจะมีลักษณะ คล้ายกับการใช้สิ่งของในชีวิตประจำวัน เช่น การใช้โทรทัศน์ เราสามารถใช้ได้โดยไม่จำเป็นต้องรู้ว่าภายในเครื่องโทรทัศน์มี "ส่วนประกอบ" อะไรบ้าง และไม่จำเป็นต้องรู้ว่าแต่ละส่วนประกอบทำงานอย่างไร รู้เพียงแค่วิธีใช้ เช่น วิธีเปิด วิธีเปลี่ยนช่อง วิธีปรับเสียง วิธีปรับสี วิธีตั้งเวลา วิธีปิดเครื่อง เป็นต้น ลักษณะของโปรแกรมแบบ OOP ก็มีลักษณะคล้ายกับการใช้โทรทัศน์ในชีวิตประจำวัน

คุณสมบัติของการพัฒนาระบบแบบเจ็ทโอเรียนเต็ด

Inheritance

เป็นคุณสมบัติที่ว่า Class ต้องสามารถสืบทอด ได้เช่นเดียวกับภาษาโปรแกรม ที่กำหนด เป็น Component ทั้งที่ มองเห็นและมองไม่เห็น ก็ต้องสืบทอดได้ โดย ดีไรว์คลาส ก็คือ Class ที่ถูกสืบทอดมานั้น สามารถเพิ่มเติม Method ได้ตามความเหมาะสม

Polymorphism

เป็นคุณสมบัติที่สามารถเปลี่ยนแปลงความสามารถของ Class ให้เข้ากับสภาพแวดล้อม ตัวอย่างง่ายๆ เช่น เราสร้าง Class ที่ชื่อว่า Shape ซึ่งจะใช้สร้าง Object เป็นรูปทรงต่างๆ เช่น วงกลม, สามเหลี่ยม และ สี่เหลี่ยม เป็นต้น แล้วเราก็มี Method Area เพื่อหาพื้นที่ของรูปทรงต่างๆ แน่นอนว่า Method Area ของการเรียกใช้งานแต่ละครั้ง ต้องคำนึงด้วยว่าระบุ Property ของรูปทรงว่าเป็นรูปทรงอะไร ซึ่งจะทำให้มีวิธีการคำนวณที่แตกต่างกันนั่นเอง

Overloading

คือ เมธอดที่มีชื่อซ้ำกันในคลาส ตัวแปรก็สามารถตั้งซ้ำกันได้โปรแกรม แต่ต้องอยู่กันคนละขอบเขต เป็นความสามารถของตัวแปรภาษา ที่จะตรวจสอบ Signature ของ Function แล้วแปลงออกมาได้อย่างถูกต้อง

Overriding

คือ การแทนที่รายละเอียดการทำงานของคลาสแม่ ด้วยรายละเอียดการทำงานของคลาสลูก

Abstract class

คือ class ที่ไม่ระบุรายละเอียดการทำงาน

Abstract method

คือ method ที่มีคำว่า 'abstract' อยู่หน้าชื่อ และมีเพียงชื่อของ method โดยไม่มีตัวโปรแกรม และหากคลาสใด มี method ใด method หนึ่งเป็น

Abstract คลาสนั้นจะต้องเป็น Abstract ด้วย (เรียกว่า Abstract class) และต้องมี คำว่า 'abstract' อยู่หน้าชื่อคลาส ในตอนกำหนดคลาสด้วย มิฉะนั้นจะไม่ผ่านการคอมไพล์

2.5 ทฤษฎีของการพัฒนาระบบแบบเอ็มวีซี (MVC)

ความหมายของการพัฒนาระบบแบบเอ็มวีซี

MVC คือหลักการออกแบบ รูปแบบหนึ่ง ซึ่งเป็นที่นิยมมากในการออกแบบ และประยุกต์ใช้กับเว็บแอปพลิเคชัน ชื่อเต็มๆของคือ Model-View-Controller ซึ่ง MVC นี้จะแบ่งแอปพลิเคชันตามบทบาทหน้าที่ (Roles of Objects) โดยแบ่งออกเป็น 3 บทบาทด้วยกันคือ Model (M), View (V) และ Controller (C)

Model

Model คือออบเจกต์ที่ทำหน้าที่เป็นตัวแทนของข้อมูล ไม่ว่าข้อมูลจะถูกจัดเก็บในรูปแบบใดในระบบฐานข้อมูลหรือในไฟล์ เมื่อข้อมูลนั้นถูกโหลดเข้ามาในแอปพลิเคชัน จะเปลี่ยนมันให้อยู่ในรูปของออบเจกต์ และเรียกบทบาทของออบเจกต์นี้ว่า "Model" ยกตัวอย่างเช่นออบเจกต์ Customer, Employee, Product เป็นต้น ฟังก์ชันแล้วออบเจกต์นี้คล้ายคลึงกับ Entity เลย จริงๆแล้วก็อันเดียวกันครับ เพียงแต่ใน Design Pattern นี้เค้าเรียกว่า Model

View

View คือออบเจกต์ที่ทำหน้าที่ในการแสดงผล เช่น แสดงหน้าฟอร์มกรอกข้อมูล, แสดงผลลัพธ์ของการค้นหา เป็นต้น ซึ่งมันจะประกอบไปด้วย Presentation Logic เป็นหลัก การแสดงผลก็มีโลจิกของ view อยู่ เช่น การเรียงลำดับสินค้าตามราคา เป็นต้น อันที่จริงแล้ว View ก็คือ User Interface (UI) นั่นเอง

Controller

Controller คือออบเจกต์ที่ทำหน้าที่รับคำสั่ง เช่น คำสั่งค้นหาข้อมูล, คำสั่งลบข้อมูล เป็นต้น เมื่อมีคำสั่ง (หรือ Request) เข้ามาจากผู้ใช้ (User) ออบเจกต์ตัวนี้จะทำหน้าที่รับคำสั่งและเรียกใช้ออบเจกต์ตัวอื่นๆ (Model และ View) ให้ทำงานร่วมกัน ดังนั้นจึงประกอบไปด้วย Control Logic

