บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องประกอบไปด้วย

2.1 ทฤษฎีด้านระบบฐานข้อมูล

- ความรู้เบื้องต้นเกี่ยวกับฐานข้อมูล

- โครงสร้างข้อมูลและสถาปัตยกรรมของระบบฐานข้อมูล

- ฐานข้อมูลเชิงสัมพันธ์

- แผนผังความสัมพันธ์ของข้อมูล

2.2 ทฤษฏีกระบวนการผลิตซอฟต์แวร์

- การบวนการผลิตซอฟต์แวร์ (Software Process)

2.3 ทฤษฏีหรือเทคโนโลยีและเทคโนโลยีที่ใช้ในการพัฒนา

- ภาษาพีเอชพี (PHP)

- ภาษาเอสคิวแอล (SQL)

2.4 งานวิจัยหรือผลงานที่เกี่ยวข้อง

2.1 ทฤษฏีด้านระบบฐานข้อมูล

ความรู้เบื้องต้นเกี่ยวกับฐานข้อมูล (INTRODUCTION TO DATABASE)

ความหมายของฐานข้อมูล

ฐานข้อมูล หมายถึง การรวบรวมข้อมูลที่เกี่ยวข้องกัน โดยกำหนดรูปแบบในการจัดเก็บอย่างเป็นระบบ เพื่อให้ผู้ใช้สามารถจัดการและเรียกใช้ข้อมูลได้อย่างมีประสิทธิภาพ

ระบบการจัดการฐานข้อมูล

จะทำหน้าที่ในการควบคุมดูแลการสร้างและการเรียกใช้ข้อมูลโดยไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างฐานข้อมูลซึ่งประกอบด้วยโครงสร้าง(Structure) และการใช้งาน (Operation)

- โครงสร้าง หมายถึง โครงสร้างของระบบฐานข้อมูล

- การใช้งาน หมายถึง วิธีการที่จะให้ผู้ใช้สามารถเรียก, แก้ไขข้อมูลในระบบฐานข้อมูลได้

จุดมุ่งหมายของระบบจัดการฐานข้อมูล

เพื่อจัดการควบคุมฐานข้อมูลรวมถึงการพัฒนาข้อมูลให้ทันสมัย ตรงกับความต้องการของผู้ใช้งาน

ลักษณะของการจัดการฐานข้อมูล

แบ่งได้ 2 ลักษณะใหญ่ๆ คือ

1. ลักษณะข้อมูลทั่วไป จะจัดเก็บแฟ้มข้อมูลธรรมดาที่มีเฉพาะค่าของข้อมูล ส่วนรายละเอียดที่เกี่ยวข้องกับตัวข้อมูล จะถูกระบุอยู่ในแฟ้มข้อมูล โปรแกรมซึ่งเขียนขึ้นโดยผู้เขียนโปรแกรมเอง
2. ลักษณะข้อมูลแบบเมทาดาตา (Meta data) จะจัดเก็บรายละเอียดข้อมูลเกี่ยวกับตัวข้อมูลไว้ที่เดียวกัน จัดแยกเป็น 2 ส่วน คือ ข้อมูลและเมทาดาตา (Meta data) ซึ่งบางโปรแกรม เรียกว่า พจนานุกรม (Data dictionary) โดยจะมีระบบจัดการฐานข้อมูลทำหน้าที่ควบคุมจัดการ

DATA  
DESCRIPTION  
--------------------

DATA

APPLICATION

DATAFILE

ภาพที่ 2.1.1 ลักษณะข้อมูลทั่วไป ในโปรแกรมประยุกต์ใช้งานทั่วไป

DATA

DESCRIPTION

PROCEDURE A

DATA

DESCRIPTION

PROCEDURE A

ภาพที่ 2.1.2 ลักษณะเมทาดาตา (Meta data) ในโปรแกรมประยุกต์ใช้ฐานข้อมูล

ประโยชน์ของฐานข้อมูล

* ลดความซับซ้อนของข้อมูล

การจัดเก็บข้อมูลแบบแฟ้มข้อมูลอาจจะทำให้ข้อมูลถูกเก็บไว้หลายๆแห่งทำให้เกิดความซับซ้อนการนำข้อมูลรวบรวมเก็บไว้ในฐานข้อมูลเดียวกันจะช่วยลดปัญหาความซับซ้อนได้

* หลีกเลี่ยงความขัดแย้งของข้อมูล

การจัดเก็บข้อมูลแบบแฟ้มข้อมูลหลายแฟ้มข้อมูล จะก่อให้เกิดความขัดแย้งของข้อมูลได้ข้อมูลจะถูกจัดเก็บรวมกัน จึงสามารถเลี่ยงการซ้ำกันของข้อมูลได้

* สามารถใช้ข้อมูลร่วมกันได้

ผู้ใช้ข้อมูลสามารถใช้ข้อมูลเดียวกันได้ซึ่งอาจมาจากแฟ้มข้อมูลที่แตกต่างกันโดยฐานข้อมูลจะมีลักษณะของการแชร์ข้อมูล (Share Data)

* การรักษาความถูกต้องเชื่อถือได้ของข้อมูล

ระบบการจัดการฐานข้อมูล สามารถที่จะระบุกฎเกณฑ์เพื่อควบคุมความผิดพลาดที่อาจจะเกิดขึ้นได้ในลักษณะข้อมูลที่ถูกต้อง (Valid Data)

* กำหนดระดับความปลอดภัยของข้อมูลได้

ผู้บริหารระบบสามารถกำหนดสิทธิ, ระดับการเรียกใช้ข้อมูลของผู้ใช้แต่ละคนได้

* กำหนดความเป็นมาตรฐานเดียวกัน

ผู้บริหารระบบฐานข้อมูลสามารถกำหนดสิทธิ, ระดับการเรียกใช้ข้อมูลของผู้ใช้แต่ละคนได้

* เกิดความเป็นอิสระของข้อมูลและโปรแกรม

ระบบฐานข้อมูล โปรแกรมประยุกต์จะใช้ทำงานอย่างมีระบบ โดยจะเชื่อมโยงฐานข้อมูลโดยที่จะสามารถใช้ข้อมูลได้อย่างอิสระโดยวิธีการแปลงส่ง(Mapping) และลักษณะวิว (View)

ข้อเสีย

* มีต้นทุนสูง
* การเสี่ยงต่อการหยุดชะงักของระบบ

เนื่องจากข้อมูลอาจถูกเก็บไว้ในลักษณะเป็นศูนย์รวม ความผิดพลาดของการทำงานบางส่วนอาจมีผลกระทบกับระบบ ทำให้ระบบหยุดชะงักได้

องค์ประกอบของระบบฐานข้อมูล

1. ฮาร์ดแวร์ ( Hardware )

ระบบฐานข้อมูล จะต้องมีฮาร์ดแวร์ที่มีประสิทธิภาพ ในเรื่องของหน่วยประมวลผลกลางที่ต้องการความเร็วสูง มีขนาดหน่วยความจำหลักที่มากพอ และหน่วยความจำสำรองที่จะจัดเก็บข้อมูลเพียงพอกับความต้องการของระบบ รวมถึงอุปกรณ์นำข้อมูลเข้าและออกรายงานตามคุณสมบัติที่ระบบต้องการ เพื่อให้ได้ระบบการประมวลผลมีประสิทธิภาพ

2. ซอฟต์แวร์ (Software) ระบบฐานข้อมูล

จะต้องมีโปรแกรมที่ทำหน้าที่ควบคุมดูแลการสร้าง การสืบค้นหาข้อมูล ปรับเปลี่ยน การแก้ไขโครงสร้าง การจัดทำรายงาน โดยสื่อกลางระหว่างผู้ใช้กับระบบงานและโปรแกรมประยุกต์ต่างๆ ที่มีอยู่รวมถึงการสื่อสารข้อมูลระหว่างเครื่องด้วย คุณสมบัติและหน้าที่ของโปรแกรมการจัดการฐานข้อมูล

- สามารถกำหนดและเก็บโครงสร้างฐานข้อมูล

- สามารถรับและจัดเก็บข้อมูลไว้ในฐานข้อมูล

- สามารถดูแลรักษาข้อมูลในฐานข้อมูล

- สามารถประสานงานกับระบบปฏิบัติการ

- สามารถจัดทำข้อมูลสำรองและการกู้คืนข้อมูล

- สามารถควบคุมความปลอดภัยในระบบฐานข้อมูล

- สามารถควบคุมการใช้งานของผู้ใช้ระบบในเวลาเดียวกัน

- สามารถทำพจนานุกรมข้อมูล

- สามารถติดต่อเชื่อมโยงในลักษณะของระบบเครือข่าย

โครงสร้างข้อมูลและสถาปัตยกรรมของระบบฐานข้อมูล

(DATA STRUCTURE AND DATABASE SYSTEM ARCHITECTURE)

ลักษณะของโครงสร้างข้อมูล สามารถแบ่งเป็นหน่วยข้อมูลได้ดังนี้

บิท (Bit)

หมายถึง หน่วยของข้อมูลที่มีขนาดเล็กที่สุดในระบบแฟ้มข้อมูล

ไบร์ท (Byte)

หมายถึง หน่วยของข้อมูลที่ประกอบด้วยบิทหลายๆ บิทมารวมกันโดยมีขอบเขตที่แทนความหมาย หมายถึงสิ่งหนึ่ง เช่น รหัสประจำตัว เป็นตัวแทนค่าตัวเลขที่มีความหมายตามขอบเขต

โครงสร้างข้อมูลและสถาปัตยกรรมของระบบฐานข้อมูล

(DATA STRUCTURE AND DATABASE SYSTEM ARCHITECTURE)

ลักษณะของโครงสร้างข้อมูล สามารถแบ่งเป็นหน่วยข้อมูลได้ดังนี้

- บิท (bit) หมายถึง หน่วยของข้อมูลที่มีขนาดเล็กที่สุดในระบบแฟ้มข้อมูล

- ไบร์ท (byte) หมายถึง หน่วยของข้อมูลที่ประกอบด้วยบิทหลายๆ บิทมารวมกันโดยมีขอบเขตที่แทนความหมาย หมายถึงสิ่งหนึ่ง เช่น รหัสประจำตัว เป็นตัวแทนค่าตัวเลขที่มีความหมายตามขอบเขต

สถาปัตยกรรมของฐานข้อมูล

ระบบฐานข้อมูล เป็นโครงสร้างข้อมูลที่ ใช้หลายกลุ่ม ต้องการเรียกใช้ข้อมูลบางส่วนหรือทั้งหมด โดยจะแตกต่างไปตามงานที่ปฏิบัติ ลักษณะของผู้ใช้ก็จะมีหลากหลายโดยเฉพาะผู้ใช้ทั่วไป (End user) เป็นผู้ใช้ที่ไม่มีความจำเป็นต้องรู้รายละเอียดของการจัดเก็บข้อมูล ดังนั้น เพื่อที่จะให้การทำงานเป็นไปตามระดับของผู้ใช้จึงสามารถแบ่งระดับของข้อมูลได้ 3 ระดับ คือ

1. ระดับภายนอก (External Level)

เป็นระดับของข้อมูลที่ผู้ใช้แต่ละคนลองเก็บข้อมูล ตามระบบงานที่ผู้ใช้ได้รับผิดชอบงานนั้น ผู้ใช้ส่วนใหญ่เป็น Operator หรือ โปรแกรมเมอร์

1. ระดับแนวคิด (Conceptual Level)

เป็นโครงร่างที่อธิบายถึงฐานข้อมูลโครงสร้างข้อมูลความสัมพันธ์ของข้อมูล กฎเกณฑ์และข้อจำกัดที่มีต่อฐานข้อมูล โดยจะผ่านการวิเคราะห์และออกแบบจาก

1. ผู้บริการฐานข้อมูล คือ DBA

เป็นโครงร่างที่อธิบายถึงการจัดเก็บข้อมูลจริงจัดเก็บในรูปแบบใดวิธีการเข้าถึงข้อมูลวิธีใด เช่น การแฮสชิ่ง (Hashing Function), อินเด็กซ์ (Indexing) เป็นต้น

ข้อมูลแต่ละระดับจะมีการเชื่อมโยงโดยจะถูกจัดการด้วยระบบการจัดการฐานข้อมูล โดยจะแปลงความหมายข้อมูลจากระดับหนึ่ง ไปยังอีกระดับหนึ่งที่เรียกกันว่า การแปลงส่ง (Mapping)

ตัวอย่าง ID\_NUMBER มีจำนวน 8 หลักเป็นต้น

- เรคคอร์ด (Record)

หมายถึง หน่วยของข้อมูลที่ประกอบด้วยหลายๆ ฟิลด์มารวมกัน ที่มีความสัมพันธ์กัน เช่น รหัสประจำตัว ชื่อ ที่อยู่ เป็นต้น

- แฟ้มข้อมูล (File)

หมายถึง หน่วยของจ้อมูลที่ประกอบด้วยหลายๆ เรคคอร์ดมารวมกัน

หน่วยการจัดเก็บของฐานข้อมูลเชิงสัมพันธ์ สามารถเปรียบเทียบกับแฟ้มข้อมูล โดยอาศัยศัพท์ดังนี้

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

ตารางที่ 2.1.3 หน่วยการจัดเก็บของฐานข้อมูลเชิงสัมพันธ์

- รีเลชั่น

หมายถึง การแทนข้อมูลของตารางในฐานข้อมูลเชิงสัมพันธ์ ซึ่งอาจเกี่ยวกับคน สถานที่ สิ่งของ เช่น รีเลชั่นของข้อมูลสินค้า เป็นต้น

- ทูเพิล

หมายถึง การแทนค่าข้อมูลในแต่ละแถว หรือ อาจเรียกกันว่าระเบียน (record)

- แอททริบิวต์

หมายถึง การแทนค่าขอบเขตของข้อมูลในแต่ละคอลัมน์

- คาร์ดินาลลิตี้

หมายถึง จำนวนแถวของข้อมูลในแต่ละตาราง

- โดเมน

หมายถึง ขอบเขตหรือค่าที่ควรจะเป็นของข้อมูลในแอททริบิวต์หนึ่ง

ฐานข้อมูลเชิงสัมพันธ์เป็นการจัดเก็บรวมรวบข้อมูลในลักษณะ 2 มิติ คือ แถวและคอลัมน์ โดยมีการจัดเก็บอย่างมีระบบและกฎเกณฑ์

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| รหัสวิชา | ชื่อวิชา | หน่วยกิต | จำนวนชั่วโมง |  |
| 013002 | การจัดการข้อมูลทั่วไป | 1 | (2) | Tuple |
| 013003 | การจัดทำเอกสาร | 3 | (1) |  |
| 013004 | การจัดการฐานข้อมูล | 3 | (2) |
|  |  |  |  |
|  | Attribute |  | |

CARDINALITY

ภาพที่ 2.1.4 ตารางตัวอย่างของการแสดงข้อมูลของรายวิชา

สถาปัตยกรรมของฐานข้อมูล

ระบบฐานข้อมูล เป็นโครงสร้างข้อมูลที่ใช้หลายกลุ่ม ต้องการเรียกใช้ข้อมูลบางส่วนหรือทั้งหมด โดยจะแตกต่างไปตามงานที่ปฏิบัติ ลักษณะของผู้ใช้ก็จะมีหลากหลายโดยเฉพาะผู้ใช้ทั่วไป (End user) เป็นผู้ใช้ที่ไม่มีความจำเป็นต้องรู้รายละเอียดของการจัดเก็บข้อมูล ดังนั้น เพื่อที่จะให้การทำงานเป็นไปตามระดับของผู้ใช้จึงสามารถแบ่งระดับของข้อมูลได้ 3 ระดับ คือ

1. ระดับภายนอก (External Level)

เป็นระดับของข้อมูลที่ผู้ใช้แต่ละคนลองเก็บข้อมูล ตามระบบงานที่ผู้ใช้ได้รับผิดชอบงานนั้น ผู้ใช้ส่วนใหญ่เป็น Operator หรือ โปรแกรมเมอร์

2. ระดับแนวคิด (Conceptual Level)

เป็นโครงร่างที่อธิบายถึงฐานข้อมูล โครงสร้างข้อมูล ความสัมพันธ์ของข้อมูล กฎเกณฑ์และข้อจำกัดที่มีต่อฐานข้อมูล โดยจะผ่านการวิเคราะห์และออกแบบจากผู้บริการฐานข้อมูล คือ DBA

3. ระดับภายใน (Internal /Physical Level)

เป็นโครงร่างที่อธิบายถึงการจัดเก็บข้อมูลจริง จัดเก็บในรูปแบบใด วิธีการเข้าถึงข้อมูลวิธีใด เช่น การแฮสชิ่ง (Hashing Function), อินเด็กซ์ (Indexing) เป็นต้น

ข้อมูลแต่ละระดับจะมีการเชื่อมโยงโดยจะถูกจัดการด้วยระบบการจัดการฐานข้อมูล โดยจะแปลงความหมายข้อมูลจากระดับหนึ่ง ไปยังอีกระดับหนึ่งที่เรียกกันว่า การแปลงส่ง (Mapping)

การแปลงส่ง (Mapping) แบบออกเป็น 2 ประเภท

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

2. การแปลงส่งระหว่างระดับแนวคิดกับระดับภายใน (Conceptual/Internal Mapping)

วิวของผู้ใช้คนที่ 1

วิวของผู้ใช้คนที่ N

ระดับภายนอก External / Conceptual Mapping

ระดับภายใน

เค้าโครงร่างภายใน

ระดับแนวคิด

เค้าโครงร่างแนวคิด

Conceptual/Internal Mapping

ภาพที่ 2.1.5 ฐานข้อมูล

ระดับแนวคิด

การแบ่งระดับของข้อมูลดังกล่าวทำให้เกิดความเป็นอิสระของข้อมูล โดยสามารถแบ่งออกเป็น 2 อย่าง ดังนี้

1. ความเป็นอิสระของข้อมูลเชิงตรรกะ (Logical Data Independence) คือ ความเป็นอิสระของข้อมูลในระดับภายนอกกับระดับแนวคิด โดยการเปลี่ยนแปลงข้อมูลที่เกิดขึ้นในระดับแนวคิด จะไม่ส่งผลกระทบกับระดับภายนอก เช่น การเพิ่มแอททริบิวต์ไม่มีผลต่อโปรแกรมประยุกต์ที่เขียนขึ้น

2. ความเป็นอิสระของข้อมูลในเชิงกายภาพ (Physical Data Independence) คือ ความอิสระของข้อมูลในระดับแนวคิดกับระดับภายใน โดยการเปลี่ยนแปลงที่เกิดขึ้นในระดับภายใน จะไม่มีผลกระทบกับระดับแนวคิด เช่น การเพิ่มประสิทธิภาพการเข้าถึงข้อมูลให้ไว้ขึ้น โดยการปรับปรุงเค้าโครงภายใน

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 หรือป้องกันข้อมูล ที่เกิดจากการใช้ข้อมูลจากผู้ใช้หลายคน

ฐานข้อมูลเชิงสัมพันธ์ (RELATIONAL DATABASE)

ฐานข้อมูลเชิงสัมพันธ์

เป็นการจัดเก็บรวบรวมข้อมูลในลักษณะตาราง 2 มิติ คือ แถวและคอลัมน์ โดยมีการจัดเก็บอย่างเป็นระบบและกฎเกณฑ์ ซึ่งได้ผ่านกระบวนการทำรีเลชั่นให้เป็นบรรทัดฐาน (Normalized) เพื่อลดปัญหาที่จะเกิดขึ้นในการจัดเก็บและค้นหาข้อมูล การจัดเก็บฐานข้อมูลเชิงสัมพันธ์ จะทำให้ข้อมูลเป็นกลุ่มตาราง ผู้ใช้จะเรียนรู้และเข้าใจง่าย โดยไม่จำเป็นต้องรู้ว่า ข้อมูลจริงๆมีการจัดเก็บอย่างไร รวมถึงวิธีการเรียกใช้ข้อมูล ภาษาที่ใช้คือเอชคิวแอล (SQL) ซึ่งเป็นภาษาที่มีลักษณะคล้ายภาษาอังกฤษ ง่ายต่อการเขียนคำสั่ง จึงเหมาะกับการใช้เป็นคำสั่งแบบสอบถาม

คุณสมบัติของตาราง

- ข้อมูลในแต่ละแถวจะไม่ซ้ำกัน โดยระบบจัดการฐานข้อมูลจะควบคุมไม่ให้เกิดความซับซ้อน

- ข้อมูลตารางในแต่ละแถวไม่มีความสำคัญในการเรียงลำดับ

- ในแต่ละคอลัมน์ของแถวๆ หนึ่งจะบรรจุข้อมูลได้เพียงค่าเดียว

- ค่าของข้อมูลในแต่ละคอลัมน์ จะบรรจุข้อมูลประเภทเดียวกัน

- คอลัมน์ทางเทคนิคจะใช้ Attribute

- แถวทางเทคนิค Tuple

ความหมายของคิย์ (Key)

ข้อมูลเชิงสัมพันธ์ สามารถเชื่อมโยงข้อมูลด้วยการใช้คีย์ ซึ่งความหมายของคีย์ ก็จะใช้คุณสมบัติในการแบ่ง โดยแบ่งออกเป็น 2 ประเภทใหญ่ๆ ดังนี้

1. คีย์หลัก (Primary Key) เป็นคีย์ที่เลือกจากแอททริบิวต์เดียวที่มีคุณสมบัติ คือ มีค่าที่ไม่ซ้ำซ้อนกัน ในแต่ละแถว (Tuple) นั่นเอง

2. คีย์นอก (Foreign Key) เป็นคีย์ที่ใช้อ้างอิง แอททริบิวต์เดียวกันในอีกตาราง (Relation) หนึ่ง ซึ่งคีย์นอกนี้ จะมีคุณสมบัติเป็นคีย์หลักในตารางอ้างอิง

ตารางตัวอย่างของการแสดงข้อมูลพนักงาน

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| รหัสพนักงาน | ชื่อพนักงาน | นามสกุล | รหัสแผนก | เงินเดือน |
| 0300141  0300142  0300143  0300144  0300145 | รัชนก  กาญจนา  ดัชนี  เพรชมณี  ไวยกรณ์ | เต็มใจ  สมใจ  คำเปง  วรรณคำ  สะกดใจ | 1001  1001  1002  1002  1003 | 8,500  8,500  9,000  9,000  6,000 |

ภาพที่ 2.1.6 ตารางบุคลากร

Foreign Key

Primary Key

Alternate Key

|  |  |  |
| --- | --- | --- |
| รหัสแผนก | ชื่อแผนก | จำนวนเงิน |
| 1001  1002  1003  Primary Key | การตลาด  คอมพิวเตอร์  การจัดการทั่วไป | 8,500  9,000  6,000 |

ภาพที่ 2.1.7 ตารางแผนก

การสร้างคีย์สามารถทำได้ในแต่ละตาราง ซึ่งแอททริบิวต์ใดไม่สามารถเป็นคีย์หลักได้ ก็จะทำการรวมแอททริบิวต์ที่เป็นคีย์รอง (ALTERNATE KEY) มารวมกัน เพื่อให้เป็นคีย์หลักซึ่งมีค่าเป็นค่าที่ไม่ซ้ำกัน (UNIQUE) โดยมีวิธีทำอยู่ 3 วิธี ดังนี้

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) มารวมกันและเมื่อตัดคีย์รอง (ALTERNATE KEY) ออกค่าก็ยังเป็นค่าที่ไม่ซ้ำกัน (UNIQUE) การนำมารวมกันเพื่อเป็นประโยชน์ในการค้นหา

3. แบบคีย์ประกอบ (COMBINED)

เป็นการนำคีย์รอง (ALTERNATE KEY) ที่มีมากกว่า 2 คีย์ขึ้นไป มารวมกัน เพื่อเป็นค่าที่ไม่ซ้ำกัน (UNIQUE) ทำเป็นคีย์หลัก (PRIMARY KEY) โดยหากขาดตัวใดไปแล้วค่าจะไม่เป็นค่าที่ไม่ซ้ำกัน (UNIQUE)

ฐานข้อมูลเชิงสัมพันธ์กับกฎของคีย์

- กฎความบูรณภาพของเอนทิตี้ (THE ENTITY INTEGRITY RULE) คอลัมน์ใดที่เป็นคีย์หลัก จะมีค่าว่าง (NULL) ไม่ได้ เพราะจะไม่ทราบค่าที่แน่นอน

- กฎความบูรณภาพอ้างอิง (THE REFERENT INTEGRITY RULE) การอ้างอิงของคีย์นอกจะต้องสามารถอ้างอิงให้ตรงกับค่าของคีย์หลัก เพื่อเชื่อมโยงข้อมูลระหว่างตารางได้

หลักการแก้ไขและลบข้อมูล

- การแก้ไขหรือลบข้อมูลแบบมีข้อจำกัด (RESTRICT)

การแก้ไขหรือลบข้อมูลได้ก็ต่อเมื่อ ไม่มีข้อมูลที่ถูกอ้างอิงโดยคีย์นอกของอีกหนึ่งตาราง

- การแก้ไขหรือลบข้อมูลแบบต่อเรียง (CASCADE)

การแก้ไขหรือลบข้อมูลแบบต่อเรียงของคีย์หลัก ระบบจะทำการลบข้อมูลของคีย์นอก ในตารางที่มีคีย์นอกอ้างอิงอยู่

- การแก้ไขหรือลบข้อมูลโดยเปลี่ยนเป็นว่าง (NULLIFY)

การแก้ไขหรือลบข้อมูลโดยเปลี่ยนเป็นว่าง จะทำได้ก็ต่อเมื่อ มีการเปลี่ยนแปลงค่าคีย์นอกในข้อมูลที่ถูกอ้างอิงเป็นว่างเสียก่อน

- การแก้ไขหรือลบข้อมูลแบบใช้ค่าโดยปริยาย (DEFAULT)

จะเป็นการปรับค่าคีย์นอกที่กำหนดขึ้น ในกรณียกเลิกวิชาที่เปิดสอน เป็นปรับเปลี่ยนค่าเป็น “0” สำหรับค่าที่ไม่มีการเปิดวิชาเรียน

ประเภทตาราง

ตารางของข้อมูลเชิงสัมพันธ์

จะมีตารางหลักที่ถูกกำหนดขึ้น เพื่อเก็บข้อมูลจริงจะทำการเรียกใช้ข้อมูลในภายหลัง

ตารางของข้อมูลเชิงสัมพันธ์แบบวิว (VIEW)

เป็นตารางที่ถูกสร้างขึ้นตามความต้องการของผู้ใช้แต่ละคน ตามงานที่ผู้ใช้กำหนดขึ้น หรือที่เราเรียกว่า “ตารางสมมติ”

แผนผังความสัมพันธ์ของข้อมูล (E-R DIAGRAM)

จะใช้เป็นตัวโครงร่างในการออกแบบฐานข้อมูล ซึ่งแบ่งเป็น 2 ส่วน ดังนี้

1. เอนทิตี (ENTITY)

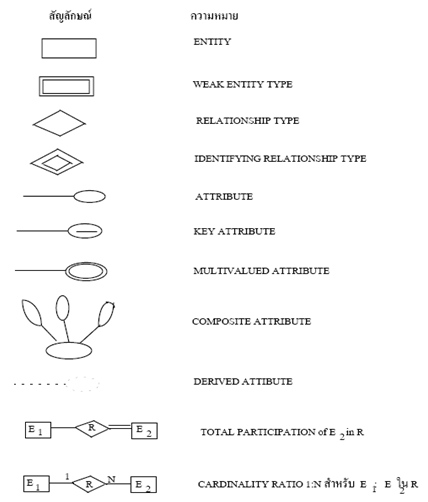
เป็นอ๊อบเจกทีฟ (OBJECTIVE) อะไรก็ได้ที่เราสนใจที่จะนำมาเป็นหลักในการทำแฟ้มข้อมูลของเรื่องนั้น ซึ่งสามารถแบ่งออกได้ว่าเป็นเอนทิตี (ENTITY) เช่น โต๊ะจะมี  
แอททริบิวต์เป็นหมายเลขครุภัณฑ์, หนังสือ จะมีแอททริบิวต์เป็นหมายเลขหมวดหมู่

2. กลุ่มของเอนทีตี (ENTITY SET)

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

วิธีการเขียนแผนผังความสัมพันธ์ของข้อมูล (E-R DIARAM)

สัญลักษณ์ที่ใช้เขียนแผนผังความสัมพันธ์ของข้อมูล มีดังนี้

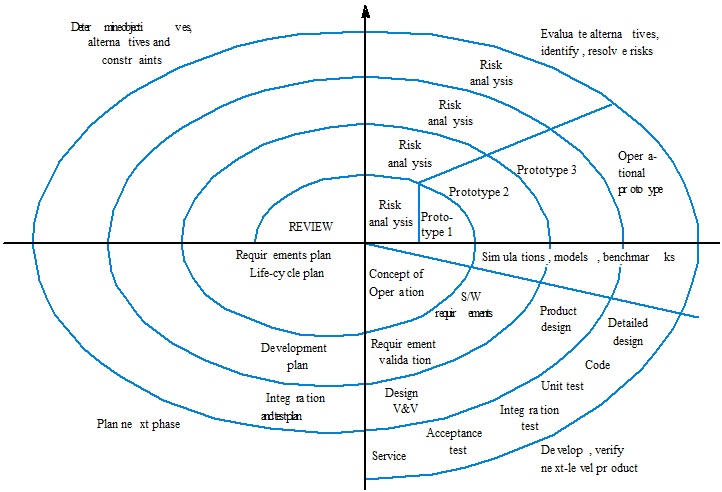


เอนทิตีอ่อนแอ (WEAK ENTITY) เป็นเอนทิตี (ENTITY) ที่เกิดขึ้นใหม่ ในภายหลังโดยอาศัยจากเอนทิตี ตี (ENTITY) อื่นๆ และจะต้องมีความสัมพันธ์อยู่ภายใต้อิทธิพลของเอนทิตี (ENTITY) นั้น

2.2 ทฤษฎีกระบวนการผลิตซอฟต์แวร์

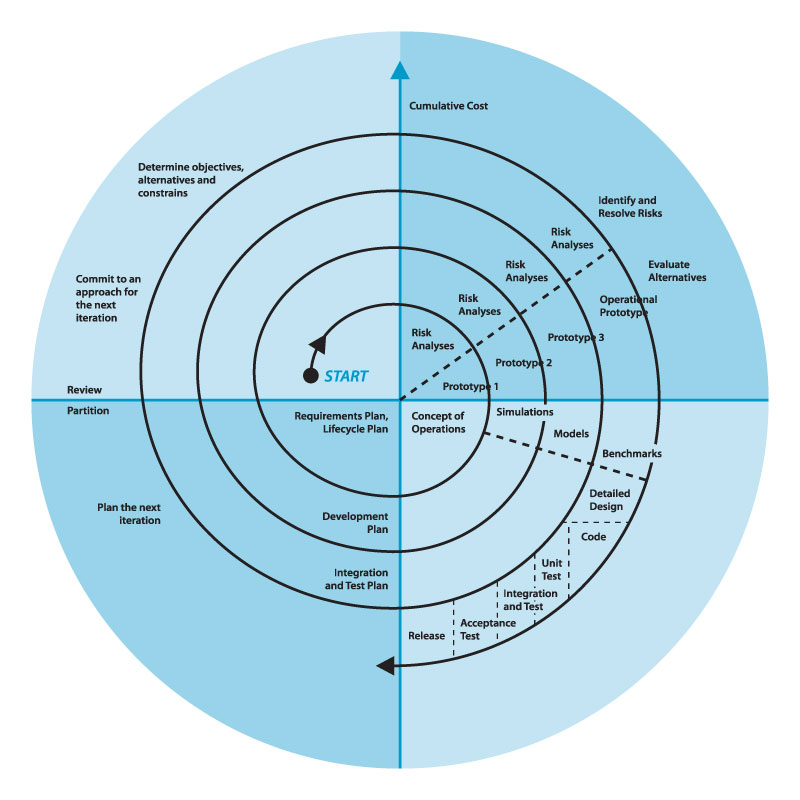
กระบวนการผลิตซอฟต์แวร์ (Soft ware Process)

โมเดลแบบก้นหอย (Spiral Model)

โมเดลแบบก้นหอย (Spiral Model) คือ กระบวนการพัฒนาซอฟต์แวร์ (Software Development Process) ที่ถูกพัฒนาขึ้นโดยเอาจุดแข็งของการพัฒนา โมเดล (Development Model) อื่นที่ดีอยู่แล้ว มาประยุกต์โมเดลแบบน้ำตก (Waterfall Model) หาวิและเพิ่มเติมส่วนของการวิเคราะห์ และตีค่าความเสี่ยงที่เกิดขึ้น เพื่อจะได้ทราบว่าจุดใดมีความเสี่ยงมากน้อยเพียงใด จะได้หาวิธีลดความเสี่ยง ซึ่งความเสี่ยงเป็นสาเหตุที่ทำให้การพัฒนาไม่ประสบความสำเร็จ การวิเคราะห์หรือต้นเหตุของความเสี่ยง ก็เพื่อที่จะหาวิธีการที่จะทำให้เกิดความเสี่ยงน้อยที่สุด รวมถึงวิธีการแก้ไขเมื่อเกิดเหตุการณ์ที่คาดคิดเกิดขึ้น ถ้าความเสี่ยงน้อยลง ก็ทำให้ต้นทุน (Cost) หรือต้นทุนที่ใช้ก็จะลดลงตามไปด้วย

ภาพที่ 2.2.1 รูปแสดงหลักการทำงานของระบบโมเดลแบบก้นหอย (Spiral Model)

กระบวนการพัฒนาของโมเดลแบบก้นหอย

 ถูกพัฒนามาจากโครงสร้างพื้นฐานของโมเดลแบบน้ำตก (Waterfall Model) ที่มีการแบ่งแยกขั้นตอน เช่น ขั้นตอนแนวคิดของการดำเนินงาน, ขั้นตอนความต้องการของซอฟต์แวร์ , ขั้นตอนการออกแบบ, ขั้นตอนการเขียนคำสั่ง, ขั้นตอนการรวบรวม, ขั้นตอนการใช้ เป็นต้นเนื่องจากในโมเดลแบบน้ำตก (Waterfall Model) สามารถส่งผลลัพธ์ที่ได้กลับไปยังขั้นตอนก่อนหน้าโดยไม่ต้องมีการแก้ไขทุกขั้นตอนใหม่หมด แต่โมเดลแบบน้ำตก (Waterfall Model) ในแต่ละขั้นตอนจะเกิดการตอบกลับ(Feedback) บ่อยครั้ง โมเดลแบบก้นหอย (Spiral Model) จึงถูกพันกับความเสี่ยงและความเป็นไปได้ที่เกิดขึ้น ตลอดจนหาแนวทางแก้ไขเมื่อเกิดข้อผิดพลาด

ภาพที่ 2.2.1 รูปกระบวนการซอฟต์แวร์ของโมเดลแบบก้นหอย (Spiral Model)

โครงสร้างของโมเดลแบบก้นหอย (Spiral Model)

โครงสร้างของโมเดลแบบก้นหอย (Spiral Model) แสดงในรูปโดย

-รัศมีของวงกลม หมายถึง ต้นทุน (Cost) ที่เกิดขึ้นในกระบวนการพัฒนาซอฟต์แวร์ ถ้าจำนวนรอบ (Cycle) ที่มากขึ้นก็จะหมายถึง ต้นทุนของการพัฒนาก็จะเพิ่มมากขึ้นเช่นกัน

- มุมของวงกลม หมายถึง ความก้าวหน้าในการปฏิบัติแต่ละขั้นตอนในแต่ละรอบได้สำเร็จ

ส่วนต่างๆ ในแต่ละรอบของโมเดลแบบก้นหอย ประกอบด้วย

1. กำหนดระยะ (Determine Phase) เป็นส่วนที่ทำหน้าที่กำหนด

- วัตถุประสงค์ (Objective) กำหนดผลลัพธ์ที่จะได้รับ

- ทางเลือก (Alternative) ที่เป็นไปได้ของการนำไปใช้ (Implement) ทางเลือกอาจมีการนำสิ่งที่มีอยู่แล้วมาใช้ใหม่ (Reuse), ทางเลือกในการจัดซื้อ

- เงื่อนไข (Constraint) เป็นเงื่อนไขที่เกิดขึ้นในการพัฒนา

2. ขั้นตอนการประเมิน (Evaluate Phase)

จากทางเลือกทั้งหมดที่กำหนดเอาไว้ในขั้นตอนการประเมิน (Determine Phase) และเงื่อนไขที่ได้ตั้งไว้ เพื่อทราบว่าจุดใดในขบวนการที่เป็นจุดสำคัญที่ทำการวิเคราะห์ หรือประเมินค่าความเสี่ยงอาจทำได้โดยการทำต้นแบบประสิทธิภาพของความเสี่ยง (Performance Risk) และความต้องการของผู้ใช้ (User Requirement), การจำลองสถานการณ์ (Simulate) เพื่อหาประสิทธิภาพ

ในการทำต้นแบบ (Prototypes) เบื้องต้น เป็นความพยายามในการแก้ไขความเสี่ยงที่เกิดขึ้น ในกรณีของความเสี่ยงการเชื่อมต่อกับผู้ใช้ (User Interface Risk) และการควบคุมความเสี่ยง (Control Risk)

3. ขั้นตอนการพัฒนา ตรวจสอบ (Develop, verify Phase)

หลังจากการทำต้นแบบ (Prototype) เพื่อวอเคราะห์หาค่าความเสี่ยงในเฟส (Phase) ถึงขั้นตอนในการพัฒนาซอฟต์แวร์ (Software Development Phase) ให้เป็นไปตามวัตถุประสงค์ที่ว่าความเสี่ยงที่เกิดจากหน้าประสานงานกับผู้ใช้งาน (User Interface) หรือตีค่าความเสี่ยงและวางแผนสำหรับรอบ (Cycle) ต่อไปของการพัฒนา ซึ่งจุดนี้เองทำให้ในขั้นตอน

การเปรียบเทียบโมเดลแบบก้นหอย (Spiral Model) กับโมเดลกระบวนการพัฒนาซอฟต์แวร์ (Software Development Process Model) อื่น

ข้อดีของโมเดลแบบก้นหอย (Spiral Model)

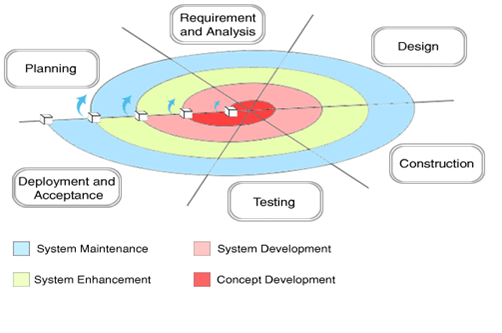
1. สนับสนุนการนำซอฟต์แวร์ (Software) กลับมาใช้อย่างเต็มตัว
2. ในแต่ละรอบ (Cycle) มีขั้นตอนประมวลผลที่สิ้นสุดภายในรอบเดียว
3. การวางแผนเพื่อกำหนดทางเดินของกระบวนการซอฟต์แวร์ (Software Process) ในรอบต่อไป
4. เนื่องจากการพัฒนาอยู่บนพื้นฐานของวัตถุประสงค์ที่ตั้งไว้ทำให้ผลลัพธ์ของผลิตภัณฑ์ซอฟต์แวร์ (Software Product) ตรงตามความต้องการของผู้ใช้

5. แก้ไขข้อผิดพลาดที่เกิดขึ้นตั้งแต่เนิ่นๆ

6. มีความเป็นอิสระต่อกัน ทางด้านการพัฒนาและการแก้ไข

ข้อเสียของโมเดลแบบก้นหอย (Spiral Model)

1. เนื่องจากโมเดลแบบก้นหอย (Spiral Model) ทุกรอบ (Cycle) ของการพัฒนามีการวิเคราะห์และตีค่า ถ้าการวิเคราะห์เกิดความผิดพลาด จะทำให้ผลิตภัณฑ์ซอฟต์แวร์ (Software Produce) ที่ออกมาผิดพลาดทำให้เกิดค่าใช้จ่ายที่เพิ่มขึ้น



ภาพที่ 2.2.2 แสดงการทำงานของแต่ละส่วนของโมเดลแบบก้นหอย (Spiral Model)

2.3 ทฤษฎีหรือเทคโนโลยีและเทคโนโลยีที่ใช้ในการพัฒนา

ภาษา พีเอชพี (PHP)

พีเอชพี (PHP) คือ ภาษาคอมพิวเตอร์ในลักษณะเซิร์ฟเวอร์-ไซด์ สคริปต์ โดยลิขสิทธิ์อยู่ในลักษณะโอเพนซอร์ส ภาษาพีเอชพีใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบHTMLโดยมีรากฐานโครงสร้างคำสั่งมาจากภาษา ภาษาซี ภาษาจาวา และ ภาษาเพิร์ล ซึ่ง ภาษาพีเอชพี นั้นง่ายต่อการเรียนรู้ ซึ่งเป้าหมายหลักของภาษานี้ คือให้นักพัฒนาเว็บไซต์สามารถเขียนเว็บเพจที่มีความตอบโต้ได้อย่างรวดเร็ว พีเอชพีรุ่นล่าสุดคือ PHP 5.4.0 ส่วนรุ่นพัฒนาคือ PHP 6.0.0-dev คุณลักษณะคือ

การแสดงผลของพีเอชพี จะปรากฏในลักษณะHTML ซึ่งจะไม่แสดงคำสั่งที่ผู้ใช้เขียน ซึ่งเป็นลักษณะเด่นที่พีเอชพีแตกต่างจากภาษาในลักษณะไคลเอนต์-ไซด์ สคริปต์ เช่น ภาษาจาวาสคริปต์ ที่ผู้ชมเว็บไซต์สามารถอ่าน ดูและคัดลอกคำสั่งไปใช้เองได้ นอกจากนี้พีเอชพียังเป็นภาษาที่เรียนรู้และเริ่มต้นได้ไม่ยาก โดยมีเครื่องมือช่วยเหลือและคู่มือที่สามารถหาอ่านได้ฟรีบนอินเทอร์เน็ต ความสามารถการประมวลผลหลักของพีเอชพี ได้แก่ การสร้างเนื้อหาอัตโนมัติจัดการคำสั่ง การอ่านข้อมูลจากผู้ใช้และประมวลผล การอ่านข้อมูลจากดาต้าเบส ความสามารถจัดการกับคุกกี้ ซึ่งทำงานเช่นเดียวกับโปรแกรมในลักษณะCGI คุณสมบัติอื่นเช่น การประมวลผลตามบรรทัดคำสั่ง (command line scripting) ทำให้ผู้เขียนโปรแกรมสร้างสคริปต์พีเอชพี ทำงานผ่านพีเอชพี พาร์เซอร์ (PHP parser) โดยไม่ต้องผ่านเซิร์ฟเวอร์หรือเบราว์เซอร์ ซึ่งมีลักษณะเหมือนกับ Cron   
(ใน ยูนิกซ์หรือลีนุกซ์) หรือ Task Scheduler (ในวินโดวส์) สคริปต์เหล่านี้สามารถนำไปใช้ในแบบ Simple text processing tasks ได้

การแสดงผลของพีเอชพี ถึงแม้ว่าจุดประสงค์หลักใช้ในการแสดงผล HTML แต่ยังสามารถสร้าง XHTML หรือ XML ได้ นอกจากนี้สามารถทำงานร่วมกับคำสั่งเสริมต่างๆ ซึ่งสามารถแสดงผลข้อมูลหลัก PDF แฟลช (โดยใช้ libswf และ Ming) พีเอชพีมีความสามารถอย่างมากในการทำงานเป็นประมวลผลข้อความ จาก POSIX Extended หรือ รูปแบบ Perl ทั่วไป เพื่อแปลงเป็นเอกสาร XML ในการแปลงและเข้าสู่เอกสาร XML เรารองรับมาตรฐาน SAX และ DOM สามารถใช้รูปแบบ XSLT ของเราเพื่อแปลงเอกสาร XML

เมื่อใช้พีเอชพีในการทำอีคอมเมิร์ซ สามารถทำงานร่วมกับโปรแกรมอื่น เช่น Cybercast payment, CyberMUT, VeriSign Pay flow Pro และ CCVS functions เพื่อใช้ในการสร้างโปรแกรมทำธุรกรรมทางการเงิน

ข้อดีของ ภาษา PHP

1. ใช้ได้ฟรี

2. PHP เป็นโปรแกรมวิ่งข้าง Sever ดังนั้นขีดความสามารถไม่จำกัด

3. Conlatfun-นั่นคือPHP วิ่งบนเครื่อง UNIX, Linux, Windows ได้หมด

4. เรียนรู้ง่าย เนืองจาก PHP ฝั่งเข้าไปใน HTML และใช้โครงสร้างและไวยากรณ์ภาษาง่ายๆ

5. เร็วและมีประสิทธิภาพ โดยเฉพาะเมือใช้กับ Apach Xerve เพราะไม่ต้องใช้โปรแกรมจากภายนอก

6. ใช้ร่วมกับ XML ได้ทันที

7. ใช้กับระบบแฟ้มข้อมูลได้

8. ใช้กับข้อมูลตัวอักษรได้อย่างมีประสิทธิภาพ ใช้กับโครงสร้างข้อมูลใช้ได้แบบ Scalar, Array, Associative array

9. ใช้กับการประมวลผลภาพได้

ข้อเสียของ ภาษา PHP

- ขาด IDE ที่เป็นมาตรฐานกลางทำให้คนเขียน ต้องหาโปรแกรมที่ใช้ในการพัฒนาเอง บางครั้งออก version ใหม่บ่อยเกินไป การเขียน ต้อง include ไม่มี บริษัท software ใหญ่ๆ เป็นป๋าดันให้ เลยไม่ดังเปรี้ยง แต่ค่อยๆดังเพราะทำความดี สะสม การเขียนติดต่อระดับ component หรือ COM+ ของ windows อาจต้อง config ยุ่งยากหน่อย (ไม่เคยเขียน) ใช้ IE เปิด web php.net ใน เครือข่าย kku แล้ว download php ยากมาก

**การใช้ตัวแปรในภาษา PHP**

PHP Variable () สำหรับการเขียนโปรแกรมสำหรับภาษาคอมพิวเตอร์ระดับสูง สิ่งที่จะขาดเสียมิได้คือ การกำหนดและใช้ตัวแปร (variable) ตัวแปรในภาษา PHP จะเหมือนกับในภาษา Perl คือเริ่มต้นด้วยเครื่องหมาย dollar ($) โดยเราไม่จำเป็นต้องกำหนดแบบของข้อมูล (data type) อย่างเจาะจงเหมือนในภาษาซี เพราะว่า ตัวแปลภาษาจะจำแนกเองโดยอัตโนมัติว่า ตัวแปรดังกล่าว ใช้ข้อมูลแบบใด ในช่วงเวลานั้นๆ เช่น ข้อความ จำนวนเต็ม จำนวนที่มีเลขจุดทศนิยม ตรรก เป็นต้น

**โครงสร้าง**

<?

$var = value;

?>

**ขอบเขตการตั้งชื่อตัวแปร**

- ขึ้นต้นด้วยเครื่องหมาย $ แล้วตามด้วยตัวอักษร A-Z, a-z

- มีความยาวไม่เกิน 255 ตัวอักษร

- ห้ามมีจุดทศนิยม หรือช่องว่าง

- จะต้องไม่ตรงกับคำสงวน และควรตั้งชื่อ ให้มีความหมายใกล้เคียงกับ ค่าที่เก็บ

- ตัวอักษรเล็กหรือใหญ่จะเป็นตัวแปรต่างกัน

- ถ้าตั้งตัวแปรมาใหม่ แล้วทับตัวแปรเก่า ค่าของตัวแปรเก่าจะหายไป

ตัวอย่าง

<?php

$a = 1234; // decimal number

$b = "Weerachai Nukitram"; // String

$c = 1234.456; // Float

$d[0] = "Win"; // Array

$d[1] = "ThaiCreate.Com"; // Array

$e = true; // Boolean

?>

การใช้ Echo เพื่อแสดงข้อความ

การแสดงผลในภาษา PHP

การแสดงผลข้อความใน PHP สามารถใช้คำสั่งได้สามรูปแบบคือ คำสั่ง echo, print และคำสั่ง printf คำสั่งทั้งสามแบบนี้เป็นคำสั่งแสดงผลข้อมูลออกทางบราวเซอร์ ลองมาศึกษาวิธีการใช้ และสรุปข้อแตกต่างกันได้

ในการเขียน php เพื่อแสดงผลออกทางเว็บบราวเซอร์นั้น เราสามารถใช้คำสั่งได้ 3 คำสั่งดังนี้

1. echo

2. print

3. printf

แบบแรก คำสั่ง echo เป็นคำสั่งที่สามารถแสดงผลข้อมูลได้หลาย ๆ ประเภท เช่น

กรณีที่ 1 ใช้ echo แสดงค่าคงที่

<?

echo "test echo "; //ข้อมูลที่ถูกแสดงผลออกบราวเซอร์คือ test echo

?>

กรณีที่ 2 echo แสดงผลการคำนวณค่าตัวเลข

<?

echo 2+3; // ข้อมูลที่ถูกแสดงผลออกบราวเซอร์คือ 5

?>

กรณีที่ 3 echo แสดงผลการคำนวณค่าของตัวแปร

<?

$a = 3;

$b = 5;

echo $a + $b; //ข้อมูลที่ถูกแสดงผลออกบราวเซอร์คือ 8

?>

กรณีที่ 4 ใช้ echo แสดงผลข้อมูลหลายประเภทโดยการแยกค่านิพจน์

<?

$a = 3;

$b = 5;

echo "3+5= ", 3+5," = ",$a+$b; // ข้อมูลที่ถูกแสดงผลออกทางเว็บบราวเซอร์ คือ 3+5 = 8 = 8

?>

แบบที่ 2 คำสั่ง print เป็นคำสั่งที่สามารถแสดงผลข้อมูลเช่นเดียวกันกับคำสั่ง echo แต่คำสั่ง print จะไม่สามารถแสดงผลข้อมูลแบบหลาย ๆ นิพจน์ หรือหลาย ๆ ค่าตัวแปร โดยการใช้ เครื่องหมาย, คั่น แบบคำสั่ง echo ได้

รูปแบบการใช้คำสั่ง print

<?

#แบบแสดงผลค่าคงที่

print "test print "; //ข้อมูลที่ถูกแสดงผลออกบราวเซอร์คือ test print

#แบบแสดงผลการคำนวนค่าตัวเลข

print 5+5; //ข้อมูลที่ถูกแสดงผลออกบราวเซอร์คือ 10

#แบบแสดงผลการคำนวนค่าของตัวแปร

$a=2;

$b=5;

print $a+$b; //ข้อมูลที่ถูกแสดงผลออกบราวเซอร์คือ 7

?>

แบบที่ 3 คำสั่ง printf เป็นคำสั่งที่มีลักษณะการใช้เหมือนกับคสั่ง printf ในภาษา C ซึ่งจะสามารถจัดรูปแบบ (Format) ของข้อความที่เราจะทำการแสดงผลออกทางเว็บบราวเซอร์ได้

รูปแบบการใช้คำสั่ง printf

<?

#แสดงผลข้อมูลเป็นแบบข้อมูลตัวเลข

printf ("5x5 = %d" , 5\*5); //ข้อมูลที่ถูกแสดงผลออกบราวเซอร์คือ 5x5 = 25

#แบบแสดงผลข้อมูลเป็นแบบทศนิยม

printf ("5/5 = %f" , 5/5); //ข้อมูลที่ถูกแสดงผลออกบราวเซอร์คือ 5/5 = 1.000000

#แบบแสดงผลข้อมูลเป็นแบบเลขฐาน 8

printf ("5+5 = %o" , 5+5); //ข้อมูลที่ถูกแสดงผลออกบราวเซอร์คือ 5+5 = 12

#แบบแสดงผลข้อมูลเป็นแบบข้อความ

printf ("5-5 = %s" , 5-5); //ข้อมูลที่ถูกแสดงผลออกบราวเซอร์คือ 5-5 = 0

?>

การใช้เงื่อนไขแบบ if-else

เงื่อนไข if

if() กำหนดเงื่อนไขที่เป็นจริงและเงื่อนไขที่เป็นเท็จ

โครงสร้าง

<?

if(expr1)

{

…//statement

}

else

{

...//statement

}

?>

ตัวอย่าง

<?php

$i=2;

if($i==2)

{

echo "Yes \$i = 2"."<br>";

}

Else

{

echo "No \$i <> 2"."<br>";

}

echo "<hr>";

if(date("Y-m-d")== "2008-08-14") // today = 2008-08-13

{

echo "Is today"."<br>";

}

Else

{

echo "Not Is today"."<br>";

}

?>

เงื่อนไข if..else()

if..else() กำหนดเงื่อนไขเป็นจริงได้หลายเงื่อนไขเพื่อเป็นทางเลือกในการตัดสินใจ

โครงสร้าง

<?

if(expr1)

{

...//statement

}\

elseif(expr2)

{

...//statement

}

?>

ตัวอย่าง

<?php

$i=3;

if($i==1)

{

echo "Yes \$i = 1"."<br>";

}

elseif($i==2)

{

echo "Yes \$i = 2"."<br>";

}

elseif($i==3)

{

echo "Yes \$i = 3"."<br>";

}

Else

{

echo "I don't no"."<br>";

}

?>

การใช้งานลูปต่าง ๆ ในภาษา PHP

คำสั่ง while() จะพิจารณาเงื่อนไขแล้วค่อยทำ ถ้าเงื่อนไขเป็นจริงจะทำ ถ้าเท็จ จะกระโดดออกจากลูปทันที

โครงสร้าง

<?

while (expr)

{

statement & condition

}

?>

ตัวอย่าง

<?

$i=1;

while($i<=5)

{

echo"<font size=$i>www.ThaiCreate.Com"."<br>";

$i++;

}

?>

คำสั่ง do.while()ทำก่อนแล้วค่อยพิจารณาเงื่อนไข ถ้าเงื่อนไขเป็นเท็จ จะกระโดดออกจากลูปทันที

โครงสร้าง

<?

do{

...

} while($condition)

?>

ตัวอย่าง

<?

$i=1;

do{

echo"<font size=$i>www.ThaiCreate.Com"."<br>";

$i++;

}while($i<=5)

?>

คำสั่ง for() จะทำงานจนเงื่อนไขนั้นเป็นเท็จ ถึงจะกระโดดออกจากลูป

โครงสร้าง

<?

for (expr1; expr2; expr3)

{

...//condition

}

?>

ตัวอย่าง

<?php

for($i=0;$i<=5;$i++)

{

echo $i." <font size=$i>www.ThaiCreate.Com</font>"."<br>";

}

?>

การใช้อาร์เรย์ (Array)

อาร์เรย์ในภาษา PHP นั้นจะแตกต่างจากอาร์เรย์ในภาษาซีหรือจาวาตรงที่ว่า อาร์เรย์ในภาษา PHP มีขนาดที่เปลี่ยนแปลงได้ หรือจะเรียกว่า dynamic array หรือ vector (สำหรับอาร์เรย์มิติเดียว) เริ่มต้นอาจจะแจ้งใช้ตัวแปรแบบอาร์เรย์ พร้อมเจาะจงขนาดเริ่มแรก เช่น มีขนาดเป็นศูนย์ก็ได้

$myarray[]=3;

$myarray[]=1.1;

$myarray[]="abc";

แต่เมื่อใช้อาร์เรย์ไป ขนาดของมันจะปรับเปลี่ยนได้ คือขยายจำนวนข้อมูลที่เก็บอยู่ภายในอาร์เรย์ ตามจำนวนข้อมูลที่เราใส่เพิ่มเข้าไป จากตัวอย่างข้างบน ในกรณีที่เรามิได้กำหนดเลขดัชนี (index) ก็หมายความว่า จะมีการขยายขนาดของอาร์เรย์เพิ่มขึ้นอีกหนึ่งโดยอัตโนมัติ ทุกครั้งที่เราใส่ข้อมูลที่อยู่ทางขวา และค่าที่เรากำหนดจากทางขวามือ และจะเก็บไว้ในที่ใหม่ของอาร์เรย์ เราไม่ต้องคำนึงถึงเรื่องการจอง หรือ ปลดปล่อยหน่วยความจำของอาร์เรย์ เหมือนอย่างในกรณีของอาร์เรย์ แบบไดนามิกในภาษาซี

นอกจากนั้นข้อมูลแต่ละตัวในอาร์เรย์ไม่จำเป็น ต้องเป็นข้อมูลชนิดเดียวกัน เช่น อาจจะมีทั้งจำนวนเต็ม เลขทศนิยม และข้อความ ปะปนกันไป ตัวอย่างเช่น

<?

$myarray[0] = 1;

echo "number of elements =".count($myarray)."<BR>\n";

$myarray[1] = "abc";

echo "number of elements =".count($myarray)."<BR>\n";

$myarray[2] = 1.3;

echo "number of elements =".count($myarray)."<BR>\n";

$myarray[]= 13+10; // the same as $myarray[3]= 13+10;

echo "number of elements =".count($myarray)."<BR>\n";

for ($i=0; $i < 4; $i++) {

echo $myarray[$i]," \n";

}

?>

ถ้าเราต้องการจะทราบจำนวนของข้อมูลที่มีอยู่ในอาร์เรย์เราจะใช้คำสั่ง count()เทคนิคหนึ่งที่ใช้ในการสร้างอาร์เรย์ที่เก็บหลายๆข้อความหรือสตริงค์ คือ แทนที่เราจะกำหนดค่าของสมาชิก ในอาร์เรย์ทีละตัว เราจะสร้างได้โดยอัตโนมัติ โดยเก็บสตริงค์เหล่านั้นไว้ในสตริงค์เพียงอันเดียวโดยมีสัญลักษณ์ | เป็นตัวแยก และก็แล้วใช้ฟังก์ชันเป็นตัวแบ่งเพื่อสร้างอาร์เรย์อีกที ตาม

ตัวอย่าง

<?

// create empty array

$a=array();

// define string containing color names separated by | (pipe)

$color\_names="red|green|blue";

// create array from string

$a=explode("|",$color\_names);

while ( $color=each($a) ) {

echo "$color[1]<BR>\n"; // note: $color[0] contains the index (0,1,2,...)

}

?>

ลองดูอีกตัวอย่างหนึ่งที่ใช้ฟังก์ชัน explode() สร้างอาร์เรย์โดยอัตโนมัตสำหรับใส่ไว้ใน FORM ในส่วนของ SELECT เป็นเมนูให้เลือก

<?

// create selection list from a given string

function str2select($str, $delim) {

$options = explode($delim,$str);

$num = count($options);

for( $i=0; $i < $num;$i++) {

echo "<option> $options[$i]</option>\n";

}

}

$select\_str="10 บาท|20 บาท|30 บาท|40 บาท|50 บาท|100 บาท|200 บาท|500 บาท|1000 บาท";

?>

<FORM>

<SELECT NAME="testform">

<? str2select($select\_str,"|"); ?>

</SELECT>

</FORM>

คำสั่ง SQL

SQL SELECT

เป็นคำสั่งที่ใช้สำหรับการเรียกดูข้อมูลในตาราง (Table) คำสั่ง SQL SELECT สามารถเรียกได้ทั้งตาราง หรือว่า สามารถระบุฟิวด์ที่ต้องการเรียกดูข้อมูลได้

โครงสร้าง

SELECT Column1, Column2, Column3,... FROM [Table-Name]

ตัวอย่าง 1 การเลือกข้อมูลที่ระบุฟิวด์

SELECT CustomerID, Name, Email FROM customer

ตัวอย่าง 2 การเลือกข้อมูลทั้งหมดของ Table

SELECT \* FROM customer

SQL WHERE

เป็นคำสั่งที่ใช้สำหรับการระบุเงื่อนไขการเลือกข้อมูลในตาราง (Table) คำสั่ง SQL WHERE สามารถระบุเงื่อนไขในการเลือกข้อมูลได้ 1 เงื่อนไข หรือมากกว่า 1 เงื่อนไข

โครงสร้าง

SELECT Column1, Column2, Column3,... FROM Table-Name WHERE [Field] = 'Value'

ตัวอย่าง 1 การเลือกข้อมูลโดยใช้ Operators = (เท่ากับ)

SELECT \* FROM customer WHERE CountryCode = 'US'

หรือ แบบ 2 เงื่อนไข ใช้ and เข้ามาเชื่อม วลี

SELECT \* FROM customer WHERE CountryCode = 'US' and Budget = '4000000'

ตัวอย่าง 2 การเลือกข้อมูลโดยใช้ Operators != (ไม่เท่ากับ)

SELECT \* FROM customer WHERE CountryCode != 'US'

หรือ แบบ 2 เงื่อนไข ใช้ and เข้ามาเชื่อม วลี

SELECT \* FROM customer WHERE CountryCode != 'US' and CountryCode != 'EN' หรือจะใช้ or SELECT \* FROM customer WHERE CountryCode != 'US' or Budget = '1000000'

SQL UPDATE

เป็นคำสั่งที่ใช้สำหรับแก้ไขข้อมูลในตาราง (Table) โดยสามารถทำการแก้ไขได้หลายฟิวด์และหลาย Record ภายในคำสั่ง 1 คำสั่ง ทั้งนี้ขึ้นอยู่กับ Where ที่ผู้ใช้ได้เขียนขึ้น

โครงสร้าง

UPDATE [Table-Name] SET Column1='Value1',Column2='Value2',... WHERE clause

ตัวอย่าง 1 การแก้ไขข้อมูลลงใน Table

UPDATE country SET CountryCode = 'JP',CountryName='Japan' WHERE CountryCode = 'CH'

SQL DELETE

เป็นคำสั่งที่ใช้สำหรับลบข้อมูลในตาราง (Table) โดยสามารถทำการลบได้หลาย Record ภายในคำสั่งเดียว หรือว่า Record เดียว ทั้งนี้ขึ้นอยู่กับ Where ที่ผูใช้เขียนขึ้นด้วย

โครงสร้าง

DELETE FROM [Table-Name] WHERE clause

ตัวอย่าง 1 การลบข้อมูลลงใน Table

DELETE FROM country WHERE CountryCode = 'JP'

ภาษาที่ใช้สำหรับนิยามข้อมูล (Data Definition Language-DDL )

Data Definition Language (DDL) เป็นภาษาที่ใช้นิยามโครงสร้างข้อมูลเพื่อเปลี่ยนแปลง หรือยกเลิกโครงสร้างข้อมูล ตามที่ออกแบบไวโครงสรางดังกลาว คือ สคีมา (Schema) นั้นเอง

ตัวอยางเชน การกําหนดใหฐานขอมูลประกอบดวยตารางอะไรบาง ชื่ออะไร ประเภทใด มีอินเด็กซ (Index)

ภาษา DDLประกอบดวย 3 คําสั่งคือ

1. คําสั่งการสราง (Create) ไดแก การสรางตารางและอินเด็กซ

CREATE TABLE <Table name>

(Attribute 1 Type 1,

Attribute 2 Type 2,)

CREATE Unique Index on X<Table name>

เชน

CREATE TABLE S11

(SNO CHAR(5) Not NULL,

SNAME CHAR(10) ,

STATUS integer

)

CREATE Unique Index XS11 on S11(SNO)

2. คําสั่งเปลี่ยนแปลงโครงสราง

ALTER TABLE < ชื่อตารางที่ตั้งขึ้น >

<คำสั่งการเปลี่ยนแปลง> (<ชื่อคอลัมนประเภทขอมูล>);

เช่น

ALTER TABLE SUPPLIER

ADD (LAST\_SNAME Char(10));

3. คำสั่งยกเลิก (Drop) ต่างๆ

การลบโครงสร้างตาราง

DROP TABLE < ชื่อตารางที่ตั้งขึ้น >

ภาษาดังกล่าวคือ ภาษาที่ใชสรางฐานขอมูลลงในคอมพิวเตอรหลังจากที่เราไดออกแบบแลว วาฐานข้อมูลมีกี่รีเลชั่น แตละรีเลชั่นมีความสัมพันธอยางไร จากนั้นการใชภาษา DDL นี้แปลง   
รีเลชั่นตางๆ

ใหอยูในรูปภาษาสําหรับนิยามขอมูล เพื่อปอนเขาสูระบบฐานขอมูล เพื่อสรางฐานขอมูลที่แทจริง ใหเกิดขึ้นในคอมพิวเตอรภาษา DDL สามารถสรุปคําสั่งตางๆไดดังต่อไปนี้

|  |  |
| --- | --- |
| คําสั่ง | ความหมาย |
| CREATE TABLE | นิยามโครงสรางขอมูลในรูปตารางบนฐานขอมูล |
| DROP TABLE | ลบโครงสรางตารางขอมูลออกจากระบบ |
| ALTER TABLE | แกไขปรับปรุงโครงสรางตาราง |
| CREATE INDEX | สรางดัชนีของตาราง |
| DROP INDEX | ลบ ดัชนีของตารางออกจากระบบ |
| CREATE VIEW | กําหนดโครงสรางวิวของผูใช |
| DROP VIEW | ลบโครงสรางวิวออกจากระบบ |

คําสั่งนิยามโครงสรางตาราง

การสร้างตารางในฐานขอมูลแบบรีเลชั่นเนลโดยเฉพาะฐานข้อมูลขนาดใหญ่บนระบบ UNIX จะทำด้วยการป้อนคำสั่งในลักษณะเท็กซ์โหมด (Text Mode) เข้าไปในระบบฐานข้อมูล

CREATE TABLE <ชื่อตาราง >

(<ชี่อคอลัมนประเภทของขอมูล> [,<ชี่อคอลัมน์ประเภทของขอมูล>]....);

ประเภทของขอมูล

ประเภทของขอมูลแบงเปน 5 ประเภทใหญ่ๆ ขึ้นอยู่กับซอฟตแวรระบบฐานขอมูลที่ใชวาคืออะไร ตัวอย่างเช่น CHAR, INTEGER, DATE ฯลฯ

คำสั่งการลบโครงสร้างตาราง

DROP TABLE <ชื่อตารางที่ตองการลบ>

คำสั่งการเปลี่ยนแปลงโครงสรางตาราง

ในกรณีที่ตองการเปลี่ยนแปลงโครงสรางตารางที่เคยนิยามไวสามารถใช้คำสั่งต่อไปนี้

ALTER TABLE <ชื่อตารางที่ตองการเปลี่ยนแปลง>

<คําสั่งการเปลี่ยนแปลง><[,<ชื่อคอลัมน์ประเภทของข้อมูล>]>

คําสั่งดัชนี

ดัชนี (INDEX) มีความสําคัญมากตอฐานขอมูลเชิงสัมพันธ เนื่องจากระบบฐานข้อมูลแบบรีเลชั่นเนล (RDBMS) จะใชดัชนีในการคนหาระเบียนที่ตองการไดอยางรวดเร็ว โดยดัชนีที่ถูกสร้างขึ้น จะเก็บไวแยกจากตารางในพื้นที่ตางหากของคอมพิวเตอร โดยปกติถาไมมีการประกาศดัชนีไวการคนหาขอมูลในตาราง นั้นจะตองทําแบบเรียงลําดับจากแถวที่หนึ่งจนถึงแถวสุดทาย

การสรางดัชนีสําหรับตารางใดๆ จะทําไดโดยการเลือกคอลัมนใดคอลัมนหนึ่งจากตารางมา  
เปนดัชนีและตารางหนึ่งๆ สามารถมีไดหลายดัชนี นอกจากเพิ่มความรวดเร็วในการดึงขอมูลแลว ยังสามารถนําไปใชในการควบคุม คอลัมนที่นํามาสรางเปนดัชนีใหมีการเก็บขอมูลที่ไมซ้ำกัน(Unique) อีกดวย

ภาษาสําหรับการจัดการขอมูล (Data Manipulation Language-DML)

หลังจากที่เราสร้างโครงสร้างฐานข้อมูลขึ้นแลวคำสั่งต่อไปในการป้อนข้อมูลลงในฐานข้อมูล

และเปลี่ยนแปลงข้อมูลในฐานข้อมูลโดยการใช้ภาษาสำหรับจัดการข้อมูล ( Data Manipulation Language -DML) ใช้จัดการข้อมูลภายในตารางภายในฐานข้อมูลและภาษาแกไขเปลี่ยนแปลงตาราง

แบงออกเป็น 4 Statement คือ

- Select Statement: การเรียกหา (Retrieve) ข้อมูลจากฐานข้อมูล

- Insert Statement: การเพิ่มเติมข้อมูลในตาราง (Table) จากฐานข้อมูล

- Delete Statement: การลบข้อมูลออกจากตาราง (Table) จากฐานข้อมูล

- Update Statement: การเปลี่ยนแปลงข้อมูลในตาราง (Table) จากฐานข้อมูล

|  |  |
| --- | --- |
| คําสั่ง | ความหมาย |
| SELECT | เรียกค้นข้อมูลในตาราง |
| INSERT | เพิ่มแถวข้อมูลลงในตาราง |
| DELETE | ลบแถวข้อมูลในตาราง |
| UPDATE | ปรับปรุงแถวข้อมูลในตาราง |

คำสั่งค้นหาข้อมูล (Query Statement)

คำสั่ง SELECT เป็นคำสั่งการเรียกดูข้อมูลหรือค้นข้อมูลตามเงื่อนไขที่ระบุบ เนื่องจากคำสั่ง

SELECT เป็นคำสั่งที่มีรูปแบบการใช้งานที่ง่าย เพื่อช่วยในการค้นหาข้อมูลที่ซับซ้อน มีรูปแบบดังนี้

- SELECT <ชื่อคอลัมน์ที่ต้องการดูข้อมูล>

- FROM <ชื่อตาราง>

- WHERE <เงื่อนไขตามระบุบ>

- SELECT --- เป็นคำสั่งในการเรียกดูข้อมูลในคอลัมน์ที่ระบุซึ่งอาจจะมากกว่าหนึ่งก็ได้

และถ้ามีมากกว่าหนึ่งคอลัมน์ ตองคั่นด้วยเครื่องหมายคอมมา ( ,) และนอกจากนี้ยังสามารถใช้เครื่องหมายดอกจัน (\*) เพื่อแสดงถึงการขอดูข้อมูลทั้งหมดไดอีกดวย

- FROM --- เป็นส่วนประกอบของคำสั่งที่บอกถึงตารางที่ต้องการดูซึ่งอาจจะมีมากกว่าหนึ่งตารางก็ไดที่จะถูกเรียกใช้จากคำสั่ง SELECT

- WHERE--- เป็นส่วนประกอบของคำสั่ง ที่ใช้บ่งบอกเงื่อนไขที่จะใช้ในการค้นหาข้อมูล ขึ้นมาจากตารางใด ๆ ที่อยู่หลัง FROM นี้

- การเรียกดูแบบซ้อนทับกัน (Nested SELECT Statement)

- SELECT <ชื่อคอลัมน์>

- FROM <ชื่อตาราง>

- WHERE <ชื่อคอลัมน์> IN

- (SELECT <ชื่อคอลัมน์>

- FROM <ชื่อตาราง>

- WHERE <ชื่อคอลัมน์>)

คำสั่งเติมข้อมูล (Insert Statement)

INSERT INTO < ชื่อตาราง >

VALUES (< ชื่อคอลัมน์\_1> [,<ชื่อคอลัมน์\_2>]…);

คำสั่งแกไขและลบแถว (Update Statement)

UPDATE < ชื่อตาราง >

SET <ค่าที่ต้องการ>

WHERE <เงื่อนไข>

ภาษาควบคุม (Control Language)

เป็นภาษาที่ใช้ควบคุมระบบรักษาความปลอดภัยของฐานข้อมูล ประกอบด้วย 2 คำสั่งคือ

คำสั่ง GRANT เป็นคำสั่งที่ใช้กำหนดสิทธิให้กับผู้ใช้แต่ละคนให้มีสิทธิกระทำการใดกับข้อมูล เช่นการเพิ่มข้อมูล การแกไข หรือ การลบข้อมูลในตารางใดบ้าง

คำสั่ง REVOKE เป็นคำสั่งให้มีการยกเลิกสิทธินั้นหลังจากที่ได GRANT แลวคาบูลลีน (Boolean-Type Data)

AND - OR

NOT - =

> หรือ >= - <หรือ=<

<> - Bulit-InFunction

COUNT - SUM

AVG - MAX

MIN - DISTINCT

ชุดคำสั่ง GROUP BY, ORDER BY และ HAVING

เนื่องจากข้อมูลที่สนใจมักจะเป็นผลสรุปหรือข้อมูลที่ใช้ในการตัดสินใจ ภาษา SQL จึงไดออก

แบบให้มีชุดคำสั่งพิเศษ ที่ทำหน้าที่แบงออกเป็นกลุ่มๆ ไดแก่

1.1 ชุดคําสั่ง GROUP BY

GROUP BY เปนคําสั่งใหมีการจัดกลุมแถวขอมูลตามคอลัมน โดยขอมูลที่เหมือนกันจะถูกจัดใหอยูในกลุมเดียวกัน

SELECT S#, MAX( QTY)

FROM SPJ

GROUP BY S#

ผลลัพธ์

|  |  |
| --- | --- |
| S# | QTY |
| S | 700 |
| S2 | 800 |
| S3 | 500 |
| S4 | 300 |
| S5 | 800 |

1.2 ชุดคําสั่ง ORDER BY

ORDER BY เปนคําสั่งใหมีการเรียงลําดับขอมูลในแถวขอมูลตามคอลัมน โดยขอมูลที่เหมือนกันจะถูกจัดเรียงจากนอยไปมาก ถาไมการระบุบ แตถาระบุบวาเปน DESC

จะเรียงจากมากไปนอย ดังนั้นจากตัวอยางตอไปนี้

- SELECT S#, MAX (QTY)

- FROM SPJ

- GROUP BY S#

- ORDER BY 2 DESC

|  |  |
| --- | --- |
| S# | QTY |
| S5 | 800 |
| S2 | 800 |
| S1 | 700 |
| S3 | 500 |
| S4 | 300 |

หมายเหตุ ตัวเลข 2 ที่อยูหลัง ORDER BY จะเปน การบอกระบบใหทําการเรียงลําดับขอมูล  
ตามลําดับคอลัมนที่ 2 ของคําสั่ง SQL ซึ่งก็คือ MAX (QTY) นั่นเอง

1.3 ชุดคําสั่ง HAVING

HAVING เหมือนกับคําสั่ง WHERE ที่ตองตามดวยเงื่อนไข แต HAVING จะใชในกรณีที่มีการจัดกลุมหรือการใช GROUP BY

2.4 งานวิจัยหรือผลงานที่เกี่ยวข้อง

โดยปกติแล้วทางร้าน เดอะพิซซ่าคอมปานี สาขา หางดง จะไม่มีระบบการจัดลำดับพนักงานส่ง แต่เป็นการ การใส่วิน(การนำเลขลำดับประจำตัวใส่ลงในช่องเตรียมส่งสินค้า)แทนและเมื่อมีรายการสั่งจากลูกค้าเข้ามา ผู้จัดการร้านและผู้ช่วยผู้จัดการร้าน จะเป็นผู้จัดรายการสินค้าที่ต้องไปส่งลูกค้าให้พนักงาน และดูในช่องจัดลำดับว่า ณ ตอนนี้ลำดับของพนักงานอยู่ในหมายเลขไหน และให้พนักงานหมายเลขนั่น จะต้องเป็นผู้ส่งสินค้าให้แก่ลูกค้า

ซึ่งในการใส่วิน(การนำเลขลำดับประจำตัวใส่ลงในช่องเตรียมส่งสินค้า)นั่น ทำให้ทราบได้ทันทีว่า พนักงานหมายเลขไหนพร้อมถึงคิวในการส่ง และหมายเลขไหน ยังต้องรอคิวต่อไป ซึ่งถ้าหากพนักงานส่งได้เห็นใบการส่งสินค้า ว่าที่อยู่บ้านของลูกค้ามีระยะทางที่ไกลออกไปมาก ก็สามารถ ถอดวิน(การนำเลขลำดับประจำตัวออกจากช่องเตรียมส่งสินค้า)ของตัวเองออกได้ เพื่อที่จะได้ไม่ต้องไปส่งสินค้าไกลๆ และนั่นก็หมายความว่า ลำดับที่ต่อจากพนักงานที่ ถอดวิน(การนำเลขลำดับประจำตัวออกจากช่องเตรียมส่งสินค้า) จะต้องเป็นผู้ไปส่งแทน และถ้าหาก พนักงานทุกคนต่างถอดวิน (การนำเลขลำดับประจำตัวออกจากช่องเตรียมส่งสินค้า) ก็จะทำให้ ไม่มีพนักงานส่งสินค้า และทำให้การส่งสินค้ามีความล่าช้า

การทำงานของระบบการจัดลำดับพนักงานส่ง เดอะพิซซ่าคอมปานี สาขาหางดง 2 นั่นเป็นการเรียงลำดับพนักงานส่งโดยการลงชื่อ โดยพนักงานที่ลงชื่อคนแรก ก็จะได้ส่งสินค้าเป็นลำดับแรก และพนักงานส่งที่ลงชื่อต่อจากนี้ก็เป็นลำดับ ต่อไป และระบบการจัดลำดับพนักงานส่งนี้ พนักงานส่งไม่สามารถเห็นรายการสั่งสินค้าจากลูกค้าก่อนได้ และไม่สามารถ ถอดวิน (การนำเลขลำดับประจำตัวออกจากช่องเตรียมส่งสินค้า) ของตัวเองออกได้ จึงทำให้ไม่เกิดปัญหาในเรื่องของพนักงานส่ง และการส่งสินค้าถึงมือลูกค้าล่าช้า