**Web Programming**

**Unit 4 : Server Side Languages SQL**

**Server Side**

⊗ บทที่ผ่านมาเรียนรู้เกี่ยวกับการพัฒนาเว็บไซต์และการเขียนโปรแกรมในส่วนของการออกแบบ UI หน้าเว็บเพจ <front end> เป็นส่วนของ Client Side

⊗ หากเว็บไซต์มีการพัฒนาเฉพาะส่วน front end และมีการเผยแพร่ข้อมูลเฉพาะในส่วนนี้จะเรียกประเภทเว็บไซต์ว่า Static Website หรือ Static Webpage

⊗ ในส่วนเว็บไซต์ที่มีการบันทึกข้อมูล เช่น ข้อมูลสมาชิก ข้อมูลผลิตภัณฑ์หรือเนื้อหาต่างๆของเว็บไซต์ที่มีการเก็บข้อมูลเหล่านั้นและมีระบบการจัดเก็บข้อมูล ระบบบันทึกดึงข้อมูลมาแสดงกระบวนการทั้งหมดนี้จัดอยู่ในการพัฒนาเว็บไซต์ในส่วนของ back-end จะเรียกว่า Server Side Programming เป็นการออกแบบและพัฒนาในส่วนของฐานข้อมูล

⊗ ภาษาที่ใช่ เช่น SQL, PHP, JSP, ASP.NET

⊗ ปัจจุบันมี framework ที่ทำให้งานในส่วนนี้ง่ายขึ้น เช่น Node.js, Laravel

**With and without a database**

⊗ ฐานข้อมูล <Database> คือ ที่เก็บข้อมูลของเว็บไซต์มีระบบการจัดการข้อมูล เช่น การบันทึก การนำข้อมูลมาแสดง การเพิ่ม การลบข้อมูลแต่ละชนิดสามารถมีความสัมพันธ์กันได้เราเรียกฐานข้อมูลลักษณะนี้ว่า Relation Database หรือฐานข้อมูลเชิงสัมพันธ์และระบบจัดการฐานข้อมูลนี้เรียกว่า Relation Database Management System : RDBMS

⊗ เว็บไซต์ที่ไม่มีระบบการจัดเก็บข้อมูลหรือไม่มีฐานข้อมูลจะมีเฉพาะส่วนของ Client Side อาจเรียกเว็บไซต์ลักษณะนี้ว่าเป็นเว็บไซต์แบบ Static ดังที่ได้กล่าวไปในสไลด์ก่อนหน้าจะมีแค่การปฏิสัมพันธ์บ้างโดย JavaScript หากผู้ดูแลเว็บ <Web Administrator> ไม่อัพเดทเนื้อหาก็จะไม่มีความเคลื่อนไหวใดๆแต่หากเว็บไซต์ใดมีการใช้ฐานข้อมูล เช่น การสมัครสมาชิก การตั้งกระทู้ การสั่งซื้อสินค้า เรียกว่าเว็บไซต์แบบ Dynamic เพราะมีความเคลื่อนไหวของข้อมูลจากผู้ใช้และข้อมูลจะมีการเปลี่ยนแปลงอยู่เรื่อยๆหรือตลอดเวลาโดยผู้ดูแลหรือผู้ใช้

**SQL : Structured Query Language**

⊗ ภาษาสำหรับการจัดการข้อมูลในฐานข้อมูล เช่น การสร้างฐานข้อมูล การนำข้อมูลเข้าสู่ฐานข้อมูล การเรียกดูข้อมูล การลบข้อมูล การกำหนดสิทธิ์การเข้าถึงข้อมูลเป็นต้น

⊗ ข้อมูลใน RDBMS จะถูกเก็บอยู่ใน Object เรียกว่าตาราง <Table> ซึ่งข้อมูลประเภทเดียวกันจะอยู่ในตารางเดียวกัน เช่น ข้อมูลทุกคนผู้ใช้จะอยู่ในตารางชื่อ “User” ข้อมูลสินค้าทุกชิ้นจะอยู่ในตารางชื่อ “goods” โดยข้อมูลจะถูกเก็บอยู่ในรูปแบบของคอลัมน์และแถว <Column and row> และกล่าวได้ว่าทุกตารางในฐานข้อมูลสามารถมีความสัมพันธ์กันได้หมายถึงเราสามารถนำข้อมูลของแต่ละตารางมาใช้งานร่วมกันได้เพื่อให้การใช้งานเว็บไซต์นั้นมีความสามารถมากขึ้น

Diagram

Description automatically generated

⊗ ภาษา SQL เป็นภาษาที่มี syntax ที่เข้าใจได้ง่ายและไม่ซับซ้อน เช่น



⊗ แปลตามตัวอักษรได้วว่า เลือก “ชื่อลูกค้า” จากตารางชื่อ ”ลูกค้า” ซึ่งมีที่อยู่คือ ”Obere Str.57” เป็นการแสดงชื่อลูกค้าที่มีที่อยู่ดังกล่าวโดยข้อมูลนี้อยู่ในตาราง Customers โดยคำว่า ”WHERE” เป็นเงื่อนไขที่เราต้องการหากใครมีที่อยู่ก็จะแสดงชื่อของเจ้าของที่อยู่

**ตัวอย่างของตารางชื่อ “customer”**

⊗ สำหรับเก็บข้อมูลลูกค้าซึ่งจะสังเกตได้ว่าข้อมูลจะถูกเก็บอยู่ในตารางมีโครงสร้างเป็นแถว <row> และคอมลัมน์ <column>

⊗ คอมลัมน์ <column> แนวตั้ง

⊗ แถว <row> แนวนอน

⊗ แถว 1 แนวจะเก็บข้อมูลลูกค้า 1 รายเรียกว่า เรคคอร์ด <record> ซึ่งข้อมูลที่เก็บได้แก่ CustomerName (ชื่อลูกค้า), ContactName (ชื่อผู้ติดต่อ), Addresss (ที่อยู่), City (เมือง), PostalCode (รหัสไปรษณีย์), Country (ประเทศ) และจะเก็บอยู่ในแนวตั้ง

⊗ ดังนั้นในตารางนี้จะมีข้อมูลลูกค้าอยู่ด้วยกันทั้งหมด 5 record คือ มีลูกค้าทั้งหมด 5 รายด้วยกัน

Graphical user interface, application

Description automatically generated

**DBMS : Database Management System**

⊗ ชื่อย่อของระบบการจัดการฐานข้อมูล

⊗ ปัจจุบันมีหลายระบบ เช่น MySQL, Microsoft Access, MS SQL Server, Oracle ซึ่งข้อมูลต่างๆสามารถมีความสัมพันธ์กันได้ เราเรียกระบบฐานข้อมูลที่จัดการกับข้อมูลที่มีความสัมพันธ์กันว่า RDBMS : Relation Database Management System

⊗ นอกจากนี้ยังมีข้อมูลรูปแบบอื่นๆ เช่น ข้อมูลเชิงแผนที่ที่ต้องการเก็บข้อมูลพิกัดบนแผนที่ ตำแหน่ง ขอบเขตของพื้นที่ซึ่งจะอยู่ในรูปแบบกราฟและแกนต่างๆ เช่น x, y, z ของกราฟเราเรียกระบบฐานข้อมูลนี้ว่าระบบจัดการฐานข้อมูลเชิงพื้นที่ SDBMS : Spatial Database Management System

**RDBMS : Relation Database Management System**

⊗ ตารางแต่ละตารางจะมีคีย์เพื่อบ่งบอกถึงความสัมพันธ์กับตารางอื่นๆโดยจะนำข้อมูล 1 คอลัมน์ไปใช้กับตารางอื่นจะเรียกข้อมูลคอลัมน์นี้ว่าคีย์หลัก <Primary key : PK> ซึ่งแต่ละตารางจะมีคีย์หลักได้เพียง 1 คีย์เท่านั้น

⊗ คีย์นอก <Foreign Key : FK> คือ คีย์หลักของตารางอื่นที่นำมาใช้ยังตารางนี้โดยแต่ละตารางสามารถมีคีย์นอกได้มากกว่า 1 คีย์

**ตัวอย่างฐานข้อมูลของเว็บขายสินค้า**

⊗ ประกอบด้วยตาราง 2 อัน คือ customers เก็บข้อมูลลูกค้า, orders เก็บรายละเอียดคำสั่งซื้อ

⊗ สังเกตว่า PK ของ customers คือ customerNumber และ PK ของ orders คือ ordersNumber ในแต่ละ order จะต้องการระบุว่าลูกค้าคนใดเป็นคนสั่งซื้อ

⊗ ดังนั้นจึงต้องนำ customerNumber มาระบุใน orders จึงบอกได้ว่า customerNumber คือ FK ของ orders

⊗ สรุปได้ว่า PK ของตารางใดๆ คือ FK ของตารางอื่นๆที่ถูกนำมาไปใช้

Diagram

Description automatically generated

**phpMyAdmin**

⊗ เป็นระบบจัดการฐานข้อมูล RDBMS ซึ่งใช้พื้นฐานความรู้ทางด้านบานข้อมูลและภาษา SQL โดยข้อมูลจะถูกเก็บไว้เป็นตาราง เช่น ตารางลูกค้าซึ่งในเว็บไซต์หนึ่งอาจมีตารางได้หลายตารางขึ้นอยู่กับข้อมูลที่ต้องการจัดเก็บ เช่น เว็บไซต์ขายสินค้าอาจมีตารางลูกค้า สินค้า ผู้ขาย การสั่งซื้อเป็นต้น

Graphical user interface, application

Description automatically generated

**SQl Queries - SELECT**

⊗ Query แปลว่า การร้องขอ ซักถาม เรียกร้อง คือ การกระทำบางอย่างกับฐานข้อมูล เช่น create, select, insert into และคำสั่งอื่นๆที่สามารถกระทำได้ด้วยภาษา SQL คือ การติดต่อและการร้องขอไปยังฐานข้อมูลเพื่อทำงานกับข้อมูล

⊗ คำสั่งในการดึงข้อมูลหรือเข้าถึงข้อมูลในฐานข้อมูล คือ SELECT

Logo, company name

Description automatically generated

⊗ คำว่า SELECT เป็น syntax ในการเข้าถึงข้อมูลและเลือกข้อมูล

⊗ name คือ คอลัมน์ที่ต้องการเลือก

⊗ FROM คือ จากตารางใด

⊗ Customers คือ ชื่อตารางในฐานข้อมูลที่ต้องการติดต่อเพื่อเรียกข้อมูลในที่นี้คือ Customers

⊗ ดังนั้น สรุปได้ว่าโค้ดนี้ต้องการดึง name จาก Customers มาแสดง = เอาชื่อลูกค้าทั้งหมดมาแสดง

⊗ ข้อความต่อจาก SELECT จะเป็นคอลัมน์ในตารางซึ่งเราสามารถระบุเฉพาะคอลัมน์ที่ต้องการได้

A picture containing logo

Description automatically generated

⊗ ฉะนั้นจากตาราง Customers เราสามารถเขียน query ลักษณะนี้ได้ เช่น



⊗ หมายถึง ต้องการถึงข้อมูลเฉพาะชื่อและที่อยู่เท่านั้น



⊗ จากตัวอย่างจะเห็นว่าไม่ได้มีการระบุชื่อคอลัมน์แต่เป็นการใส่สัญลักษณ์ \* แทนหมายถึงการดึงคอลัมน์ทุกคอลัมน์มาแสดง

**SQl Queries - WHERE**

⊗ สามารถใส่เงื่อนไขได้ในกรณีที่ต้องการเลือกข้อมูลแบบมีเงื่อนไข เช่น หากต้องการถึงข้อมูลชื่อลูกค้าโดยแสดงเฉพาะคนที่อยู่ปะเทศ Mexico สามารถเขียนได้ดังนี้



⊗ โดยใส่คำว่า WHERE แสดงความเป็นเงื่อนไข

Chart, scatter chart

Description automatically generated

**SQl Queries – INSERT INTO**

⊗ เป็นคำสั่งสำหรับการป้อนข้อมูลเพื่อบันทึกลงไปยังตารางในฐานข้อมูล



⊗ โดยตำแหน่งและจำนวน column และ value จะต้องตรงกัน เช่น



⊗ ตำแหน่งแรกในช่อง value จะถูกบันทึกไปยังคอลัมน์ CustomerName และตำแหน่งถัดไปใน value จะถูกบันทึกในคอลัมน์ถัดไปตามที่ระบุไว้

**NULL**

⊗ ค่า Null เป็นค่าที่ระบุว่าไม่มีค่า คือ ไม่ทราบชนิดของค่า <data type> ว่าเป็น int, string, float หรือปะเภทใดๆก็ตามและไม่ใช่ค่า 0 เพราะตัวเลข 0 คือค่าศูนย์แต่ null คือไม่มีค่าและไม่ทราบค่า

⊗ ประโยชน์ของ null สามารถนำไปใช้ได้หลายรูปแบบ เช่น การออกคำนำหน้าอาจไม่บังคับให้กรอกซึ่งเมื่อไม่มีการกรอกขึ้นก็จะให้ค่าในช่องนั้นเป็น null และเมื่อผู้ดูแลระบบต้องการดูข้อมูลผู้ที่ไม่ได้ทำการกรอกคำนำหน้าชื่อก็สามารถ query โดยใส่เงื่อนไขเป็นค่า null เช่น



⊗ ผลลัพธ์จะแสดงเฉพาะชื่อผู้ที่ไม่ได้กรอกคำนำหน้าสามารถใช้



⊗ ในกรณีที่ต้องการเฉพาะชื่อคนที่มีคำนำหน้า

**ตัวอย่างการ query โดยมีค่า Null เป็นเงื่อนไข**

Graphical user interface, text, application, email

Description automatically generated

**AUTO INCREMENT**

⊗ Relational Database เป็นระบบฐานข้อมูลที่ต้องการคีย์หลัก <PK, FK> เพื่อนำไปใช้ในตารางอื่นเพื่อให้เกิดความสัมพันธ์และการเชื่อมโยงเพื่ออ้างอิงโดยจะต้องระบุคอลัมน์ 1 คอลัมน์เพื่อใช้เป็น PK โดยมีข้อกำหนดวาคอลัมน์ที่เป็น PK จะต้องมีข้อมูลที่ไม่ซ้ำกันในแต่ละ record

Graphical user interface, application

Description automatically generated

⊗ จากภาพจะเห็นได้ว่าในแต่ละคอลัมน์สามารถมีข้อมูลที่ซ้ำกันได้คนสามารมีชื่อ ที่อยู่ ประเภทเพศที่ซ้ำกันได้จังต้องมีการกำหนดรหัส <ID> เพื่อนำไปใช้เป็น PK และ ID จะไม่ซ้ำกันเลย

⊗ กำหนด primary key สามารถทำได้ตั้งแต่ขั้นตอนการสร้างคอมลัมน์โดยการใส่เครื่องหมายถูกที่ A\_I <Auto Increment>

⊗ เมื่อทำการตั้งค่าแล้วเมื่อมีการ insert ข้อมูลเข้ามาและมีการสร้าง record ใหม่คำในคอลัมน์ที่เป็น PK จะถูกใส่อัตโนมัติเป็นแบบตัวเลขและเรียงเพิ่มขึ้นเรื่อยๆจาก record ล่าสุด

Graphical user interface

Description automatically generated

**PHP**

⊗ PHP เป็นภาษา script เหมือน JavaScript ที่สามารถฝังลงในภาษา HTML ได้เพียงแต่ PHP นั้นมีหน้าที่แตกต่างออกไป

⊗ PHP เป็นทางเลือกที่ใช้กันอย่างแพร่หลาย ฟรีและมีประสิทธิภาพ

⊗ ภาษาที่เป็นคู่แข่งสำหรับ PHP น ASP ของ Microsoft และ JPS ของ google

Chart, scatter chart

Description automatically generated

**เว็บไซต์ที่มีการใช้งาน PHP**

⊗ WordPress

⊗ Wikipedia

⊗ Slack

⊗ Etsy

⊗ Flickr

⊗ iStockPhoto

⊗ Facebook เคยใช้

⊗ canva

Chart, bubble chart

Description automatically generated

⊗ ก่อนการเขียน PHP ทุกครั้งจะต้องมีการประกาศด้วยเครื่องหมาย <?php และปิดด้วย ?>

⊗ ลักษณะการแทรกโค้ด PHP จะเป็นลักKณะเดียวกับการแทรก JavaScript และภาษา PHP จะแทรกอยู่ตำแหน่งใดของไฟล์ HTML ก็ได้ซึ่งหากไฟล์ HTML โดยมีคำสั่งภาษา PHP ฝังอยู่จะต้องตั้งชื่อนามสกุลไฟล์นั้นเป็น “.php”

A picture containing text

Description automatically generated

**PHP คือออะไร**

⊗ PHP เป็นตัวย่อสำหรับ “PHP: Hypertext Preprocessor”

⊗ PHP เป็นใช้กันอย่างแพร่หลายมาเปิดภาษา script

⊗ PHP script จะดำเนินการบน server

⊗ PHP สามารถนำมาใช้ได้ฟรี

⊗ PHP มีประสิทธิภาพมากพอที่จะเป็นภาษาหลักในการขับเคลื่อนระบบของเว็บบล็อกที่ใหญ่ที่สุดในโลกบนเว็บ <WordPress> และก็ยังมีคงามลึกซึ้งพอที่จะใช้ใน social network ที่ใหญ่ที่สุด <Facebook> นอกจากนี้ยังง่ายพอที่จะเริ่มต้นครั้งแรกของ server ภาษาด้าน

**Cookie**

⊗ cookie มักจะถูกนำมาใช้เพื่อระบุผู้ใช้ เช่น พฤติกรรมการท่องเว็บของเว็บใดเว็บหนึ่งการคลิกจากลิงค์หนึ่งไปยังลิงค์หนึ่งและหลังจากนั้นไปยังเพจใดต่อเพื่อเหตุผลทางการตลาดโดยนำออกมาเป็นสถิติหลังจากรวบรวมข้อมูลของผู้ใช้หลายๆคนรวมกัน

⊗ cookie เป็นไฟล์ขนาดเล็กฝังในเครื่องคอมพิวเตอร์ของผู้ใช้แต่ละครั้งที่คอมพิวเตอร์เครื่องเดียวกันร้องขอหน้าเว็บเบราว์เซอร์จะส่ง cookie ไปด้วยคำสั่ง PHP

⊗ cookie จะเก็บข้อมูลหลักๆอยู่ 3 ประเภทคือ

1. CookieName คือ ชื่อ cookie

2. Value คือ ค่าของข้อมูลที่ cookie เก็บ เช่น ข้อมูลการเข้าเพจต่างๆขของ user

3. ข้อมูลประกอบอื่นๆ เช่น เวลาหมดอายุ, ข้อมูลเกี่ยวกับ Host, Path

**สร้าง Cookie**

⊗ จากภาพเป็นการสร้าง cookie ที่มีชื่อว่า user มีค่า John Doe cookie จะหมดอายุหลังจาก 30 วัน (86400\*30) เครื่องหมาย “/” คือ path หมายความว่า cookie นี้ใช้กับทุกเพจในเว็บนั้นๆหรือกำหนด directory เพื่อใช้กับบางเพจ จากนั้นเราจะเรียกใช้ค่า cookie user โดยใช้ตัวแปรแบบ global$\_COOKIE

Graphical user interface, application

Description automatically generated

**Session**

⊗ เป็นวิธ๊การในการจัดเก็บข้อมูลในรูปแบบตัวแปรที่จะนำมาใช้ในหลายๆเพจซึ่งแตกต่างจาก cookie ข้อมูลที่ไม่ได้เก็บไว้บนคอมพิวเตอร์ของผู้ใช้

⊗ เมื่อทำงานกับโปรแกรมหรือเปิดคอมพิวเตอร์เพื่อทำการเปลี่ยนแปลงบางอย่างและปิดมันลักษณะนี้เป็นเหมือน session คอมพิวเตอร์รู้ว่าผู้ใช้จาก username ที่กรอกแต่บนระบบ internet webserver ไม่ทราบว่าผู้ใช้เป็นใครเพราะโปรโตคอล HTTP นั้นไม่มีการเก็บสถานะของผู้ใช้

⊗ session เข้ามาแก้ปัญหานี้โดยการจัดเก็บข้อมูลของผู้ใช้ที่จะนำมาใช้ในหลายๆเพจ เช่น session มีอายุจนถึงผู้ใช้ปิดเบราว์เซอร์

⊗ ดังนั้นตัวแปร session เก็บข้อมูลเกี่ยวกับผู้ใช้เดียวและมีอยู่ในทุกเพจในเว็บไซต์หนึ่ง

**Session - ประกาศใช้งาน**

⊗ session จะเริ่มต้นด้วยคำสั่ง session\_start() ฟังก์ชันและจะต้องประกาศก่อนแท็ก <html>

⊗ ตัวแปร session ถูกกำหนดด้วยตัวแปร $\_SESSION

⊗ การกำหนดค่าให้กับตัวแปร session สามารถทำได้โดย $\_SESSION [“ชื่อตัวแปร”] = “ค่าตัวแปร”; ดังภาพ

Text

Description automatically generated

**Session - การรับค่าจากเพจอื่น**

⊗ การรับค่าแปรนั้นสามารถทำได้ด้วยการประกาศ session\_start() ซึ่งจะทำการรับค่าตัวแปรจากเพจก่อนหน้าโดยอัตโนมัติโดยไม่จำเป็นต้องประกาศใหม่

A picture containing graphical user interface

Description automatically generated

**Session – การเปลี่ยนแปลงค่า**

⊗ การเปลี่ยนค่าในตัวแปร $\_SESSION สามารถทำได้ด้วยการประกาศตัวแปรใหม่และใช้ชื่อตัวแปรเดิมแต่ระบุค่าใหม่ลงไปเสมือนเป็นการกำหนดค่าใหม่ทับค่าเดิม

A picture containing graphical user interface

Description automatically generated

**Session – การยกเลิก session**

⊗ ในการลบตัวแปร session สามารถใช้คำสั่ง session\_unset() และการยกเลิกระบบ session หรือการทำลาย session การใช้ให้ใช้คำสั่ง session\_destroy():

Graphical user interface, text

Description automatically generated