



**โครงการ
ระบบสร้างโปรแกรมบัญชีโรงเรียน**

**เสนอ
ภราดาอาวุธ สีดาเกษ**

**จัดทำโดย
นายดอน วิภา
งานเทคโนโลยีและสารสนเทศ**

โรงเรียนอัสสัมชัญเทคนิคนครพนม

บทที่ 1

บทนำ

1. หลักการและเหตุผล

การบัญชี หมายถึง การจดบันทึกเหตุการณ์ทางการเงินไว้เป็นหมวดหมู่ การวิเคราะห์ การสรุปผล และการแปลความหมาย ซึ่งต้องอาศัยความรู้ในการทำบัญชี การจัดทำรายงานทางบัญชี การวางระบบบัญชี การตรวจสอบบัญชี และการจัดการทางบัญชี โดยยึดถือหลักปฏิบัติตามกรอบของแม่บทการบัญชี กฎหมายที่เกี่ยวข้องกับวิชาชีพบัญชี และมาตรฐานการบัญชีที่นักบัญชีโดยทั่วไปยอมรับและถือปฏิบัติสืบต่อกันมา ทั้งนี้ไม่ว่ากิจการจะมีลักษณะเป็นรูปแบบใด การทำบัญชีจะมีประโยชน์ในการช่วยให้เจ้าของกิจการควบคุมดูแลสินทรัพย์ได้ ช่วยให้ทราบถึงผลการดำเนินงานและฐานะทางการเงินของกิจการ ช่วยในการวางแผนการดำเนินงานได้ ซึ่งนอกจากจะเป็นประโยชน์ต่อเจ้าของกิจการ องค์กรแล้ว ยังมีประโยชน์ต่อการสรุปแผนงาน กิจกรรมต่าง ๆ ดังนั้น

2. วัตถุประสงค์

1. เพื่อบันทึกเหตุการณ์ทางการเงินไว้เป็นหมวดหมู่ การวิเคราะห์ การสรุปผล และการแปลความหมาย
2. เพื่อจัดทำรายงานทางบัญชี การวางระบบบัญชี การตรวจสอบบัญชี และการจัดการทางบัญชี
3. เพื่อวิเคราะห์ฐานะทางการเงิน ในการวางแผนการดำเนินงาน

3. ขอบเขตในการดำเนินงาน

1. ภาษา PHP
2. ฐานข้อมูล MySQL

4. ขั้นตอนในการดำเนินงาน

- 1.) นำเสนอโครงการ
- 2.) ศึกษาและเก็บรวบรวมข้อมูล
- 3.) วิเคราะห์และติดตั้งโปรแกรม
- 4.) ทดสอบโปรแกรม และแก้ไขข้อผิดพลาด
- 5.) นำเสนอผลงานต่อคณะกรรมการ
- 6.) จัดทำเอกสาร

5. ตารางการปฏิบัติงาน

กิจกรรม	เดือน 2553							
	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
นำเสนอโครงการ	→							
ศึกษาและเก็บรวบรวมข้อมูล	→							
วิเคราะห์และติดตั้งโปรแกรม		→						
ทดสอบโปรแกรม และแก้ไขข้อผิดพลาด		→			→			
นำเสนอผลงานต่อคณะกรรมการ					→			
จัดทำเอกสาร					→			→

6. ผลคาดว่าจะได้รับ

1. มีระบบบันทึกเหตุการณ์ทางการเงินไว้เป็นหมวดหมู่ การวิเคราะห์ การสรุปผล และ

การแปลความหมาย

2. มีระบบจัดทำรายงานทางบัญชี การวางระบบบัญชี การตรวจสอบบัญชี และการจัดการทางบัญชี

3. สามารถนำข้อมูลมาวิเคราะห์ฐานะทางการเงิน ในการวางแผนการดำเนินงาน

บทที่ 2

ความรู้พื้นฐาน

2.1 ความรู้พื้นฐานในการใช้งานภาษาที่ใช้เป็นเครื่องมือในการเขียน (PHP)

ประวัติความเป็นมาของภาษา PHP

PHP เป็นภาษาจำพวก scripting language คำสั่งต่างๆจะเก็บอยู่ในไฟล์ที่เรียกว่าสคริปต์ (script) และเวลาใช้งานต้องอาศัยตัวแปลชุดคำสั่ง ตัวอย่างของภาษาสคริปต์ก็เช่น JavaScript, Perl เป็นต้น ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือ PHP ได้รับการพัฒนาและออกแบบมาเพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถ สอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ ดังนั้นจึงกล่าวได้ว่า PHP เป็นภาษาที่เรียกว่า server-side หรือ HTML-embedded scripting language เป็นเครื่องมือที่สำคัญชนิดหนึ่ง ที่ช่วยให้เราสามารถสร้างเอกสารแบบ Dynamic HTML ได้อย่างมีประสิทธิภาพและมีลูกเล่นมากขึ้น ถ้าใครรู้จัก Server Side Include (SSI) ก็จะสามารถเข้าใจการทำงานของ PHP ได้ไม่ยาก สมมติว่า เราต้องการจะแสดงวันเวลาปัจจุบันที่ผู้เข้ามาเยี่ยมชมเว็บไซต์ในขณะนั้น ในตำแหน่งใดตำแหน่งหนึ่งภายในเอกสาร HTML ที่เราต้องการ อาจจะใช้คำสั่งในรูปแบบนี้ เช่น `<!--#exec cgi="date.pl"-->` ไว้ในเอกสาร HTML เมื่อ SSI ของ web server มาพบคำสั่งนี้ ก็จะกระทำคำสั่ง date.pl ซึ่งในกรณีนี้ เป็นสคริปต์ที่เขียนด้วยภาษา perl สำหรับอ่านเวลาจากเครื่องคอมพิวเตอร์ แล้วใส่ค่าเวลาเป็นเอาพุท (output) และแทนที่คำสั่งดังกล่าว ลงในเอกสาร HTML โดยอัตโนมัติ ก่อนที่จะส่งไปยังผู้อ่านอีกทีหนึ่ง

อาจจะกล่าวได้ว่า PHP ได้รับการพัฒนาขึ้นมา เพื่อแทนที่ SSI รูปแบบเดิมๆ โดยให้มีความสามารถ และมีส่วนเชื่อมต่อกับเครื่องมือชนิดอื่นมากขึ้น เช่น ติดต่อกับคลังข้อมูลหรือ database เป็นต้น

PHP ได้รับการเผยแพร่เป็นครั้งแรกในปี.ศ.1994 จากนั้นก็มีการพัฒนาต่อมาตามลำดับ เป็นเวอร์ชัน 1 ในปี 1995 เวอร์ชัน 2 (ตอนนั้นใช้ชื่อว่า PHP/FI) ในช่วงระหว่าง 1995-1997 และเวอร์ชัน 3 ช่วง 1997 ถึง 1999 จนถึงเวอร์ชัน 4 ในปัจจุบัน

PHP เป็นผลงานที่เติบโตมาจากกลุ่มของนักพัฒนาในเชิงเปิดเผยรหัสต้นฉบับ หรือ OpenSource ดังนั้น PHP จึงมีการพัฒนาไปอย่างรวดเร็ว และแพร่หลายโดยเฉพาะอย่างยิ่งเมื่อใช้ร่วมกับ Apache Webserver ระบบปฏิบัติการอย่างเช่น Linux หรือ FreeBSD เป็นต้น ในปัจจุบัน PHP สามารถใช้ร่วมกับ Web Server หลายๆตัวบนระบบปฏิบัติการอย่างเช่น Windows 95/98/NT เป็นต้น

เนื่องจากว่า PHP ไม่ได้เป็นส่วนหนึ่งของตัว Web Server ดังนั้นถ้าจะใช้ PHP ก็จะต้องดูก่อนว่า Web server นั้นสามารถใช้สคริปต์ PHP ได้หรือไม่ ยกตัวอย่างเช่น PHP สามารถใช้ได้กับ Apache WebServer และ Personal Web Server (PWP) สำหรับระบบปฏิบัติการ Windows 95/98/NT

ในกรณีของ Apache เราสามารถใช้ PHP ได้สองรูปแบบคือ ในลักษณะของ CGI และ Apache Module ความแตกต่างอยู่ตรงที่ว่า ถ้าใช้ PHP เป็นแบบโมดูล PHP จะเป็นส่วนหนึ่งของ Apache หรือเป็นส่วนขยายในการทำงานนั่นเอง ซึ่งจะทำงานได้เร็วกว่าแบบที่เป็น CGI เพราะว่า ถ้าเป็น CGI แล้ว ตัวแปลชุดคำสั่งของ PHP ถือว่าเป็นแค่โปรแกรมภายนอก ซึ่ง Apache จะต้องเรียกขึ้นมา

ทำงานทุกครั้ง ที่ต้องการใช้ PHP ดังนั้น ถ้ามองในเรื่องของประสิทธิภาพในการทำงาน การใช้ PHP แบบที่เป็นโมดูลหนึ่งของ Apache จะทำงานได้มีประสิทธิภาพมากกว่า

ต่อไปเราจะมาทำความรู้จักกับภาษา PHP และทำความเข้าใจการทำงาน รวมถึงคำสั่งพื้นฐานต่าง ๆ

2.1.2 การสอดแทรกคำสั่งภาษา PHP ในเอกสาร HTML

เพื่อเป็นการบ่งบอกให้รู้ว่า ส่วนใดเป็นคำสั่ง PHP ที่อยู่ภายในเอกสาร HTML จึงได้มีการกำหนดสัญลักษณ์ไว้ดังนี้ ซึ่งสามารถทำได้หลายรูปแบบ เช่น

- ▶ `<? ... ?>` (SGML style)
- ▶ `<?php ... ?>` (XML style)
- ▶ `<script language="php"> ... </script>` (JavaScript style)
- ▶ `<% ... %>` (ASP style)

ที่นิยมก็คือแบบแรก โดยเริ่มต้นด้วย `<?>` และจบด้วย `?>` และตรงกลางจะเป็นคำสั่งในภาษา PHP เราสามารถวางคำสั่ง PHP ไว้ภายในเอกสาร HTML ตามที่ต้องการได้ อาจจะสลับกับ Tag ของภาษา HTML ก็ได้ ตัวอย่างเช่น

```
<HTML>
<HEAD><TITLE> My Homepage </TITLE></HEAD>
<BODY BGCOLOR=#FFFFFF>
<H1><? echo "Hello World"; ?></H1>
Your web browser is <? echo $HTTP_USER_AGENT; ?>.
</BODY>
</HTML>
```

คำสั่งแรกที่ย่างที่สุดสำหรับการเรียนรู้ ก็คือคำสั่ง `echo` แล้วตามด้วยข้อความหรือสตริงค์ (string) ข้อความในภาษา PHP จะเริ่มต้นและจบด้วย double quote (") เหมือนในภาษาซี

2.1.3 การใช้ตัวแปรในภาษา PHP

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

```
$mystring = "Hello World!";
```

```
$myinteger = 1031;
```

```
$myfloat = 3.14;
```

ถ้าเราต้องการจะแสดงค่าของตัวแปร ก็อาจจะใช้คำสั่ง echo ได้ ตัวอย่างเช่น

```
echo "$mystring\n";
```

```
echo "$myinteger\n";
```

```
echo "$myfloat\n";
```

สัญลักษณ์ `\n` หมายถึงการขึ้นบรรทัดใหม่ เป็น escape character ตัวหนึ่ง (สำหรับตัวอื่นๆ โปรดดูในตาราง) เมื่อพิมพ์ข้อความเป็นอาพุด และโปรดสังเกตว่า สำหรับการใช้งานภายในเอกสาร HTML การขึ้นบรรทัดใหม่โดยใช้ `\n` จะแตกต่างจากการขึ้นบรรทัดโดยใช้ `
` ใน HTML

```
<?
```

```
$mystring = "Hello World!";
```

```
$myinteger = 1031;
```

```
$myfloat = 3.14;
```

```
echo "$mystring<BR>\n";
```

```
echo "$myinteger<BR>\n";
```

```
echo "$myfloat<BR>\n";
```

```
?>
```

Escaped characters `\n` newline

`\r` carriage

`\t` horizontal tab

`\\` backslash

`\$` dollar sign

`\"` double-quote

`%%` percent

ตัวแปรตัวหนึ่ง อาจจะมีข้อมูลหลายแบบในช่วงเวลาที่ต่างกัน แต่การจะใช้งานบ้างครั้งจะต้องดูด้วยว่าเมื่อไหร่จะใช้เป็นตัวเลขเท่านั้น และไม่ใช้กับข้อความเป็นต้น ตัวอย่างเช่น

```
<?
```

```
$x = 10;
```

```
$y = $x + 15.5;
```

```
echo "$x, $y \n";
```

```

$x = "abc";
echo "$x \n";
$z = $x + 15.5;
echo "$x, $z \n";
echo ("100.5" - 16);
echo (0xef + 007);
?>

```

ในกรณีนี้เรากำหนดในตอนแรกว่า \$x ให้เก็บค่า 10 ซึ่งเป็นจำนวนเต็ม ถ้านำมาบวกกับ 15.5 ผลที่ได้ก็จะเป็น 25.5 ซึ่งกลายเป็นเลขทศนิยม แล้วเก็บไว้ในตัวแปร \$y ต่อมากำหนดให้ตัวแปร \$x เก็บสตริงที่เก็บข้อความ "abc" ถ้านำมาบวกกับ 15.5 กรณีนี้จะให้ผลที่ได้ไม่ถูกต้อง เนื่องจากไม่สามารถนำข้อความมาบวกกับตัวเลขได้

แต่ PHP อนุญาตให้เราทำเช่นนั้นได้ในบางกรณี สมมุติว่า สตริงก็มีเฉพาะตัวเลขและสามารถเปลี่ยนเป็น เลขจำนวนเต็ม หรือจำนวนจริงได้โดยอัตโนมัติ เราก็นำสตริงนี้มาบวกลบคูณหรือหารกับตัวแปรที่เก็บเป็นตัวเลขได้

ค่าคงที่สำหรับเลขจำนวนเต็ม อาจจะอยู่ในรูปของเลขฐานแปดหรือสิบหกก็ได้ ถ้าเป็นเลขฐานแปดจะมีเลขศูนย์นำ ถ้าเป็นเลขฐานสิบหกจะมี 0x นำหน้า

2.1.4 การอ่านและแปลงแบบข้อมูลในตัวแปรหรือค่าคงที่แบบเจาะจง

เราสามารถแปลงแบบข้อมูลจากแบบหนึ่งไปยังอีกแบบหนึ่ง (type casting) เช่น แปลงจากข้อความที่มีเฉพาะตัวเลขให้กลายเป็นเลขจำนวนเต็ม (int) หรือทศนิยม (double), (float), (real) หรืออาจจะใช้คำสั่ง settype() ทำได้ตามตัวอย่างต่อไปนี้

```

<?
$x = ((double)"100.1") + 0.3e+3;
echo $x," <BR>\n";
echo ($x=(int)$x)," <BR>\n";
$x = "P".$x."\n";
echo $x," <BR>\n";
$x= ceil(13.45); /* get integer part */
echo $x," <BR>\n";
if (!settype( $x, "integer" ) ) {
    echo "error\n";
}

```

```

}
echo $x, " $x%5=",($x%5)," <BR>\n"; ?>

```

2.1.5 การอ่านแบบข้อมูลของตัวแปรหรือค่าคงที่

ถ้าต้องการเช็คว่า ตัวแปรมีข้อมูลแบบใด เราสามารถใช้คำสั่ง `gettype()` ได้ ค่าที่ได้จากฟังก์ชันก็จะเป็น "integer" "double" หรือ "string" เป็นต้น

```

<?
echo gettype(0),"\n";
echo gettype(1.1),"\n";
echo gettype((1==1)),"\n";
$var="abc";
if ( gettype($var)=="string" ) {
    echo "this is a string\n";
}
?>

```

เราอาจจะไม่ใช่ `gettype()` ก็ได้ แต่เลือกใช้ฟังก์ชัน `is_long()` สำหรับเช็คค่าที่เป็นเลขจำนวนเต็ม, `is_string()` สำหรับเช็คค่าที่เป็นสตริงค์, `is_double()` สำหรับค่าที่เป็นเลขทศนิยม, `is_array()` สำหรับค่าที่เป็นอาร์เรย์ หรือ `is_object()` สำหรับค่าที่เป็นออบเจกจากคลาสแทน ซึ่งจะให้ค่าเท่ากับ true (1) ถ้าตัวแปรมีแบบข้อมูล ตรงตามที่กำหนด

```

<?
unset($a);
$a="hello";
if (is_string($a) == true) {
    echo "\$a is a string <BR>\n";
}
unset($a);
$a[]="red";
$a[]="green";
$a[]="blue";
if (is_array($a) == true) {
    echo "\$a is an array of size ",count($a),"<BR>\n";
}

```



```
} ?>
```

2.1.6 ตัวอย่างการใช้ echo เพื่อแสดงข้อความ (เพิ่มเติม)

การพิมพ์ค่าใดๆที่เก็บอยู่ในตัวแปร ถ้าชื่อของตัวแปรอยู่ในสตริงระหว่าง double quote เวลาสร้างเอาพุดแล้ว จะอ่านค่าของตัวแปรนั้นก่อนแล้วจึงแทนที่ลงในข้อความ แต่ถ้านำหน้าด้วย backslash (\) ก็จะไม่มีการอ่านค่าของตัวแปร เช่น "\\$a" จะให้ผลต่างจาก "\$a" สังเกตได้จากตัวอย่างต่อไปนี้

```
<?
$a=1;
echo "\$a=$a <BR>\n";
$test = "test";
echo "$test$test$test<BR>\n";
echo $test,$test,$test,"<BR>\n";

$a = 1;
$b = 2;
echo $a,"+",$b,"=",$a+$b,"<BR>\n";
echo $a,"+",$b,"=",$a+$b,"<BR>\n";
?>
```

สำหรับข้อความในภาษา PHP เราอาจจะใช้ single quote แทน double quote ได้ แต่เวลาใช้งานร่วมกับ echo หรือ print() จะให้ผลต่างกัน ซึ่งสังเกตได้จากตัวอย่างต่อไปนี้

```
<?
$a = "aaa";
$b = 'bbb';
echo "$a $b<BR>\n";
echo ' $a $b<BR>\n';
?>
```

ตัวแปลคำสั่งจะมองข้ามชื่อตัวแปรและรวมถึงพวก escape sequence ต่างๆด้วยที่อยู่ในข้อความที่ใช้ single quote

2.1.7 คำอธิบาย (หมายเหตุ) ในภาษา PHP

ถ้าเราต้องการเขียนคำอธิบายในส่วนใดๆก็ตามของสคริปต์ เราก็จะสามารถทำได้โดยใช้ /* ... */ เหมือนในภาษาซี หรือ // เหมือนในภาษาจาวา หรือ # เหมือน shell script โปรดสังเกตว่า // ใช้เขียนนำคำอธิบายในภายหลังบรรทัดหนึ่งๆ เท่านั้น ส่วน # ใช้เริ่มต้นของบรรทัดที่เขียนคำอธิบาย

```
<?
# comment
$a = 41; // set $a to 41.
$b =10; // set $b to 10.
$b += $a; /* add $a to $b */
echo $b," \n";
?>
```

2.1.8 ค่าการใช้คำสั่งสำหรับคำนวณเลขคณิต

บวก	(+)	เช่น $x + y$
ลบ	(-)	เช่น $x - y$
คูณ	(*)	เช่น $x * y$
หาร	(/)	เช่น x / y

ตารางที่ 2-1 ค่าการใช้คำสั่งสำหรับคำนวณเลขคณิต

หาเศษจากการหาร (%) หรือโมดูลัส เช่น $x \% y$ การเศษจากการหารโดยปรกติจะใช้กับเลขจำนวนเต็มเท่านั้น ถ้าใช้กับเลขมีจุดทศนิยม จะมีการปัดทิ้งเป็นจำนวนเต็มก่อน กำหนดให้ x มีค่าเท่ากับ 7 และ y มีค่าเท่ากับ 4

$x + y$	11
$x - y$	3
$x * y$	28
x / y	1.75

$\$x \% \y	3
--------------	---

ตารางที่ 2-2 ตัวอย่างการใช้งานการคำนวณเลขคณิต
กำหนดให้ $\$x$ มีค่าเท่ากับ 2.5 และ $\$y$ มีค่าเท่ากับ 4

$\$x + \y	6.5
$\$x - \y	-1.5
$\$x * \y	1.0
$\$x / \y	0.615
$\$x \% \y	2

ตารางที่ 2-3 ตัวอย่างการใช้งานการคำนวณเลขแบบทศนิยม

2.1.9 การเพิ่มหรือลดค่าของตัวเลขในตัวแปรที่ละหนึ่งตามแบบภาษาซีหรือจาวา

$\$x++$	เพิ่มค่าขึ้นอีกหนึ่ง
$++\$x$	เพิ่มค่าขึ้นอีกหนึ่ง
$\$x--$	ลดค่าลงอีกหนึ่ง

ตารางที่ 2-4 การเพิ่มลดค่าที่ละหนึ่ง

$--\$x$ ลดค่าลงอีกหนึ่ง ความแตกต่างของการวาง $++$ หรือ $--$ ไว้ข้างหน้าหรือข้างหลัง คือดูว่า จะอ่านค่าของตัวแปรก่อน (ในกรณีที่มีการอ่านค่าของตัวแปร) หรืออ่านค่าหลังจากการเพิ่มหรือลด
โปรดลองทำตามตัวอย่างแล้วสังเกตผลลัพธ์ที่ได้ในแต่ละกรณี

```
<?

```

```
$x=3;

```

```
echo $x++, "<BR>\n";

```

```
echo $x, "<BR>\n";

```

```
$x=3;

```

```
echo ++$x, "<BR>\n";

```

```
echo $x, "<BR>\n";

```

```

$x=3;
echo $x--,"<BR>\n";
echo $x,"<BR>\n";
$x=3;
echo --$x,"<BR>\n";
echo $x,"<BR>\n";
?>

```

2.1.10 การกำหนดค่าของตัวแปรที่เป็นตัวเลขหรือสตริงค์โดยใช้ assignment operators

การกำหนดค่า (assignment) หรือเปลี่ยนแปลงค่าให้แก่ตัวแปร จะใช้โอเปอเรเตอร์ (assignment operators) ได้ในหลายรูปแบบ เหมือนอย่างที่ใช้ในภาษาซี ตามตัวอย่างต่อไปนี้

```

$x=0;
$x += 1; // the same as $x = $x + 1;
$x--; // the same as $x = $x - 1;
$x *= 3; // the same as $x = $x * 3;
$x /= 2; // the same as $x = $x / 2;
$x %= 4; // the same as $x = $x % 4;
$x="";
$x .= 'A'; // append char to an existing string
$x .= "BC"; // append string to an existing string

```

จากตัวอย่างข้างบน ในกรณีของการต่อสตริงค์ เราจะใช้จุด (.) เป็นโอเปอเรเตอร์

2.1.11 การใช้ตัวแปรเป็นชื่อของตัวแปร

ภาษา PHP เปิดโอกาสให้เราสามารถเลือกหรือเปลี่ยนชื่อของตัวแปรได้ ตัวอย่างเช่น

```
<?
$a = "var1";
$$a = 10.3;
echo "$a ${$a} $$a <BR>\n";
echo "$var1 <BR>\n"; ?>
```

จากตัวอย่างข้างบน เรากำหนดให้ตัวแปร \$a เก็บสตริงค์ "var1" และจะใช้เป็นชื่อของตัวแปรอีกตัวหนึ่ง โดยทางอ้อม \$\$a เป็นการอ้างถึงตัวแปรที่มีชื่อเดียวกับค่าของตัวแปร \$a (ในกรณีนี้คือ var1) ดังนั้นถ้าเราเขียนว่า \$\$a หรือ \$var1 ก็หมายถึงตัวแปรตัวเดียวกัน ถ้าต้องการแสดงค่าของ \$\$a โดยใช้คำสั่ง echo โดยอยู่ในสตริงค์ (ระหว่าง double quotations) เราจะต้องเขียน \${\$a} ไม่ใช่ \$\$a เพราะว่า ถ้าเขียนตามแบบหลัง ตัวแปรคำสั่งจะอ่านค่า \$a ก่อนแล้วแทนที่ลงในข้อความ ซึ่งจะได้ \$var1 แทนที่จะเป็นการอ่านค่าของ \$var1 เทคนิคนี้ยังสามารถใช้ได้กับฟังก์ชัน ตัวอย่างเช่น

```
<?
function foobar() {
    echo "foobar<BR>\n";
}
function callFunc ($f) {
    if ( is_string($f) == true) {
        $f();
    }
}
callFunc("foobar");
?>
```

ตัวอย่างข้างบนอาจจะทำให้เกิดปัญหาถ้าสมมุติว่า \$f เป็นชื่อของฟังก์ชันที่ไม่มีอยู่จริง วิธีตรวจสอบคือ การใช้ฟังก์ชัน function_exists() ดังต่อไปนี้

```
<?
function MyFunc() {
    print ("ok..<BR>\n");
}
```

```

}
$f="myFunc";
if ( function_exists($f) ) {
    $f();
}
else {
    echo "$f does not exist!";
}
?>

```

2.1.12 การกำหนดค่าคงที่

ในภาษา PHP มีการทำสัญลักษณ์ให้เก็บค่าคงที่ เช่น อาจจะเป็นสตริงหรือตัวเลขก็ได้ สามารถทำได้โดยใช้ คำสั่ง DEFINE() สัญลักษณ์ที่กำหนดโดยคำสั่ง DEFINE() จะเหมือนกันตัวแปร ทั่วๆ ไป แต่แตกต่างกันที่ว่า เมื่อนิยามแล้วจะเปลี่ยนแปลงค่าอีกไม่ได้

```

<?
define(PI, 3.141592654);
define(YES, true);
define(NO, false);
define("AUTHOR", "RWS");
echo (PI/3),"<BR>\n";
echo "AUTHOR=".AUTHOR."<BR>\n";
echo "YES=".YES."<BR>\n";
?>

```

นอกจากสัญลักษณ์ที่ผู้ใช้นิยามขึ้นมาได้เองแล้วยังมีสัญลักษณ์กลุ่มหนึ่งที่ได้มีการนิยามไว้ก่อนแล้วใน ภาษา PHP ตัวอย่างเช่น

__FILE__	เก็บชื่อของไฟล์สคริปต์
__LINE__	เก็บเลขบรรทัดภายในสคริปต์ในตอนที่ใช้
TRUE	มีค่าเป็นจริง
FALSE	มีค่าเป็นเท็จ
PHP_VERSION	เก็บเวอร์ชันของ PHP

PHP_OS เก็บชื่อระบบปฏิบัติการที่ใช้ เช่น Linux

2.1.13 การทำขั้นตอนซ้ำหรือวนลูป

การวนลูปหรือสร้างลูปเพื่อทำงานซ้ำเป็นส่วนประกอบสำคัญของโปรแกรมคอมพิวเตอร์ ในภาษา PHP ก็จะใช้โครงสร้างเหมือนภาษาซี ดังต่อไปนี้

while-do loop

do-while loop

for-loop

ตัวอย่างการใช้ while-do loop เพื่อคำนวณค่า เลขยกกำลังสอง ซึ่งมีเลขฐานตั้งแต่ 1 ถึง 10

```
<?
$x = 1;
while ($x <= 10) {
    echo $x*$x,"\n";
    $x++;
}
?>
```

เริ่มต้นด้วยการกำหนดตัวแปร \$x ให้มีค่าเป็นหนึ่ง ซึ่งในกรณีนี้เราใช้เป็นเลขฐาน ในการคำนวณ เลขยกกำลังสอง เมื่อเข้าสู่การวนลูปแบบ while-do จะมีการตรวจสอบเงื่อนไข ของการวนลูปในแต่ละครั้งว่า เงื่อนไขเป็นจริงอยู่หรือไม่ ในกรณีนี้เรากำหนดเงื่อนไขในการวนลูปไว้ว่า ถ้าค่าของ \$x มีค่าน้อยกว่าหรือ เท่ากับ 10 ก็ให้ทำคำสั่งที่อยู่ภายในลูป ซึ่งก็คือ echo \$x*\$x,"\n"; โดยจะพิมพ์ค่าของผลคูณซึ่งหมายถึงเลขยก กำลังสองนั่นเอง หลังจากนั้น ก็ให้เพิ่มค่าของ \$x ทีละหนึ่งในการวนลูปแต่ละครั้ง ค่าของ \$x จะเพิ่มขึ้นเรื่อยๆจนมีค่ามากกว่า 10 เมื่อถึงเวลานั้น ก็จะเป็นการจบการวนลูป เพราะว่า เราจะได้ว่า เงื่อนไข (\$x <= 10) มีค่าเป็นเท็จ สมมุติว่า ถ้าเปลี่ยนจาก \$x++ เป็น \$x-- ปัญหาที่จะเกิดตามมาเวลาใช้งาน คือ แทนที่จะวนลูปแค่ สิบครั้ง ก็กลับกลายเป็นว่า เป็นการวนลูปนับครั้งไม่ถ้วน เพราะว่า ค่าของ \$x จะลดลงเรื่อยๆในการวนลูปแต่

ละครั้ง คือเป็นลบ และค่าเป็นลบจะน้อยกว่า 10 เสมอ (ยกเว้นแต่ว่า เมื่อถึงจุดเวลาหนึ่งค่าเป็นลบมากๆ จะกระโดดกลับเป็นบวก)

ตัวอย่างการใช้ do-while loop เพื่อคำนวณค่าเลขยกกำลังสอง ซึ่งมีเลขฐานตั้งแต่ 1 ถึง 10

```
<?
$x = 1;
do {
    echo $x*$x,"<BR>\n";
    $x++;
} while ($x < 10);
?>
```

โปรดสังเกตความแตกต่างระหว่างการใช้ while-do และ do-while โดยเฉพาะตรงเงื่อนไข ในการจบการวนลูป ในกรณีของ do-while เราจะกระทำขั้นตอนในลูปก่อนหนึ่งครั้ง แล้วค่อยตรวจสอบว่า เงื่อนไขในการวนลูปเป็นจริงหรือไม่ ความแตกต่างนี้ เราสามารถจำได้ง่ายๆ คือว่า ถ้าใช้ do-while จะต้องมีการทำคำสั่งภายในลูปหนึ่งครั้งเสมอ แม้ว่าเงื่อนไขโดยเริ่มต้นจะเป็นเท็จก็ตาม ซึ่งแตกต่างจาก while-do ถ้าเงื่อนไขเป็นเท็จตั้งแต่เริ่ม ก็จะไม่มีการทำคำสั่งที่อยู่ในลูป

อีกแบบหนึ่งสำหรับการวนลูปคือ ใช้ for-loop ทำได้ตามตัวอย่างต่อไปนี้

```
<?
for ($x = 1; $x <=10; $x++) {
    echo $x*$x,"<BR>\n";
}
?>
```

ในบรรทัดที่เริ่มต้นด้วย for ระหว่างวงเล็บเปิดและปิด จะถูกแบ่งเป็นสามส่วนโดยเครื่องหมาย semicolon (;) ในส่วนแรกเราสามารถใส่คำสั่งที่ต้องการจะกระทำก่อนเข้าลูป ส่วนแรกนี้จะมีหรือไม่มีก็ได้ ในส่วนที่สองจะเป็นเงื่อนไขสำหรับการทำ loop และในส่วนที่สามจะคำสั่งที่จะต้องทำเป็นการจบท้ายลูปในแต่ละครั้ง หลักการทำงานของ for-loop จะคล้ายกับ while-do-loop

การใช้งาน for-loop และวางตำแหน่งส่วนต่างๆ อาจจะไม่จำเป็นต้องทำเหมือนกันแต่ให้ผลเหมือนกัน เช่น

```
<?
$x=1;
for (; $x <=10; $x++) {
    echo $x*$x,"<BR>\n";
```



```

}
$x=1;
for (; $x <=10; ) {
    echo $x*$x,"<BR>\n";
    $x++;
}
?>

```

จากตัวอย่างข้างบนที่ผ่านมา เป็นการวนลูปจะใช้การนับเลขเพิ่มขึ้นทีละหนึ่ง เรายังสามารถเขียนใหม่โดยเป็นการนับเลขลดลง ยกตัวอย่างเช่น เราต้องการจะพิมพ์ตัวเลขเรียงลำดับจาก 10,9,8...,1 ก็อาจจะเขียนคำสั่งได้ดังนี้

```

<?
for ($x=10 ; $x >0; $x--) {
    echo $x,"<BR>\n";
}
?>

```

การใช้งาน for-loop ก็จะเหมือนกับเวลาใช้ในภาษาซี ในหลายๆเรื่อง เช่น เราสามารถใส่คำสั่งได้ มากกว่าหนึ่งโดยใช้เครื่องหมาย (,) เป็นตัวแยก ตัวอย่างเช่น

```

<?
for ($x=1, $y=0 ; $x < 10; $x++, $y--) {
    echo "$x $y<BR>\n";
}
?>

```

2.1.14 การแบ่งสายงาน โดยจำแนกตามเงื่อนไขแบบ if-else

ในบางครั้งมีความจำเป็นต้องจำแนกเงื่อนไขในการทำงาน โดยแต่ละเงื่อนไขจะกำหนดกรณีเพื่อทำคำสั่งหรือกลุ่มของคำสั่ง ซึ่งอาจจะแตกต่างจากคำสั่งในกรณีอื่น ในภาษา PHP จะใช้ โครงสร้าง if หรือ if-else ในการจำแนกกรณีตามเงื่อนไข

```

<?
if ($x == 0)
    echo $x," is zero<BR>\n";
else if ($x > 0)

```

```

    echo $x, " is positive<BR>\n";
else
    echo $x, " is negative<BR>\n";
?>

```

จากตัวอย่าง ถ้า \$x มีค่าเป็นศูนย์ตามเงื่อนไข ก็จะทำคำสั่ง echo \$x, " is zero
\n"; ถ้าเงื่อนไขแรกเป็นเท็จ ก็จะเงื่อนไขที่สองว่า \$x มีค่ามากกว่าศูนย์หรือไม่ ถ้าใช่ ก็ทำคำสั่ง echo \$x, " is positive
\n"; ถ้าเงื่อนไขสองเป็นเท็จอีก ก็ให้ทำคำสั่งในกรณีสุดท้ายคือ \$x จะต้องมีค่าเป็นลบ ถ้าในแต่ละกรณีต้องมีการทำคำสั่งมากกว่าหนึ่ง คือ เป็นกลุ่มคำสั่ง จะต้องใช้ { } มากำหนดขอบเขต (scope) เช่น

```

<?
if($x == 0) {
    echo "$x is zero.<BR>\n";
}
else if ($x > 0) {
    echo "$x is positive.<BR>\n";
}
else {
    echo "$x is negative.<BR>\n";
}
?>

```

โปรดสังเกตว่า { } ไม่ต้องมีเครื่องหมาย ; ต่อท้าย ในภาษา PHP มีการกำหนด elseif (เงื่อนไข) ขึ้นมาใช้ ซึ่งไม่มีอะไรแตกต่างจาก else if (เงื่อนไข) โครงสร้างแบบ (เงื่อนไข) ? นิพจน์ นิพจน์ แบบที่ใช้กันในภาษาซีนั้น ก็ใช้ได้เช่นกัน ตัวอย่างเช่น

```

<?
$x= -0.1035;
echo (($x < 0) ? -$x $x),"<BR>\n";
?>

```

2.1.15 การใช้ break และ continue ภายในลูป

คำสั่ง break และ continue ภายในลูปอย่างที่ใช้กันในภาษาซี ก็นำมาใช้กับภาษา PHP ได้ ตัวอย่างเช่น

```
<?
unset($a);
$a[]=1; $a[]=2; $a[]=3; $a[]="red";$a[]="green"; $a[]="blue";
$a[]="none";
$i=0;
$found="not found";
for ($i=0; $i < count($a); $i++) {
    if ( is_long($a[$i]) ) { // skip all integer elements
        continue;
    }
    if ($a[$i] == "blue") {
        $found=$a[$i];
        break;
    }
}
echo $found."<BR>\n";
?>
```

คำสั่ง continue บังคับให้ไปเริ่มต้นทำขั้นตอนในการวนลูปครั้งต่อไป ส่วน break นั้นส่งผลให้หยุดการทำงานของลูป

2.1.16 การแบ่งสายงานโดยจำแนกตามเงื่อนไขแบบ switch-case

นอกเหนือจากการใช้ if-else ในการจำแนกกรณีตามเงื่อนไขแล้ว เรายังสามารถใช้โครงสร้างแบบ switch-case ได้ ตัวอย่างเช่น

```
switch ($day) {
case 1
    echo "Monday<BR>\n";
    break;
case 2
```

```

    echo "Tuesday<BR>\n";
    break;
case 3
    echo "Wednesday<BR>\n";
    break;
case 4
    echo "Thursday<BR>\n";
    break;
case 5
    echo "Friday<BR>\n";
    break;
case 6
    echo "Saturday<BR>\n";
    break;
case 7
    echo "Sunday<BR>\n";
    break;
default
    echo "error<BR>\n";
}

```

ถ้าตัวแปร \$day มีค่าที่อยู่ระหว่าง 1 ถึง 7 ก็จะพิมพ์ชื่อวันเป็นภาษาอังกฤษ ถ้าตัวแปรมีค่านอกเหนือจากนั้น ซึ่งในกรณีจะเป็น default ในโครงสร้างแบบ switch-case ก็จะพิมพ์คำว่า error เพื่อให้ผู้ใช้ทราบ โปรดสังเกตว่า ในแต่ละกรณี จะต้องจบด้วยคำสั่ง break; ยกเว้นแต่ของ default ซึ่งจะมีหรือไม่ก็ได้ ถ้าเราไม่ได้ใส่คำสั่ง break; เอาไว้ โปรแกรมก็จะกระทำคำสั่งทุกคำสั่งในกรณีที่อยู่ถัดมา

การจำแนกกรณีไม่จำเป็นต้องอาศัยเฉพาะตัวแปรที่เก็บค่าจำนวนเต็มเท่านั้น ข้อมูลแบบอื่นก็ใช้ได้ เช่น ใช้ชื่อความเป็นตัวจำแนกกรณี เช่น

```

switch ($answer) {
case "yes"
    echo "The user said 'yes'.\n";
    break;

```

```

case "no"
    echo "The user said 'no'.\n";
    break;
default
    echo "The user said neither 'yes' nor 'no'.\n";
}

```

โปรดสังเกตว่า การจำแนกโดยใช้ข้อความนี้ จะดูความแตกต่างระหว่างตัวพิมพ์เล็กหรือใหญ่ด้วย ในบางครั้งเราอาจจะไม่จำเป็นต้องใส่ break; ก็ได้ ตัวอย่างเช่น

```

switch ($answer) {
case "yes"
case "no"
    echo "The user said '", $answer, "'.\n";
    break;
default
    echo "The user said neither 'yes' nor 'no'.\n";
}

```

2.1.17 การคำนวณเลขคณิตในระดับบิต

การคำนวณแบบบิตที่ใช้ในภาษาซี ก็ใช้ได้กับภาษา PHP ตามตารางข้างล่างนี้

$\$x \& \y	AND
$\$x \y	OR
$\$x \wedge \y	XOR
$\sim \$x$	NOT
$\$x \ll \y	SHIFT LEFT
$\$x \gg \y	SHIFT RIGHT

ตารางที่ 2-5 การคำนวณเลขคณิตในระดับบิต

2.2 ความรู้พื้นฐานในการใช้งานฐานข้อมูล

2.2.1 Structured Query Language; SQL

SQL จัดเป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งเป็นภาษาที่สามารถใช้งานได้บนคอมพิวเตอร์หลายระดับด้วยกัน ไม่ว่าจะเป็นระดับเมนเฟรมคอมพิวเตอร์จนถึงไมโครคอมพิวเตอร์ ชุดคำสั่ง หรือ ภาษา SQL นั้นถูกพัฒนาจากแนวความคิดทางคณิตศาสตร์ คือ Relational Algebra และ Relation Calculus ตามแนวคิดของเทคโนโลยีฐานข้อมูลเชิงสัมพันธ์ ที่ E.F. Codd เป็นผู้คิดค้นขึ้นเมื่อปี ค.ศ. 1970 และต่อมาบริษัท IBM จึงเริ่มทำการวิจัยพัฒนาเมื่อ ปี ค.ศ. 1974 โดยใช้ชื่อว่า “SEQUEL” (Structured English Query Language) จากนั้นจึงมีการปรับปรุงและเปลี่ยนชื่อมาเป็น SQL หลังจากปี ค.ศ. 1970 เป็นต้นมา ระบบฐานข้อมูล ORACLE ซึ่งถูกพัฒนาโดยบริษัท ORACLE Corporation และถือเป็นก้าวแรกในเชิงพาณิชย์สำหรับการพัฒนาระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS) ที่ตั้งอยู่บนพื้นฐานของ SQL และต่อมาก็มีผลิตภัณฑ์อื่นๆ พัฒนามา เช่น INGRESS เมื่อมีผลิตภัณฑ์จากผู้ผลิตต่างๆ มากขึ้น จึงทำให้เกิด SQL หลากๆรูปแบบจากผลิตภัณฑ์ต่างๆ ดังนั้นในราวปี ค.ศ. 1982 ทาง ANSI จึงได้คิดค้นและร่างมาตรฐานของชุดคำสั่ง SQL เพื่อให้ผู้ผลิตรายต่างๆ สร้างชุดคำสั่งดังกล่าวให้อยู่ภายใต้มาตรฐานเดียวกัน แต่ในปัจจุบันแต่ละผลิตภัณฑ์ต่างก็มีการเพิ่มคุณสมบัติพิเศษเพิ่มเติมเพื่อให้มีประสิทธิภาพยิ่งขึ้นและถือเป็นจุดขายของผลิตภัณฑ์ แต่ทั้งนี้ โดยหลักการแล้วชุดคำสั่งดังกล่าวยังคงตั้งอยู่บนพื้นฐานที่ทาง ANSI บัญญัติไว้ โดยในปัจจุบันมีระบบการจัดการฐานข้อมูลต่างๆ อาทิเช่น ORACLE , DB2 , SYBASE , Informix , MS-SQL , MS-Access ตลอดจน MS-FoxPro เป็นต้น ซึ่งการใช้งานภาษา SQL ในปัจจุบันมี 2 ลักษณะ คือ แบบโต้ตอบ (Interactive SQL) และ แบบฝังตัวในโปรแกรม (Embedded SQL)

2.2.1.1 วัตถุประสงค์ของ SQL

- สร้างฐานข้อมูลและโครงสร้างรีเลชัน
- สนับสนุนงานด้านการจัดการฐานข้อมูลพื้นฐาน เช่น การเพิ่ม การปรับปรุงการลบข้อมูลจากรีเลชัน
- สนับสนุนการค้นหา สืบถาม หรือคิวรีข้อมูลและการแปลงข้อมูลให้อยู่ในรูปแบบสารสนเทศ

2.2.1.2 ประเภทของคำสั่งภาษา SQL

- ภาษานิยามข้อมูล (Data Definition Language DDL) เป็นกลุ่มคำสั่งที่ใช้ในการสร้างฐานข้อมูล การกำหนดโครงสร้างข้อมูลว่ามีคอลัมน์หรือแอตทริบิวต์ใด ชนิดข้อมูลเป็นประเภทใด รวมทั้งการจัดการด้านการเพิ่ม แก้ไข ลบ แอตทริบิวต์ต่างๆในรีเลชัน และการสร้างดัชนี

- ภาษากำหนดข้อมูล (Data Manipulation Language DML) เป็นกลุ่มคำสั่งที่ถือเป็นแกนสำคัญของภาษา SQL โดยกลุ่มคำสั่งเหล่านี้จะใช้ในการ Update เพิ่ม ปรับปรุงและการ Query ข้อมูลในฐานข้อมูล ซึ่งอาจเป็นชุดคำสั่งในลักษณะ Interactive SQL หรือ Embedded SQL ก็ได้

- ภาษาควบคุมข้อมูล (Data Control Language DCL) ซึ่งเป็นกลุ่มคำสั่งที่จะช่วยให้ผู้บริหารฐานข้อมูล (DBA) สามารถควบคุมฐานข้อมูลเพื่อกำหนดสิทธิการอนุญาต (Grant) หรือการยกเลิกการเข้าใช้ (Revoke) ฐานข้อมูล ซึ่งเป็นกระบวนการป้องกันความปลอดภัยในฐานข้อมูล รวมทั้งการจัดการทรานแซกชัน (Transaction Management) แต่ละ DBMS จะมีการกำหนดชนิดข้อมูลซึ่งประกอบไปด้วยตัวแปรต่างๆ เช่น numeric , string , date , time เป็นต้น ดังตารางที่ 2-6 ดังนี้

- ภาษานิยามข้อมูล (Data Definition Language DDL)

CREATE คำสั่งที่ใช้ในการสร้าง เช่น CREATE TABLE สร้างตาราง

CREATE VIEW สร้างวิว

DROP คำสั่งที่ใช้ในการลบ เช่น DROP TABLE ลบตาราง ,

DROP VIEW ลบวิว

ALTER คำสั่งที่ใช้ในการเปลี่ยนแปลงโครงสร้างตาราง

- ภาษากำหนดข้อมูล (Data Manipulation Language DML)

SELECT คำสั่งที่ใช้ในการคิวรีข้อมูลในฐานข้อมูล

INSERT คำสั่งที่ใช้เพิ่มข้อมูลในตาราง

UPDATE คำสั่งที่ใช้ปรับปรุงข้อมูล (Update) ในตาราง

DELETE คำสั่งที่ใช้ลบข้อมูลออกจากตาราง

2.2.1.3 Select Statement

คำสั่ง Select เป็นคำสั่งที่ใช้เรียกดูข้อมูลจากตารางข้อมูลตามความต้องการของผู้ใช้ โดยสามารถใช้ร่วมกับ Function ทางคณิตศาสตร์ หรือเงื่อนไขประกอบต่างๆ ใน SQL Command ได้ อีกทั้งยังสามารถเรียกดูข้อมูลจากหลายๆตารางพร้อมกัน ซึ่งประโยค SELECT จะมีรูปแบบการใช้งานดังนี้

```

SELECT [DISTINCT|ALL]{*|column_expression[AS new_name][,...]}
FROM table_name [alias] [... ]
[WHERE condition ]
[GROUP BY column_list ] [HAVING condition ]
[ORDER BY column_list ];

```

Column_expression เป็นการแทนค่าชื่อคอลัมน์ หรือ expression , table_name คือชื่อของตารางที่ต้องการเข้าถึงเพื่อวิวข้อมูล และ alias คือชื่อของตารางที่ใช้แทน table_name โดยลำดับของการประมวลผล ชุดคำสั่ง SELECT มีดังนี้

- ▶ FROM กำหนดตารางที่ต้องการใช้งาน
- ▶ WHERE สร้างเงื่อนไขเพื่อการกรอง (filter) แถวที่ต้องการ
- ▶ GROUP BY จัดกลุ่ม แถว ที่มีค่าคอลัมน์เดียวกัน
- ▶ HAVING กรองกลุ่มเนื้อหาในบางเงื่อนไขจาก GROUP BY
- ▶ ORDER BY กำหนดให้เรียงผลลัพธ์ โดยที่ ASC คือการเรียงลำดับจากน้อย

ไปมาก (Default) และ DESC คือการเรียงลำดับจากมากไปน้อยใน SQL สามารถใช้ เครื่องหมายเปรียบเทียบ (Comparison Operators) ต่างๆ ได้ดังนี้

- ▶ BETWEEN / NOT BETWEEN ใช้แสดงข้อมูลเป็นช่วงๆ หรือระหว่างข้อมูล A ถึง B
- ▶ IN / NOT IN ใช้แสดงตำแหน่งข้อมูลหรือกำหนดตำแหน่ง
- ▶ LIKE / NOT LIKE เป็น โอเปอเรเตอร์ที่ใช้ในการค้นหาข้อมูลด้วยตัวอักษรด้วย

เครื่องหมาย Wildcard เช่น % และ _ คล้ายๆ การใช้ * ใน DOS

- ▶ IS NULL / IS NOT NULL ใช้ค้นหาข้อมูลที่บางคอลัมน์ของตาราง มี/ไม่มี ข้อมูล

มี ข้อมูล

- การใช้ฟังก์ชันต่างๆ ใน SQL
- COUNT เป็นฟังก์ชันใช้ในการนับจำนวน
- SUM เป็นฟังก์ชันหาผลรวม
- AVG เป็นฟังก์ชันหาค่าเฉลี่ย
- MIN เป็นฟังก์ชันหาค่าต่ำสุด
- MAX เป็นฟังก์ชันหาค่าสูงสุด และยังมี การใช้ SQL Command สำหรับการจัดการข้อมูลใน

ลักษณะอื่นๆ อีกมากมาย เช่น NESTED QUERY การ JOIN, การ UNION, INTERSECT,

EXCEPT ตลอดจนการ INSERT DELETE และ UPDATE ข้อมูลในตาราง ซึ่งสามารถคิดตามได้จากหนังสือหรือ คู่มือภาษา SQL ทั่วๆ ไป

2.2.1.4 ภาษาควบคุมข้อมูล (Data Control Language; DCL)

การควบคุมข้อมูลถือเป็นการควบคุมความปลอดภัยในการใช้งานข้อมูลจากฐานข้อมูล กล่าวคือ ฐานข้อมูลมีวัตถุประสงค์ที่ต้องการให้ผู้ใช้งานหลายๆคนสามารถเข้าถึงข้อมูลส่วนกลางได้ ซึ่งหากปราศจาก ควบคุมความปลอดภัยในฐานข้อมูลแล้ว ย่อมก่อให้เกิดปัญหาต่างๆตามมา ดังนั้นผู้บริหารฐานข้อมูล จึง จำเป็นต้องทำการกำหนดสิทธิการใช้งานที่ควรจะเป็นให้แก่ User ในแต่ละระดับว่าจะสามารถเข้าถึง ข้อมูลได้มากน้อยอย่างไร

ความปลอดภัยในฐานข้อมูลถือเป็นสิ่งสำคัญที่ไม่ควรมองข้าม เพราะข้อมูลอาจเกิดความเสียหายได้ถ้าไม่มีการควบคุมความปลอดภัย ดังนั้นเมื่อมีการควบคุมความปลอดภัย นั้นหมายถึงการป้องกันมิให้ผู้ที่ไม่มียุติสิทธิในการเข้าถึงข้อมูลจัดการกับข้อมูลในฐานข้อมูลได้ เช่น การกำหนดให้ผู้ใช้งานในแผนก แคนเซียร์สามารถเข้าถึงข้อมูลเพียงแค่เรียกดูราคาสินค้าเท่านั้น ไม่สามารถปรับปรุงข้อมูลราคาสินค้า เหล่านั้นได้ แต่พนักงานการตลาดที่ดูแลเรื่องราคาสินค้าและ โปรโมชัน มีสิทธิเข้าถึงข้อมูลราคาและ เปลี่ยนแปลงราคาสินค้าได้ เป็นต้น การจัดการดังกล่าวจะทำให้ฐานข้อมูลในองค์กรมีความปลอดภัยต่อการ บุกรุกไม่ว่าจะเป็นการจงใจหรือไม่ตั้งใจก็ตาม และช่วยลดความเสี่ยงต่อการสูญหายของข้อมูล ด้วย นอกจากความปลอดภัยในการกำหนดสิทธิการใช้งานของผู้ใช้แล้ว ยังรวมไปถึงการควบคุมภาวะการ เข้าพร้อมกัน (Concurrency Control) ด้วยซึ่งหากปราศจากการควบคุมอาจก่อให้เกิดความผิดพลาดใน การ อัปเดตข้อมูลและส่งผลต่อความเสียหายในข้อมูลได้ เช่น การบันทึกข้อมูลบัญชีของธนาคารแห่งหนึ่ง ซึ่งวันวันหนึ่งมีลูกค้ามากมายหลายคนเข้าใช้บริการตู้ ATM จากหลายๆ สถานที่ ถ้าหากไม่มีการควบคุม ภาวะการเข้าถึงข้อมูลพร้อมกัน ลูกค้า ตั้งแต่ 2 คน ขึ้นไปใช้ ATM พร้อมกัน ข้อมูลทางบัญชีอาจรับข้อมูลของ ลูกค้าคนหนึ่งแต่อีกคนหนึ่งอาจไม่รับ ทำให้ข้อมูลเสียหายไม่ตรงกับความเป็นจริงได้ จึงต้องมีการควบคุม ภาวะการเข้าพร้อมกันด้วยการให้ข้อมูลของฝ่ายใดฝ่ายหนึ่งเกิดการรอ (wait) จนกว่ารายการของลูกค้าคนที่ เข้าถึงข้อมูลได้ก่อนจะแล้วเสร็จ แล้วจึงค่อยดำเนินการต่อไปคำสั่ง GRANT เป็นคำสั่งที่ใช้ในการกำหนด สิทธิในการเข้าถึงข้อมูลว่าจะให้ผู้ใ้รายใดเข้าถึงฐานข้อมูล สามารถจัดการข้อมูลในตาราง หรือวิ ใดบ้าง เช่น กำหนดให้ผู้ใ้บางคนสามารถ SELECT ข้อมูลได้อย่างเดียว ในขณะที่บางคนสามารถ ปรับปรุงเปลี่ยนแปลงแก้ไข INSERTDELETEGRANT {privilege_list|ALLPRIVILEGE}

ONObject_name

TO {authorization_id_list|PUBLIC}

[WITHGRANTOPTION]

คำสั่ง REVOKE เมื่อสามารถกำหนดสิทธิการใช้งานแก่ผู้ใช้ได้ ก็สามารถยกเลิกสิทธิบางสิทธิ์ได้

REVOKE[GRANTOPTIONFOR]{privilege_list|ALLPRIVILEGES}

ONObject_name

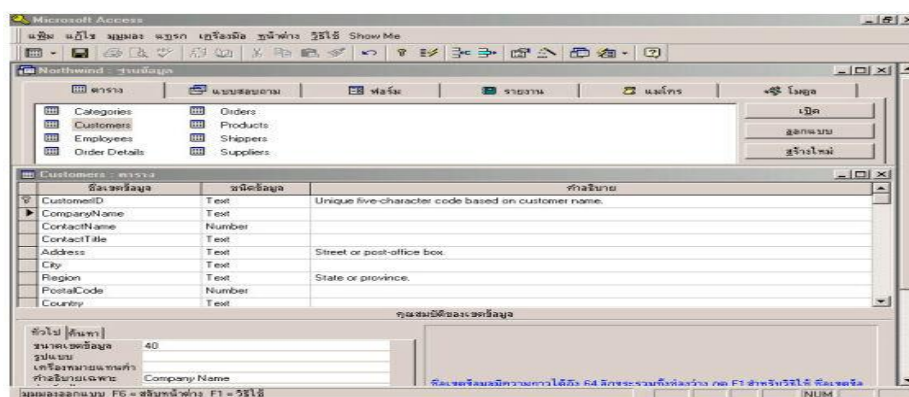
FROM{authorization_id_list|PUBLIC}[RESTRICT|CASCADE];

2.2.1.5 ตัวอย่างโปรแกรมสำเร็จรูปในการจัดการฐานข้อมูล (Microsoft Access , Microsoft SQL Server และ MySQL)

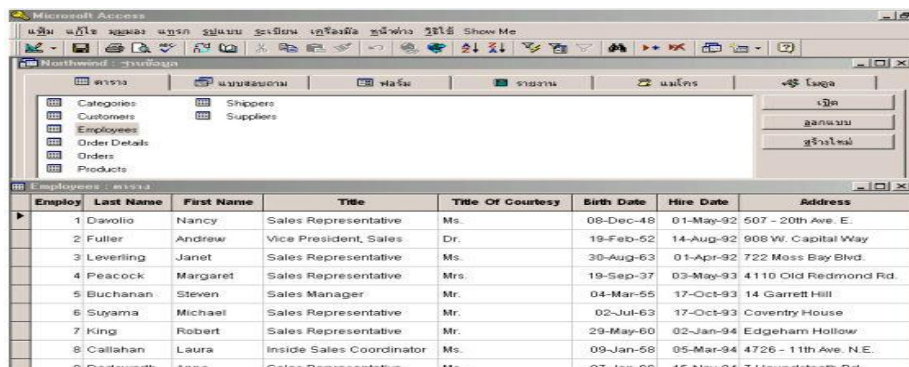
- โปรแกรม Microsoft Access

เป็นโปรแกรมการจัดการฐานข้อมูลในช่วงต้นๆ ของ กลุ่มบริษัทไมโครซอฟท์ ซึ่งออกแบบมาให้ใช้งานจัดการฐานข้อมูลได้ง่าย ไม่ซับซ้อน ซึ่งประกอบไปด้วยองค์ประกอบหลักของแถบเครื่องมือ (tool bar) ตามโครงสร้างของโปรแกรมไมโครซอฟท์ออฟฟิศทั่วไป ส่วนของการจัดการฐานข้อมูลประกอบไปด้วย ตาราง แบบสอบถาม φόρม รายงาน แมโครและโมดูล ดังภาพที่ 2-1

ในภาพที่ 2-1 แสดงถึงมุมมองหนึ่งใน Microsoft Access ที่แสดงโครงสร้างของข้อมูล Customers ในระบบการจัดการฐานข้อมูลของบริษัท Northwind ว่าประกอบไปด้วยกลุ่มของฟิลด์ข้อมูล ไตบ้างและชนิดข้อมูล (Data types) เป็นอย่างไร พร้อมกับคำอธิบายข้อมูล (Data description)



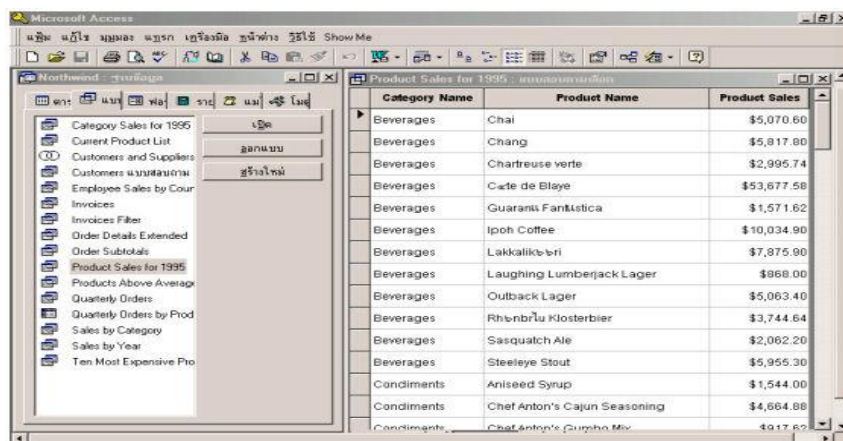
รูปที่ 2-1 ภาพแสดงมุมมอง(View) โครงสร้างฐานข้อมูล



Query คือวิธี เป็นองค์ประกอบที่สำคัญมากของระบบการจัดการฐานข้อมูล เพราะหมายถึงการ ระบุที่ 2-2 ภาพแสดงมุมมอง (TableView) โครงสร้างฐานข้อมูล สอบถามหรือเรียกดูข้อมูลตามความต้องการของผู้ใช้ (Customized data inquiry) ซึ่งผู้ใช้สามารถตั้งคำถาม ได้ตามต้องการเช่น ต้องการทราบยอดขายของสินค้าในปี 1995 ดังตัวอย่างในภาพที่ -----

- Microsoft Access ก็จะแสดงข้อมูลออกมาให้เห็น

ตารางขวามือ ลักษณะการเรียกดูข้อมูลตามวัตถุประสงค์ใช้งานแบบนี้เรียกว่า “แบบสอบถาม” หรือ “QUERY

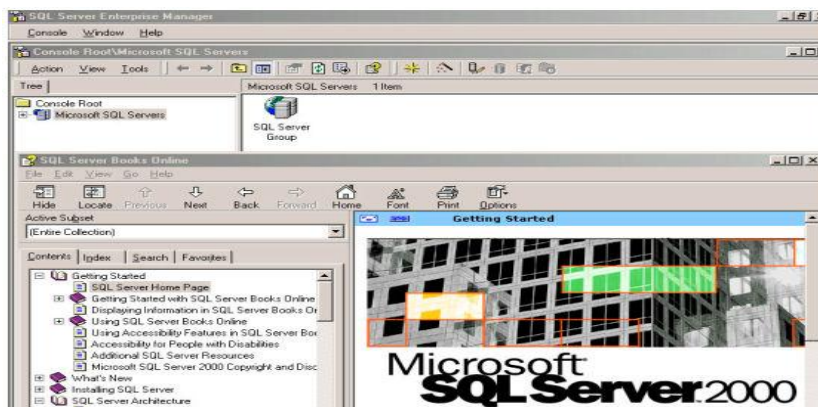


รูปที่ 2-3 ภาพแสดงวิธีการสอบถาม (Query) ข้อมูลยอดขายสินค้าของบริษัทในปี1995

การจัดการฐานข้อมูลด้วยไมโครซอฟท์เอกเซล จึงเป็นโปรแกรมที่ผู้เริ่มต้นนิยมใช้ โดยมากใช้กับการจัดการฐานข้อมูล Stand Alone ที่ไม่ได้เชื่อมต่อเครือข่าย แต่ก็สามารถนำไปพัฒนาเชื่อมต่อกับระบบเครือข่าย Client / Server ได้ ซึ่งผู้ที่สนใจสามารถศึกษาต่อเพิ่มเติมได้ จากคู่มือการใช้งาน Microsoft Access ทั่วไป

- Microsoft SQL Server

เป็นโปรแกรมการจัดการฐานข้อมูลในตระกูล Microsoft อีกเช่นกัน แต่เน้นใช้งานในระบบเครือข่าย Client / Server โดยเฉพาะ ดังนั้นการใช้งานฐานข้อมูล SQL Server ต้องมีการสร้างระบบเครือข่าย และให้ Sever เป็นฐานข้อมูล ส่วนที่ Clients จะมีเพียงแต่ตัวโปรแกรมเท่านั้น เป็นการประหยัดทรัพยากรหน่วยความจำในเครื่อง Clients และต้องการให้มีศูนย์กลางข้อมูลอยู่เพียงแหล่งเดียวเพื่อการจัดการและควบคุมการเข้าใช้ได้อย่างมีประสิทธิภาพ โปรแกรม MS-SQL Server จึงถูกพัฒนาขึ้นภายใต้การใช้ภาษา SQL ที่เป็นสากล ที่สามารถใช้งานร่วมกับฐานข้อมูลอื่นที่มีโครงสร้างภาษาเดียวกัน เช่น ฐานข้อมูลที่สร้างขึ้นจาก Microsoft Access ก็สามารถใช้งานร่วมกันกับ SQL Server ได้เลย คำสั่งต่างๆ ที่ใช้ใน SQL Server กับ Access จึงเป็นคำสั่งเดียวกัน และออกแบบให้งานได้ง่ายด้วยหน้าต่าง Windows ที่ผู้ใช้ส่วนใหญ่คุ้นเคย การใช้งาน SQL Server จะต้องอยู่บนสถาปัตยกรรมโครงสร้างฐานข้อมูลแบบ Client / Server และมีการเชื่อมต่อ (connect) กับฐานข้อมูลทุกครั้งที่มีการใช้งาน

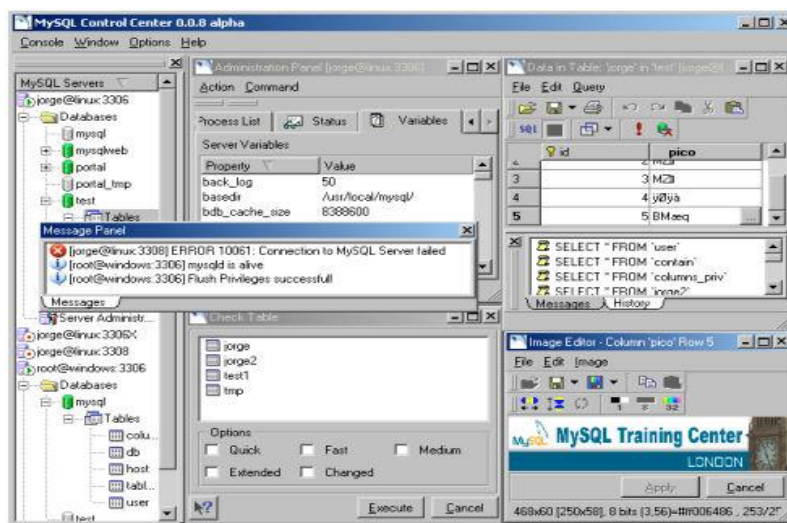


รูปที่ 2-4 ภาพแสดงหน้าต่างโปรแกรมการจัดการฐานข้อมูล Microsoft SQL Server 2000

- MySQL

โปรแกรม MySQL เป็นโปรแกรมการจัดการฐานข้อมูลอีกโปรแกรมหนึ่งที่ใช้ภาษา SQL เป็นที่นิยมและรู้จักกันในวงการ DBMS อย่างกว้างขวาง ซึ่งนักบริหารฐานข้อมูลนิยมใช้กันมากในปัจจุบัน เพราะใช้งานได้ดีในระบบปฏิบัติการทุกชนิด โดยเฉพาะอย่างยิ่งในระบบปฏิบัติการลินุกซ์ ซึ่งรองรับ MySQL ได้เป็นอย่างดี MySQL เป็นโปรแกรมบริหารจัดการด้านฐานข้อมูล (Database Management) MySQL ทำงานในลักษณะฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System RDBMS) คำว่า ฐานข้อมูลเชิงสัมพันธ์ ก็คือ ฐานข้อมูลที่แยกข้อมูลไปเก็บเอาไว้ในหน่วยย่อย ซึ่งเรียกว่าตารางข้อมูล (Table) แทนที่จะเก็บข้อมูลทั้งหมดรวมกันเอาไว้แห่งเดียว และแต่ละหน่วยย่อยที่เก็บข้อมูลต่างมีความสัมพันธ์เชื่อมโยงกันอยู่ ยกตัวอย่างเช่น ข้อมูลสินค้า ซึ่งสามารถจัดเก็บแยกกันได้ แล้วอาศัย

รหัสของสินค้าในการเรียกค้นข้อมูลที่จัดเก็บแยกกันเอาไว้ การที่เราจะเข้าไปจัดการกับข้อมูล ต้องอาศัยภาษาคอมพิวเตอร์ที่เรียกว่า SQL ซึ่งย่อมาจาก Structure Query Language ชื่อ MySQL ก็คือให้ทราบว่าเกี่ยวกับภาษา SQL อยู่แล้ว ดังนั้น MySQL จึงทำงานตามคำสั่งภาษา SQL ได้ อันเป็นไปตามมาตรฐานของโปรแกรมทางด้านฐานข้อมูลที่จะต้องมีความสามารถรองรับคำสั่งที่เป็นภาษา SQL MySQL เป็นโปรแกรมที่เปิดเผยแพร่รายละเอียดซอร์สโค้ดต่อบุคคลทั่วไป (Open Source Software) ซึ่งหมายความว่า ใครก็ตามที่มีความรู้ทางด้านภาษาคอมพิวเตอร์อย่างดีก็สามารถนำเอาซอร์สโค้ดของโปรแกรม MySQL ซึ่งเขียนด้วยภาษา C ไปดัดแปลง ปรับปรุง แก้ไข ให้ตรงกับที่ต้องการได้ทันทีโดยไม่ผิดกฎหมาย ใช้ได้กับระบบปฏิบัติการหลากหลายระบบ เช่น Linux, Solaris, Mac OS X Server, OS/2 Warp, Sun OS, Windows 9x/NT/2000/XP และระบบตระกูล Unix อีกมากมาย ซึ่งทุกระบบปฏิบัติการสามารถเชื่อมโยงและแลกเปลี่ยนข้อมูลกันได้เป็นอย่างดี



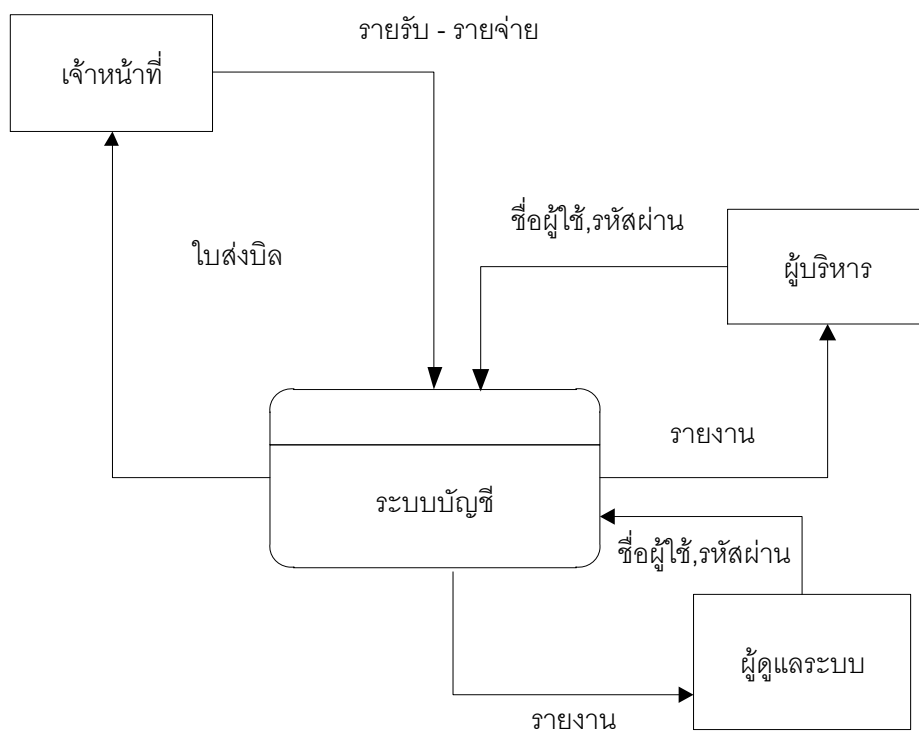
รูปที่ 2-5 ตัวอย่างหน้าต่างของโปรแกรม MySQL (www.MySQL.com)

ยังมี DBMS อีกมากมายที่กล่าวไม่หมดในที่นี้ ในการประยุกต์ใช้งานระบบการจัดการฐานข้อมูลเหล่านี้กับงานด้านการตลาดสามารถประยุกต์ใช้ได้ทันที แต่ปัจจัยสำคัญอยู่ที่การออกแบบฐานข้อมูล ซึ่งผู้ใช้งานต้องเข้าใจถึงโครงสร้างเชิงตรรกะของข้อมูลด้วย จึงจะนำไปประยุกต์ใช้งานได้อย่างเหมาะสม

บทที่ 3

การวิเคราะห์และออกแบบระบบ

3.1 ความต้องการของระบบ



ภาพที่ 3-1 คอนเท็กซ์ไดอะแกรมระบบบัญชี

ATSN : Assumption Technical School Nakhonphanom :: - Windows Internet Explorer

http://61.19.117.243/index.php?mod=expenses&file=index&op=show_month&year=2554

File Edit View Favorites Tools Help

Google

ATSN : Assumption Techn... 61.19.117.243 / localhost / d... Asterisk Configuration โปรแกรมบัญชี - ค้นหาคีย์ Google

หน้าหลัก E-Learning ระบบอินเทอร์เน็ตโรงเรียน E-mail ข่าวเพลง เว็บวัดเดิม

โรงเรียนอัสสัมชัญเทคนิคนครพนม

โรงเรียนการกุศล ในสังกัด มูลนิธิคณะเซนต์คาเบรียลแห่งประเทศไทย

หน้าหลัก E-Learning ระบบอินเทอร์เน็ตโรงเรียน E-mail ข่าวเพลง เว็บวัดเดิม

User Navigator

- บัญชีแยกตามวัน
- บัญชีแยกตามเดือน
- บัญชีแยกตามปีการศึกษา
- บัญชีแยกตามฝ่าย

นางสอน วิภา

+++ ทว่าพลัก +++
Report
เข้าห้องประชุม
ภาพจากกล้อง
Upload ClipVDO
สรุปบัญชี
ระบบบริหารการ
ประชุม
สรุปบัญชี (สรุป)
Upload Download
ตรวจสอบข้อมูล
แผนงาน โครงการ กิ
รรม
กรงกัญญ์การไร้พียง
ส่งโครงการ นร.
ส่งงานวิจัย
ส่งโครงการสอน

ปีการศึกษา 2554

ลำดับ	เดือน	รายงานส่งคณะ	งบประมาณที่ใช้ไป
1	พฤษภาคม พ.ศ.2554	รายงานส่งคณะ	1,007,004.22 -
2	มิถุนายน พ.ศ.2554	รายงานส่งคณะ	5,650.00 -
3	กรกฎาคม พ.ศ.2554	รายงานส่งคณะ	0.00 -
4	สิงหาคม พ.ศ.2554	รายงานส่งคณะ	0.00 -
5	กันยายน พ.ศ.2554	รายงานส่งคณะ	0.00 -
6	ตุลาคม พ.ศ.2554	รายงานส่งคณะ	0.00 -
7	พฤศจิกายน พ.ศ.2554	รายงานส่งคณะ	0.00 -
8	ธันวาคม พ.ศ.2554	รายงานส่งคณะ	0.00 -
9	มกราคม พ.ศ.2555	รายงานส่งคณะ	0.00 -
10	กุมภาพันธ์ พ.ศ.2555	รายงานส่งคณะ	0.00 -

start ATSN : Assumpti... Documents Media Get EdkPlus - [C:\Ap... 2 SSH Secure S... 7 Microsoft Word Macromedia Drea... EN 12:47

ATSN : Assumption Technical School Nakhonphanom :: - Windows Internet Explorer

http://61.19.117.243/index.php?mod=expenses&file=index&op=report&year=2554

File Edit View Favorites Tools Help

Google

ATSN : Assumption Techn... 61.19.117.243 / localhost / d... Asterisk Configuration โปรแกรมบัญชี - ค้นหาคีย์ Google

หน้าหลัก E-Learning ระบบอินเทอร์เน็ตโรงเรียน E-mail ข่าวเพลง เว็บวัดเดิม

Assumption Technical School Nakhon Phanom A Charitable school of St.Gabriel's Foundation Thailand.

โรงเรียนอัสสัมชัญเทคนิคนครพนม

โรงเรียนการกุศล ในสังกัด มูลนิธิคณะเซนต์คาเบรียลแห่งประเทศไทย

หน้าหลัก E-Learning ระบบอินเทอร์เน็ตโรงเรียน E-mail ข่าวเพลง เว็บวัดเดิม

User Navigator

- บัญชีแยกตามวัน
- บัญชีแยกตามเดือน
- บัญชีแยกตามปีการศึกษา
- บัญชีแยกตามฝ่าย

นางสอน วิภา

+++ ทว่าพลัก +++
Report
เข้าห้องประชุม
ภาพจากกล้อง
Upload ClipVDO
สรุปบัญชี
ระบบบริหารการ
ประชุม
สรุปบัญชี (สรุป)
Upload Download
ตรวจสอบข้อมูล
แผนงาน โครงการ กิ
รรม

ทำรายงานประจำปี 2554

MS Word PDF

ทำรายงานประจำปี

รหัสบัญชี	รายการ	รายจ่าย	รับจริง	คงเหลือ
510101	เงินเดือนครู 18 คน (12x261,117)	3,133,404.00 -	235,317.00 -	2,898,087.00 -
510201	เงินเดือนพนักงาน 12x95,650	1,151,400.00 -	89,650.00 -	1,061,550.00 -
510301	เงินเดือนครู 18 คน	261,117.00 -	0.00 -	261,117.00 -
510302	เงินเดือนพนักงาน 12 คน	95,650.00 -	0.00 -	95,650.00 -

start ATSN : Assumpti... Documents Media Get EdkPlus - [C:\Ap... 2 SSH Secure S... 7 Microsoft Word Macromedia Drea... EN 12:48

บทที่ 4

บทสรุป

4.1 สรุปผลการดำเนินงาน

การพัฒนาโปรแกรมบัญชีนี้ ได้พัฒนาตามแผนการดำเนินการตั้งแต่ การกำหนดปัญหาจนกระทั่งถึงการปรับปรุงแก้ไขระบบ ซึ่งทำให้ได้รับความรู้และประสบการณ์มากมายทั้ง ในด้านการวิเคราะห์ปัญหา การออกแบบ การพัฒนาระบบ การทดสอบและการปรับปรุงแก้ไขระบบ โดยใช้ โปรแกรม MySQL เป็นตัวจัดการฐานข้อมูล และใช้ภาษา PHP เป็นตัวพัฒนาโปรแกรม จนกระทั่งได้ โปรแกรมบัญชีออนไลน์ ที่มีความเหมาะสม

4.2 ปัญหาและอุปสรรค

การทำโครงการได้พบปัญหาและอุปสรรคต่าง ๆ โดยสรุป ดังนี้

4.2.1 มีความผิดพลาดในการใส่ตัวอักษร

4.2.2 ข้อจำกัดเรื่องการกำหนดชื่อตัวแปรระหว่างตัวภาษาอังกฤษพิมพ์เล็กกับพิมพ์ใหญ่ไม่

เหมือนกัน

4.2.3 ข้อจำกัดเรื่องฐานข้อมูล โปรแกรมจะทำงานได้กับฐานข้อมูลที่ได้สร้างไว้ และกำหนดให้ชื่อ ตารางที่ใช้เก็บข้อมูลมีชื่อตามชื่อแบบฟอร์มเอกสารที่ผู้จัดทำสร้าง

4.3 ข้อเสนอแนะ

4.3.1 ผู้ที่ต้องการพัฒนาและผู้ศึกษาให้ควรใช้ระบบปฏิบัติการขึ้นไป เพื่อเป็นเครื่องมือช่วยในการ จำลองเซิร์ฟเวอร์ พัฒนาแอปพลิเคชันด้วย PHP จึงสามารถทำได้ง่าย เพราะสามารถทำเว็บเพจและ ทดสอบการทำงานที่เครื่องจำลองได้เลย แลวจึงอัปโหลดไฟล์ต่าง ๆ ไปยังเครื่องเซิร์ฟเวอร์ โดยที่ผลการ ทำงานที่เกิดบนเครื่องจำลองจะมีความเหมือนกับที่ทำงานบน

เครื่องเซิร์ฟเวอร์มาก

4.3.2 โปรแกรมบัญชีเป็น Web Application ที่สร้างขึ้นเพื่อเชื่อมต่อกับระบบอินเทอร์เน็ตเพื่ออำนวยความสะดวกให้ผู้ใช้สามารถตรวจสอบค่าใช้จ่าย หรือประกอบการตัดสินใจ เนื้อหาที่จัดทำเน้นเรื่องที่น่าไป ใช้งานได้จริง หากผู้ใดที่สนใจที่จะพัฒนาควรเลือกเนื้อหาและระบบการจัดให้เหมาะสมกับความ ต้องการเพื่อให้เกิดประโยชน์และ การใช้งานสูงสุด

ภาคผนวก

ภาคผนวก ก

การติดตั้ง AppServ

การติดตั้ง AppServ (PHP แอปพลิเคชันเซิร์ฟเวอร์)

AppServ คือ ชุดติดตั้งโปรแกรม PHP แอปพลิเคชันเซิร์ฟเวอร์สำหรับติดตั้งบนระบบปฏิบัติการ

Windows

ในชุดติดตั้ง AppServ นี้ ประกอบด้วยโปรแกรมต่างๆ ดังต่อไปนี้

- Apache สำหรับทำหน้าที่เป็นเว็บเซิร์ฟเวอร์
- PHP สำหรับทำหน้าที่เป็นตัวแปลภาษา PHP
- MySQL สำหรับทำหน้าที่เป็นดาต้าเบสเซิร์ฟเวอร์
- phpMyAdmin สำหรับทำหน้าที่เป็นโปรแกรมบริหารจัดการฐานข้อมูลของ MySQL

ติดตั้งโปรแกรม AppServ (PHP แอปพลิเคชันเซิร์ฟเวอร์) ซึ่งสามารถดาวน์โหลดได้ที่
เว็บไซต์ <http://appserv.sourceforge.net>

ขั้นตอนการติดตั้งโปรแกรม AppServ

ดับเบิลคลิกไฟล์ \AppServ\2.4.1\appserv-win32-2.4.1.exe หลังจากนั้นคลิก Next

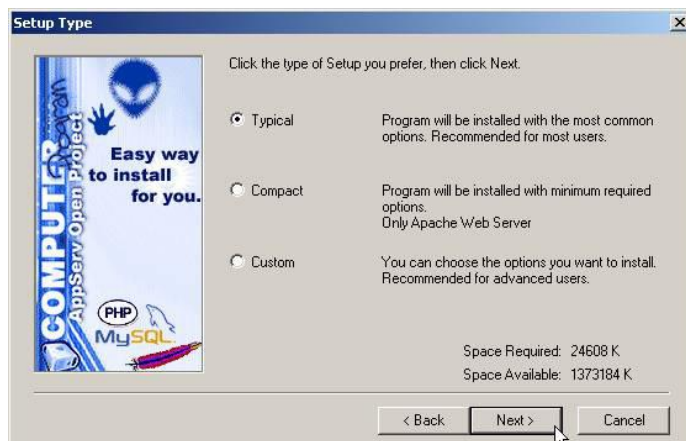


รูปที่ ก.1 ภาพแสดงเริ่มการติดตั้ง

กำหนดไดเรกทอรีที่จะติดตั้งและคลิก Next (ค่า default คือ C:\AppServ)



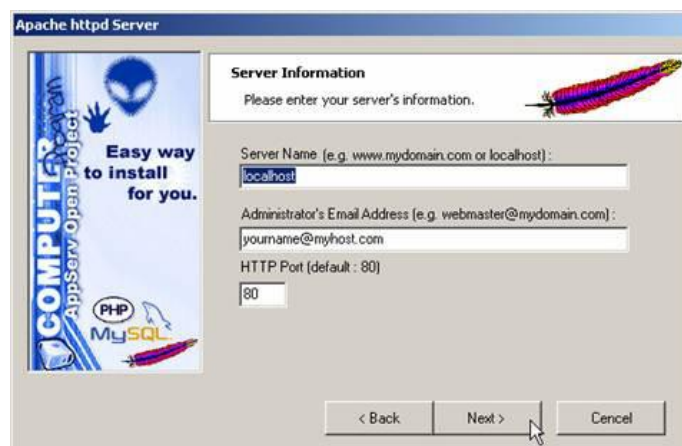
รูปที่ ก.2 ภาพเลือกไดเรกทอรีติดตั้ง
เลือกการติดตั้งแบบ Typical หลังจากนั้นคลิก Next



รูปที่ ก.3 ภาพแสดงรูปแบบการติดตั้ง
กำหนดชื่อเว็บเซิร์ฟเวอร์ หลังจากนั้นคลิก Next

Server Name: localhost

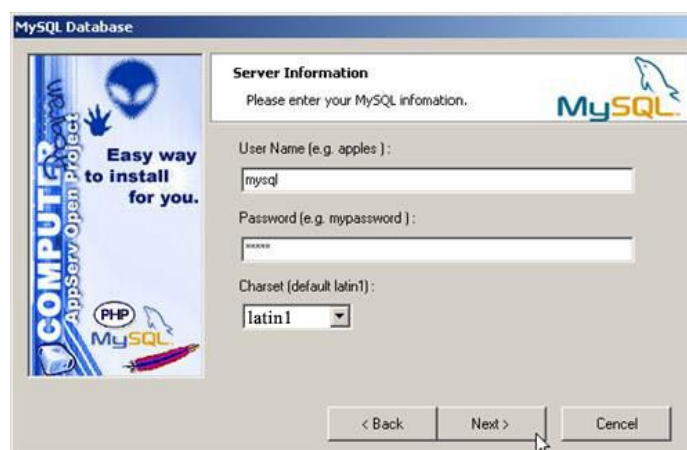
HTTP Port: 80



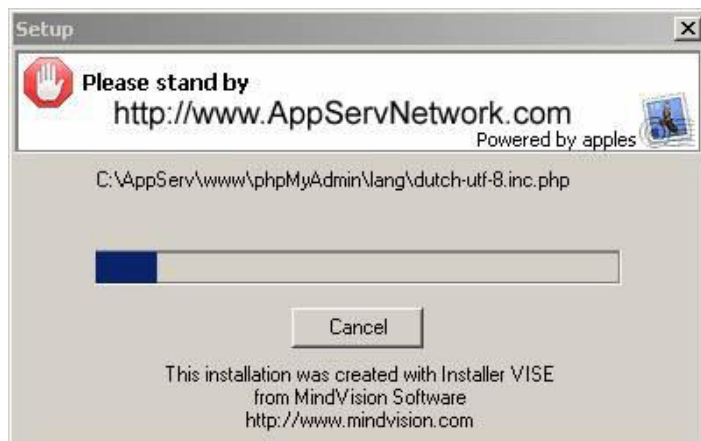
รูปที่ ก.4 ภาพแสดงการกำหนดรูปแบบ Server

กำหนด username, password และ Character set ของ MySQL หลังจากนั้นคลิก Next

User Name: mysql
Password: mysql
Charset: latin1

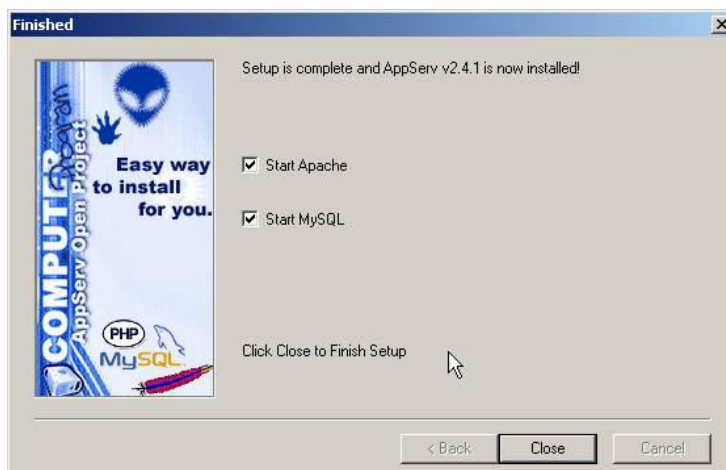


รูปที่ ก.5 ภาพแสดงการกำหนดUsername และ Password



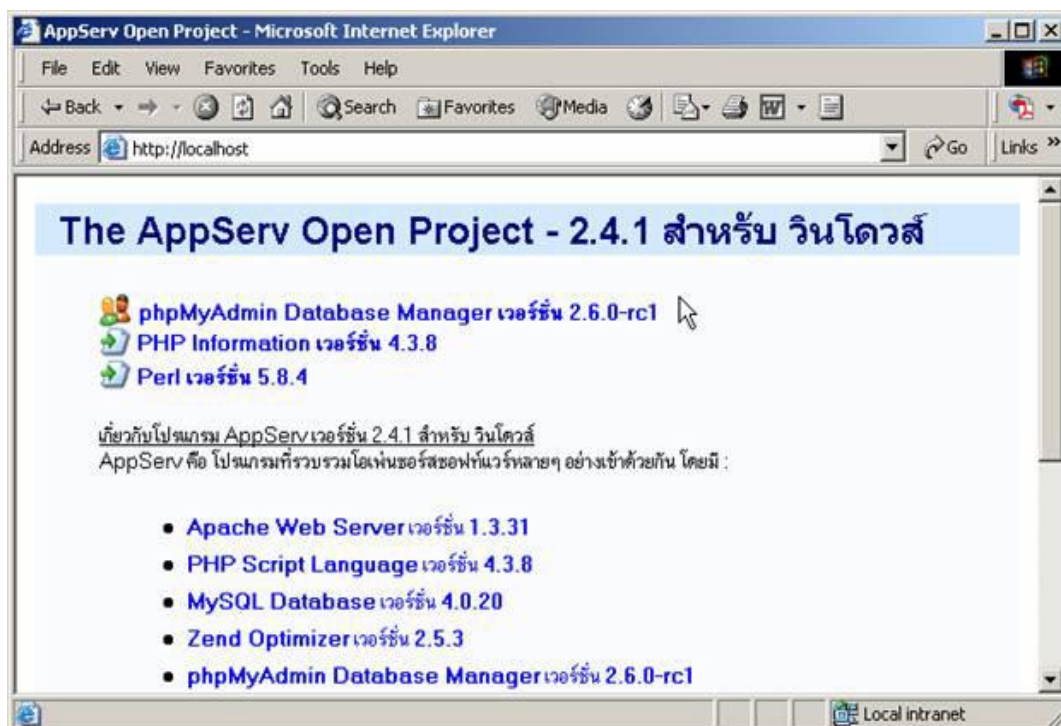
รูปที่ ก.6 ภาพแสดงการดำเนินการติดตั้ง
คลิกเพื่อเช็คเลือกให้สตาร์ท Apache และ MySQL เมื่อเปิดเครื่องใหม่ทุกครั้ง หลังจากนั้นคลิก

Close



รูปที่ ก.7 ภาพแสดงการ Start Apache และ MySQL

ทดสอบการติดตั้งโดยเปิดเว็บเบราว์เซอร์ (เช่น Internet Explorer) ไปที่ <http://localhost>



รูปที่ ก.8 ภาพแสดงภาพทดสอบการติดตั้ง

การเริ่มต้นใช้งาน (Start) PHP แอปพลิเคชันเซิร์ฟเวอร์

โดยปกติเมื่อเปิดเครื่องคอมพิวเตอร์ โปรแกรม Apache เว็บเซิร์ฟเวอร์ และ MySQL ดาต้าเบส เซิร์ฟเวอร์จะทำงานโดยอัตโนมัติ แต่หากโปรแกรมไม่ทำงาน เราสามารถสั่งให้เริ่มต้นทำงานได้โดยวิธีการดังนี้

ขั้นตอนการเริ่มต้นใช้งาน (Start):

รัน Apache เว็บเซิร์ฟเวอร์ โดยคลิกที่เมนู

Start -> Programs -> AppServ -> Apache Control Server -> Start

รัน MySQL ดาต้าเบสเซิร์ฟเวอร์ โดยคลิกที่เมนู

Start -> Programs -> AppServ -> WinMySQLAdmin

จะปรากฏไอคอน  ที่ System Tray (ด้านล่าง-ขวาของ Windows)

การหยุดใช้งาน (Stop) PHP แอปพลิเคชันเซิร์ฟเวอร์

หากต้องการหยุดการทำงานของโปรแกรม Apache เว็บเซิร์ฟเวอร์ และ MySQL ดาต้าเบส เซิร์ฟเวอร์ สามารถสั่งให้หยุดทำงานได้ โดยวิธีการดังนี้


ขั้นตอนการหยุดใช้งาน (Stop):

หยุดการทำงาน Apache เว็บเซิร์ฟเวอร์ โดยคลิกที่เมนู


Start -> Programs -> AppServ -> Apache Control Server -> Stop

หยุดการทำงาน MySQL ดาต้าเบสเซิร์ฟเวอร์ โดยคลิกที่เมนู

สำหรับเครื่องที่ใช้ Windows 9x:

- คลิกไอคอน  ที่ด้านล่าง-ขวาของ Windows (System Tray)
- เลือก **Win 9x > Shutdown the Server**

สำหรับเครื่องที่ใช้ Windows XP/2000:

- คลิกไอคอน  ที่ System Tray (ด้านล่าง-ขวาของ Windows)
- เลือก **Win NT > Stop the Service**
- เลือก **Win NT > ShutDown this Tool**

บรรณานุกรม

ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ. บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), 2540 หน้า 84-95.

กิตติ ภัคดีวัฒนะกุล. คัมภีร์ PHP พิมพ์ครั้งที่ 6 กรุงเทพฯ.

ดร.อำไพ พรประเสริฐ. การวิเคราะห์และออกแบบระบบ (System Analysis and Design).

โอภาส เอี่ยมสิริวงศ์ .การวิเคราะห์และออกแบบระบบ (System Analysis and Design). สุภชัย สมพานิช. Database Programming. พิมพ์โดย บริษัท ไอเดีย อินโฟ เซนเตอร์ จำกัด ,พิมพ์ครั้งที่ 1 . กรุงเทพฯ 10310.

<http://www.php.net>

<http://www.mysql.com>

<http://www.webthaid.com>