

1 Generate a new primary key value for a new database record

Program : genNextUserNo.php

```
<?php require_once('Connections/conn.php'); ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Algorithm to generate next userNo</title>
  </head>
  <body>
    <?php
      mysql_select_db($database_conn, $conn);
      $query_USERrs = "SELECT max(userNo) AS lastUserNo FROM user";
      $USERrs = mysql_query($query_USERrs, $conn) or die(mysql_error());
      $row_USERrs = mysql_fetch_assoc($USERrs);
      $lastUserNo = $row_USERrs['lastUserNo'];
      if ($lastUserNo == null)
        echo '$lastUserNo = null<br />'; // for debug
      else
        echo '$lastUserNo = '.$lastUserNo."<br />"; // for debug
      $nextUserNo = sprintf("U%04d", substr($lastUserNo,1)+1);
      echo '$nextUserNo = '.$nextUserNo."<br />";
      mysql_free_result($USERrs);
    ?>
  </body>
</html>
```

- If User table has only one record with **userNo = U0001**, the output from PHP :

```
$lastUserNo = U0001
$nextUserNo = U0002
```

- If User table has **no record**, the output from PHP :

```
$lastUserNo = null
$nextUserNo = U0001
```

Program : genNextUserNo_function.php

```
<?php require_once('Connections/conn.php'); ?>
<?php require_once('common.php'); ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Algorithm to generate next userNo</title>
  </head>
  <body>
    <?php
      echo getNextPK('U', 'userNo', 'user')."<br />"; // pass column name, table name
      echo getNextPK('A', 'adminNo', 'admin')."<br />"; // pass column name, table name
    ?>
  </body>
</html>
```

Program : common.php

```
<?php
function getNextPK($prefix, $column, $table) {
```

```

global $database_conn, $conn; // access global variables
mysql_select_db($database_conn, $conn);
$query_rs = "SELECT max($column) AS lastPK FROM $table";
$rs = mysql_query($query_rs, $conn) or die(mysql_error());
$row_rs = mysql_fetch_assoc($rs);
$lastPK = $row_rs['lastPK'];
mysql_free_result($rs);
return sprintf("$prefix%04d", substr($lastPK,1)+1);
}
?>

```

Further Reading :

printf --- 輸出格式化字串
 sprintf --- 傳回格式化字串

本篇要探討的重點，就是『如何格式化』？後面僅以 printf 作為範例。

printf() 函數的調用格式為：

printf("<格式化字串>", <參量表>);

其中 格式化字串 包括兩部分內容：一部分是 正常字符，這些字符將按原樣輸出；另一部分是 格式化規定字符，以 "%" 開始，後跟一個或幾個規定字符，用來確定輸出內容格式。 參量表 是需要輸出的一系列參數，其個數必須與格式化字串所說明的輸出參數個數一樣多，各參數之間用 "," 分開，且順序一一對應，否則將會出現意想不到的錯誤。

1. 格式化規定符

格式化規定符如下：

符號	作用	符號	作用
%d	十進制有符號整數	%e	指數形式的浮點數
%u	十進制無符號整數	%x、%X	無符號以十六進製表示的整數
%f	浮點數	%o	無符號以八進製表示的整數
%s	字符串	%b	二進制整數
%c	單字元	%g	自動選擇合適的表示法
%p	指針的值		

說明：

(1). 可以在 "%" 和字母之間插進數字表示 最大場寬。

例如：%3d 表示輸出 3 位整型數，不夠 3 位右對齊。%9.2f 表示輸出場寬為 9 的浮點數，其中小數位為 2，整數位為 6，小數點占一位，不夠 9 位右對齊。%8s 表示輸出 8 個字符的字符串，不夠 8 個字符右對齊。如果字符串的長度、或整型數位數超過說明的場寬，將按其實際長度輸出。但對浮點數，若整數部分位數超過了說明的整數位寬度，將按實際整數位輸出；若小數部分位數超過了說明的小數位寬度，則按說明的寬度以四捨五入輸出。

另外，若想在輸出值前加一些 0，就應在場寬項前加個 0。

例如：%04d 表示在輸出一個小於 4 位的數值時，將在前面補 0 使其總寬度為 4 位。