

*Jean ENGELS*

# PHP 5 – Cours et exercices

## Corrigés des exercices du livre

### Exercices du chapitre 2

#### Exercice 1

Parmi les variables suivantes, lesquelles ont un nom valide :  
mavar, \$mavar, \$var5, \$\_mavar, \$\_5var, \$\_\_élément1,  
\$hotel4\* ?

Les noms \$mavar, \$var5, \$\_mavar, \$\_5var, \$\_\_élément1 respectent les conventions. Les autres ne sont pas valides : mavar ne commence pas par le caractère \$ et \$hotel4\* se termine par le caractère \*

#### Exercice 2

Donner les valeurs de \$x, \$y, \$z à la fin du script suivant :

```
$x="PostgreSQL";  
$y="MySQL";  
$z=&$x;  
$x="PHP 5";  
$y=&$x;
```

les variables \$x, \$y et \$z valent toutes "PHP 5"

#### Exercice 3

Lire les valeurs des variables du script de l'exercice 2 à l'aide du tableau \$GLOBALS.

```

<?php
$x="PostgreSQL";
$y="MySQL";
$z=&$x;
$x="PHP 5";
$y=&$x;
echo $GLOBALS['x'], "<br />";
echo $GLOBALS['y'], "<br />";
echo $GLOBALS['z'], "<br />";
?>

```

## Exercice 4

Déterminer le numéro de version de PHP, le nom du système d'exploitation de votre serveur ainsi que la langue du navigateur du poste client.

```

<?php
echo "Version de PHP : ",PHP_VERSION, "<br />";
echo "Système d'exploitation du serveur : ",PHP_OS, "<br />";
echo "Langue du navigateur client :
",$_SERVER["HTTP_ACCEPT_LANGUAGE"], "<br />";
?>

```

J'obtiens le résultat suivant (sur mon poste) :

```

Version de PHP : 5.0.2
Système d'exploitation du serveur : WINNT
Langue du navigateur client : fr

```

## Exercice 5

Donner la valeur de chacune des variables pendant et à la fin du script suivant et vérifier l'évolution du type de ces variables :

```

$x="PHP5";
$a[]=&$x;
$y=" 5 eme version de PHP";
$z=$y*10;
$x.=$y;
$y*=$z;
$a[0]="MySQL";

```

Solution : Script affichant les valeurs et les types

```

<?php
$x="PHP5";
echo "\$x vaut : $x et est de type ", gettype($x), "<br />";
$a[]=&$x;
echo "\$a[0] vaut : $a[0] et est de type ", gettype($a), "<br />";
$y=" 5 eme version de PHP";
echo "\$y vaut : $y et est de type ", gettype($y), "<br />";
$z=$y*10;
echo "\$z vaut : $z et est de type ", gettype($z), "<br />";
$x.=$y;
echo "\$x vaut : $x et est de type ", gettype($x), "<br />";
$y*=$z;
echo "\$y vaut : $y et est de type ", gettype($y), "<br />";

```

```
$a[0]="MySQL";
echo "\$a[0] vaut : {$a[0]} et est de type ", gettype($a), "<br />";
?>
```

Résultat affiché :

```
$x vaut : PHP5 et est de type string
$a[0] vaut : PHP5 et est de type array
$y vaut : 5 eme version de PHP et est de type string
$z vaut : 50 et est de type integer
$x vaut : PHP5 5 eme version de PHP et est de type string
$y vaut : 250 et est de type integer
$a[0] vaut : MySQL et est de type array
```

## Exercice 6

Donner la valeur des variables \$x, \$y, \$z à la fin du script :

```
$x="7 personnes";
$y=(integer) $x;
$x="9E3";
$z=(double) $x;
```

Solution :

```
$x vaut : 9E3
$y vaut : 7
$z vaut : 9000
```

## Exercice 7

Donner la valeur booléenne des variables \$a, \$b, \$c, \$d, \$e et \$f :

```
$a="0";
$b="TRUE";
$c=FALSE;
$d=($a OR $b);
$e=($a AND $c);
$f=($a XOR $b);
```

Solution : même dans sa dernière version, PHP continue à afficher 1 pour TRUE et rien (une chaîne vide !) pour FALSE. Pour réaliser l'évaluation booléenne des variables et afficher le résultat en clair (avec les mots TRUE ou FALSE) nous pouvons écrire le code suivant :

```
<?php
function bool($val)
{
if($val) echo "TRUE <br />";
else echo "FALSE <br />";
}
//*****
$a="0";
echo "\$a vaut : ",bool($a);
$b="TRUE";
echo "\$b vaut : ",bool($b);
```

```

$c=FALSE;
echo "\$c vaut : ",bool($c);
$d=($a OR $b);
echo "\$d vaut : ",bool($d);
$e=($a AND $c);
echo "\$e vaut : ",bool($e);
$f=($a XOR $b);
echo "\$f vaut : ",bool($f);
?>

```

Nous obtenons l'affichage ci-dessous :

```

$a vaut : FALSE
$b vaut : TRUE
$c vaut : FALSE
$d vaut : TRUE
$e vaut : FALSE
$f vaut : TRUE

```

## Exercices du chapitre 3

### Exercice 1

Rédiger une expression conditionnelle pour tester si un nombre est à la fois un multiple de 3 et de 5.

```

<?php
$x=1245;
if($x%3==0 AND $x%5==0)
{
echo "$x est multiple de 3 et de 5 <br />";
}
else
{
echo "$x n'est pas multiple de 3 et de 5 <br />";
}
?>

```

N.B : En pratique un multiple de 3 et de 5 est un multiple de 15 !

### Exercice 2

Écrire une expression conditionnelle utilisant les variables \$age et \$sexe dans une instruction if pour sélectionner une personne de sexe féminin dont l'âge est compris entre 21 et 40 ans et afficher un message de bienvenue approprié.

```

<?php
$sexe="F";
$age=43;
if($sexe=="F" AND $age>=21 AND $age <= 40)

```

```

{
echo "Bonjour Madame vous avez entre 21 et 40 ans <br />";
}
else
{
echo "Désolé, vous ne remplissez pas les conditions <br />";
}
?>

```

### Exercice 3

Effectuer une suite de tirages de nombres aléatoires jusqu'à obtenir une suite composée d'un nombre pair suivi de deux nombres impairs.

```

<?php
$compteur=0;
do
{
$x=rand(0,1000);
$y=rand(0,1000);
$z=rand(0,1000);
$compteur++;
echo $x, ", " , $y, ", " , $z,"<br />";
}
while($x%2==1 OR $y%2==0 OR $z%2==0);
echo "Résultat obtenu en $compteur coups";
?>

```

Exemple de résultat obtenu :

```

194,285,494
435,759,162
237,292,768
366,533,397
Résultat obtenu en 4 coups

```

### Exercice 4

Créer et afficher des numéros d'immatriculation automobile (pour Paris, par exemple) en commençant au numéro 100 PHP 75. Effectuer ensuite la même procédure en mettant en réserve les numéros dont le premier groupe de chiffres est un multiple de 100. Stocker ces numéros particuliers dans un tableau.

Si on réalise le script complet, il affiche plusieurs millions de numéros de 100 PHP 75 à 999 ZZZ 75. L'exécution est donc très longue et risque de bloquer le serveur. Pour effectuer un test, les valeurs des chiffres sont limitées ci-dessous entre 100 et 120.

```

<?php
$initx=80;//lettre P
$inity=72;//lettre H
$initz=80;//lettre P

```

```

//Boucles imbriquées
for($x=$initx;$x<=90;$x++)
{
  for($y=$inity;$y<=90;$y++)
  {
    for($z=$initz;$z<=90;$z++)
    {
      for($num=100;$num<120;$num++)
      {
        echo "Numéro : $num ",chr($x),chr($y),chr($z)," 75<br />";
      }
      $initz=65;//on repart à la lettre A pour le troisième
caractère
    }
    $inity=65;//on repart à la lettre A pour le deuxième caractère
  }
}
?>

```

Pour ne conserver que les nombres multiples de 100 nous remplaçons l'instruction echo par le stockage des numéros dans un tableau. Il est affiché à la fin de toutes les boucles à l'aide de la fonction print\_r(). On constate alors qu'il contient déjà 65150 éléments !

```

<?php
$initx=80;//lettre P
$inity=72;//lettre H
$initz=80;//lettre P
//Boucles imbriquées
for($x=$initx;$x<=90;$x++)
{
  for($y=$inity;$y<=90;$y++)
  {
    for($z=$initz;$z<=90;$z++)
    {
      for($num=100;$num<1000;$num+=100)
      {
        $tab[]=$num".chr($x).chr($y).chr($z)." 75";
      }
      $initz=65;//on repart à la lettre A pour le troisième
caractère
    }
    $inity=65;//on repart à la lettre A pour le deuxième caractère
  }
}
print_r($tab);
?>

```

## Exercice 5

Choisir un nombre de trois chiffres. Effectuer ensuite des tirages aléatoires et compter le nombre de tirages nécessaire pour obtenir le nombre initial. Arrêter les tirages et afficher le nombre de coups réalisés. Réaliser ce script d'abord avec l'instruction `while` puis avec l'instruction `for`.

Avec une boucle `while`

```
<?php
//Nombre à trouver
$nb=789;
//compteur
$coup=0;
//boucle de tirage
while($x!=$nb)
{
    $x=rand(0,1000);
    $coup++;
    //echo $x,"<br />";//pour afficher tous les tirages
}
echo "$nb trouvé en $coup coups ";
?>
```

Avec une boucle `for`

```
<?php
//Nombre à trouver
$nb=789;
//boucle de tirage
for($coup=1;$x!=$nb;$coup++)
{
    $x=rand(0,1000);
    //echo $x,"<br />";//pour afficher tous les tirages
}
echo "$nb trouvé en $coup coups ";
?>
```

Notez qu'il est rare d'obtenir le nombre cherché en moins de 100 tirages.

## Exercice 6

Créer un tableau dont les indices varient de 11 à 36 et dont les valeurs sont des lettres de A à Z. Lire ensuite ce tableau avec une boucle `for` puis une boucle `foreach` et afficher les indices et les valeurs (la fonction `chr(n)` retourne le caractère dont le code ASCII vaut `n`).

```

<?php
for ($i=11;$i<=36;$i++)
{
    $stab[$i]=chr(54+$i);
}
//Lecture avec for
for ($i=11;$i<=36;$i++)
{
    echo "Elément d'indice $i : $stab[$i] <br />";
}
echo "<hr />";
//Lecture avec foreach
foreach($stab as $cle=>$valeur)
{
    echo "Elément d'indice $cle : $valeur <br />";
}
?>

```

## Exercice 7

Utiliser une boucle `while` pour déterminer le premier entier obtenu par tirage aléatoire qui soit un multiple d'un nombre donné. Écrire la variante utilisant la boucle `do...while`.

Avec une boucle `while`, notez qu'il faut initialiser `$x` avant la boucle.

```

<?php
$nb=57;
$compteur=0;
$x=rand(0,1000);
//Boucle while
while ($x%$nb!=0)
{
    $x=rand(0,1000);
    $compteur++;
}

echo "$x est multiple de $nb: Résultat obtenu en $compteur coups";
?>

```

Avec une boucle `do...while` il n'est pas nécessaire d'initialiser `$x`.

```

<?php
$nb=57;
$compteur=0;
do
{
    $x=rand(0,1000);
    $compteur++;
}
while ($x%$nb!=0);
echo "$x est multiple de $nb: Résultat obtenu en $compteur coups";
?>

```

## Exercice 8

Rechercher le PGCD (plus grand commun diviseur) de deux nombres donnés. Gérer au moyen d'une exception le cas où au moins un des nombres n'est pas entier.

Si vous avez oublié l'algorithme d'Euclide pour calculer le PGCD visitez le site : <http://villemin.gerard.free.fr/ThNbDemo/AlgoEucl.htm>

```
<?php
//Il faut $A > $B
$A=56952;
$B=3444;
try
{
    if(!(is_integer($A) OR is_integer($b)))
    {throw new Exception("Nombre(s) non entiers !",99);}
    else
    {
        echo "Le PGCD de $A et $B est : ";
        do
        {
            $R=$A%$B;
            $A=$B;
            $B=$R;
        }
        while($R!=0);
        echo $A ;
    }
}
catch(Exception $except)
{
    echo $except->getMessage();
}
?>
```

Le résultat affiché est ici :

```
Le PGCD de 56952 et 3444 est : 84
```

# Exercices du chapitre 4

## Exercice 1

Transformez une chaîne écrite dans des casses différentes afin que chaque mot ait une initiale en majuscule.

```
<?php
$ch="TransFormeZ unE Chaîne éCRITe dans des cASSes diFFérenTes afin
qUe chaQUe MOT ait une inITiale en MAJUSCULE";
```

```
echo ucwords(strtolower($ch));
?>
```

Le résultat est le suivant :

```
Transformez Une Chaîne Écrite Dans Des Casses Différentes Afin Que
Chaque Mot Ait Une Initiale En Majuscule
```

Remarquez que la fonction `ucwords()` permet d'avoir des majuscules accentuées.

## Exercice 2

En utilisant la fonction `strlen()` écrivez une boucle qui affiche chaque lettre de la chaîne PHP 5 sur une ligne différente.

```
<?php
$ch="PHP 5";
for($i=0;$i<strlen($ch);$i++)
{
echo "<h3> $ch[$i] </h3>";
}
?>
```

## Exercice 3

Formatez l'affichage d'une suite de chaînes contenant des nom et prénom en respectant les critères suivants : un prénom et un nom par ligne affichés sur 20 caractères; toutes les initiales des mots doivent se superposer verticalement.

```
<?php
$nom1="Azerky";
$prenom1="Sophia";
echo sprintf("<tt>%'_-20s %'_-20s </tt><br />", $nom1, $prenom1);
$nom2="Bazertudoh";
$prenom2="Jean-Michel";
echo sprintf("<tt>%'_-20s %'_-20s </tt><br />", $nom2, $prenom2);
?>
```

Pour obtenir de bons résultats il est préférable d'utiliser une police à espacement fixe en incluant le texte de chaque ligne dans un élément HTML `<tt>`. Le résultat obtenu est le suivant :

```
Azerky_____ Sophia_____
Bazertudoh_____ Jean-Michel_____
```

Nous pouvons également utiliser la fonction `vsprintf()` et passer les chaînes à afficher dans un tableau multidimensionnel.

```
<?php
$stab[0][]="Azerky";
$stab[0][]="Sophia";
$stab[1][]="Bazertudoh";
$stab[1][]="Jean-Michel";
```

```

for($i=0;$i<count($stab);$i++)
{
    echo vsprintf("<tt>%'_-20s %'_-20s </tt><br />", $stab[$i]);
}
?>

```

## Exercice 4

Utilisez les fonctions adéquates afin que la chaîne `<form action="script.php">` soit affichée telle quelle et non comme du code HTML.

```

<?php
$ch='<form action="script.php">';
echo htmlentities($ch);
?>

```

Le code HTML de l’affichage est le suivant :

```

<lt;form action=&quot;script.php&quot;&gt;&gt;

```

## Exercice 5

À partir de deux chaînes quelconques contenues dans des variables, effectuez une comparaison entre elles pour pouvoir les afficher en ordre alphabétique naturel.

Nous utilisons la fonction `strtolower()` avant d’opérer la comparaison, sinon tous les caractères de A à Z sont avant les caractères a à z.

```

<?php
$ch1="alpha";
$ch2="Azéma";
if(strtolower($ch1)<strtolower($ch2)) echo $ch1 ," est avant ",
$ch2;
else echo $ch2 ," est avant ", $ch1;
?>

```

## Exercice 6

Effectuez une censure sur des textes en n’affichant pas ceux qui contiennent le mot `zut`.

```

<?php
$ch="Zut je me suis trompé";
$modele="zut";
if(eregi($modele,$ch)) echo "Un Mot censuré";
else echo $ch
?>

```

## Exercice 7

Créez une fonction de validation d’une adresse HTTP ou FTP en vous inspirant de l’exemple 4-12.

Le modèle doit répondre à la définition suivante :

1. Commencer par « www »
2. Suivi par des lettres puis éventuellement un point ou un tiret suivis d'un deuxième groupe de lettres
3. Se terminer par un point suivi de l'extension qui peut avoir de 2 à 4 caractères.

Par exemple, les adresses `www.machin.com` ou `www.machin-truc.uk` sont valides.

```
<?php
//Création de la fonction de validation
function validhttp($ch)
{
    $modele="^(www)\.([a-z0-9]+)(\.|-)?([a-z0-9]*)\.([a-z]{2,4}$)";
    $ch=strtolower($ch);
    if (ereg($modele, $ch))
    {
        echo "$ch est valide <br />";
    }
    return TRUE;
}
else
{
    echo "$ch est invalide <br />";
}
return FALSE;
}
}
//Utilisation de la fonction de validation
$url1="www.laposte2.info";
$url2="www.pierre-plus-loin.info";
$url3="www.engels.funphp.com";
validhttp($url1);
validhttp($url2);
validhttp($url3);
?>
```

Les résultats obtenus sont les suivants :

```
www.laposte2.info est valide
www.pierre-plus-loin.info est invalide
www.engels.funphp.com est valide
```

## Exercice 8

Créez une expression régulière pour valider un âge inférieur à 100 ans.

```
<?php
$modelle="^[0-9]?([0-9]?)$";
$age="84";
if (ereg($modelle, $age))
{
    echo "$age est un age valide <br />";
}
```

```

    return TRUE;
}
else
{
    echo "$age est un age invalide <br />";
    return FALSE;
}
?>

```

Le résultat obtenu est le suivant :

```
84 est un age valide
```

## Exercice 9

Dans la chaîne PHP 5 \n est meilleur \n que ASP \n et JSP \n réunis, remplacez les caractères \n par <br /> en utilisant deux méthodes différentes (une fonction ou une expression régulière).

1. En utilisant la fonction nl2br()

```

<?php
$ch="PHP 5 \n est meilleur \n que ASP \n et JSP \n réunis";
echo "<p>",nl2br($ch),"</p>";
?>

```

2. En utilisant une expression régulière et la fonction ereg\_replace()

```

<?php
$ch="PHP 5 \n est meilleur \n que ASP \n et JSP \n réunis";
echo ereg_replace("\n","<br />",$ch);
?>

```

Dans les deux cas on obtient le même résultat suivant:

```

PHP 5
est meilleur
que ASP
et JSP
réunis

```

# Exercices du chapitre 5

## Exercice 1

Écrivez un tableau multidimensionnel associatif dont les clés sont des noms de personne et les valeurs des tableaux indicés contenant le prénom, la ville de résidence et l'âge de la personne.

```

<?php
$stab=array("Dupont"=>array("Paul","Paris",27),"Schmoll"=>array("Kir
k","Berlin",35),"Smith"=>array("Stan","Londres",45));

```

```
print_r($tab);
?>
```

La fonction `print_r()` affiche la structure du tableau :

```
Array ( [Dupont] => Array ( [0] => Paul [1] => Paris [2] => 27 )
 [Schmoll] => Array ( [0] => Kirk [1] => Berlin [2] => 35 ) [Smith]
=> Array ( [0] => Stan [1] => Londres [2] => 45 ) )
```

## Exercice 2

Écrivez un tableau multidimensionnel associatif dont les clés sont des noms de personne et les valeurs des tableaux associatifs dont les clés sont le prénom, la ville de résidence et l'âge de la personne avec une série de valeurs associées.

```
<?php
$tab=array("Dupont"=>array("prenom"=>"Paul", "ville"=>"Paris", "age"=
>27),
"Schmoll"=>array("prenom"=>"Kirk", "ville"=>"Berlin", "age"=>35),
"Smith"=>array("prenom"=>"Stan", "ville"=>"Londres", "age"=>45));
print_r($tab);
?>
```

La fonction `print_r()` affiche la structure du tableau :

```
Array ( [Dupont] => Array ( [prenom] => Paul [ville] => Paris [age]
=> 27 ) [Schmoll] => Array ( [prenom] => Kirk [ville] => Berlin
[age] => 35 ) [Smith] => Array ( [prenom] => Stan [ville] =>
Londres [age] => 45 ) )
```

## Exercice 3

Utilisez une boucle `foreach` pour lire les tableaux des exercices 1 et 2.

Lecture du tableau de l'exercice 1 :

```
<?php
$tab=array("Dupont"=>array("Paul", "Paris", 27), "Schmoll"=>array("Kir
k", "Berlin", 35), "Smith"=>array("Stan", "Londres", 45));
foreach($tab as $cle=>$valeur)
{
    echo "<b>Elément $cle :</b><br />";
    foreach($valeur as $ind=>$val)
    {
        echo "élément $ind :", $val, "<br />";
    }
}
?>
```

Le résultat obtenu est le suivant :

```
Elément Dupont :
élément 0 :Paul
élément 1 :Paris
élément 2 :27
Elément Schmoll :
élément 0 :Kirk
élément 1 :Berlin
élément 2 :35
```

```
Elément Smith :  
élément 0 :Stan  
élément 1 :Londres  
élément 2 :45
```

Lecture du tableau de l'exercice 2 :

```
<?php  
$stab=array("Dupont"=>array("prenom"=>"Paul", "ville"=>"Paris", "age"=  
>27),  
"Schmoll"=>array("prenom"=>"Kirk", "ville"=>"Berlin", "age"=>35),  
"Smith"=>array("prenom"=>"Stan", "ville"=>"Londres", "age"=>45));  
foreach($stab as $cle=>$valeur)  
{  
    echo "<b>Element $cle :</b><br />";  
    foreach($valeur as $cle2=>$val)  
    {  
        echo " $cle2 :", $val, "<br />";  
    }  
}  
?>
```

Le résultat obtenu est le suivant :

```
Element Dupont :  
prenom :Paul  
ville :Paris  
age :27  
Element Schmoll :  
prenom :Kirk  
ville :Berlin  
age :35  
Element Smith :  
prenom :Stan  
ville :Londres  
age :45
```

## Exercice 4

Utilisez une boucle `while` pour lire les tableaux des exercices 1 et 2.

Lecture du tableau de l'exercice 1 : nous utilisons une boucle `while` et la fonction `each()`.

```
<?php  
$stab=array("Dupont"=>array("Paul", "Paris", 27), "Schmoll"=>array("Kir  
k", "Berlin", 35), "Smith"=>array("Stan", "Londres", 45));  
while($element=each($stab))  
{  
    echo "Personne: {$element['key']} <br />";  
    while($coord=each($element[1]))  
    {  
        echo "clé {$coord[0]} valeur {$coord[1]} <br />";  
    }  
    echo"<hr />";  
}  
?>
```

Le résultat obtenu est le suivant :

```
Personne: Dupont
clé 0 valeur Paul
clé 1 valeur Paris
clé 2 valeur 27

-----

Personne: Schmoll
clé 0 valeur Kirk
clé 1 valeur Berlin
clé 2 valeur 35

-----

Personne: Smith
clé 0 valeur Stan
clé 1 valeur Londres
clé 2 valeur 45
```

Lecture du tableau de l'exercice 2 : nous utilisons une boucle while et la fonction each().

```
<?php
$stab=array("Dupont"=>array("prenom"=>"Paul", "ville"=>"Paris", "age"=
>27),
"Schmoll"=>array("prenom"=>"Kirk", "ville"=>"Berlin", "age"=>35),
"Smith"=>array("prenom"=>"Stan", "ville"=>"Londres", "age"=>45));

while($element=each($stab))
{
    echo "Personne: {$element['key']} <br />";
    while($coord=each($element[1]))
    {
        echo "{$coord[0]}:{$coord[1]} <br />";
    }
    echo"<hr />";
}
?>
```

Le résultat obtenu est le suivant :

```
Personne: Dupont
prenom:Paul
ville:Paris
age:27

-----

Personne: Schmoll
prenom:Kirk
ville:Berlin
age:35

-----

Personne: Smith
prenom:Stan
ville:Londres
age:45

-----
```

## Exercice 5

Créez un tableau contenant une liste d'adresses de sites recommandés, puis créez un lien aléatoire vers le premier site de la liste après avoir trié le tableau en ordre aléatoire.

La fonction `shuffle()` mélange effectivement les éléments d'un tableau mais ne conserve pas les clés, elle n'est donc pas adaptée pour récupérer la clé et la valeur du tableau `$tab`. Nous utilisons la fonction `array_rand()` qui retourne la clé de l'élément pris au hasard. Cette clé permet de lire le nom du site et son adresse URL.

```
<?php
$tab=array("PHP"=>"http://www.php.net", "MySQL"=>"http://www.mysql.o
rg", "SQLite"=>"http://www.sqlite.org");
$site=array_rand($tab);
echo "Site recommandé : <a href=\"\$tab[$site]\" > ",$site,"</a>";
?>
```

### Exemple de résultat affiché

Site recommandé : <u>MySQL</u>
--------------------------------

## Exercice 6

Créez un tableau d'entiers variant de 1 à 63, puis à partir de celui-ci un autre tableau de nombres variant de 0 à 6.3. Créez ensuite un tableau associatif dont les clés  $X$  varient de 0 à 6.3 et dont les valeurs sont  $\sin(X)$ . Affichez le tableau de valeurs dans un tableau HTML.

```
<?php
//Tableau ayant pour valeurs les entiers de 0 à 63
$tab=range(0,63);
//Tableau ayant pour valeurs les décimaux de 0 à 6.3
foreach($tab as $ind=>$val)
{
    $tab[$ind]=$tab[$ind]/10;
}
//Tableau dont les clés sont X et les valeurs sin(X)
foreach($tab as $ind=>$val)
{
    $val= (string) $val;
    $tabsin[$val]= sin($val);
}
//Création du tableau HTML
echo "<table border=\"1\" width=\"50%\" >";
echo " <caption><b>Tableau de valeurs de la fonction
sinus</b></caption>";
echo "<tr> <th> X </th> <th> sin( X )</th> </tr>";
```

```

foreach($tabsin as $cle=>$val)
{
    echo "<tr><td>$cle</td> <td>$val</td></tr>";
}
echo "</table>";
echo "<hr />";
?>

```

Le résultat affiché est le tableau HTML suivant :

**Tableau de valeurs de la fonction sinus**

<b>X</b>	<b>sin( X )</b>
0	0
0.1	0.0998334166468
0.2	0.198669330795
0.3	0.295520206661
0.4	0.389418342309
0.5	0.479425538604
0.6	0.564642473395
0.7	0.644217687238
0.8	0.7173560909
0.9	0.783326909627
1	0.841470984808
1.1	0.891207360061
1.2	0.932039085967
1.3	0.963558185417
1.4	0.985449729988
1.5	0.997494986604
1.6	0.999573603042
1.7	0.991664810452
1.8	0.973847630878
1.9	0.946300087687
2	0.909297426826
2.1	0.863209366649
2.2	0.80849640382
2.3	0.745705212177
2.4	0.675463180551
2.5	0.598472144104
2.6	0.515501371821
2.7	0.427379880234
2.8	0.334988150156
2.9	0.239249329214

3	0.14112000806
3.1	0.0415806624333
3.2	-0.0583741434276
3.3	-0.157745694143
3.4	-0.255541102027
3.5	-0.35078322769
3.6	-0.442520443295
3.7	-0.529836140908
3.8	-0.611857890943
3.9	-0.687766159184
4	-0.756802495308
4.1	-0.818277111064
4.2	-0.871575772414
4.3	-0.916165936749
4.4	-0.95160207389
4.5	-0.977530117665
4.6	-0.993691003633
4.7	-0.999923257564
4.8	-0.996164608836
4.9	-0.982452612624
5	-0.958924274663
5.1	-0.925814682328
5.2	-0.88345465572
5.3	-0.832267442224
5.4	-0.772764487556
5.5	-0.70554032557
5.6	-0.631266637872
5.7	-0.550685542598
5.8	-0.464602179414
5.9	-0.37387666483
6	-0.279415498199
6.1	-0.182162504272
6.2	-0.0830894028175
6.3	0.0168139004843

## Exercice 7

Créez un tableau contenant une liste d'adresses e-mail. Extrayez le nom de serveur de ces données, puis réalisez des statistiques sur les occurrences de chaque fournisseur d'accès.

```
<?php
//Tableau des adresses mail
$tab=array("php5@free.com","jean556@fiscali.fr","machine@waladoo.fr",
"webmestre@waladoo.fr","pauldeux@fiscali.fr","macafi@fiscali.fr")
;
//Récupération des noms de domaine
foreach($tab as $ind=>$val)
{
$dom=explode("@",$val);
$domaine[]=$dom[1];
}
//Compte du nombre d'occurrences de chaque domaine
$stat=array_count_values ($domaine);
//Nombre total d'adresses
$total=count ($tab);
//Ou encore
//$total=array_sum($stat);
//Calcul des pourcentages
foreach($stat as $fourn=>$nb)
{
$pourcent [$fourn]=$nb/$total*100;
echo "Fournisseur d'accès : $fourn =
",round($pourcent [$fourn],2)," % <br />";
}
?>
```

Le résultat obtenu ici est le suivant :

```
Fournisseur d'accès : free.com = 16.67 %
Fournisseur d'accès : fiscali.fr = 50 %
Fournisseur d'accès : waladoo.fr = 33.33 %
```

# Exercices du chapitre 6

## Exercice 1

Créer un formulaire comprenant un groupe de champs ayant pour titre "Adresse client". Le groupe doit permettre la saisie du nom, du prénom, de l'adresse, de la ville et du code postal. Les données sont ensuite traitées par un fichier PHP séparé récupérant les données et les affichant dans un tableau HTML.

Le formulaire est le fichier ch6exo1.htm qui ne contient que du code HTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>Exercice chapitre 6 n° 1</title>
</head>
<body>
<form action="ch6exo1.php" method="post">
  <fieldset>
    <legend><b>Saisissez vos coordonnées </b></legend>
<table border="0" >
  <tr>
    <td>Nom : </td>
    <td><input type="text" name="nom" /></td>
  </tr>
  <tr>
    <td>Prénom : </td>
    <td><input type="text" name="prenom" /></td>
  </tr>
  <tr>
    <td>Adresse : </td>
    <td><input type="text" name="adresse" /></td>
  </tr>
  <tr>
    <td>Ville : </td>
    <td><input type="text" name="ville" /></td>
  </tr>
  <tr>
    <td>Code postal : </td>
    <td><input type="text" name="code" maxlength="5"/></td>
  </tr>
  <tr>
    <td>CONFIRMER</td>
    <td><input type="submit" value="ENVOI" /></td>
  </tr>
</table>
</fieldset>
</form>
</body>
</html>

```

Le fichier `ch6exo1.php` de traitement des données est un fichier PHP.

```

<?php
echo "<table border=\\"1\\" >";
echo "<caption><b>Confirmation de vos coordonnées</b></caption>";
foreach($_POST as $cle=>$val)
{
  echo "<tr> <td> $cle : &nbsp;  </td> <td>".stripslashes($val)
  . "</td></tr>";
}
echo "</table>";
?>

```

La fonction `stripslashes()` permet de supprimer le caractère d'échappement `\` qui est automatiquement ajouté dans les chaînes saisies. Si on ne l'utilise pas on obtient dans l'exemple suivant l'affichage « l'Odéon » au lieu de « l'Odéon ». Le résultat obtenu est le suivant :

Confirmation de vos coordonnées	
nom :	Lagarde
prenom :	Richard
adresse :	556 Rue de l'Odéon
ville :	Paris
code :	75006

## Exercice 2

Améliorer le script précédent en vérifiant l'existence des données et en affichant une boîte d'alerte JavaScript si l'une des données est manquante.

Le fichier `ch6exo2.htm` est identique au fichier `ch6exo1.htm` mis à part la ligne :

```
<form action="ch6exo2.php" method="post">
```

qui désigne le fichier de traitement `ch6exo2.php`.

Le fichier `ch6exo2.php` de traitement des données :

```
<?php
if(!empty($_POST['nom']) AND !empty($_POST['prenom']) AND
!empty($_POST['adresse']) AND !empty($_POST['ville']) AND
!empty($_POST['code']))
{
    echo "<table border=\"1\" >";
    echo "<caption><b>Confirmation de vos coordonnées
2</b></caption>";
    foreach($_POST as $cle=>$val)
    {
        echo "<tr> <td> $cle : &nbsp;</td> <td>".stripslashes($val)
.</td></tr>";
    }
    echo "</table>";
}
else
{
    echo "<script type=\"text/javascript\">alert('Le formulaire est
incomplet');document.location='ch6exo2.htm' </script>";
}
?>
```

## Exercice 3

Le fichier suivant peut-il être enregistré avec l'extension **.php** ou **.htm** ? Où se fait le traitement des données ?

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 //EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title> Insertion des données </title>
</head>
<body>
<form method="post" action="ajout.php" >
//Suite du formulaire
</form>
</body>
</html>
```

Le fichier ne contient que du code HTML, il peut donc être enregistré avec l'extension **.htm**. Il peut cependant être enregistré avec l'extension **.php** mais cela est inutile. Le traitement des données saisies est fait par le code PHP du fichier externe « ajout.php ».

#### Exercice 4

Comment faire pour que les données soient traitées par le même fichier que celui qui contient le formulaire ? Proposer deux solutions.

Première solution : l'attribut `action` de l'élément `<form>` doit contenir le nom du fichier qui crée le formulaire.

Deuxième solution : l'attribut `action` de l'élément `<form>` contient le code suivant :

```
action= "<?=$_SERVER["PHP_SELF"] ?>"
```

#### Exercice 5

Créer un formulaire de saisie d'adresse e-mail contenant un champ caché destiné à récupérer le type du navigateur de l'utilisateur. Le code PHP affiche l'adresse mail et le nom du navigateur dans la même page après vérification de l'existence des données.

La page de saisie du mail et de traitement des données (fichier `ch6exo5.php`) :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>Exercice chapitre 6 n° 5</title>
</head>
```

```

<body>
<form action="ch6exo5.php" method="post">
  <fieldset>
    <legend><b>Saisissez votre e-mail </b></legend>
<table border="0" >
  <tr>
    <td>mail : </td>
    <td><input type="text" name="mail" /><input type="hidden"
name="navigateur" value="<?=$_SERVER['HTTP_USER_AGENT'] ?>" /></td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td><input type="submit" value="ENVOI" /></td>
  </tr>
</table>
</fieldset>
</form>
<?php
if(!empty($_POST['mail']) AND !empty($_POST['navigateur'])) )
{
  echo "<table border=\"1\" >";
  echo "<caption><b>Votre mail et votre navigateur</b></caption>";
  foreach($_POST as $cle=>$val)
  {
    echo "<tr> <td> $cle : &nbsp;</td> <td>".stripslashes($val)
."</td></tr>";
  }
  echo "</table>";
}
else
{
  echo "<script type=\"text/javascript\">alert('Le formulaire est
incomplet');document.location='ch6exo5.php' </script>";
}
?>
</body>
</html>

```

Résultat obtenu :

Votre mail et votre navigateur	
mail :	jan@funphp.com
navigateur :	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)

## Exercice 6

Créer un formulaire demandant la saisie d'un prix HT et d'un taux de TVA. Le script affiche le montant de la TVA et le prix TTC dans deux zones de texte créées dynamiquement. Le formulaire maintient les données saisies.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>

```

```

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>Exercice chapitre 6 n° 5</title>
</head>
<body>
<form action="<?=$_SERVER['PHP_SELF'] ?>" method="post">
  <fieldset>
    <legend><b>Saisissez le prix HT et le taux de TVA </b></legend>
    <table border="0" >
      <tr>
        <td>Prix Hors Taxes : </td>
        <td><input type="text" name="prixht" value="<?=$_
$_POST['prixht']?>" /></td>
      </tr>
      <tr>
        <td>Taux de TVA (en %) : </td>
        <td><input type="text" name="tva" value="<?=$_
$_POST['tva']?>" /></td>
      </tr>
    </table>
    <?php
    if(!empty($_POST['prixht']) AND !empty($_POST['tva'])) )
    {
      echo "<tr><td>Montant de la TVA : </td><td><input type=\"text\"
value=\"\".round($_POST['prixht']*$_POST['tva']/100,2)
.\"\"/></td></tr>";
      echo "<tr><td>Prix TTC : </td><td><input type=\"text\"
value=\"\".round($_POST['prixht']*(1+$_POST['tva']/100),2)
.\"\"/></td></tr>";
    }
    else
    {
      echo "<b>Le formulaire est incomplet!</b>";
    }
    ?>

    <tr>
      <td>&nbsp;</td>
      <td><input type="submit" value="ENVOI" /></td>
    </tr>
  </table>
</fieldset>
</form>

</body>
</html>

```

## Exercice 7

Créer un formulaire n'effectuant que le transfert de fichiers ZIP et d'une taille limitée à 1 Mo. Le script affiche le nom du fichier du poste client ainsi que la taille du fichier transféré et la confirmation de réception.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>

```

```

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>Exercice chapitre 6 n° 7</title>
</head>
<body>
<form action="<?=$_SERVER['PHP_SELF'] ?>" method="post"
enctype="multipart/form-data">
  <fieldset>
    <legend><b>Transférez un fichier ZIP </b></legend>
<table border="0" >
  <tr>
    <td>Choisissez un fichier </td>
    <td><input type="file" name="fich" /></td>
    <td><input type="hidden" name="MAX_FILE_SIZE"
accept="application/zip" value="1000000"/></td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td><input type="submit" value="ENVOI" /></td>
  </tr>
</table>
</fieldset>
</form>
</body>
</html>

<?php
if(isset($_FILES['fich']))
{
  if($_POST["MAX_FILE_SIZE"] <$_FILES["fich"]["size"])
  {
    echo "<b>Taille trop grande </b><hr />";
    echo "Taille maximale autorisée :",$_POST["MAX_FILE_SIZE"],"
octets<hr />";
    echo "Taille du fichier transféré :",$_FILES["fich"]["size"],"
octets<hr />";
  }
  else
  {
    //Enregistrement et renommage du fichier

$result=move_uploaded_file($_FILES["fich"]["tmp_name"],"monfichier.
zip");
    if($result==TRUE)
    {
      echo "<b>Vous avez bien transféré le fichier</b><hr />";
      echo "Le nom du fichier est : ",$_FILES["fich"]["name"],"<hr
/>";
      echo "Votre fichier a une taille de
",$_FILES["fich"]["size"],"<hr />";
    }
    else {echo "<hr /> Erreur de transfert
n°",$_FILES["fich"]["error"];}
  }
}
?>

```

## Exercice 8

Dans la perspective de création d'un site d'agence immobilière, créer un formulaire comprenant trois boutons Submit nommés « Vendre », « Acheter » et « Louer ». En fonction du choix effectué par le visiteur, le rediriger vers une page spécialisée dont le contenu réponde au critère choisi.

La page qui crée le formulaire (fichier ch6exo8.php)

```
<?php
switch($_POST["choix"])
{
    case "Vendre":
        header("location:pagevente.htm");
        break;
    case "Acheter":
        header("location:pageachat.htm");
        break;
    case "Louer":
        header("location:pagelocation.htm");
        break;
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>Exercice chapitre 6 n° 8</title>
</head>
<body>
<form action="<?= $_SERVER['PHP_SELF'] ?>" method="post" >
    <fieldset>
        <legend><b>Faites votre choix </b></legend>
        <table border="0" >
            <tr>
                <td><input type="submit" name="choix" value="Vendre" /></td>
                <td><input type="submit" name="choix" value="Acheter"
/></td>
                <td><input type="submit" name="choix" value="Louer" /></td>
            </tr>
        </table>
    </fieldset>
</form>
</body>
</html>
```

Un exemple de page cible d'un bouton d'envoi (fichier pagevente.htm). La méthode JavaScript `window.history.back()` permet de rediriger le visiteur vers la page précédente.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <meta      http-equiv="Content-Type"      content="text/html;
charset=iso-8859-1" />
    <title> Page des vendeurs immobiliers </title>
  </head>
  <body>
    <h1>Vous voulez vendre</h1>
    <a href="javascript:window.history.back()">RETOUR</a>
  </body>
</html>

```

## Exercices du chapitre 7

### Exercice 1

Créez une fonction PHP qui affiche une boîte d'alerte à partir de la fonction JavaScript dont la syntaxe est `alert("chaîne de caractères")`. Cette fonction peut être appelée avec comme paramètre le texte du message à afficher. Elle est particulièrement utile pour afficher des messages d'erreur de manière élégante, sans que ces derniers restent écrits dans la page.

La fonction retourne la valeur TRUE. Ceci n'est pas obligatoire mais peut permettre un contrôle d'exécution.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <meta      http-equiv="Content-Type"      content="text/html;
charset=iso-8859-1" />
    <title> Chapitre 7 Exercice 1 </title>
  </head>
  <body>
    <?php
function alerte($ch)
{
  echo "<script type=\"text/javascript\"> alert('$ch'); </script>";
  return TRUE;
}
//Utilisation
if(alerte("Bonjour à tous")) echo "La fonction alerte() fonctionne
bien";
alerte('RAAAAAAAAAAAAAAAAAAAAA');
?>
  </body>
</html>

```

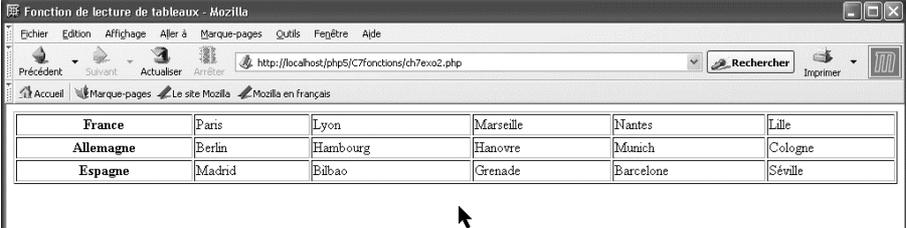
### Exercice 2

Écrivez une fonction de lecture de tableaux multidimensionnels en vous inspirant de l'exemple 7.3. L'affichage se fait sous forme de tableau HTML dont les titres sont les clés des tableaux.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
<title>Fonction de lecture de tableaux multidimensionnels</title>
</head>
<body>
<?php
//Définition de la fonction
function tabmulti($tab,$bord)
{

echo "<table border=\"\$bord\" width=\"100%\"><tbody>";
foreach($tab as $cle=>$tab2)
{
echo "<tr><th>$cle</th> ";
foreach($tab2 as $ind=>$val)
{
echo "<td>$val </td>";
}
echo "</tr>";
}
echo "</tbody> </table>";
}
//Définition des tableaux
$tab1
array("France"=>array("Paris", "Lyon", "Marseille", "Nantes", "Lille"),
"Allemagne"=>array("Berlin", "Hambourg", "Hanovre", "Munich", "Cologne"
),
"Espagne"=>array("Madrid", "Bilbao", "Grenade", "Barcelone", "Séville")
);
//Appel de la fonction
tabmulti($tab1,1);
?>
</body>
</html>
```

Le résultat affiché est le suivant :



<b>France</b>	Paris	Lyon	Marseille	Nantes	Lille
<b>Allemagne</b>	Berlin	Hambourg	Hanovre	Munich	Cologne
<b>Espagne</b>	Madrid	Bilbao	Grenade	Barcelone	Séville

### Exercice 3

Écrivez une fonction qui retourne la somme de la série de terme général  $u_n = x^{2n+1}/n!$ . Les paramètres de la fonction sont  $n$  pour le nombre d'itérations et  $d$  pour le nombre de décimales affichées pour le résultat. Il est possible de réutiliser la fonction `prod()` présentée dans ce chapitre pour calculer la factorielle  $n!$ .

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
<title>Fonction de calcul de la somme des termes d'une
suite</title>
</head>
<body>
<?php
//Définition de la fonction
function somme($n,$d,$x)
{
    $som=0;
    $fact=1;
    for($i=0;$i<=$n;$i++)
    {
//Calcul de factorielle de $i
if($i==0) {$fact=1 ;}
else {$fact*=$i;}
//Calculde la somme
        $som+= (pow($x,2*$i+1))/$fact;
//echo $som,"<br />";
    }
    return round($som,$d);
}
//Utilisation
echo somme(10,6,1);
?>
</body>
</html>
```

#### Exercice 4

Écrivez une fonction dont le paramètre passé par référence est un tableau de chaînes de caractères et qui transforme chacun des éléments du tableau de manière que le premier caractère soit en majuscule et les autres en minuscules, quelle que soit la casse initiale des éléments, même si elle est mixte.

```
<?php
//Définition de la fonction
function initmaj(&$stab)
{
    foreach($stab as $ind=>$val)
    {
        $stab[$ind]=ucfirst(strtolower($val));
    }
    return $stab;
}
```

```

}
//Utilisation
$stabch= array("AzertToTO","Sous peAu","sARtHES jp");
print_r(initmaj($stabch));
?>

```

## Exercice 5

À partir de la fonction sinus de PHP, écrivez une fonction qui donne le sinus d'un angle donné en radian, en degré ou en grade. Les paramètres sont la mesure de l'angle et l'unité est symbolisée par une lettre. Le deuxième paramètre doit avoir une valeur par défaut correspondant aux radians.

```

<?php
//Définition de la fonction
function sinus($x,$unit)
{
    switch($unit)
    {
        case "R":
        case "r":
            return sin($x);
            break;
        case "D":
        case "d":
            return sin($x/180*PI());
            break;
        case "G":
        case "g":
            return sin($x/200*PI());
            break;
    }
}
//Utilisation
echo sinus(PI()/6,R),"<br />";
echo sinus(60,d),"<br />";
echo sinus(100,g),"<br />";
?>

```

Résultats affichés :

```

0.5
0.866025403784
1

```

## Exercice 6

Créez une fonction de création de formulaires comprenant une zone de texte, une case d'option (*radio button*), un bouton Submit et un bouton Reset. Choisissez comme paramètres les attributs des différents éléments HTML en cause. Chaque appel de la fonction doit incorporer le code HTML du formulaire à la page.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

```





```

<?php
//Inclusion des fonctions
include('ch7exo7.php');
//Utilisation
$code= form("machin.php","post","Complétez le formulaire");
$code.= text("Votre nom","nom");
$code.= text("Votre prénom","prenom");
$code.= radio("Paris","ville","paris");
$code.= radio("Lyon","ville","lyon");
$code.= submit("Envoyer","Effacer");
$code.= finform();
echo $code;
?>
</body>
</html>

```

Nous verrons dans les exercices du chapitre 9 que l'utilisation d'objets permet le même genre d'opération de création de formulaire d'une façon plus élégante.

## Exercices du chapitre 8

### Exercice 1

Après avoir consulté le résultat affiché par l'exemple 8-1, déterminer la date et l'heure de l'exécution de ce script.

Le timestamp en question vaut 1098480456 .

```

<?php
echo date("\Le d / M / Y à H:m:s",1098480456);
?>

```

Le résultat affiché est : « Le 22 / Oct / 2004 à 23:10:36 »

Notez l'utilisation du caractère d'échappement pour afficher normalement la lettre « L ».

### Exercice 2

Calculez votre age à l'instant actuel à la seconde près.

Le script suivant ne fonctionne pas en local sous Windows et affiche le message suivant : « Warning: mktime() [function.mktime]: Windows does not support negative values for this function in c:\wamp\www\php5\C8dates\ch8exo2.php on line 2 ». En effet Windows n'accepte pas les timestamps négatifs et dans l'exemple la date de naissance est antérieure à 1970. En revanche, sur un serveur externe sous Linux, le script fonctionne.

```

<?php

```

```

$datenaiss= mktime(9,15,45,1,21,1962);
$aujourd'hui=time();
$age=$aujourd'hui - $datenaiss;
echo "Mon age est de $age secondes";
?>

```

On obtient par exemple : Mon age est de 1352626403 secondes

### Exercice 3

Vérifiez si la date du 29 Février 1962 a existé.

```

<?php
if(checkdate(2,29,1962)) {echo "Date valide";}
else{echo "Date non valide";}
?>

```

Le résultat est : « Date non valide »

### Exercice 4

Quel jour de la semaine était le 3 mars 1993 ? Affichez le résultat en français.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Afficher le jour de la semaine en français</title>
</head>
<body>
<?php
//Date en français
$jour = mktime(0,0,0,3,3,1993);
$semaine = array(" dimanche "," lundi "," mardi "," mercredi ","
jeudi "," vendredi "," samedi ");
echo "<h2>Le ",date("d M Y ",$jour)," était un ",
$semaine[date('w')], "</h2>";
?>
</body>
</html>

```

Le résultat est : Le 03 Mars 1993 était un mercredi

### Exercice 5

Affichez toutes les années bissextiles comprises entre 2005 et 2052.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Afficher les années bissextiles </title>
</head>
<body>
<?php
for($i=2005;$i<=2052;$i++)
{
    $date=mkttime(0,0,0,1,1,$i);
    if(date("L",$date)==1) {echo "L'année $i est bissextile<br />";}
}
?>
</body>
</html>

```

Les résultats affichés sont :

```

L'année 2008 est bissextile
L'année 2012 est bissextile
L'année 2016 est bissextile
L'année 2020 est bissextile
L'année 2024 est bissextile
L'année 2028 est bissextile
L'année 2032 est bissextile
L'année 2036 est bissextile

```

L'intervalle de validité d'un timestamp va généralement du Vendredi 13 Décembre 1901 20:45:54 GMT au Mardi 19 Janvier 2038 03:14:07 GMT. Donc les résultats affichés s'arrêtent à 2036. En local des alertes sont affichées pour les années postérieures à 2038.

## Exercice 6

Déterminez quel jour de la semaine seront tous les premier Mai des années comprises entre 2005 et 2010. Si le jour est un samedi ou un dimanche, affichez le message « Désolé ! ». Si le jour est un vendredi ou un lundi affichez « Week end prolongé ! ».

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Afficher jours de 1er mai </title>
</head>
<body>
<p>
<?php
for($i=2005;$i<=2010;$i++)
{
    $date=mkttime(0,0,0,5,1,$i);
    if(date("w",$date)==6 OR date("w",$date)==0 ) {echo
"1<sup>er</sup> Mai $i : Désolé<br />";}
    elseif(date("w",$date)==5 OR date("w",$date)==1 ) {echo
"1<sup>er</sup> Mai $i : Week end prolongé<br />";}
}
?>

```

```
</p>
</body>
</html>
```

Les résultats affichés sont :

```
1er Mai 2005 : Désolé
1er Mai 2006 : Week end prolongé
1er Mai 2009 : Week end prolongé
1er Mai 2010 : Désolé
```

## Exercice 7

L'Ascension est le quarantième jour après Pâques (Pâques compris dans les 40 jours). Calculez les dates de l'Ascension pour les années 2005 à 2010.

```
<?php
for($i=2004;$i<2010;$i++)
{
    echo "Jour de Paques : ",date("d M Y ",easter_date($i)),"<br />";
    echo "    Jour de l'Ascension    ",date("d M Y ",easter_date($i)+39*86400),"<br />";
}
?>
```

Les résultats obtenus sont les suivants :

```
Jour de Paques : 11 Apr 2004
Jour de l'Ascension 20 May 2004
Jour de Paques : 27 Mar 2005
Jour de l'Ascension 05 May 2005
Jour de Paques : 16 Apr 2006
Jour de l'Ascension 25 May 2006
Jour de Paques : 08 Apr 2007
Jour de l'Ascension 17 May 2007
Jour de Paques : 23 Mar 2008
Jour de l'Ascension 01 May 2008
Jour de Paques : 12 Apr 2009
Jour de l'Ascension 21 May 2009
```

# Exercices du chapitre 9

## Exercice 1

Écrivez une classe représentant une ville. Elle doit avoir les propriétés `nom` et `département` et une méthode affichant « la ville X est dans le département Y ». Créez des objets `ville`, affectez leurs propriétés, et utilisez la méthode d'affichage.

```
<?php
class ville
{
    public $nom;
    public $depart;
    public function getinfo()
    {
```

```

$texte="La ville de $this->nom est dans le département : $this-
>depart <br />";
return $texte;
}
}
//Création d'objets
$ville1 = new ville();
$ville1->nom="Nantes";
$ville1->depart="Loire Atlantique";
$ville2 = new ville();
$ville2->nom="Lyon";
$ville2->depart="Rhône";
echo $ville1->getinfo();
echo $ville2->getinfo();
?>

```

Le résultat affiché est :

```

La ville de Nantes est dans le département : Loire Atlantique
La ville de Lyon est dans le département : Rhône

```

## Exercice 2

Modifiez la classe précédente en la dotant d'un constructeur. Réalisez les mêmes opérations de création d'objets et d'affichage.

```

<?php
class ville
{
    private $nom;
    private $depart;
    public function __construct($nom,$depart)
    {
        $this->nom=$nom;
        $this->depart=$depart;
    }
    public function getinfo()
    {
        $texte="La ville de $this->nom est dans le département : $this-
>depart <br />";
        return $texte;
    }
}
//Création d'objets
$ville1 = new ville("Nantes", "Loire Atlantique");
$ville2 = new ville("Lyon", "Rhône");
echo $ville1->getinfo();
echo $ville2->getinfo();
?>

```

Le résultat affiché est :

```

La ville de Nantes est dans le département : Loire Atlantique
La ville de Lyon est dans le département : Rhône

```

On constate encore une fois que l'utilisation d'un constructeur facilite la création d'objet et l'affectation de ses propriétés. Une

deuxième possibilité consiste à isoler le code de création de la classe ville de l'exercice 2 dans un fichier séparé, de l'inclure dans le code de l'exercice 3 et d'étendre la classe ville à l'aide du mot-clé `extends` puis de lui ajouter le constructeur.

### Exercice 3

Créez une classe représentant une personne. Elle doit avoir les propriétés nom, prénom et adresse, ainsi qu'un constructeur et un destructeur. Une méthode `getpersonne()` doit retourner les coordonnées complètes de la personne. Une méthode `setadresse()` doit permettre de modifier l'adresse de la personne. Créez des objets `personne`, et utilisez l'ensemble des méthodes.

```
<?php
class personne
{
    private $nom;
    private $prénom;
    private $adresse;
    //Constructeur
    public function __construct($nom,$prénom,$adresse)
    {
        $this->nom=$nom;
        $this->prénom=$prénom;
        $this->adresse=$adresse;
    }
    //Destructeur
    public function __destruct()
    {
        echo "<script type=\"text/javascript\">alert('La personne nommée
        $this->prénom $this->nom \\nest supprimée de vos
        contacts')</script>";
    }
    //
    public function getpersonne()
    {
        $texte=" $this->prénom $this->nom <br /> $this->adresse <br />";
        return $texte;
    }
    //
    public function setadresse($adresse)
    {
        $this->adresse=$adresse;
    }
}
//Création d'objets
$client = new personne("Geelsen","Jan"," 145 Rue du Maine Nantes");
echo $client->getpersonne();
//Modification de l'adresse
$client->setadresse("23 Avenue Foch Lyon");
//Suppression explicite du client, donc appel du destructeur
unset($client);
```

```
//Fin du script
echo "Fin du script";
?>
```

Notez l'ordre d'apparition de la boîte d'alerte créée par le destructeur après l'appel de la fonction `unset()` et de l'affichage de la fin du script.

## Exercice 4

Créez une classe nommée `form` représentant un formulaire HTML. Le constructeur doit créer le code d'en-tête du formulaire en utilisant les éléments `<form>` et `<fieldset>`. Une méthode `settext()` doit permettre d'ajouter une zone de texte. Une méthode `setsubmit()` doit permettre d'ajouter un bouton d'envoi. Les paramètres de ces méthodes doivent correspondre aux attributs des éléments HTML correspondants. La méthode `getform()` doit retourner tout le code HTML de création du formulaire. Créez des objets `form`, et ajoutez-y deux zones de texte et un bouton d'envoi. Testez l'affichage obtenu.

Le fichier contenant la définition de la classe `form` (`ch9exo4.php`) est indépendant ce qui permet son inclusion dans d'autres scripts en vue de l'utilisation de la classe ou de son extension.

```
<?php
class form
{
    protected $code;
    protected $codeinit;
    protected $codetext;
    protected $codesubmit;
    public function __construct($action,$titre,$methode="post")
    {
        $this->codeinit="<form action=\"\$action\" method=\"\$methode\">";
        $this->codeinit.="<fieldset><legend><b>$titre</b></legend>";
        //$this->action = $action;
        //$this->methode=$methode;
    }
    //*****
    public function settext($name,$libelle,$methode="post")
    {
        $this->codetext.="<b>$libelle</b><input                                type=\"text\"
        name=\"\$name\" /><br /><br />";
    }
    //*****
    public function setsubmit($name="envoi",$value="Envoyer")
    {
        $this->codesubmit="<input                                type=\"submit\"                                name=\"\$name\"
        value=\"Envoyer\" /><br />";
    }
    //*****
    public function getform()
    {
        $this->code="";
    }
}
```

```

$this->code.=$this->codeinit;
    $this->code.=$this->codetext;
$this->code.=$this->codesubmit;
    $this->code.="</fieldset></form>";
echo $this->code;
}
}
?>

```

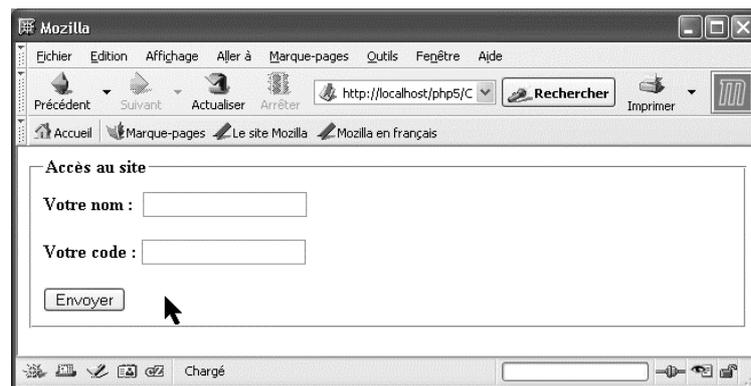
Le fichier `ch9exo4b.php` inclus le fichier précédent et crée un objet de type `form`.

```

<?php
include('ch9exo4.php');
//*****
$myform = new form("traitement.php","Accès au site","post");
$myform->settext("nom","Votre nom : &nbsp;");
$myform->settext("code","Votre code : ");
$myform->setsubmit();
$myform->getform();
?>

```

Le résultat obtenu est le suivant :



## Exercice 5

Créez une sous-classe nommée `form2` en dérivant la classe `form` de l'exercice 4. Cette nouvelle classe doit permettre de créer des formulaires ayant en plus des boutons radio et des cases à cocher. Elle doit donc avoir les méthodes supplémentaires qui correspondent à ces créations. Créez des objets, et testez le résultat.

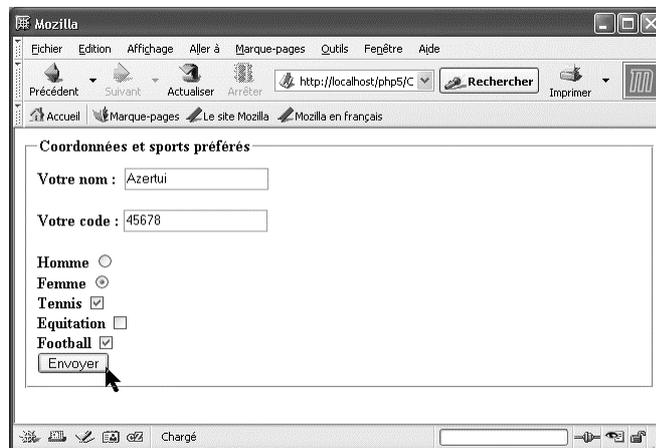
```

<?php
include('ch9exo4.php');
class form2 extends form
{
    protected $coderradio;
    protected $codecase;
    public function __construct($action,$titre,$methode="post")
    {
        parent::__construct($action,$titre,$methode="post");
    }
}

```

```
//*****
public function setradio($name,$libelle,$value)
{
    $this->coderadio.="<b>$libelle</b><input                type=\"radio\"
name=\"\$name\" value=\"\$value\" /><br />";
}
//*****
public function setcase($name,$libelle,$value)
{
    $this->codecase.="<b>$libelle</b><input                type=\"checkbox\"
name=\"\$name\" value=\"\$value\" /><br />";
}
//*****
public function getform()
{
    $this->code="";
    $this->code.=$this->codeinit;
    $this->code.=$this->codetext;
    $this->code.=$this->coderadio;
    $this->code.=$this->codecase;
    $this->code.=$this->codesubmit;
    $this->code.="</fieldset></form>";
    echo $this->code;
}
}
//*****
$myform = new form2("traitementb.php","Coordonnées et sports
préférés","post");
$myform->settext("nom","Votre nom : &nbsp;");
$myform->settext("code","Votre code : ");
$myform->setradio("sexe"," Homme ","homme");
$myform->setradio("sexe"," Femme ","femme");
$myform->setcase("loisir"," Tennis ","tennis");
$myform->setcase("loisir"," Equitation ","équitaion");
$myform->setcase("loisir"," Football ","football");
$myform->setsubmit();
$myform->getform();
?>
```

Le résultat obtenu est le suivant :

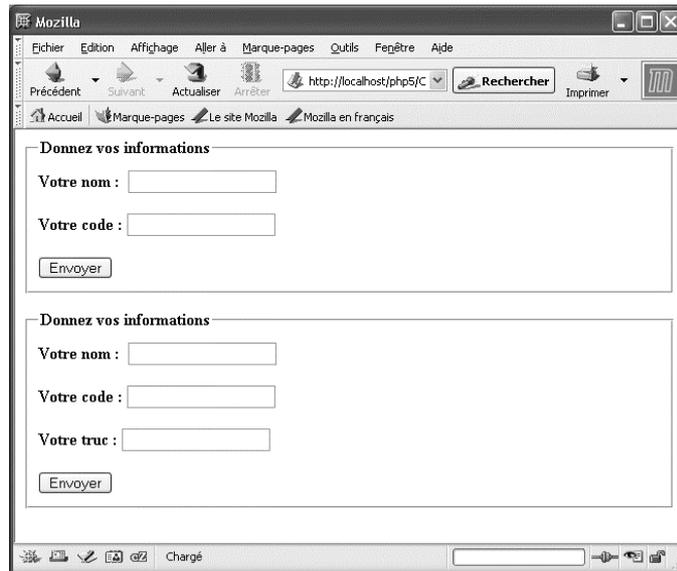


## Exercice 6

Créez un objet à partir de la classe `form2` de l'exercice 5, puis créez-en un clone. Modifiez certaines caractéristiques de cet objet, et affichez les deux formulaires obtenus.

```
<?php
include('ch9exo4.php');
class form2 extends form
{
    protected $coderadio;
    protected $codecase;
    public function __construct($action,$titre,$methode="post")
    {
        parent::__construct($action,$titre,$methode="post");
    }
    /*******
    public function setradio($name,$libelle,$value)
    {
        $this->coderadio.="<b>$libelle</b><input                type=\"radio\"
name=\"\$name\" value=\"\$value\" /><br />";
    }
    /*******
    public function setcase($name,$libelle,$value)
    {
        $this->codecase.="<b>$libelle</b><input                type=\"checkbox\"
name=\"\$name\" value=\"\$value\" /><br />";
    }
    /*******
    public function getform()
    {
        $this->code="";
        $this->code.=$this->codeinit;
        $this->code.=$this->codetext;
        $this->code.=$this->coderadio;
        $this->code.=$this->codecase;
        $this->code.=$this->codesubmit;
        $this->code.="</fieldset></form>";
        echo $this->code;
    }
}
/*******
$myform = new form2("traitementb.php","Donnez vos
informations","post");
$myform->settext("nom","Votre nom : &nbsp;");
$myform->settext("code","Votre code : ");
$myform->setsubmit();
$myform->getform();
$myclone = clone $myform;
$myclone->settext("truc","Votre truc : ");
$myclone->getform();
?>
```

Le résultat affiché est le suivant :



## Exercice 7

Créez une classe abstraite représentant une personne. Elle déclare les propriétés `nom` et `prénom` et un constructeur. Créez une classe `client` dérivée de la classe `personne` en y ajoutant la propriété `adresse` et une méthode `setcoord()` qui affiche les coordonnées complètes de la personne. Créez une classe `électeur` dérivée de la même classe abstraite, et ajoutez-y deux propriétés `bureau_de_vote` et `vote`, ainsi qu'une méthode `avoter()`, qui enregistre si une personne a voté dans la propriété `vote`.

Les propriétés `nom` et `prenom` qui sont déclarées `private` dans la classe abstraite (repères ❶ et ❷) sont redéfinies comme `public` dans la classe `electeur` (repères ❸ et ❹) car sinon nous ne pourrions pas les lire lors du contrôle du vote (repères ❺ et ❻) car la classe `electeur` ne possède pas de méthode `getinfo()`.

```
<?php
//Classe personne
abstract class personne
{
    private $nom; ←❶
    private $prenom; ←❷
    abstract protected function __construct() ;
}
//Classe client
class client extends personne
{
    private $adresse;
    public function __construct($nom, $prenom, $adresse)
    {
        $this->nom=$nom;
        $this->prenom=$prenom;
        $this->adresse=$adresse;
    }
}
```

```

    }
    public function getcoord()
    {
        $info="Le client $this->prenom $this->nom habite $this-
>adresse <br />";
        return $info;
    }
}
//Classe electeur
class electeur extends personne //
{
    public $nom; ← ③
    public $prenom; ← ④
    public $bureau_de_vote;
    public $vote;
    function __construct ($nom, $prenom, $bureau_de_vote)
    {
        $this->nom=$nom;
        $this->prenom=$prenom;
        $this->bureau_de_vote=$bureau_de_vote;
    }
    public function avoter()
    {
        $this->vote=TRUE;
    }
}
//Création d'objets
$client1 = new client("Delmas","Jacquou","Bordeaux");
echo "<h4>", $client1->getcoord()," </h4>";
$selecteur1 = new electeur("Tinitin","Milan","Brussel 5");
//L'électeur vote
$selecteur1->avoter();
//Controle du vote
if($selecteur1->vote)
{echo "L'électeur $selecteur1->prenom $selecteur1->nom inscrit au
bureau $selecteur1->bureau_de_vote a voté <br />";} ← ⑤
else
{echo "L'électeur $selecteur1->prenom $selecteur1->nom inscrit au
bureau $selecteur1->bureau_de_vote peut encore voter <br />";} ← ⑥
?>

```

## Exercices du chapitre 10

### Exercice 1

Créez une image de 500 × 300 pixels avec une couleur de fond rouge. Écrivez un texte de bienvenue en blanc avec une police PHP.

```

<?php
header ("Content-type: image/png");
$idimg=imagecreate(500,300);

```

```

$fond=imagecolorallocate($idimg,255,0,0);
$blanc=imagecolorallocate($idimg,255,255,255);
$texte="Bienvenus chez PHP 5";
imagestring($idimg,5,160,140,$texte,$blanc);
imagepng($idimg,"bienvenu.png");
imagepng($idimg);
imagedestroy($idimg);
?>

```

## Exercice 2

Créez une image de  $400 \times 200$  pixels avec un fond transparent. Dessinez une suite de rectangles emboîtés de couleurs différentes.

```

<?php
header ("Content-type: image/png");
$idimg=imagecreate(400,200);
$fond=imagecolorallocate($idimg,150,160,170);
$id=imagecolortransparent($idimg,$fond);
//Création d'un tableau des couleurs
$stabcolor[]=imagecolorallocate($idimg,255,0,255);
$stabcolor[]=imagecolorallocate($idimg,255,255,255);
$stabcolor[]=imagecolorallocate($idimg,0,0,255);
$stabcolor[]=imagecolorallocate($idimg,0,255,0);
$stabcolor[]=imagecolorallocate($idimg,255,0,0);
$stabcolor[]=imagecolorallocate($idimg,0,0,0);
//Dessin des rectangles
for($i=0,$j=0,$k=0;$i<200,$j<100;$i+=40,$j+=20,$k++)
{
    imagerectangle($idimg,200-$i,100-$j,200+$i,100+$j,$stabcolor[$k]);
}
imagepng($idimg,"rectangle.png");
imagepng($idimg);
imagedestroy($idimg);
?>

```

## Exercice 3

Créez une image de  $800 \times 600$  pixels avec une couleur de fond verte. Tracez un trapèze isocèle rempli de jaune, et écrivez le mot « trapèze » au centre.

```

<?php
header ("Content-type: image/gif");
$idimg=imagecreate(800,600);
$vert=imagecolorallocate($idimg,0,255,0);
$jaune=imagecolorallocate($idimg,255,255,0);
//Dessin des rectangles
//Coordonnées du quadrilatère
$stab=array (200,150,600,150,700,450,100,450);
//Tracé du quadrilatère
imagefilledpolygon($idimg,$stab,4,$jaune);
$texte="TRAPEZE";
imagestring($idimg,5,360,296,$texte,$vert);

```

```

imagegif($idimg, "trapeze.gif");
imagegif($idimg);
imagedestroy($idimg);
?>

```

## Exercice 4

Créez une image de  $601 \times 601$  pixels avec un fond transparent. Déterminez le centre O de l'image, et tracez des cercles concentriques centrés en O avec des rayons variant de 30 pixels jusqu'au bord de l'image. Attribuez à chaque cercle une couleur différente.

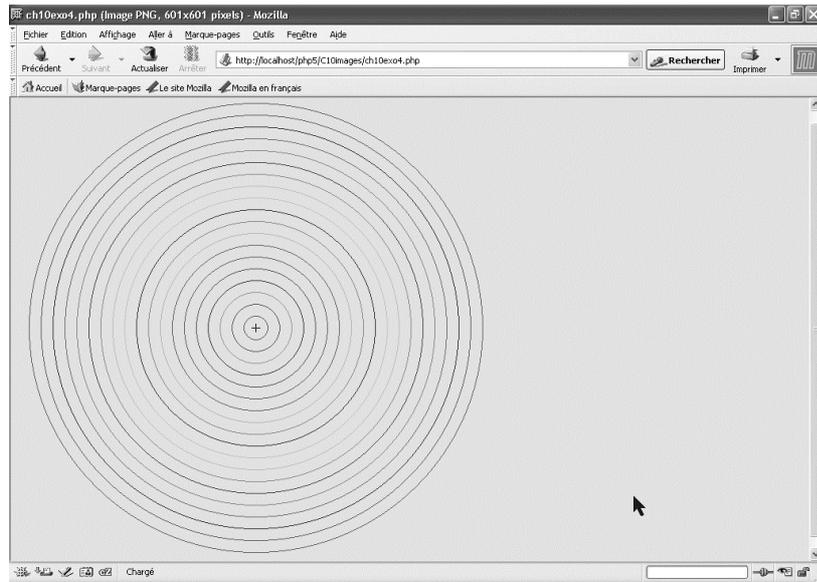
Comme nous devons tracer vingt cercles, nous créons un tableau de couleurs aléatoires à l'aide d'une boucle (repère ❶). Les coordonnées du centre sont obtenues en utilisant les fonctions `imagesx()` et `imagesy()` qui fournissent les dimensions de l'image (repères ❷ et ❸). Si nous changeons la taille de l'image nous aurons ainsi toujours son centre. Le tracé des cercles est réalisé au moyen d'une boucle double, la variable `$i` désignant le rayon et la variable `$j` le numéro de la couleur.

```

<?php
header ("Content-type: image/png");
$idimg=imagecreate(601,601);
$fond=imagecolorallocate($idimg,225,5,5);
imagecolortransparent($idimg,$fond);
$noir=imagecolorallocate($idimg,0,0,0);
//Création d'un tableau de 20 couleurs aléatoires
for($i=0;$i<20;$i++) ←❶
{
    $R= rand(0,255);
    $V= rand(0,255);
    $B= rand(0,255);
    $tabcolor[]=imagecolorallocate($idimg,$R,$V,$B);
}
//Coordonnées du centre de l'image
$XO=imagesx($idimg)/2; ←❷
$YO=imagesy($idimg)/2; ←❸
//Tracé du centre
imageline($idimg,$XO-5,$YO,$XO+5,$YO,$noir);
imageline($idimg,$XO,$YO-5,$XO,$YO+5,$noir);
//Tracé des cercles
for($i=0,$j=0;$i<600,$j<20;$i+=30,$j++) ←❹
{
    imageellipse($idimg,$XO,$YO,$i,$i,$tabcolor[$j]);
}
imagepng($idimg, "cercles.png");
imagepng($idimg);
imagedestroy($idimg);
?>

```

Le résultat obtenu est le suivant :

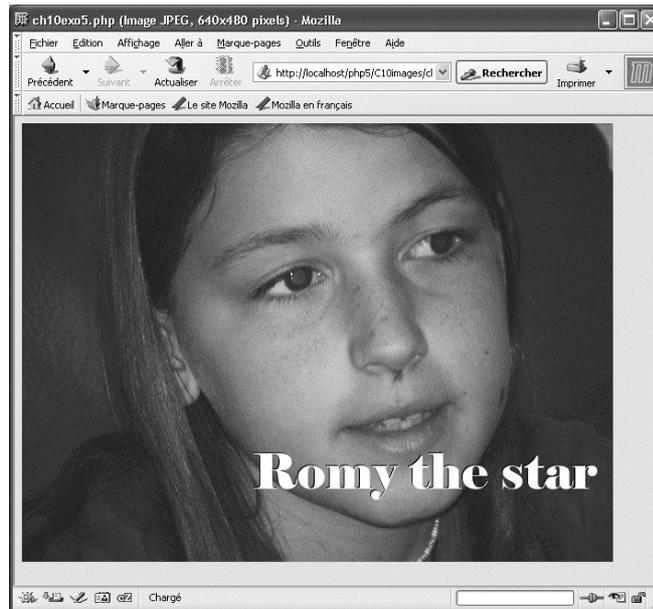


## Exercice 5

Créez une image à partir d'un fichier JPEG existant sur votre poste. Écrivez une légende de votre choix, d'abord en noir puis dans une autre couleur, en la décalant de 1 pixel en X et en Y afin de créer un effet d'ombre.

```
<?php
header ("Content-type: image/jpeg");
$idimg=imagecreatefromjpeg("romy.jpg");
$fond=imagecolorallocate($idimg,225,5,5);
$blanc=imagecolorallocate($idimg,255,255,255);
$noir=imagecolorallocate($idimg,0,0,0);
$titre="Romy the star";
imaggottext($idimg,40,0,249,399,$noir,"Elephnt.ttf",$titre);
imaggottext($idimg,40,0,250,400,$blanc,"Elephnt.ttf",$titre);
imagejpeg($idimg,"romy2.jpg");
imagejpeg($idimg);
imagedestroy($idimg);
?>
```

Le texte est affiché avec une police TrueType. Notez que le deuxième (en blanc) affichage recouvre le premier (en noir). Le résultat affiché est le suivant :



## Exercice 6

Créez une image de  $1\,024 \times 768$  pixels. Tracez la fonction  $f(x) = x^2$ , avec  $x$  compris entre  $-50$  et  $+50$ , et tracez les axes. Le tracé doit occuper la plus grande surface possible de l'image.

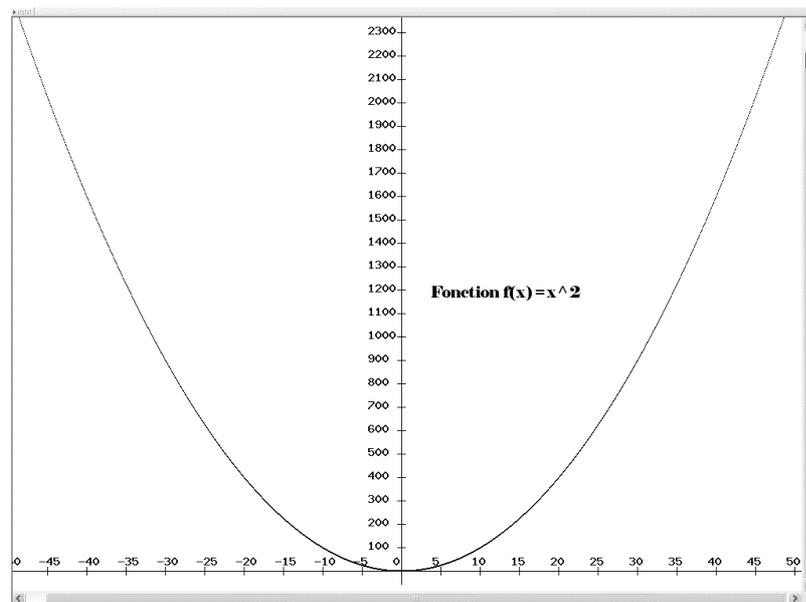
```
<?php
header ("Content-type: image/png");
$dimimg=imagecreate(1024,768);
$blanc=imagecolorallocate($dimimg,255,255,255);
$noir=imagecolorallocate($dimimg,0,0,0);
//Tracé des axes
imageline($dimimg,0,750,1024,750,$noir);
imageline($dimimg,512,768,512,0,$noir);
//Graduations des abscisses
for($x=-500;$x<=500;$x+=50)
{
    $X=512+$x;
    imageline($dimimg,$X,745,$X,755,$noir);
    $texte=(string) ($x/10);
    imagestring($dimimg,5,$X-10,730,$texte,$noir);
}
//Graduations des ordonnées
for($y=30;$y<=750;$y+=30)
{
    $Y=750-$y;
    imageline($dimimg,507,$Y,517,$Y,$noir);
    $texte=(string) ($y*10/3);
    imagestring($dimimg,5,470,$Y-10,$texte,$noir);
}
//Tracé de la courbe point par point
for($x=-500;$x<500;$x+=1)
{
    $X=512+$x;
    $Y=(250000-$x*$x)*0.003;
```

```

imagesetpixel($idimg,$X,$Y,$noir);
//Doublement du trait pour une meilleure visibilité
imagesetpixel($idimg,$X,$Y+1,$noir);
}
//Affichage du titre
$titre="Fonction f(x)=x^2";
imagettftext($idimg,15,0,550,400,$noir,"Elephnt.ttf",$titre);
imagepng($idimg,"carre.png");
imagepng($idimg);
imagedestroy($idimg);
?>

```

Le résultat obtenu est le suivant :



## Exercices du chapitre 11

### Exercice 1

Créer un fichier pour enregistrer la date de chaque connexion à votre site. Procéder ensuite à la lecture des données puis calculer des statistiques sur ces dates.

```

<?php
//Ecriture des dates
$idfile=fopen("connex.txt","a");
$date=time();
flock($idfile,2);
fwrite($idfile,$date);
flock($idfile,3);
fclose($idfile);
//Lecture des données
$idfile=fopen("connex.txt","r");
flock($idfile,1);
$i=1;

```

```

while($date=fgets($idfile,11))
{
    $tab[]=$date;
}
//Elimination des doublons
$tabstat=array_values(array_unique($tab));
//Affiche les dates et les heures des connexions
foreach($tabstat as $valeur)
{
    echo date("d/M H:m:s",$valeur),"<br />";
}
flock($idfile,3);
fclose($idfile);
?>

```

## Exercice 2

```

<?php
$nav = $_SERVER['HTTP_USER_AGENT'];
$explorer = 'MSIE';
$pos = strpos($nav, $explorer);
// Notez l'utilisation de ===. Un simple == ne donnerait pas le
résultat escompté
// car la lettre 'a' est à la position 0 (la première).
if ($pos !== false)
{
    echo "Navigateur Internet Explorer";
}
else
{
    echo "Navigateur Mozilla ou Netscape";
}
?>

```

## Exercice 3

En vous inspirant de l'exemple 11-5 créer un livre d'or qui n'affiche que les cinq derniers avis donnés par les visiteurs du site.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=iso-
8859-1">
<title>Le livre est d'or </title>
</head>
<body style="background-color: #ffcc00;">
<form action="<?php echo $PHP_SELF ?>" method="post" >
<fieldset>
<legend><b>Donnez votre avis sur PHP 5 ! </b></legend>
<b>Nom : &nbsp;<input type="text" name="nom" width="60" /> <br>
Mail : &nbsp;<input type="text" name="mail" width="60" /> <br>
Vos commentaires sur le site</b><br>
<textarea name="comment" rows="10" cols="50">Ici </textarea> <br>

```



```

        echo "<tr > <td colspan=\" 3 \">", stripslashes($tab5[$i][3])
        , "</td> </tr> ";
    }

    fclose($id_file);
}
echo "</tbody></table> ";
}
else{ echo "<h2>Donnez votre avis puis cliquez sur 'envoyer' !
</h2> ";}
?>
</body>
</html>

```

## Exercices du chapitre 12

### Exercice 1

Créer un formulaire de saisie des deux codes couleur préférés du visiteur du site pour la couleur de fond et le texte de la page. Les enregistrer dans deux cookies valables deux mois. À l'ouverture de la page d'accueil, récupérer ces valeurs et créer un style utilisant ces données.

```

<?php
if(!isset($_COOKIE['fond']) AND !isset($_COOKIE['texte'])) )
{
    $fond=$_POST['fond'];
    $texte=$_POST['texte'];
    $expir=time() + 2*30*24*3600;
    setcookie("fond",$fond,$expir);
    setcookie("texte",$texte,$expir);
}
else
{
    $fond=$_COOKIE['fond'];
    $texte=$_COOKIE['texte'];
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>Couleurs du site</title>
<style type="text/css" >
<!--
body{background-color: <?php echo $fond ?> ; color: <?php echo
$texte ?> ;}
legend{font-weight:bold;font-family:cursive;}

```







```

//En pratique ils proviennent d'une base de données
$login="machin";
$motpass="123456";
//Récupération des valeurs saisies
$nom=$_POST['nom'];
$pass=$_POST['pass'];
//Vérification
if($nom==$login AND $pass==$motpass)
{
    $expir=time() + 20;
    //Ecriture des cookies
    setcookie("nom",$nom,$expir);
    setcookie("pass",$pass,$expir);
    //Redirection vers la page à accès réservé
    echo "<script type=\"text/javascript\">
window.location='ch12exo3b.php' </script>";
}
else
{
    //Redirection vers la page de saisie du code
    echo "<script type=\"text/javascript\">
window.location='ch12exo3a.php' </script>";
}
?>

```

## Exercice 4

Enregistrer le nom de la page du site préférée du visiteur dans un cookie. Lors de sa connexion, il devra être redirigé automatiquement vers cette page.

```

<?php
//Détection du cookie et redirection éventuelle vers la page
préférée
if(isset($_COOKIE['mapage']))
{
    $page=$_COOKIE['mapage'];
    header("Location:$page");
}
//
if(isset($_POST['mapage']))
{
    setcookie("mapage",$_POST['mapage'],time()+10);
    header("Location:$page");
    echo "Vous avez choisi ",$_POST['mapage'],"<br />";
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>Choisissez votre page préférée</title>
</head>

```





```

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>Couleurs du site avec des sessions</title>
<style type="text/css" >
<!--
body{background-color: <?php echo $fond ?> ; color: <?php echo
$texte ?> ;}
legend{font-weight:bold;font-family:cursive;}
label{font-weight:bold;font-style:italic;}
-->
</style>
</head>
<body>
<p>Contenu de la page B
<a href="ch12exo7.php">Lien vers la page principale</a>
</p>
</body>
</html>

```

## Exercice 8

Transformer le script de l'exemple 12-5 (commande en ligne) en permettant les saisies à partir de pages différentes et en créant sur chacune un bouton provoquant l'affichage de l'ensemble du panier à chaque demande.

La modification est très simple à réaliser et peut permettre de créer autant de page que l'on désire, chacune étant spécialisée dans une gamme de produits. La première page dédiée à l'achat de livres (fichier ch12exo8.php) reprend l'essentiel de l'exemple 12.5 en ajoutant simplement un lien vers la deuxième page (repère ❶).

```

<?php
session_start();
//AJOUTER
if($_POST["envoi"]=="AJOUTER" && $_POST["code"]!=" &&
$_POST["article"]!=" && $_POST["prix"]!="")
{
    $code=$_POST["code"];
    $article= $_POST["article"];
    $prix= $_POST["prix"];
    $_SESSION['code'] []= $code;
    $_SESSION['article'] []= $article;
    $_SESSION['prix'] []= $prix;
}
//VERIFIER
if($_POST["envoi"]=="VERIFIER")
{
    echo "<table border=\"1\" >";
    echo " <tr><td colspan=\"3\"><b>Récapitulatif de votre
commande</b></td>";

```



```

<tr>
<th>article : </th>
<td><input type="text" name="article" /></td>
</tr>
<tr>
<th>prix :</th>
<td><input type="text" name="prix" /></td>
</tr>
<tr>
<td colspan="3">
<input type="submit" name="envoi" value="AJOUTER" />
<input type="submit" name="envoi" value="VERIFIER" />
<input type="submit" name="envoi" value="ENREGISTRER" />
<input type="submit" name="envoi" value="LOGOUT" />
</td>
</tr>
</tbody>
</table>
</fieldset>
</form>
</body>
</html>

```

La deuxième page dédiée à l'achat de disques (fichier ch12exo8b.php) est pratiquement identique, seuls les intitulés changent ainsi qu'un lien vers la première page (repère ❶). Le code PHP commun aux deux pages permet l'affichage des articles commandés et le récapitulatif de toute la commande à partir de n'importe quelle page. Comme indiqué plus haut nous pourrions donc créer sans difficulté, autant de pages que le site comporterait de familles d'articles.

```

<?php
session_start();
//AJOUTER
if($_POST["envoi"]=="AJOUTER"      &&      $_POST["code"]!="      &&
$_POST["article"]!="      && $_POST["prix"]!="")
{
    $code=$_POST["code"];
    $article= $_POST["article"];
    $prix= $_POST["prix"];
    $_SESSION['code'][]= $code;
    $_SESSION['article'][]= $article;
    $_SESSION['prix'][]= $prix;
}
//VERIFIER
if($_POST["envoi"]=="VERIFIER")
{
    echo "<table border=\"1\" >";
    echo "    <tr><td colspan=\"3\"><b>Récapitulatif de votre
commande</b></td>";
    echo "    <tr><th>&nbsp;&nbsp;&nbsp;code&nbsp;&nbsp;&nbsp;</th><th>&nbsp;&nbsp;&nbsp;article&nbsp;&nbsp;&nbsp;</
th><th>&nbsp;&nbsp;&nbsp;prix&nbsp;&nbsp;&nbsp;</th>";
    for($i=0;$i<count($_SESSION["code"]);$i++)
    {

```

```

        echo          "<tr>          <td>{$_SESSION['code']}[$i]</td>
<td>{$_SESSION['article']}[$i]
</td><td>{$_SESSION['prix']}[$i]</td>";
        $prixtotal+=$_SESSION['prix'][$i];
    }
    echo "<tr>          <td colspan=2>  PRIX  TOTAL  </td>  <td>".
sprintf("%01.2f", $prixtotal)."</td>";
    echo "</table>";
}
//ENREGISTRER
if($_POST["envoi"]=="ENREGISTRER")
{
    $idfile=fopen("commande.txt",w);
    for($i=0;$i<count($_SESSION["code"]);$i++)
    {
        fwrite($idfile,          $_SESSION["code"][$i]."          ;
".$_SESSION["article"][$i].". " ; ".$_SESSION["prix"][$i].". "; \n");
    }
    fclose($idfile);
}
//LOGOUT
if($_POST["envoi"]=="LOGOUT")
{
    session_unset();
    session_destroy();
    echo "<h3>La session est terminée</h3>";
}
$_POST["envoi"]="";
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
<title>Gestion de panier</title>
</head>
<body>
<div>
<a href="ch12exo8.php"><big>Page des livres</big></a><br />
</div>
<form action="<?=$_SERVER['PHP_SELF'] ?>" method="post"
enctype="application/x-www-form-urlencoded">
<fieldset>
<legend><b>Commande de disques</b></legend>
<table>
<tbody>
<tr>
<th>code : </th>
<td> <input type="text" name="code" /></td>
</tr>
<tr>
<th>article : </th>
<td><input type="text" name="article" /></td>
</tr>

```

```

<tr>
<th>prix :</th>
<td><input type="text" name="prix" /></td>
</tr>
<tr>
<td colspan="3">
<input type="submit" name="envoi" value="AJOUTER" />
<input type="submit" name="envoi" value="VERIFIER" />
<input type="submit" name="envoi" value="ENREGISTRER" />
<input type="submit" name="envoi" value="LOGOUT" />
</td>
</tr>
</tbody>
</table>
</fieldset>
</form>
</body>
</html>

```

## Exercices du chapitre 13

### Exercice 1

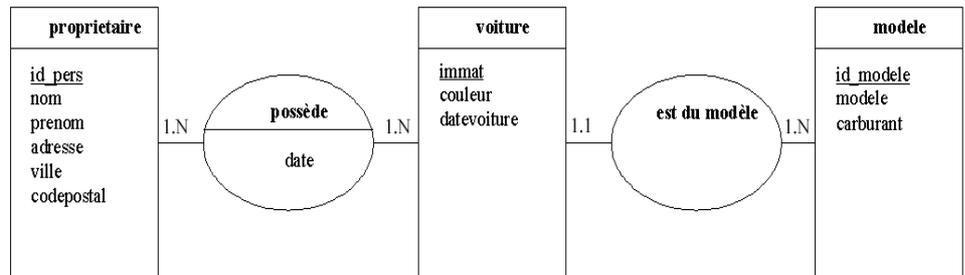
Créez le MCD d'une base de données voiture qui enregistre les certificats d'immatriculation des véhicules en circulation (carte grise). Elle doit répondre aux contraintes suivantes :

- Un véhicule est d'un modèle donné identifié par un numéro de type.
- Un véhicule peut avoir un ou plusieurs propriétaires simultanément (copropriété).
- Les recherches effectuées sur la base doivent permettre de retrouver, par exemple, tous les véhicules d'une personne, la ou les personnes propriétaires d'un véhicule dont on connaît l'immatriculation et tous les propriétaires d'un modèle de voiture donné.

Cardinalités :

- Un propriétaire peut avoir une ou plusieurs voitures : pour l'association possède la cardinalité du côté de l'entité propriétaire est donc 1.N.
- Une voiture peut être la propriété d'une ou plusieurs personnes : pour l'association possède la cardinalité du côté de l'entité voiture est donc 1.N.

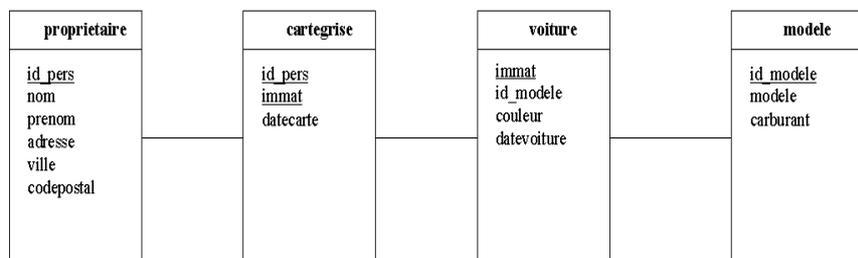
- Une voiture est d'un seul modèle : pour l'association `est du modèle` la cardinalité du côté de l'entité `voiture` est donc 1.1.
- A un modèle peuvent correspondre une ou plusieurs voitures : pour l'association `est du modèle` la cardinalité du côté de l'entité `modèle` est donc 1.N.



## Exercice 2

Créez le MLD de la base `voiture` à partir du MCD de l'exercice 1. Vérifiez la conformité du modèle par rapport aux formes normales.

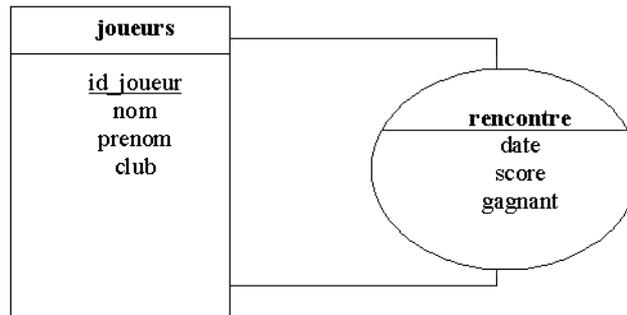
En application des règles nous obtenons le MLD suivant :



## Exercice 3

Créez le MCD d'une base de données `tournoi` permettant d'enregistrer les participants à un tournoi de tennis et l'ensemble des matches joués en trois sets au maximum. La base doit enregistrer les participants d'un match donné, ainsi que le gagnant et le score de chaque set.

La relation `rencontre` est réflexive.

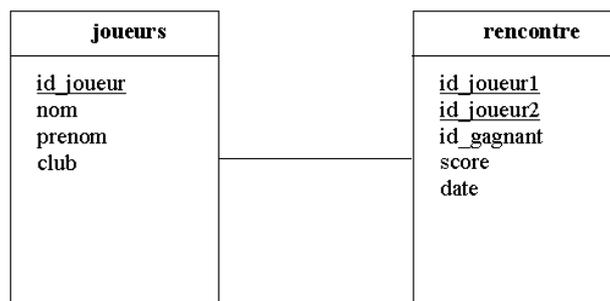


Il existerait également une autre solution en créant une entité `joueurs` et une entité `match` reliées par l'association `jouer`.

#### Exercice 4

Créez le MLD de la base `tournoi`, et vérifiez sa conformité.

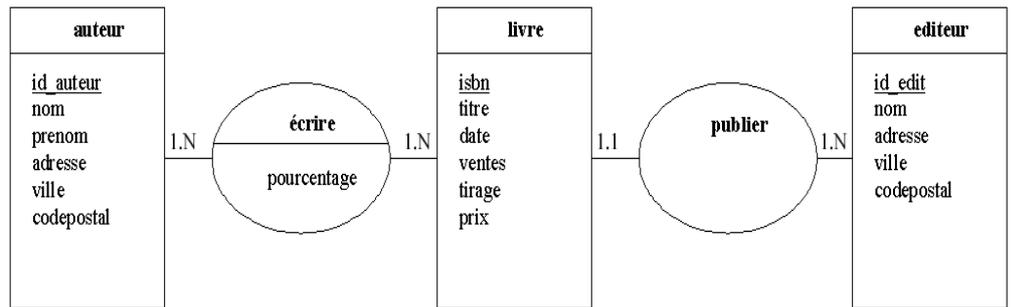
Le MLD correspondant est :



#### Exercice 5

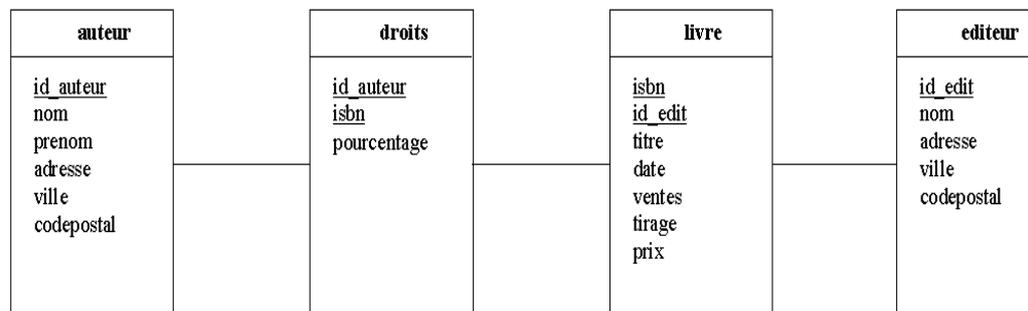
Créez le MCD d'une base permettant à un groupe de gérer les droits d'auteur des livres publiés par ses différentes maisons d'édition. Elle doit répondre aux contraintes suivantes :

- Un livre peut être écrit par un ou plusieurs auteurs. Un auteur peut écrire un ou plusieurs livres. Chaque auteur touche un pourcentage des droits totaux d'un livre en fonction de sa participation.
- Un livre est publié par un seul éditeur.



## Exercice 6

Créez le MLD correspondant à la base de l'exercice 5, et vérifiez sa conformité.



# Exercices du chapitre 14

Sauf indication contraire, toutes les opérations de sélection des exercices sont à réaliser sur la base magasin.

## Exercice 1

Créer une base nommée *voitures*. Créer ensuite les tables de la base *voitures* selon le modèle logique défini dans les exercices du chapitre 13. Omettre volontairement certaines colonnes et faire volontairement quelques erreurs de type de colonne. Une fois les tables créées, ajouter les colonnes manquantes et corriger les erreurs. Vérifier la structure de chaque table.

Nous utilisons bien sûr phpMyAdmin pour créer la base puis les tables.

- Création de la table personne (en omettant volontairement le champ codepostal)

```
CREATE TABLE `proprietaire` (  
  `id_pers` SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT ,  
  `nom` VARCHAR( 30 ) NOT NULL ,  
  `prenom` VARCHAR( 30 ) NOT NULL ,  
  `adresse` VARCHAR( 50 ) NOT NULL ,  
  `ville` VARCHAR( 40 ) NOT NULL ,  
  PRIMARY KEY ( `id_pers` )  
) TYPE = MYISAM ;
```

Nous ajoutons le champ codepostal «oublié» lors de la création de la table.

```
ALTER TABLE `proprietaire` ADD `codepostal` MEDIUMINT( 5 ) UNSIGNED NOT NULL ;
```

Nous modifions le type du champ `id_pers` pour avoir un éventail de valeurs plus grand.

```
ALTER TABLE `proprietaire` CHANGE `id_pers` `id_pers` MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT
```

- Création de la table cartegrise

```
CREATE TABLE `cartegrise` (  
  `id_pers` MEDIUMINT UNSIGNED NOT NULL ,  
  `immat` VARCHAR( 6 ) NOT NULL ,  
  `datecarte` DATE NOT NULL ,  
  PRIMARY KEY ( `id_pers` , `immat` )  
) ;
```

- Nous créons la table voiture.

```
CREATE TABLE `voitures` (  
  `immat` VARCHAR( 6 ) NOT NULL ,  
  `id_modele` VARCHAR( 10 ) NOT NULL ,  
  `couleur` ENUM( 'claire', 'moyenne', 'foncée' ) NOT NULL ,  
  `datevoiture` DATE NOT NULL ,  
  PRIMARY KEY ( `immat` )  
) ;
```

- Nous créons la table modele

```
CREATE TABLE `modele` (  
  `id_modele` VARCHAR( 10 ) NOT NULL ,  
  `modele` VARCHAR( 30 ) NOT NULL ,  
  `carburant` ENUM( 'essence', 'diesel', 'gpl', 'électrique' ) NOT NULL ,  
  PRIMARY KEY ( `id_modele` )  
) ;
```

## Exercice 2

Exporter les tables de la base voitures dans des fichiers SQL.

Nous obtenons les fichiers suivants :

- Le fichier `proprietaire.sql` :

```
-- phpMyAdmin SQL Dump
```

```

-- version 2.6.0-rc3
-- http://www.phpmyadmin.net
--
-- Serveur: localhost
-- Généré le : Mercredi 15 Décembre 2004 à 18:21
-- Version du serveur: 4.0.21
-- Version de PHP: 5.0.2
--
-- Base de données: `voitures`
--
-----
--
-- Structure de la table `proprietaire`
--
CREATE TABLE `proprietaire` (
  `id_pers` mediumint(8) unsigned NOT NULL auto_increment,
  `nom` varchar(30) NOT NULL default '',
  `prenom` varchar(30) NOT NULL default '',
  `adresse` varchar(50) NOT NULL default '',
  `ville` varchar(40) NOT NULL default '',
  `codepostal` mediumint(5) unsigned NOT NULL default '0',
  PRIMARY KEY (`id_pers`)
) TYPE=MyISAM AUTO_INCREMENT=1 ;

```

- Le fichier `cartegrise.sql` :

```

-- phpMyAdmin SQL Dump
-- version 2.6.0-rc3
-- http://www.phpmyadmin.net
--
-- Serveur: localhost
-- Généré le : Mercredi 15 Décembre 2004 à 18:26
-- Version du serveur: 4.0.21
-- Version de PHP: 5.0.2
--
-- Base de données: `voitures`
--
-----
--
-- Structure de la table `cartegrise`
--
CREATE TABLE `cartegrise` (
  `id_pers` mediumint(8) unsigned NOT NULL default '0',
  `immat` varchar(6) NOT NULL default '',
  `datecarte` date NOT NULL default '0000-00-00',
  PRIMARY KEY (`id_pers`,`immat`)
) TYPE=MyISAM;

```

- Le fichier `voiture.sql`

```

-- phpMyAdmin SQL Dump
-- version 2.6.0-rc3
-- http://www.phpmyadmin.net
--
-- Serveur: localhost
-- Généré le : Mercredi 15 Décembre 2004 à 18:27

```

```

-- Version du serveur: 4.0.21
-- Version de PHP: 5.0.2
--
-- Base de données: `voitures`
--
-- -----
--
-- Structure de la table `voitures`
--
CREATE TABLE `voiture` (
  `immat` varchar(6) NOT NULL default '',
  `id_modele` varchar(10) NOT NULL default '',
  `couleur` enum('claire','moyenne','foncée') NOT NULL default
'claire',
  `datevoiture` date NOT NULL default '0000-00-00',
  PRIMARY KEY (`immat`)
) TYPE=MyISAM;

```

- Le fichier modele.sql

```

-- phpMyAdmin SQL Dump
-- version 2.6.0-rc3
-- http://www.phpmyadmin.net
--
-- Serveur: localhost
-- Généré le : Mercredi 15 Décembre 2004 à 18:29
-- Version du serveur: 4.0.21
-- Version de PHP: 5.0.2
--
-- Base de données: `voitures`
--
-- -----
--
-- Structure de la table `modele`
--
CREATE TABLE `modele` (
  `id_modele` varchar(10) NOT NULL default '',
  `modele` varchar(30) NOT NULL default '',
  `carburant` enum('essence','diesel','gpl','électrique') NOT NULL
default 'essence',
  PRIMARY KEY (`id_modele`)
) TYPE=MyISAM;

```

### Exercice 3

Supprimer toutes les tables de la base voitures.

Le code SQL est le suivant :

```

DROP TABLE `proprietaire`
DROP TABLE `cartegrise`
DROP TABLE `voiture`
DROP TABLE `modele`

```

## Exercice 4

Recréer les tables de la base `voitures` en utilisant les fichiers SQL précédents.

Pour recréer avec phpMyAdmin, les tables détruites, choisir successivement la base, puis l'onglet « SQL », « Emplacement du fichier texte », « Parcourir » pour désigner l'emplacement du fichier `.sql`, et enfin « Exécuter ». Les tables sont alors recréées l'une après l'autre.

Dans l'exercice 2, nous avons également la possibilité d'exporter l'ensemble de la base `voitures` en un seul fichier `.sql`. Nous aurions obtenu alors le fichier `voitures.sql` suivant :

```
-- phpMyAdmin SQL Dump
-- version 2.6.0-rc3
-- http://www.phpmyadmin.net
--
-- Serveur: localhost
-- Généré le : Mercredi 15 Décembre 2004 à 18:41
-- Version du serveur: 4.0.21
-- Version de PHP: 5.0.2
--
-- Base de données: `voitures`
--
-- -----
--
-- Structure de la table `cartegrise`
--
CREATE TABLE `cartegrise` (
  `id_pers` mediumint(8) unsigned NOT NULL default '0',
  `immat` varchar(6) NOT NULL default '',
  `datecarte` date NOT NULL default '0000-00-00',
  PRIMARY KEY (`id_pers`,`immat`)
) TYPE=MyISAM;

-- -----
--
-- Structure de la table `modele`
--
CREATE TABLE `modele` (
  `id_modele` varchar(10) NOT NULL default '',
  `modele` varchar(30) NOT NULL default '',
  `carburant` enum('essence','diesel','gpl','électrique') NOT NULL
  default 'essence',
  PRIMARY KEY (`id_modele`)
) TYPE=MyISAM;

-- -----
--
-- Structure de la table `proprietaire`
--
CREATE TABLE `proprietaire` (
```

```

`id_pers` mediumint(8) unsigned NOT NULL auto_increment,
`nom` varchar(30) NOT NULL default '',
`prenom` varchar(30) NOT NULL default '',
`adresse` varchar(50) NOT NULL default '',
`ville` varchar(40) NOT NULL default '',
`codepostal` mediumint(5) unsigned NOT NULL default '0',
PRIMARY KEY (`id_pers`)
) TYPE=MyISAM AUTO_INCREMENT=1 ;
-----
--
-- Structure de la table `voiture`
--
CREATE TABLE `voiture` (
  `immat` varchar(6) NOT NULL default '',
  `id_modele` varchar(10) NOT NULL default '',
  `couleur` enum('claire','moyenne','foncée') NOT NULL default
'claire',
  `datevoiture` date NOT NULL default '0000-00-00',
  PRIMARY KEY (`immat`)
) TYPE=MyISAM;

```

## Exercice 5

Insérer des données dans la table propriétaire de la base voitures puis en vérifier la bonne insertion.

Exemple de code d'insertion :

```

INSERT INTO `proprietaire` ( `id_pers` , `nom` , `prenom` ,
`adresse` , `ville` , `codepostal` )
VALUES (
  '', 'Zouk', 'Julia', '56 Boulevard Nez', 'Paris', '75011'
);

```

## Exercice 6

Créer un fichier texte contenant une liste de modèles de voitures avec autant de données par ligne que de colonnes dans la table modele de la base voitures. Insérer ces données dans la base.

Exemple de fichier texte contenant des modèles: le fichier modele.txt

```

"17C92853AZ";"Citroën C5";"diesel"
"178524ER45";"Citroën Picasso";"essence"
"7499RF5679";"Renault Mégane Scénic";"diesel"
"33356677PO";"Peugeot 206";"électrique"
"563339GH56";"Citroën C3";"essence"
"83321TY455";"Renault Espace";"diesel"

```

Pour revoir la méthode d'insertion à partir d'un fichier texte avec phpMyAdmin, voir la page 382 et suivantes.

## Exercice 7

Créer un fichier Excel ou OpenOffice contenant une liste de modèles de voitures avec autant de données par ligne que de colonnes dans la table `modele`. L'enregistrer au format CSV et insérer les données dans la base.

La feuille du tableur à l'aspect type suivant :

	A	B	C
1	AZER67455T	Peugeot 307	essence
2	FHT55432GH	Renault Twingo	essence
3	DSQS455674	Renault Adventime	diesel
4	485228FGD7	Volkswagen Golf	diesel
5	ZER627864K	Ferrari GT 40	essence
6			
7			

L'insertion des données se fait selon la même procédure que celle utilisée pour un fichier texte. Après l'insertion la table `modele` a le contenu suivant :

id_modele	modele	carburant
17C92853AZ	Citroën C5	diesel
178524ER45	Citroën Picasso	essence
7499RF5679	Renault Mégane Scénic	diesel
33356677PO	Peugeot 206	électrique
563339GH56	Citroën C3	essence
83321TY455	Renault Espace	diesel
AZER67455T	Peugeot 307	essence
FHT55432GH	Renault Twingo	essence
DSQS455674	Renault Adventime	diesel
485228FGD7	Volkswagen Golf	diesel
ZER627864K	Ferrari GT 40	essence

## Exercice 8

Insérer des données dans les autres tables de la base voitures. Effectuer des mises à jour en modifiant certaines valeurs.

Trivial avec phpMyAdmin.

## Exercice 9

Dans la base `magasin`, sélectionner les articles dont le prix est inférieur à 1 500 €.

Requête SQL :

```
SELECT id_article, designation, prix
FROM article
WHERE prix <1500
```

## Exercice 10

Dans la base `magasin`, sélectionner les articles dont le prix est compris entre 100 et 500 €.

Requête SQL :

```
SELECT id_article, designation, prix
FROM article
WHERE prix
BETWEEN 100
AND 500
```

## Exercice 11

Dans la base `magasin`, sélectionner tous les articles de marque Nikon (dont la désignation contient ce mot).

Requête SQL :

```
SELECT id_article, designation, prix
FROM article
WHERE designation LIKE '%Nikon%'
```

## Exercice 12

Dans la base `magasin`, sélectionner tous les caméscopes, leur prix et leur référence.

Requête SQL :

```
SELECT id_article, designation, prix
FROM article
WHERE designation LIKE '%caméscope%'
```

On peut également écrire :

```
SELECT id_article, designation, prix
FROM article
WHERE categorie = 'vidéo'
```

## Exercice 13

Dans la base `magasin`, sélectionner tous les produits de la catégorie `informatique` et afficher leur code, leur désignation et leur prix par ordre décroissant de prix.

Requête SQL :

```
SELECT id_article, designation, prix
FROM article
WHERE categorie = 'informatique '
ORDER BY prix DESC
```

## Exercice 14

Dans la base `magasin`, sélectionner tous les clients de moins de 40 ans et ordonner les résultats par ville en ordre alphabétique.

Requête SQL :

```
SELECT nom, prenom, age, ville
FROM CLIENT WHERE age <40
ORDER BY ville ASC
```

## Exercice 15

Dans la base magasin, calculer le prix moyen de tous les articles.

Requête SQL :

```
SELECT avg( prix )
FROM article
```

## Exercice 16

Dans la base magasin, calculer le nombre d'e-mails non NULL et distincts l'un de l'autre.

Requête SQL :

```
SELECT count( DISTINCT mail) FROM client
```

## Exercice 17

Dans la base magasin, afficher les coordonnées des clients ayant la même adresse (même adresse et même ville).

Requête SQL :

```
SELECT nom, prenom, adresse, ville, mail FROM client WHERE adresse='75
Bd Hochimin' AND ville='Lille'
```

Avec PHP, si l'adresse et la ville étaient contenues respectivement dans les variables \$adresse et \$ville on aurait le code suivant :

```
SELECT      nom, prenom, adresse, ville, mail      FROM      client      WHERE
adresse='$adresse' AND ville='$ville'
```

## Exercice 18

Dans la base magasin, sélectionner tous les articles commandés par chaque client.

Requête SQL :

```
SELECT nom, prenom, article.id_article, designation
FROM `client`, commande, article, ligne
WHERE client.id_client=commande.id_client
AND ligne.id_comm=commande.id_comm
AND ligne.id_article=article.id_article
ORDER BY nom
```

## Exercice 19

Dans la base magasin, sélectionner tous les clients dont le montant d'une commande dépasse 1 500 €.

Requête SQL :

```
SELECT nom, prenom, ligne.id_comm, sum(prixunit*quantite) AS 'total'
FROM client, ligne, commande
WHERE ligne.id_comm=commande.id_comm
AND commande.id_client=client.id_client
GROUP BY ligne.id_comm
HAVING sum(prixunit*quantite)>1500
```

## Exercice 20

Dans la base magasin, sélectionner tous les clients dont le montant total de toutes les commandes dépasse 5 000 €.

Requête SQL :

```
SELECT client.id_client, ligne.id_comm, sum(prixunit*quantite)
FROM client, ligne, commande
WHERE ligne.id_comm=commande.id_comm
AND commande.id_client=client.id_client
GROUP BY client.id_client
HAVING sum(prixunit*quantite)>5000
```

## Exercice 21

Dans la base voitures, sélectionner tous les véhicules d'une personne donnée.

Requête SQL : Nous cherchons par exemple tous les véhicules de M. Algout.

```
SELECT cartegrise.immat, modele, proprietaire.id_pers
FROM voiture, modele, proprietaire, cartegrise
WHERE proprietaire.nom='Algout'
AND proprietaire.id_pers=cartegrise.id_pers
AND cartegrise.immat=voiture.immat
AND voiture.id_modele=modele.id_modele
```

## Exercice 22

Dans la base voitures, sélectionner toutes les personnes ayant le même modèle de voiture.

Requête SQL : Nous cherchons par exemple tous les propriétaires de véhicules de type « Picasso ».

```
SELECT
proprietaire.nom, proprietaire.prenom, modele.modele, modele.carburant
FROM voiture, modele, proprietaire, cartegrise
WHERE modele LIKE '%Picasso'
```

```
AND voiture.id_modele=modele.id_modele
AND cartegrise.immat=voiture.immat
AND proprietaire.id_pers=cartegrise.id_pers
```

## Exercice 23

Dans la base `voitures`, sélectionner tous les véhicules ayant plusieurs copropriétaires.

Requête SQL :

```
SELECT cartegrise.immat FROM cartegrise
GROUP BY immat
HAVING count(*) >1
```

# Exercices du chapitre 15

Tous les exercices ci-dessous portent sur la base de données `voitures` créée aux chapitres 13 et 14.

## Exercice 1

Créer un script permettant d'afficher le contenu de la table `modele` dans un tableau HTML. Les résultats doivent être triés par marque.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
<title>Lecture de la table modele</title>
<style type="text/css" >
table {border-style:double;border-width: 3px;border-color:
blue;background-color: #FFCCEE;}
</style>
</head>
<body>
<?php
include("connex.inc.php");
$idcom=connex("voitures","myparam");
$requete="SELECT * FROM modele ORDER BY modele";
$result=@mysql_query($requete,$idcom);
if(!$result)
{
echo "Lecture impossible";
}
else
{
$nbcol=mysql_num_fields($result);
$nbmod=mysql_num_rows($result);
```

```

echo "<h3> Tous les $nbmod modèles de voitures</h3>";
echo "<table border=\"1\">";
echo " <th>Type du modele</th> <th>Marque et modèle</th>
<th>Carburant</th>";
while($ligne=mysql_fetch_array($result,MYSQL_NUM))
{
    echo "<tr>";
    foreach($ligne as $valeur)
    {
        echo "<td> $valeur </td>";
    }
    echo "</tr>";
}
echo "</table>";
}
?>
</body>
</html>

```

## Exercice 2

Créer un formulaire permettant l'insertion de nouvelles données dans la table modele.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<html>
<head>
<title>Saisissez les caractéristiques du modèle</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
</head>
<body>
<form action= "<?php echo $_SERVER['PHP_SELF'];?>" method="post"
enctype="application/x-www-form-urlencoded">
<fieldset>
<legend><b>Vos coordonnées</b></legend>
<table>
<tr><td>Code du type : </td><td><input type="text" name="id_modele"
size="40" maxlength="10"/></td></tr>
<tr><td>Marque et modèle : </td><td><input type="text"
name="modele" size="40" maxlength="30"/></td></tr>
<tr><td>Carburant : </td>
<td><select name="carburant">
<option value="essence">Essence</option>
<option value="diesel">Diesel</option>
<option value="électrique">Electrique</option>
</select></td>
</tr>
<tr>
<td><input type="reset" value=" Effacer "></td>
<td><input type="submit" value=" Envoyer "></td>
</tr>
</table>
</fieldset>

```

```

</form>
<?php
include('connex.inc.php');
if(!empty($_POST['id_modele'])&& !empty($_POST['modele'])&&
!empty($_POST['carburant']))
{
    $id_modele=mysql_escape_string($_POST['id_modele']);
    $modele=mysql_escape_string($_POST['modele']);
    $carburant=mysql_escape_string($_POST['carburant']);
    //Requête SQL
    $requete="INSERT INTO modele
VALUES('$id_modele','$modele','$carburant')";
    $idcom=connex('voitures','myparam');
    $result=mysql_query($requete,$idcom);

    if(!$result)
    {
        echo mysql_errno();
        echo mysql_error();
        echo "<script type=\"text/javascript\">
alert('Erreur : ".mysql_error()."')</script>";
    }
    else
    {
        echo "<script type=\"text/javascript\">";
        echo "alert('Le modèle est enregistré ')</script>";
    }
}
else {echo "<h3>Formulaire à compléter!</h3>";}
?>
</body>
</html>

```

### Exercice 3

Créer un formulaire permettant l'insertion simultanée des coordonnées d'une personne dans les tables propriétaire et cartegrise. Il doit contenir les zones de saisie des coordonnées de la personne et la liste des modèles d'une marque créée dynamiquement à partir de la saisie de la marque.

Notez qu'il s'agit bien d'un exercice d'insertion de données dans plusieurs tables, et que la gestion de la situation réelle serait plus complexe car il faudrait s'assurer que la personne ne figure pas déjà dans la table « propriétaire », donc parcourir toute cette table. L'insertion des données se fait également dans la table « voiture »

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<html>
<head>
<title>Saisissez les caractéristiques du modèle</title>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
</head>
<body>
<form action= "<?php echo $_SERVER['PHP_SELF'];?>" name="form1"
method="post"
enctype="application/x-www-form-urlencoded">
<fieldset>
<legend><b>Enregistrement d'un véhicule</b></legend>
<table>
<tr colspan="2"><td><b>Propriétaire</b></td></tr>
<tr><td>Nom : </td><td><input type="text" name="nom" size="40"
maxlength="10"/></td></tr>
<tr><td>Prénom : </td><td><input type="text" name="prenom"
size="40" maxlength="30"/></td></tr>
<tr><td>Adresse : </td><td><input type="text" name="adresse"
size="40" maxlength="30"/></td></tr>
<tr><td>Ville : </td><td><input type="text" name="ville" size="40"
maxlength="30"/></td></tr>
<tr><td>Code postal : </td><td><input type="text" name="codepostal"
size="40" maxlength="5"/></td></tr>
<tr colspan="2"><td><b>Modèle</b></td></tr>
<tr><td>Marque : </td><td><input type="text" name="marque"
size="40" maxlength="30" />
<input type="submit" name="cherchemod" value=" Chercher les
modèles " /></td>
</tr>
<?php
if(isset($_POST['cherchemod']))
{
include_once('connex.inc.php');
$marque=mysql_escape_string($_POST['marque']);
//Requête SQL
$requete="SELECT DISTINCT id_modele,modele FROM modele WHERE
modele LIKE '%$marque%'";
$idcom=connex('voitures','myparam');
$result=mysql_query($requete,$idcom);
echo "<tr><td>Les modèles</td> <td><select name=\"id_modele\">";
while($ligne=mysql_fetch_array($result,MYSQL_NUM))
{
echo " <option value=\"$ligne[0]\">$ligne[1]</option>";
}
echo "</select></td></tr>";
}
?>
<tr><td>Carburant : </td>
<td><select name="carburant">
<option value="essence">Essence</option>
<option value="diesel">Diesel</option>
<option value="électrique">Electrique</option>
</select></td>
</tr>
<tr colspan="2"><td><b>Voiture</b></td></tr>
<tr>
<td>Numéro d'immatriculation</td>
<td><input type="text" name="immat" /></td>
</tr>
<tr><td>Couleur : </td>

```

```

<td><select name="couleur">
<option value="claire">Claire</option>
<option value="moyenne">Moyenne</option>
<option value="foncée">Foncée</option>
</select></td>
</tr>
<tr>
<td>Date 1ere immatriculation AAAA-MM-JJ</td>
<td><input type="text" name="datevoiture" /></td>
</tr>
<td>Date de la carte grise AAAA-MM-JJ</td>
<td><input type="text" name="datecarte" /></td>
</tr>
<tr>
<td><input type="reset" value=" Effacer " /></td>
<td><input type="submit" value="ENREGISTRER" name="enreg" /></td>
</tr>
</table>
</fieldset>
</form>
<?php
include_once('connex.inc.php');
if(isset($_POST['enreg']))
{
//Récupération des valeurs du formulaire
$id_modele=mysql_escape_string($_POST['id_modele']);
$carburant=mysql_escape_string($_POST['carburant']);
$immat=mysql_escape_string($_POST['immat']);
$couleur=mysql_escape_string($_POST['couleur']);
$datevoiture=mysql_escape_string($_POST['datevoiture']);
$datecarte=mysql_escape_string($_POST['datecarte']);
$nom=mysql_escape_string($_POST['nom']);
$prenom=mysql_escape_string($_POST['prenom']);
$adresse=mysql_escape_string($_POST['adresse']);
$ville=mysql_escape_string($_POST['ville']);
$codepostal=mysql_escape_string($_POST['codepostal']);
//Connexion
$idcom=connex('voitures','myparam');
//Requête SQL: insertions dans la table voiture
$requete="INSERT INTO voiture
VALUES('$immat','$id_modele','$couleur','$datevoiture')";
$result1=mysql_query($requete,$idcom);
//Requête SQL: insertions dans la table proprietaire
$requete="INSERT INTO proprietaire
VALUES('','$nom','$prenom','$adresse','$ville','$codepostal')";
$result2=mysql_query($requete,$idcom);
$id_pers=mysql_insert_id($idcom);
//Requête SQL: insertions dans la table cartegrise
$requete="INSERT INTO cartegrise
VALUES('$id_pers','$immat','$datecarte')";
$result3=mysql_query($requete,$idcom);

```

```

//Requête SQL: insertions dans la table
if(!$result1)
{
    echo mysql_errno();
    echo mysql_error();
    echo "<script type=\"text/javascript\">
    alert('Erreur : ".mysql_error()."')</script>";
}
else
{
    echo "<script type=\"text/javascript\">";
    echo "alert('La carte grise est enregistrée ')</script>";
}
}
else {"<h3>Formulaire à compléter!</h3>";}
?>
</body>
</html>

```

## Exercice 4

Créer un formulaire de recherche permettant de retrouver tous les propriétaires d'un type de véhicule de marque et de modèle donnés. Afficher les résultats sous forme de tableau HTML.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<html>
<head>
<title>Recherche des propriétaires d'un modèle</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
</head>
<body>
<form action="<?php echo $_SERVER['PHP_SELF'];?>" name="form1"
method="post"
enctype="application/x-www-form-urlencoded">
<fieldset>
<legend><b>Choisir le modèle</b></legend>
<table>
<tr><td>Marque et modèle</td><td><input type="text" name="modele"
maxlength=" " /></td></tr>
<tr><td><input type="submit" value="Chercher" /></td>
</tr>
</table>
</fieldset>
</form>
<?php
if(isset($_POST['modele']))
{
    include_once('connex.inc.php');
    $modele=mysql_escape_string($_POST['modele']);
    //Requête SQL
    $requete="SELECT proprietaire.nom,proprietaire.prenom FROM
voiture,modele,proprietaire,cartegrise WHERE modele LIKE
'%"$modele%" AND voiture.id_modele=modele.id_modele AND

```

```

cartegrise.immat=voiture.immat AND
proprietaire.id_pers=cartegrise.id_pers";
$idcom=connex('voitures', 'myparam');
$result=mysql_query($requete,$idcom);
echo "<table border=\"1\" >";
while($ligne=mysql_fetch_array($result,MYSQL_NUM))
{
    echo " <tr><td>$ligne[0]</td><td> $ligne[1]</td>";
}
echo "</table>";
}
else {"<h3>Formulaire à compléter!</h3>";}
?>
</body>
</html>

```

## Exercice 5

Créer un formulaire de recherche permettant de retrouver tous les véhicules possédés par une personne donnée. Afficher les résultats sous forme de tableau HTML.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<html>
<head>
<title>Recherche des voitures d'une personne</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
</head>
<body>
<form action= "<?php echo $_SERVER['PHP_SELF'];?>" name="form1"
method="post"
enctype="application/x-www-form-urlencoded">
<fieldset>
<legend><b>Coordonnées de la personne</b></legend>
<table>
<tr><td>Nom : </td><td><input type="text" name="nom" /></td></tr>
<tr><td>Prénom : </td><td><input type="text" name="prenom"
/></td></tr>
<tr><td><input type="submit" value="Chercher" /></td>
</tr>
</table>
</fieldset>
</form>
<?php
if(isset($_POST['nom']) && isset($_POST['prenom']))
{
    include_once('connex.inc.php');
    $nom=mysql_escape_string($_POST['nom']);
    $prenom=mysql_escape_string($_POST['prenom']);
    //Requête SQL
    $requete="SELECT voiture.immat,modele.modele FROM
voiture,modele,proprietaire,cartegrise WHERE
proprietaire.nom='$nom' AND proprietaire.prenom='$prenom' AND
proprietaire.id_pers=cartegrise.id_pers AND

```

```

voiture.id_modele=modele.id_modele AND
cartegrise.immat=voiture.immat";
$idcom=connex('voitures', 'myparam');
$result=mysql_query($requete,$idcom);
echo "<h3>Liste des véhicules de $prenom $nom </h3>";
echo "<table border=\"1\" >";
while($ligne=mysql_fetch_array($result,MYSQL_NUM))
{
echo " <tr><td>&nbsp;",$ligne[0], " 75 &nbsp;</td><td>&nbsp;",$
$ligne[1], "&nbsp;</td>";
}
echo "</table>";
}
else {"<h3>Formulaire à compléter!</h3>";}
?>
</body>
</html>

```

Le formulaire de recherche et un exemple de résultats obtenus :



## Exercice 6

Réécrire entièrement le code de l'exercice 5 en récupérant tous les résultats dans des objets et en manipulant leurs propriétés.

Les seules modifications sont dans l'utilisation de la fonction `mysql_fetch_object()` (repère ❶) et dans la lecture des résultats (repère ❷)

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<html>
<head>
<title>Recherche des voitures d'une personne</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
</head>
<body>
<form action= "<?php echo $_SERVER['PHP_SELF'];?>" name="form1"
method="post"
enctype="application/x-www-form-urlencoded">

```

```

<fieldset>
<legend><b>Coordonnées de la personne</b></legend>
<table>
<tr><td>Nom : </td><td><input type="text" name="nom" /></td></tr>
<tr><td>Prénom : </td><td><input type="text" name="prenom"
/></td></tr>
<td><input type="submit" value="Chercher" /></td>
</tr>
</table>
</fieldset>
</form>
<?php
if(isset($_POST['nom']) && isset($_POST['prenom']))
{
    include_once('connex.inc.php');
    $nom=mysql_escape_string($_POST['nom']);
    $prenom=mysql_escape_string($_POST['prenom']);
    //Requête SQL
    $requete="SELECT voiture.immat,modele.modele FROM
voiture,modele,proprietaire,cartegrise WHERE
proprietaire.nom='$nom' AND proprietaire.prenom='$prenom' AND
proprietaire.id_pers=cartegrise.id_pers AND
voiture.id_modele=modele.id_modele AND
cartegrise.immat=voiture.immat";
    $idcom=connex('voitures','myparam');
    $result=mysql_query($requete,$idcom);
    echo "<h3>Liste des véhicules de $prenom $nom </h3>";
    echo "<table border='1' >";
    while($ligne=mysql_fetch_object($result)) ←❶
    {
        echo " <tr><td>&nbsp;",$ligne->immat," 75 &nbsp;</td><td>&nbsp;",$
        $ligne->modele," &nbsp;</td>"; ←❷
    }
    echo "</table>";
}
else {"<h3>Formulaire à compléter!</h3>";}
?>
</body>
</html>

```

## Exercices du chapitre 16

Le but de ces exercices est de réaliser avec SQLite la même application que celle réalisée avec MySQL au cours des exercices du chapitre 14.

### Exercice 1

Créez une base nommée `voitures` à l'aide de `SQLiteManager`. Créez ensuite les tables de la base `voitures` selon le modèle logique défini dans les exercices du chapitre 13 (en cas de problème, voir le corrigé des exercices de ce chapitre). Vérifiez la structure de chaque table.

## Création de la table « propriétaire »

**Propriétés de la nouvelle table : propriétaire**

Champ	Type	Longueur	Null	Défaut	Primaire
id_pers	MEDIUMINT	8	not null		<input checked="" type="radio"/>
nom	VARCHAR	30	not null		<input type="radio"/>
prenom	VARCHAR	30	null		<input type="radio"/>
adresse	VARCHAR	50	not null		<input type="radio"/>
ville	VARCHAR	40	not null		<input type="radio"/>
codepostal	MEDIUMINT	5	not null		<input type="radio"/>

Le code SQL de création de la table :

```
CREATE TABLE propriétaire (  
  id_pers INTEGER PRIMARY KEY,  
  nom VARCHAR(30) NOT NULL,  
  prenom VARCHAR(30) ,  
  adresse VARCHAR(50) NOT NULL,  
  ville VARCHAR(40) NOT NULL,  
  codepostal MEDIUMINT(5) NOT NULL  
);
```

## Création de la table « cartegrise »

**Propriétés de la nouvelle table : cartegrise**

Champ	Type	Longueur	Null	Défaut	Primaire
id_pers	MEDIUMINT	8	not null		<input checked="" type="radio"/>
immat	VARCHAR	6	not null		<input checked="" type="radio"/>
datecarte	DATE		not null		<input type="radio"/>

Le code SQL de création de la table :

```
CREATE TABLE cartegrise (  
  id_pers MEDIUMINT(8) NOT NULL PRIMARY KEY,  
  immat VARCHAR(6) NOT NULL PRIMARY KEY,  
  datecarte DATE NOT NULL  
);
```

## Création de la table « voiture »

**Propriétés de la nouvelle table : voiture**

Champ	Type	Longueur	Null	Défaut	Primaire
immat	VARCHAR	6	not null		<input checked="" type="radio"/>
id_modele	VARCHAR	10	not null		<input checked="" type="radio"/>
couleur	VARCHAR	10	not null		<input type="radio"/>
datevoiture	DATE		not null		<input type="radio"/>

Le code SQL de création de la table :

```
CREATE TABLE voiture (  
  immat VARCHAR(6) NOT NULL PRIMARY KEY,  
  id_modele VARCHAR(10) NOT NULL,  
  couleur VARCHAR(10) NOT NULL,  
  datevoiture DATE NOT NULL  
);
```

Création de la table « modele »

Propriétés de la nouvelle table : modele

Champ	Type	Longueur	Null	Défaut	Primaire
id_modele	VARCHAR	10	not null		<input checked="" type="radio"/>
modele	VARCHAR	30	not null		<input type="radio"/>
carburant	VARCHAR	10	not null		<input type="radio"/>

Enregistrer

Le code SQL de création de la table :

```
CREATE TABLE modele (  
  id_modele VARCHAR(10) NOT NULL PRIMARY KEY,  
  modele VARCHAR(30) NOT NULL,  
  carburant VARCHAR(10) NOT NULL  
);
```

## Exercice 2

Créez un formulaire permettant l'insertion des coordonnées d'une personne dans la table `proprietaire` en utilisant la méthode objet.

```
<?php  
if(isset($_POST['nom']) && isset($_POST['adresse']) &&  
isset($_POST['ville']) && isset($_POST['codepostal']))  
{  
  //Récupération des valeurs saisies  
  $nom=$_POST['nom'];  
  $prenom=$_POST['prenom'];  
  $adresse=$_POST['adresse'];  
  $ville=$_POST['ville'];  
  $codepostal=$_POST['codepostal'];  
  $db=new  
  SQLiteDatabase("C:/wamp/www/sqlitemanager/voitures",0666,$erreur);  
  //Requête SQL  
  $requete="INSERT INTO proprietaire  
VALUES(NULL,$nom,$prenom,$adresse,$ville,$codepostal)";  
  if($result=$db->queryExec($requete)) echo "DONNEES INSEREES";  
}  
else {"<h3>Formulaire à compléter!</h3>";}  
?>  
  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"  
"http://www.w3.org/TR/REC-html40/strict.dtd">  
<html>  
<head>
```

```

<title>Enregistrement d'une personne</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
</head>
<body>
<form action= "<?php echo $_SERVER['PHP_SELF'];?>" method="post"
enctype="application/x-www-form-urlencoded">
<fieldset>
<legend><b>Coordonnées de la personne</b></legend>
<table>
<tr><td>Nom : </td><td><input type="text" name="nom" /></td></tr>
<tr><td>Prénom : </td><td><input type="text" name="prenom"
/></td></tr>
<tr><td>Adresse : </td><td><input type="text" name="adresse"
/></td></tr>
<tr><td>Ville : </td><td><input type="text" name="ville"
/></td></tr>
<tr><td>Code postal : </td><td><input type="text" name="codepostal"
/></td></tr>
<tr colspan="2"><td><input type="submit" value="Enregistrer"
/></td></tr>
</table>
</fieldset>
</form>
</body>
</html>

```

### Exercice 3

Insérez des données dans la table modele en utilisant SQLiteManager, puis créez un script qui affiche la liste de tous les modèles de voiture dans un tableau HTML en effectuant un tri par marque. Utilisez la méthode procédurale.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<html>
<head>
<title>Affichage de tous les modèles de voitures</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
</head>
<body>
<?php
if ($id_base=
sqlite_open("C:/wamp/www/sqlitemanager/voitures",0666,$erreur))
{
echo "<table border=\"1\">";
echo "<thead><th>Type</th><th> Modele </th></thead>";
//Requête SQL
$requete="SELECT id_modele,modele FROM modele ORDER BY modele
ASC";
$result=sqlite_query($id_base,$requete);
while($ligne=sqlite_fetch_array($result))
{
echo "<tr><td>&nbsp;",$ligne[0]," &nbsp;</td><td> ",$ligne[1],"
&nbsp;</td></tr>";
}
}

```

```

    echo "</table>";
}
?>
</body>
</html>

```

Le résultat affiché est le suivant :

Type	Modele
563339GH56	Citroën C3
17C92853AZ	Citroën C5
178524ER45	Citroën Picasso
ZER627864K	Ferrari GT 40
33356677PO	Peugeot 206
AZER67455T	Peugeot 307
DSQS455674	Renault Adventime
83321TY455	Renault Espace
QSDFGH7777	Renault Modus
7499RF5679	Renault Mégane Scénic
FHT55432GH	Renault Twingo
485228FGD7	Volkswagen Golf

## Exercice 4

Adaptez le script de l'exercice 3 en utilisant la méthode objet. Les données de chaque ligne du résultat doivent aussi être retournées sous forme d'objet.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<html>
<head>
<title>Affichage de tous les modèles de voitures</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
</head>
<body>
<?php
if($db=new
SQLiteDatabase("C:/wamp/www/sqlitemanager/voitures",0666,$erreur))
{
//Requête SQL
$requete="SELECT id_modele,modele FROM modele ORDER BY modele
ASC";
$result=$db->query($requete);
echo "<table border='1'>";
echo "<thead><th>Type</th><th> Modele </th></thead>";
while($ligne=$result->fetchObject())

```

```

    {
    echo "<tr><td>&nbsp;",$ligne->id_modele," &nbsp;</td><td> ",$ligne-
    >modele," &nbsp;</td></tr>";
    }
    echo "</table>";
}
?>
</body>
</html>

```

## Exercice 5

Créez dynamiquement un formulaire contenant une liste de sélection HTML (avec les éléments `<select>` et `<option>`) qui donne la liste de tous les modèles présents dans la table `modele`. Ajoutez manuellement à l'aide de `SQLiteManager` un ou plusieurs modèles à la table, et vérifiez que la liste de sélection prend bien en compte ces ajouts. Utilisez successivement la méthode procédurale et la méthode objet.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<html>
<head>
<title>Création d'une liste de sélection de tous les modèles de
voitures</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
</head>
<body>
<h2>METHODE PROCEDURALE</h2>
<form method="post" action="ch16exo5.php">
  <fieldset>
    <legend>Liste des modèles</legend>
    <select name="modeles">
<?php
if($idbase=sqlite_open("C:/wamp/www/sqlitemanager/voitures",0666,$e
rreur))
{
  //Requête SQL
  $requete="SELECT id_modele,modele FROM modele ORDER BY modele
ASC";
  $result=sqlite_query($idbase,$requete);
  while($ligne=sqlite_fetch_object($result))
  {
    echo "<option value=\"\$ligne->id_modele\"> \$ligne-
>modele</option>";
  }
  sqlite_close($idbase);
}
?>
  </select>
  <input type="submit" value="Choisir" />
</fieldset>
</form>

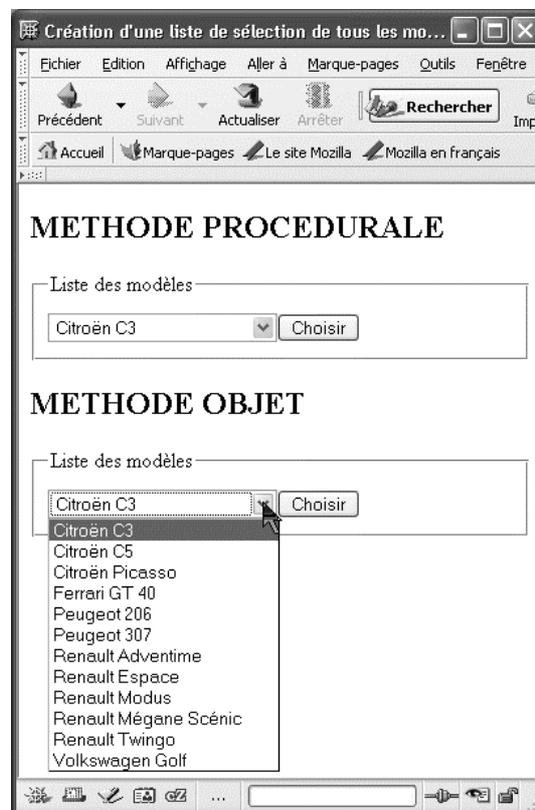
```

```

<h2>METHODE OBJET</h2>
<form method="post" action="ch16exo5.php">
  <fieldset>
    <legend>Liste des modèles</legend>
    <select name="\modeles\">
<?php
if($db=new
SQLiteDatabase("C:/wamp/www/sqlitemanager/voitures",0666,$erreur))
{
  //Requête SQL
  $requete="SELECT id_modele,modele FROM modele ORDER BY modele
ASC";
  $result=$db->query($requete);
  while($ligne=$result->fetchObject())
  {
    echo "<option value=\"\$ligne->id_modele\">\$ligne-
>modele</option>";
  }
}
?>
  </select>
  <input type="submit" value="Choisir" />
</fieldset>
</form>
</body>
</html>

```

Le résultat obtenu est :



## Exercice 6

Utilisez le mécanisme des transactions pour insérer simultanément et en toute sécurité des données dans les tables `proprietaire` et `cartegrise`. Utilisez la méthode procédurale puis la méthode objet.

### Méthode procédurale : fichier `ch16exo6.php`

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<html>
<head>
<title>Enregistrement d'un véhicule </title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
</head>
<body>
<form action= "<?php echo $_SERVER['PHP_SELF'];?>" name="form1"
method="post"
enctype="application/x-www-form-urlencoded">
<fieldset>
<legend><b>Saisissez les caractéristiques du véhicule</legend>
<table>
<tr colspan="2"><td><b>Propriétaire</td></tr>
<tr><td>Nom : </td><td><input type="text" name="nom" size="40"
maxlength="10"/></td></tr>
<tr><td>Prénom : </td><td><input type="text" name="prenom"
size="40" maxlength="30"/></td></tr>
<tr><td>Adresse : </td><td><input type="text" name="adresse"
size="40" maxlength="30"/></td></tr>
<tr><td>Ville : </td><td><input type="text" name="ville" size="40"
maxlength="30"/></td></tr>
<tr><td>Code postal : </td><td><input type="text" name="codepostal"
size="40" maxlength="5"/></td></tr>
<tr colspan="2"><td><b>Modèle</td></b></tr>
<tr><td>Nom du modèle</td>
<td>
<select name="modele">
<?php
if($idbase=mysqli_open("C:/wamp/www/sqlitemanager/voitures",0666,$e
rreur))
{
//Requête SQL
$requete="SELECT id_modele,modele FROM modele ORDER BY modele
ASC";
$result=mysqli_query($idbase,$requete);
while($ligne=mysqli_fetch_object($result))
{
echo "<option value=\"\$ligne->id_modele\"> \$ligne->modele</option>";
}
mysqli_close($idbase);
}
?>
</select></tr>
<tr><td>Carburant : </td>
```

```

<td><select name="carburant">
<option value="essence">Essence</option>
<option value="diesel">Diesel</option>
<option value="électrique">Electrique</option>
</select></td>
</tr>
<tr colspan="2"><td><b>Voiture</b></td></tr>
<tr>
<td>Numéro d'immatriculation</td>
<td><input type="text" name="immat" /></td>
</tr>
<tr><td>Couleur : </td>
<td><select name="couleur">
<option value="claire">Claire</option>
<option value="moyenne">Moyenne</option>
<option value="foncée">Foncée</option>
</select></td>
</tr>
<tr>
<td>Date lere immatriculation AAAA-MM-JJ</td>
<td><input type="text" name="datevoiture" /></td>
</tr>
<td>Date de la carte grise AAAA-MM-JJ</td>
<td><input type="text" name="datecarte" /></td>
</tr>
<tr>
<td><input type="reset" value=" Effacer " /></td>
<td><input type="submit" value="ENREGISTRER" name="enreg" /></td>
</tr>
</table>
</fieldset>
</form>
<?php
if(isset($_POST['enreg']))
{
//Récupération des valeurs du formulaire
$id_modele=$_POST['modele'];
$carburant=sqlite_escape_string($_POST['carburant']);
$immat=sqlite_escape_string($_POST['immat']);
$couleur=sqlite_escape_string($_POST['couleur']);
$datevoiture=sqlite_escape_string($_POST['datevoiture']);
$datecarte=sqlite_escape_string($_POST['datecarte']);
$nom=sqlite_escape_string($_POST['nom']);
$prenom=sqlite_escape_string($_POST['prenom']);
$adresse=sqlite_escape_string($_POST['adresse']);
$ville=sqlite_escape_string($_POST['ville']);
$codepostal=(integer) $_POST['codepostal'];
$id_base=
sqlite_open("C:/wamp/www/sqlitemanager/voitures",0666,$erreur);
//Requête SQL
$requete="BEGIN;";
//Requête SQL: insertions dans la table propriétaire
$requete="INSERT INTO propriétaire
VALUES (NULL, '$nom', '$prenom', '$adresse', '$ville', '$codepostal');"

```

```

$result=sqlite_exec($id_base,$requete);
//Récupération de l'identifiant de la personne enregistrée
$id_pers=sqlite_last_insert_rowid($id_base);
//Requête SQL: insertions dans la table cartegrise
$requete="INSERT INTO cartegrise
VALUES('$id_pers', '$immat', '$datecarte');"
//Requête SQL: insertions dans la table voiture
$requete.="INSERT INTO voiture
VALUES('$immat', '$id_modele', '$couleur', '$datevoiture');"
$requete.="COMMIT;";
sqlite_exec($id_base,$requete);
}
?>
</body>
</html>

```

## Méthode objet : fichier ch16exo6b.php

Les seuls modifications par rapport au fichier précédent sont signalées aux repères ❶, ❷, ❸, ❹, ❺ et ❻.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<html>
<head>
<title>Enregistrement d'un véhicule </title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
</head>
<body>
<form action= "<?php echo $_SERVER['PHP_SELF'];?>" name="form1"
method="post"
enctype="application/x-www-form-urlencoded">
<fieldset>
<legend><b>Saisissez les caractéristiques du véhicule</legend>
<table>
<tr colspan="2"><td><b>Propriétaire</td></tr>
<tr><td>Nom : </td><td><input type="text" name="nom" size="40"
maxlength="10"/></td></tr>
<tr><td>Prénom : </td><td><input type="text" name="prenom"
size="40" maxlength="30"/></td></tr>
<tr><td>Adresse : </td><td><input type="text" name="adresse"
size="40" maxlength="30"/></td></tr>
<tr><td>Ville : </td><td><input type="text" name="ville" size="40"
maxlength="30"/></td></tr>
<tr><td>Code postal : </td><td><input type="text" name="codepostal"
size="40" maxlength="5"/></td></tr>
<tr colspan="2"><td><b>Modèle</td></b></tr>
<tr><td>Nom du modèle</td>
<td>
<select name="modele">
<?php
$db=new SQLiteDatabase("C:/wamp/www/sqlitemanager/voitures",
0666,$erreur);//❶
{
//Requête SQL
$requete="SELECT id_modele,modele FROM modele ORDER BY modele
ASC";
$result=$db->query($requete);//❷

```

```

while($ligne=$result->fetchObject())//←❶
{
    echo "<option value=\"\$ligne->id_modele\">$ligne-
>modele</option>";
}
}
?>
</select></tr>
<tr><td>Carburant : </td>
<td><select name="carburant">
<option value="essence">Essence</option>
<option value="diesel">Diesel</option>
<option value="électrique">Electrique</option>
</select></td>
</tr>
<tr colspan="2"><td><b>Voiture</td></tr>
<tr>
<td>Numéro d'immatriculation</td>
<td><input type="text" name="immat" /></td>
</tr>
<tr><td>Couleur : </td>
<td><select name="couleur">
<option value="claire">Claire</option>
<option value="moyenne">Moyenne</option>
<option value="foncée">Foncée</option>
</select></td>
</tr>
<tr>
<td>Date 1ere immatriculation AAAA-MM-JJ</td>
<td><input type="text" name="datevoiture" /></td>
</tr>
<td>Date de la carte grise AAAA-MM-JJ</td>
<td><input type="text" name="datecarte" /></td>
</tr>
<tr>
<td><input type="reset" value=" Effacer " /></td>
<td><input type="submit" value="ENREGISTRER" name="enreg" /></td>
</tr>
</table>
</fieldset>
</form>
<?php
if(isset($_POST['enreg']))
{
//Récupération des valeurs du formulaire
$id_modele=$_POST['modele'];
$carburant=sqlite_escape_string($_POST['carburant']);
$immat=sqlite_escape_string($_POST['immat']);
$couleur=sqlite_escape_string($_POST['couleur']);
$datevoiture=sqlite_escape_string($_POST['datevoiture']);
$datecarte=sqlite_escape_string($_POST['datecarte']);
$nom=sqlite_escape_string($_POST['nom']);
$prenom=sqlite_escape_string($_POST['prenom']);
$adresse=sqlite_escape_string($_POST['adresse']);

```

```

$ville=sqlite_escape_string($_POST['ville']);
$codepostal=(integer) $_POST['codepostal'];
//Requête SQL
$requete="BEGIN;";
//Requête SQL: insertions dans la table proprietaire
$requete.="INSERT INTO proprietaire
VALUES (NULL, '$nom', '$prenom', '$adresse', '$ville', '$codepostal');"
$result=$db->queryExec($requete); // ← ④
//Récupération de l'identifiant de la personne enregistrée
$id_pers=$db->lastInsertRowid(); // ← ⑤
//Requête SQL: insertions dans la table cartegrise
$requete="INSERT INTO cartegrise
VALUES ('$id_pers', '$immat', '$datecarte');"
//Requête SQL: insertions dans la table voiture
$requete.="INSERT INTO voiture
VALUES ('$immat', '$id_modele', '$couleur', '$datevoiture');"
$requete.="COMMIT;";
$result=$db->queryExec($requete); // ← ⑥
unset($db);
}
?>
</body>
</html>

```

## Exercice 7

Créez un formulaire de recherche permettant de trouver toutes les personnes propriétaires d'un modèle de véhicule donné. Affichez les résultats sous forme de tableau HTML. Utilisez l'exercice 5 pour créer la liste de sélection des modèles.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<html>
<head>
<title>Recherche de tous les propriétaires d'un même modèle
</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
</head>
<body>
<form action="<?php echo $_SERVER['PHP_SELF'];?>" name="form1"
method="post"
enctype="application/x-www-form-urlencoded">
<fieldset>
<legend><b>Choisissez le nom du modèle</legend>
<table>
<tr><td>Nom du modèle</td>
<td>
<select name="idmodele">
<?php
if($id_base=sqlite_open("C:/wamp/www/sqlitemanager/voitures",0666,$
erreur))
{
//Requête SQL
$requete="SELECT id_modele,modele FROM modele ORDER BY modele
ASC";

```

```

$result=sqlite_query($id_base,$requete);
while($ligne=sqlite_fetch_object($result))
{
echo "<option value=\"\$ligne->id_modele\">$ligne->modele</option>";
}
sqlite_close($idbase);
}
?>
</select>
</tr>
<tr>
<td><input type="reset" value=" Effacer " /></td>
<td><input type="submit" value="CHERCHER" name="cherche" /></td>
</tr>
</table>
</fieldset>
</form>
<?php
if(isset($_POST['idmodele']))
{
//Récupération des valeurs du formulaire
$idmodele=$_POST['idmodele'];
//Requête SQL
$requete="SELECT      proprietaire.nom,      proprietaire.prenom,
voiture.immat,      modele.modele
FROM      voiture,      modele,      proprietaire,      cartegrise
WHERE      voiture.id_modele=      ' $idmodele'
AND      voiture.id_modele=modele.id_modele
AND      cartegrise.immat=voiture.immat
AND      proprietaire.id_pers=cartegrise.id_pers";
$id_base=sqlite_open("C:/wamp/www/sqlitemanager/voitures",0666,$err
eur);
$result=sqlite_query($id_base,$requete);
echo "<table border=\"1\">";
while($ligne=sqlite_fetch_array($result))
{
echo "<tr><td>$ligne[0] </td><td>$ligne[1] </td><td>$ligne[2]
</td>";
}
echo "</table>";
sqlite_close($id_base);
}
else echo "PROBLEME !!";
?>
</body>
</html>

```

# Exercices du chapitre 17

## Exercice 1

Créez un fichier XML nommé **iut.xml**, dont l'élément racine est `<iut>`. Les éléments principaux nommés `<etudiant>` ont comme attributs `id` (numéro d'inscription) et `nom`. Chaque élément `<etudiant>` peut contenir autant d'éléments `<uv>` que désiré. Chaque UV doit avoir un nom, une durée et une note enregistrés dans des sous-éléments. Visualisez ce fichier dans un navigateur pour vérifier qu'il est bien formé.

Exemple de fichier `iut.xml` :

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>
<iut>
  <etudiant id="123" nom="Lemaire">
    <uv>
      <nom>Programmation PHP 5</nom>
      <durée>4 semaines</durée>
      <note>14</note>
    </uv>
    <uv>
      <nom>MySQL</nom>
      <durée>6 semaines</durée>
      <note>16</note>
    </uv>
  </etudiant>
  <etudiant id="345" nom="Legros">
    <uv>
      <nom>Programmation PHP 5</nom>
      <durée>4 semaines</durée>
      <note>12</note>
    </uv>
    <uv>
      <nom>SQLite</nom>
      <durée>6 semaines</durée>
      <note>16</note>
    </uv>
  </etudiant>
  <etudiant id="456" nom="Legrand">
    <uv>
      <nom>Programmation PHP 5</nom>
      <durée>5 semaines</durée>
      <note>14</note>
    </uv>
    <uv>
      <nom>SQLite</nom>
      <durée>3 semaines</durée>
      <note>15</note>
    </uv>
  </etudiant>
</iut>
```

```

    </uv>
</etudiant>
<etudiant id="567" nom="Lebeau">
  <uv>
    <nom>Programmation PHP 5</nom>
    <durée>5 semaines</durée>
    <note>17</note>
  </uv>
  <uv>
    <nom>MySQL</nom>
    <durée>6 semaines</durée>
    <note>13</note>
  </uv>
</etudiant>
</iut>

```

## Exercice 2

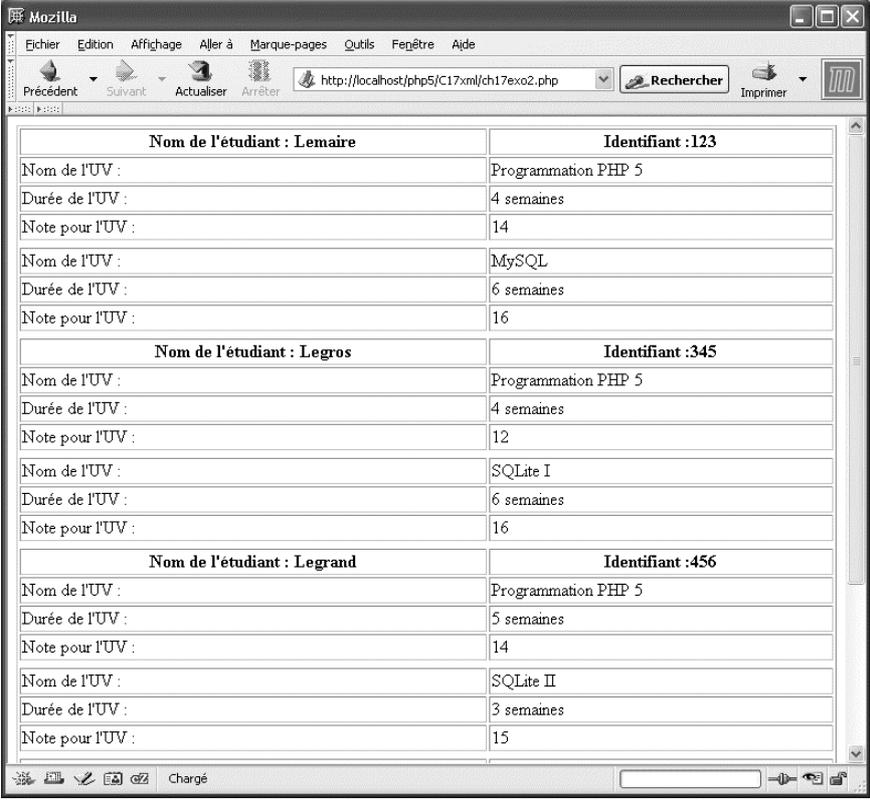
Lisez les éléments et les attributs du fichier **iut.xml**, et affichez-les dans un tableau HTML.

```

<?php
$xml=simplexml_load_file("iut.xml");
echo "<table border=\"1\" width=\"80%\">";
foreach ($xml->etudiant as $valeur)
{
  echo " <tr><th>Nom de l'étudiant : ",$valeur['nom'],"</th>
<th>Identifiant : ",$valeur['id'],"</th></tr>";
  foreach ($valeur->uv as $valeur)
  {
    echo "<tr><td> Nom de l'UV : </td><td>",$valeur-
>nom,"</td></tr>";
    echo "<tr><td> Durée de l'UV : </td><td>",$valeur-
>duree,"</td></tr>";
    echo "<tr><td> Note pour l'UV : </td><td>",$valeur-
>note,"</td></tr>";
    echo "<tr colspan=2 ><td></td></tr>";
  }
}
echo "</table>";
?>

```

Résultat obtenu :



Nom de l'étudiant : Lemaire		Identifiant :123
Nom de l'UV :		Programmation PHP 5
Durée de l'UV :		4 semaines
Note pour l'UV :		14
Nom de l'UV :		MySQL
Durée de l'UV :		6 semaines
Note pour l'UV :		16
Nom de l'étudiant : Legros		Identifiant :345
Nom de l'UV :		Programmation PHP 5
Durée de l'UV :		4 semaines
Note pour l'UV :		12
Nom de l'UV :		SQLite I
Durée de l'UV :		6 semaines
Note pour l'UV :		16
Nom de l'étudiant : Legrand		Identifiant :456
Nom de l'UV :		Programmation PHP 5
Durée de l'UV :		5 semaines
Note pour l'UV :		14
Nom de l'UV :		SQLite II
Durée de l'UV :		3 semaines
Note pour l'UV :		15

### Exercice 3

Créez un formulaire permettant d'insérer des données dans le fichier **iut.xml**. Le script doit permettre la visualisation éventuelle du fichier après l'insertion.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<html>
<head>
<title>Enregistrement en XML</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
</head>
<body>
  <form action= "<? $_SERVER['PHP_SELF'] ?>" method="post"
  enctype="application/x-www-form-urlencoded">
    <fieldset>
      <legend><b>Saisie de données</b></legend>
      <table><tbody>
        <tr colspan=2><th>Etudiant</th></tr>
        <tr>
          <td>Identifiant : </td>
          <td><input type="text" name="id" /></td>
        </tr>
        <tr>
          <td>Nom de l'étudiant : </td>
          <td><input type="text" name="nomet" /></td>
        </tr>
      </tbody></table>
    </fieldset>
  </form>
</body>
</html>
```

```

<tr colspan=2><th>UV 1</th></tr>
<tr>
  <td>Nom de l'UV 1 : </td>
  <td><input type="text" name="nomuv1" /></td>
</tr>
<tr>
  <td>Durée de l'UV 1 : </td>
  <td><input type="text" name="duree1" /></td>
</tr>
<tr>
  <td> Note de l'UV 1</td>
  <td><input type="text" name="note1" /></td>
</tr>
<tr colspan=2><th>UV 2</th></tr>
<tr>
  <td>Nom de l'UV 2 : </td>
  <td><input type="text" name="nomuv2" /></td>
</tr>
<tr>
  <td>Durée de l'UV 2 : </td>
  <td><input type="text" name="duree2" /></td>
</tr>
<tr>
  <td> Note de l'UV 2</td>
  <td><input type="text" name="note2" /></td>
</tr>
<tr>
<td>
<input type="submit" name="envoi" value="Enregistrer"/>
<a href="ch17exo2.php"><button type="button"> VOIR </button></a>
</td>
</tr>
</tbody></table>
</fieldset>
</form>
</body>
</html>
<?php
//Enregistrement d'un étudiant
if(isset($_POST['envoi']) && !empty($_POST['id']) &&
!empty($_POST['nomet']))
{
  $id= htmlspecialchars($_POST['id']);
  $nomet= htmlspecialchars($_POST['nomet']);
  $xml=simplexml_load_file("iut.xml");
  $chxml = $xml->asxml();
  $chxml = str_replace("</iut>", "", $chxml);
  $chxml.= "<etudiant id=\"\$id\" nom=\"\$nomet\">\n";
  //Enregistrement de l'UV 1
  if(!empty($_POST['nomuv1']) && !empty($_POST['duree1']) &&
!empty($_POST['note1']))
  {
    $nomuv1= htmlspecialchars($_POST['nomuv1']);
    $duree1= htmlspecialchars($_POST['duree1']);

```

```

        $note1= htmlspecialchars($_POST['note1']);
        $chxml.= "<uv>\n <nom>$nomuv1</nom>\n <duree>$duree1</duree>\n
<note>$note1</note>\n</uv>\n";
    }
    //Enregistrement de l'UV 2
    if(!empty($_POST['nomuv2']) && !empty($_POST['duree2']) &&
!empty($_POST['note2']))
    {
        $nomuv2= htmlspecialchars($_POST['nomuv2']);
        $duree2= htmlspecialchars($_POST['duree2']);
        $note2= htmlspecialchars($_POST['note2']);
        $chxml.= "<uv>\n <nom>$nomuv2</nom>\n <duree>$duree2</duree>\n
<note>$note2</note>\n</uv>\n";
    }
    //Fermeture des éléments <etudiant> et <iut>
    $chxml.=" </etudiant>\n</iut>\n";
    $verif=file_put_contents("iut.xml",$chxml);
    echo $verif;
}
?>

```

## Exercice 4

Créez un formulaire de recherche permettant d'afficher à la demande les noms des étudiants par ordre alphabétique, ainsi que la liste des UV et leur nom.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<html>
<head>
<title>Bibliographie XML</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
</head>
<body>
    <form action= "<? $_SERVER['PHP_SELF'] ?>" method="post"
    enctype="application/x-www-form-urlencoded">
        <fieldset>
            <legend><b>Recherche</b></legend>
            <table><tbody>
                <tr>
                    <td>Rechercher tous les : </td>
                    <td>
                        <select name="choix">
                            <option value="/iut/etudiant/@nom">Etudiants</option>
                            <option value="etudiant/uv/nom">Nom des UV</option>
                        </select>
                    </td>
                    <td>
                        <input type="submit" name="envoi" value="OK"/>
                    </td>
                </tr>
            </tbody></table>
        </fieldset>
    </form>
</body>
</html>

```

```

<?php
if(isset($_POST['envoi']))
{
    $choix= $_POST['choix'];
    $xml=simplexml_load_file("iut.xml");
    $result= $xml->xpath($choix);
    //Eliminer les doublons
    $result=array_unique($result);
    echo "<h3>Résultats de la recherche</h3>";
    //Affichage sous forme de liste
    echo "<ol>";
    foreach($result as $valeur)
    {
        echo "<li> $valeur </li>";
    }
    echo "</ol>";
}
?>

```

## Exercice 5

Transférez toutes les données de la base voitures créée aux chapitres 14 et 15 dans un fichier XML d'abord en utilisant phpMyAdmin puis en écrivant un script PHP. La répartition des données dans des attributs ou comme contenu des éléments est libre.

Exportation réalisée avec phpMyAdmin (fichier voitures.xml)

```

<?xml version="1.0" encoding="iso-8859-1" ?>
<!--
-
- phpMyAdmin XML Dump
- version 2.6.0-rc3
- http://www.phpmyadmin.net
-
- Serveur: localhost
- Généré le : Dimanche 26 Décembre 2004 à 00:09
- Version du serveur: 4.0.21
- Version de PHP: 5.0.2
-->

<!--
- Base de données: `voitures`
-->
<voitures>
  <!-- Table cartegrise -->
  <cartegrise>
    <id_pers>1</id_pers>
    <immat>178ZRX</immat>
    <datecarte>2004-05-15</datecarte>
  </cartegrise>
  <cartegrise>
    <id_pers>2</id_pers>

```

```
<immat>179ZRX</immat>
  <datecarte>2004-05-17</datecarte>
</cartegrise>
<cartegrise>
  <id_pers>2</id_pers>
  <immat>188ZRX</immat>
  <datecarte>2004-05-27</datecarte>
</cartegrise>
<cartegrise>
  <id_pers>3</id_pers>
  <immat>180ZRX</immat>
  <datecarte>2004-05-18</datecarte>
</cartegrise>
<cartegrise>
  <id_pers>4</id_pers>
  <immat>181ZRX</immat>
  <datecarte>2004-05-19</datecarte>
</cartegrise>
<cartegrise>
  <id_pers>4</id_pers>
  <immat>182ZRX</immat>
  <datecarte>2004-05-20</datecarte>
</cartegrise>
<cartegrise>
  <id_pers>5</id_pers>
  <immat>181ZRX</immat>
  <datecarte>2004-05-19</datecarte>
</cartegrise>
<cartegrise>
  <id_pers>5</id_pers>
  <immat>183ZRX</immat>
  <datecarte>2004-05-22</datecarte>
</cartegrise>
<cartegrise>
  <id_pers>6</id_pers>
  <immat>184ZRX</immat>
  <datecarte>2004-05-23</datecarte>
</cartegrise>
<cartegrise>
  <id_pers>7</id_pers>
  <immat>185ZRX</immat>
  <datecarte>2004-05-24</datecarte>
</cartegrise>
<cartegrise>
  <id_pers>7</id_pers>
  <immat>189ZRX</immat>
  <datecarte>2004-05-28</datecarte>
</cartegrise>
<cartegrise>
  <id_pers>8</id_pers>
  <immat>186ZRX</immat>
  <datecarte>2004-05-25</datecarte>
</cartegrise>
<cartegrise>
  <id_pers>8</id_pers>
```

```

        <immat>187ZRX</immat>
        <datecarte>2004-05-25</datecarte>
    </cartegrise>
    <cartegrise>
        <id_pers>9</id_pers>
        <immat>200XJR</immat>
        <datecarte>2004-11-23</datecarte>
    </cartegrise>
<!-- Table modele -->
    <modele>
        <id_modele>17C92853AZ</id_modele>
        <modele>Citroën C5</modele>
        <carburant>diesel</carburant>
    </modele>
    <modele>
        <id_modele>178524ER45</id_modele>
        <modele>Citroën Picasso</modele>
        <carburant>essence</carburant>
    </modele>
    <modele>
        <id_modele>7499RF5679</id_modele>
        <modele>Renault Mégane Scénic</modele>
        <carburant>diesel</carburant>
    </modele>
    <modele>
        <id_modele>33356677P0</id_modele>
        <modele>Peugeot 206</modele>
        <carburant>électrique</carburant>
    </modele>
    <modele>
        <id_modele>563339GH56</id_modele>
        <modele>Citroën C3</modele>
        <carburant>essence</carburant>
    </modele>
    <modele>
        <id_modele>83321TY455</id_modele>
        <modele>Renault Espace</modele>
        <carburant>diesel</carburant>
    </modele>
    <modele>
        <id_modele>AZER67455T</id_modele>
        <modele>Peugeot 307</modele>
        <carburant>essence</carburant>
    </modele>
    <modele>
        <id_modele>FHT55432GH</id_modele>
        <modele>Renault Twingo</modele>
        <carburant>essence</carburant>
    </modele>
    <modele>
        <id_modele>DSQS455674</id_modele>
        <modele>Renault Adventime</modele>
        <carburant>diesel</carburant>
    </modele>

```

```

<modele>
  <id_modele>485228FGD7</id_modele>
  <modele>Volkswagen Golf</modele>
  <carburant>diesel</carburant>
</modele>
<modele>
  <id_modele>ZER627864K</id_modele>
  <modele>Ferrari GT 40</modele>
  <carburant>essence</carburant>
</modele>
<modele>
  <id_modele>QSDFGH7777</id_modele>
  <modele>Renault Modus</modele>
  <carburant>essence</carburant>
</modele>
<!-- Table proprietaire -->
<proprietaire>
  <id_pers>1</id_pers>
  <nom>Zulgru</nom>
  <prenom>Mickey</prenom>
  <adresse>23 rue Newton</adresse>
  <ville>Paris</ville>
  <codepostal>75006</codepostal>
</proprietaire>
<proprietaire>
  <id_pers>2</id_pers>
  <nom>Algout</nom>
  <prenom>Michel</prenom>
  <adresse>562 rue Pascal</adresse>
  <ville>Paris</ville>
  <codepostal>75018</codepostal>
</proprietaire>
<proprietaire>
  <id_pers>3</id_pers>
  <nom>Dupont</nom>
  <prenom>Jean</prenom>
  <adresse>3 Rue d'Alésia</adresse>
  <ville>Paris</ville>
  <codepostal>75015</codepostal>
</proprietaire>
<proprietaire>
  <id_pers>4</id_pers>
  <nom>Azerty</nom>
  <prenom>Paulo</prenom>
  <adresse>5 Rue Compoint</adresse>
  <ville>Paris</ville>
  <codepostal>75018</codepostal>
</proprietaire>
<proprietaire>
  <id_pers>5</id_pers>
  <nom>Azerty</nom>
  <prenom>Marie</prenom>
  <adresse>5 Rue Compoint</adresse>
  <ville>Paris</ville>
  <codepostal>75018</codepostal>

```

```

</proprietaire>
<proprietaire>
  <id_pers>6</id_pers>
  <nom>Duval</nom>
  <prenom>Emilie</prenom>
  <adresse>543 avenue Foché</adresse>
  <ville>Paris</ville>
  <codepostal>75016</codepostal>
</proprietaire>
<proprietaire>
  <id_pers>7</id_pers>
  <nom>Zouk</nom>
  <prenom>Julia</prenom>
  <adresse>56 Boulevard Nez</adresse>
  <ville>Paris</ville>
  <codepostal>75011</codepostal>
</proprietaire>
<proprietaire>
  <id_pers>8</id_pers>
  <nom>Val</nom>
  <prenom>Phil</prenom>
  <adresse>34 rue des Champs</adresse>
  <ville>Paris</ville>
  <codepostal>75017</codepostal>
</proprietaire>
<proprietaire>
  <id_pers>9</id_pers>
  <nom>Zébulon</nom>
  <prenom>Alfred</prenom>
  <adresse>Place de la Concorde</adresse>
  <ville>Paris</ville>
  <codepostal>75008</codepostal>
</proprietaire>
<!-- Table voiture -->
<voiture>
  <immat>200XJR</immat>
  <id_modele>17C92853AZ</id_modele>
  <couleur>foncée</couleur>
  <datevoiture>2003-03-19</datevoiture>
</voiture>
<voiture>
  <immat>178ZRX</immat>
  <id_modele>17C92853AZ</id_modele>
  <couleur>claire</couleur>
  <datevoiture>2002-09-15</datevoiture>
</voiture>
<voiture>
  <immat>179ZRX</immat>
  <id_modele>178524ER45</id_modele>
  <couleur>foncée</couleur>
  <datevoiture>2004-05-17</datevoiture>
</voiture>
<voiture>
  <immat>188ZRX</immat>

```

```
<id_modele>FHT55432GH</id_modele>
  <couleur>moyenne</couleur>
  <datevoiture>2004-05-27</datevoiture>
</voiture>
<voiture>
  <immat>180ZRX</immat>
  <id_modele>178524ER45</id_modele>
  <couleur>claire</couleur>
  <datevoiture>2004-05-18</datevoiture>
</voiture>
<voiture>
  <immat>181ZRX</immat>
  <id_modele>7499RF5679</id_modele>
  <couleur>claire</couleur>
  <datevoiture>2004-05-19</datevoiture>
</voiture>
<voiture>
  <immat>182ZRX</immat>
  <id_modele>485228FGD7</id_modele>
  <couleur>foncée</couleur>
  <datevoiture>2001-11-25</datevoiture>
</voiture>
<voiture>
  <immat>183ZRX</immat>
  <id_modele>33356677PO</id_modele>
  <couleur>moyenne</couleur>
  <datevoiture>2004-05-22</datevoiture>
</voiture>
<voiture>
  <immat>184ZRX</immat>
  <id_modele>563339GH56</id_modele>
  <couleur>foncée</couleur>
  <datevoiture>2004-05-23</datevoiture>
</voiture>
<voiture>
  <immat>185ZRX</immat>
  <id_modele>83321TY455</id_modele>
  <couleur>claire</couleur>
  <datevoiture>2004-05-24</datevoiture>
</voiture>
<voiture>
  <immat>186ZRX</immat>
  <id_modele>AZERT67455</id_modele>
  <couleur>moyenne</couleur>
  <datevoiture>2004-05-25</datevoiture>
</voiture>
<voiture>
  <immat>187ZRX</immat>
  <id_modele>DSQS455674</id_modele>
  <couleur>foncée</couleur>
  <datevoiture>2004-05-25</datevoiture>
</voiture>
<voiture>
  <immat>189ZRX</immat>
  <id_modele>ZER627864K</id_modele>
```

```

        <couleur>claire</couleur>
        <datevoiture>2004-05-28</datevoiture>
    </voiture>
</voitures>

```

En utilisant un script, nous pouvons choisir librement la répartition des champs des tables MySQL en tant qu'attributs ou éléments du document XML. Dans la base « voitures » la table « modele » porte le même nom que le champ « modele » qu'elle contient. Ceci ne posait pas de problème avec MySQL et permettait de montrer que la notation `modele.modele` dans une requête SQL donnait bien accès à la colonne `modele` de la table `modele` sans risque de confusion. En revanche, en XML il risque d'y avoir une confusion si deux éléments de structure différente portent le même nom. Nous avons donc choisi de créer un élément `<modeles>` et un élément `<modele>` imbriqué dans le précédent.

```

<?php
include("connex.inc.php");
$idcom=connex("voitures","myparam");
$requete="SELECT * FROM proprietaire";
//Table proprietaire
$result1=mysql_query($requete,$idcom);
$xml="<?xml version='1.0' encoding='iso-8859-1'
standalone='yes'?'>\n<voitures>\n";
while($ligne=mysql_fetch_array($result1,MYSQL_ASSOC))
{
    $xml.="<proprietaire id='\"{$ligne['id_pers']}\">\n";
    $xml.=" <nom>{$ligne['nom']} </nom>\n";
    $xml.=" <prenom>{$ligne['prenom']} </prenom>\n";
    $xml.=" <adresse>{$ligne['adresse']} </adresse>\n";
    $xml.=" <ville>{$ligne['ville']} </ville>\n";
    $xml.=" <codepostal>{$ligne['codepostal']} </codepostal>\n";

    $xml.=" </proprietaire>\n";
}
//Table cartegrise
$result2=mysql_query($requete,$idcom);
while($ligne=mysql_fetch_array($result2,MYSQL_ASSOC))
{
    $xml.="<cartegrise id_pers='\"{$ligne['id_pers']}\" \n
immat='\"{$ligne['immat']}\">\n";
    $xml.=" <datecarte>{$ligne['datecarte']} </datecarte>\n";
    $xml.=" </cartegrise>\n";
}
//Table voiture
$result3=mysql_query($requete,$idcom);
while($ligne=mysql_fetch_array($result3,MYSQL_ASSOC))
{
    $xml.="<voiture immat='\"{$ligne['immat']}\">\n
id_modele='\"{$ligne['id_modele']}\" ";
    $xml.=" <couleur>{$ligne['datecarte']} </couleur>\n";
    $xml.=" <datevoiture>{$ligne['datecarte']} </datevoiture>\n";
    $xml.=" </voiture>\n";
}
}

```

```
//Table modele
$result4=mysql_query($requete,$idcom);
while($ligne=mysql_fetch_array($result4,MYSQL_ASSOC))
{
    $chxml.="<modeles id_modele=\"{$ligne['id_modele']}\">> ";
    $chxml.=" <modele>{$ligne['modele']} </modele>\n";
    $chxml.=" <carburant>{$ligne['carburant']} </carburant>\n";
    $chxml.=" </modeles>\n";
}
$chxml.="</voitures>";
file_put_contents("voitures2.xml",$chxml);
echo "<a href=\"voitures2.xml\"><button type=\"button\"> Voir le
fichier XML </button></a>";
?>
```

## Exercice 6

Transférez les données du fichier **iut.xml** de l'exercice 3 dans une table MySQL. Créez la table auparavant en lui donnant comme clé primaire la valeur de l'attribut **id** de l'étudiant.

Code SQL de création de la table « étudiant » réalisée avec phpMyAdmin :

```
CREATE TABLE `etudiant` (
  `id` int(10) unsigned NOT NULL default '0',
  `nom` varchar(30) NOT NULL default '',
  `uv1` varchar(30) default NULL,
  `duree1` tinyint(4) default NULL,
  `note1` tinyint(4) default NULL,
  `uv2` varchar(30) default NULL,
  `duree2` tinyint(4) default NULL,
  `note2` tinyint(4) default NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `uv1` (`uv1`,`uv2`)
) TYPE=MyISAM;
```

### Note :

Si nous voulions respecter les règles énoncées au chapitre 13, il serait grandement préférable de définir un MCD comprenant une entité « étudiant », une entité « uv » et une association « passe » reliant ces entités. Le MLD correspondant amènerait à la création de trois tables.

Script d'insertion des données dans la table « étudiant ».

```
<?php
$xml=simplexml_load_file("iut.xml");
//Connexion à la base
include_once("connex.inc.php");
$idcom= connex("iut","myparam");
//Lecture du contenu des éléments
for($i=0;$i<5;$i++)
{
    echo "Insertion n° $i";
    $id=$xml->etudiant[$i][id];
    $nom=$xml->etudiant[$i][nom];
    $uv1=$xml->etudiant[$i]->uv[0]->nom;
```

```

$duree1=$xml->etudiant[$i]->uv[0]->duree;
$note1=$xml->etudiant[$i]->uv[0]->note;

$uv2=$xml->etudiant[$i]->uv[1]->nom;
$duree2=$xml->etudiant[$i]->uv[1]->duree;
$note2=$xml->etudiant[$i]->uv[1]->note;

$requete= "INSERT INTO
etudiant (id,nom,uv1,duree1,note1,uv2,duree2,note2)
VALUES ('$id','$nom','$uv1','$duree1','$note1','$uv2','$duree2','$note2')";
//Envoi de la requête d'insertion
$verif=mysql_query($requete,$idcom);
if(!$verif) echo "NON INSERE";
}
?>

```

## Exercices du chapitre 18

### TP n° 1. Un site de rencontres

Ce premier thème, déjà ébauché au chapitre 16, est le plus simple.

Son cahier des charges est le suivant :

- Le site offre aux internautes sportifs la possibilité d'entrer en contact avec d'autres personnes pratiquants ou supporters d'un même sport dans un département donné.
- Le site est réalisé avec PHP et une base SQLite.
- Chaque page affiche un en-tête commun.
- Pour avoir accès aux informations du site, chaque visiteur s'enregistre au préalable en tant que pratiquant ou supporter d'au moins un sport.
- L'identification se fait par le biais de l'e-mail du visiteur. Une fois identifié, le visiteur a accès à une page de recherche affichant les coordonnées des personnes qui répondent aux critères qu'il a définis. L'autorisation d'accès et l'e-mail sont stockés dans un cookie.
- Un visiteur non enregistré souhaitant accéder à la page de recherche est redirigé automatiquement vers la page d'inscription.

- Si un visiteur déjà identifié veut s'inscrire pour un autre sport que celui de sa première inscription, le formulaire affiche ses coordonnées automatiquement dans le formulaire afin de lui faciliter la saisie.

## L'interface

L'interface comprend trois pages, la page d'accueil, la page d'inscription et la page de recherche, chacune dotée de fonctionnalités spécifiques.

### La page d'accueil

Nommée `index.php`, la page d'accueil contient les éléments suivants :

- En-tête commun.
- Liste des sports existants dans la base.
- Zone de saisie de l'e-mail pour identifier le visiteur. Si l'e-mail figure déjà dans la base, un message de bienvenue s'affiche avec le nom du visiteur, et les données personnelles du visiteur sont enregistrées dans un cookie. Deux nouveaux liens sont créés dynamiquement, un vers la page de recherche et un vers la page d'inscription, permettant de s'enregistrer pour un nouveau sport. Si l'e-mail ne figure pas dans la base, le visiteur est redirigé automatiquement vers la page d'inscription.
- Lien vers la page d'inscription pour les personnes non encore enregistrées.

En-tête commun à toutes les pages : fichier `entete.htm`

```
<!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>
<title> Rencontres sportives </title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
</head>
<body>
<div style="border-bottom-width:4px; border-bottom-color:red;
border-bottom-style:groove;">

</div>
```

La page d'accueil : le fichier `index.php`

```
<?php
//Vérification de l'existence des cookies
```

```

if(isset($_COOKIE['nom']))
{
    $nom=$_COOKIE['nom'];
    $prenom=$_COOKIE['prenom'];
    $message= "<h3>Bonjour $prenom $nom </h3>";
    $lien1="<a href=\"recherche.php\" title=\"Recherche\"><button
type=\"button\">Rechercher des partenaires </button></a>";
    $lien2="<a href=\"ajout.php\" title=\"S'inscrire\"><button
type=\"button\">S'inscrire pour un sport </button></a>";
}
if(isset($_POST['mail']))
{
    //Récupération du mail
    $mail=strtolower($_POST['mail']);
    //Ouverture de la base
    $id_base=mysqli_open('C:/wamp/www/sqlitemanager/sportifs',
0666,$erreur);
    $requete="SELECT id_personne,nom,prenom FROM personne WHERE
mail='$mail'";
    $result=mysqli_query($id_base,$requete);
    $ligne=mysqli_fetch_array($result,MYSQL_NUM);
    if(!empty($ligne))
    {
        $message= "<h3>Bonjour $ligne[2] $ligne[1] </h3>";
        $lien1="<a href=\"recherche.php\" title=\"Recherche\"><button
type=\"button\">Rechercher des partenaires </button></a>";
        $lien2="<a href=\"ajout.php\" title=\"S'inscrire\"><button
type=\"button\">S'inscrire pour un sport </button></a>";
    }
    //Création des cookies
    setcookie('nom',$ligne[1],time()+30*24*3600);
    setcookie('prenom',$ligne[2],time()+30*24*3600);
}
mysqli_close($id_base);
}
include_once('entete.htm');
$id_base=mysqli_open('C:/wamp/www/sqlitemanager/sportifs',
0666,$erreur);
//Lecture de la table sport
$tabresult = mysqli_array_query($id_base,'SELECT design FROM sport
ORDER BY design ASC',MYSQL_NUM);
echo "<div style=\" position:absolute;
left:0px;width:20%;background-color:#CCFF66\">";
echo "<b> Liste des sports existants</b><br /><br />";
mysqli_close($id_base);
foreach ($tabresult as $valeur)
{
    echo ucfirst($valeur[0]),"<br />";
}
?>
<form method="post" action="<?=$_SERVER['PHP_SELF'] ?>">
    <fieldset>
</fieldset>
<legend> Identification </legend>

<label>Votre mail :
    <input type="text" name="mail" />
</label>

```



## La page d'inscription : le fichier ajout.php.

```
<?php
include_once('entete.htm');
?>
<form method="post" action="ajout.php" >
<!-- Premier groupe -->
<fieldset>
<legend>Vos coordonnées </legend>
<table>
<tbody>
<tr>
<td>Nom : </td>
<td><input type="text" name="nom" size="30" /> </td>
</tr>
<tr>
<td>Prénom : </td>
<td><input type="text" name="prenom" size="30" /> </td>
</tr>
<tr>
<td>Département : </td>
<td><input type="text" name="depart" size="30" /> </td>
</tr>
<tr>
<td>Mail : </td>
<td><input type="text" name="mail" size="30" /> </td>
</tr>
</tbody>
</table>
</fieldset>
<!-- Deuxième groupe -->
<fieldset>
<legend>Vos pratiques sportives </legend>
<table>
<tbody>
<!-- Sélection des sports -->
<tr>
<td>Sport pratiqué : </td>
<td>
<select name="design">
<option value="NULL">Choisissez!</option> ";
<?php
//Création dynamique de la liste de sélection
// Connexion
$id_base=sqlite_open('C:/wamp/www/sqlitemanager/sportifs',
0666,$erreur);
//Lecture de la table sport
$stabresult = sqlite_array_query($id_base,'SELECT id_sport,design
FROM sport ORDER BY design',SQLITE_ASSOC);
foreach ($stabresult as $valeur)
{
echo "<option value=\"\",$valeur[\"id_sport\"],\"\" >\",
$valeur[\"design\"],\"</option>\";
}
//Fermeture
```

```

sqlite_close($id_base);
?>
</select>
    OU : Ajouter un sport à la liste <input type="text"
name="nomsport" size="30"/><input type="submit" name="ajout"
value="Ajouter"/>
</td>
</tr>
<tr>
<td>Niveau : </td>
<td>
<select name="niveau">
<option value="1">Débutant</option>
<option value="2">Confirmé</option>
<option value="3">Pro</option>
<option value="4">Supporter</option>
</select>
</td>
</tr>
<tr>
<td><input type="submit" name="envoi" value=" Envoyer " /> </td>
<td><input type="reset" name="efface" value=" Effacer " /> </td>
</tr>
</tbody>
</table>
</fieldset>
</form>
<a href="index.php" title="Accueil"><button type="button"> Accueil
</button></a>
<!-- CODE PHP -->
<?php
// Connexion
if(!$id_base=@sqlite_open('C:/wamp/www/sqlitemanager/sportifs',
0666,$erreur)) {echo "ERREUR : $erreur <br />";}
//else{echo "Base ouverte";}
if(isset($_POST["envoi"]) )
{
//*****
//insert dans personne
$req_pers="INSERT INTO personne
(id_personne,nom,prenom,depart,mail) VALUES (NULL, '
".$_POST['nom'].'.','".$_POST['prenom'].'.','".$_POST['depart'].'.','
".$_POST['mail'].'.)";
if(sqlite_query($id_base,$req_pers))
{$id_personne=sqlite_last_insert_rowid($id_base);}
//*****
//Insertion dans la table pratique
$id_sport=$_POST["design"];
$niveau=$_POST["niveau"];
$req_pratique="INSERT INTO pratique (id_personne,id_sport,niveau)
VALUES ('$id_personne','$id_sport','$niveau)";
if(sqlite_query($id_base,$req_pratique))
{
echo "<a href=\"recherche.php\" title=\"Recherche\"><button
type=\"button\">Rechercher des partenaires </button></a><br />";
echo "Votre numéro d'enregistrement : ",$id_personne,"<br />";
}
}
}

```

```
//AJOUT DE SPORT
if(isset($_POST["ajout"]) )
{
$req_sport="INSERT INTO sport (id_sport,design) VALUES (NULL,
'".$_POST['nomsport'].')";
if(sqlite_query($id_base,$req_sport))
{
echo "DONNEES INSEREES dans sport";
echo "<script type=\"text/javascript\"> window.location.href=
\"ajout.php\" </script>";
}
}
sqlite_close($id_base);
?>
</body>
</html>
```

## La page de recherche

Nommée `recherche.php`, la page de recherche contient les éléments suivants :

- En-tête commun.
- Formulaire de saisie contenant trois listes de sélection :
  - La première contient la liste des sports existants. Elle est construite dynamiquement à partir des données de la table `sport`, comme dans la page d'inscription.
  - La deuxième indique le niveau des pratiquants et comporte les mêmes valeurs que dans le formulaire d'inscription.
  - La troisième contient la liste des départements dans lesquels il existe des personnes inscrites. Elle est construite dynamiquement à partir des données de la table `personne`.
- Lien vers la page d'accueil.
- Lien vers la page d'inscription.
- Script traitant les informations saisies et affichant la liste des partenaires correspondants dans un tableau HTML. Les données saisies sont réaffichées dans le formulaire afin de faciliter une éventuelle nouvelle recherche de l'utilisateur.

La page de recherche : le fichier `recherche.php`

```
<?php
include_once('entete.htm');
?>
<h3>Le site des rencontres sportives</h3>
```

```

<form method="post" action="recherche.php" >
<!-- Premier groupe -->
<fieldset>
<legend>Rechercher des partenaires </legend>
<table>
<tbody>
<!-- Ligne 1 -->
<tr>
<td>Sport pratiqué : </td>

<td>
  <select name="design">
<option value="NULL">Choisissez!</option>  ";
<?php
//Création dynamique de la liste de sélection
// Connexion
$id_base=mysqli_open('C:/wamp/www/sqlitemanager/sportifs',
0666,$erreur);
//Lecture de la table sport
$stabresult = mysqli_array_query($id_base,'select id_sport,design
from sport',MYSQLI_ASSOC);
foreach ($stabresult as $valeur)
{
echo "<option value=\"", $valeur["id_sport"], "\" >",
$valeur["design"] , "</option>";
}
//Fermeture
mysqli_close($id_base);
?>
  </select>

</td>
</tr>
<!-- Ligne 2 -->
<tr>
<td>Niveau : </td>
<td>
<select name="niveau">
<option value="1">Débutant</option>
<option value="2">Confirmé</option>
<option value="3">Pro</option>
<option value="4">Supporter</option>
  </select>
</td>
</tr>
<!-- Ligne 3 -->
<tr>
<td>Département : </td>
<td>
  <select name="depart">
<option value="NULL">Choisissez!</option>
<?php
//Création dynamique de la liste de sélection des départements
// Connexion
$id_base=mysqli_open('C:/wamp/www/sqlitemanager/sportifs',
0666,$erreur);

```

```

//Lecture de la table personne
$result=sqlite_query($id_base,'select depart from
personne',SQLITE_NUM);
//Création du tableau des départements existants
for($i=0;$i<sqlite_num_rows($result);$i++)
{
    $stabdepart[$i]=sqlite_fetch_single($result);
}
//Elimination des doublons
$stabdepart=array_unique($stabdepart);
//Tri des départements
sort($stabdepart);
//Création de la liste d'options
for($i=0;$i<count($stabdepart);$i++)
{
    echo "<option value=\"", $stabdepart[$i], "\" >", $stabdepart[$i]
    , "</option>\n";
}
//Fermeture
sqlite_close($id_base);
?>
</select>

</td>
</tr>
<tr>
<td><input type="submit" name="envoi" value=" Recherche "/> </td>
<td><input type="reset" name="efface" value=" Effacer "/> </td>
</tr>
</tbody>
</table>
</fieldset>
</form>
<!-- CODE PHP RECHERCHE-->
<?php
// Connexion
if(isset($_POST["envoi"]))
{
    if(!$id_base=@sqlite_open('C:/wamp/www/sqlitemanager/sportifs',
0666,$erreur)) {echo "ERREUR : $erreur <br />";}
    else
    {
        $requete="SELECT nom,prenom,mail,design FROM
personne,pratique,sport WHERE
personne.id_personne=pratique.id_personne AND
sport.id_sport=pratique.id_sport AND pratique.niveau='".
$_POST['niveau']."' AND sport.id_sport='".$_POST['design']."' AND
personne.depart='".$_POST['depart']."'";
        $result=sqlite_query($requete,$id_base);

        echo "<table border=\"1\" rules=\"rows\" width=\"100%\" >";
        echo "<tr><th colspan=\"3\">Liste des partenaires
disponibles</th>";

        while($ligne=sqlite_fetch_array($result,SQLITE_ASSOC))
    {

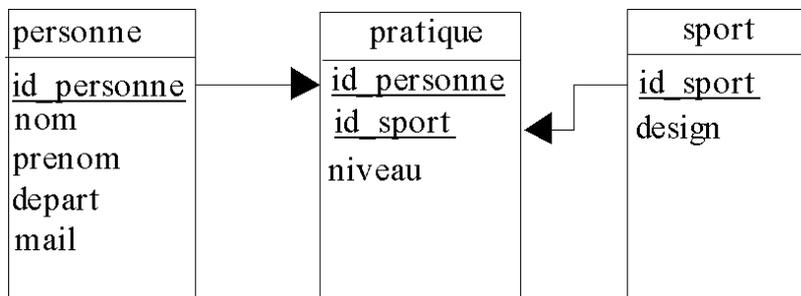
```



primaire des tables qu'elle associe. Elle a de plus un attribut niveau.

Figure 18-4

Le modèle logique de données (MLD) de la base



## TP n° 2. Dictionnaire de citations interactif

Ce TP est une première mise en œuvre simple d'une base de données MySQL. Le projet consiste à créer un dictionnaire de citations littéraires interactif en ligne. Il ne s'agit pas donc d'une banque de données statique mise en consultation. Chaque visiteur peut en enrichir le contenu avec ses citations préférées, qui sont ensuite rendues accessibles à tous. Le concept du site se rapproche de celui d'un forum puisque les données ne sont pas figées.

### L'interface

Pour créer une unité dans le site, chaque page doit incorporer les mêmes en-tête et pied de page. L'interface comprend trois pages, la page d'accueil, la page d'affichage des résultats et la page d'insertion.

### La page d'accueil

Nommée `index.php`, la page d'accueil comporte, outre les éléments décoratifs laissés à votre libre choix, les éléments suivants :

- Bandeau contenant la citation du jour tirée au sort dans la base et affichée lors de chaque connexion.
- Formulaire de recherche contenant une zone de saisie de texte dans laquelle le visiteur saisit un mot-clé de recherche d'une

citation. Il peut aussi préciser sa recherche en choisissant dans une liste de sélection parmi les auteurs présents dans la base. Cette liste est construite dynamiquement en interrogeant la base. Un dernier critère de sélection est constitué d'une seconde liste de sélection permettant de choisir le siècle des citations, du XVI<sup>e</sup> au XXI<sup>e</sup>. Le critère de tri des résultats se fait par auteur ou par siècle en fonction du choix effectué à l'aide de boutons radio. Aucun de ces critères n'étant obligatoire, chaque choix doit posséder une valeur par défaut consistant à afficher l'ensemble des citations. Le script de traitement de ce formulaire se trouve sur la page d'affichage des résultats.

- Lien vers la page d'insertion de nouvelles citations.

L'en-tête commun à toutes les pages : le fichier `entete.htm`

```
<!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>
<title> Dictionnaire de citations interactif </title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
<style type="text/css" title="">
    h1{font-size:50px;margin:0px;text-align:center;}
</style>
</head>
<body>
<div style="border-bottom-width:4px;border-bottom-color:red;border-
bottom-style:groove;">
<h1>&nbsp;
Dictionnaire de citations interactif</h1>
</div>
```

La page d'accueil : le fichier `index.php`

```
<?php
include_once('entete.htm');
include_once("connex.inc.php");
$idcom=connex("dico","myparam");
//Sélection de tous les identifiants des citations
$requete = "SELECT idcit FROM citation";
$result = mysql_query($requete,$idcom);
//Calcul du nombre total de citations
$nbcitation=mysql_num_rows($result);
$tirage=rand(1,$nbcitation);
$requetecit = "SELECT texte,nom,prenom,siecle FROM citation,auteurs
WHERE idcit='$tirage' AND citation.idauteur=auteurs.idauteur";
//$requetecit = "SELECT texte FROM citation WHERE idcit='$tirage'
";
$resultcit = mysql_query($requetecit,$idcom);
$citdujour=mysql_fetch_object($resultcit);
mysql_close($idcom);
echo "<hr />";
echo "<h4>\$"citdujour->texte \" ";
echo "<cite>$citdujour->prenom $citdujour->nom ($citdujour->siecle
<sup>ème</sup> siècle)</cite></h4>";
```

```

echo "<hr />";
?>
<form action="affichecit.php" method="post">
<fieldset>
<legend><b>Rechercher une citation</b></legend>
<table width="100%" border="1" align="left" bordercolor="#800000"
summary="Recherche">
<tr>
<td colspan="2" align="center"><h2>Recherche de citations</h2></td>
</tr>
<tr>
<td>Mot-clé </td>
<td><input type="text" name="motcle" size="40" maxlength="256">
</td>
</tr>
<tr>
<td>Auteur</td>
<td>
<?php
$idcom=connex("dico","myparam");
$requete = "SELECT nom FROM auteurs ORDER BY nom ASC";
$result = mysql_query($requete,$idcom);
echo "<select name=\"auteur\" size=\"1\" >";
echo "<option value=\"tous\"> Tous</option>";
while($stab = mysql_fetch_assoc($result))
{
echo "<option value=\"".$stab["nom"]."\" >".
$stab["nom"]."</option>";
//echo $stab["nom"],"<br />";
}
echo "</select>";
mysql_close($idcom);
?>
</td>
</tr>
<tr>
<td>Siècle</td>
<td>
<select name="siecle" size="1">
<option value=""> Tous</option>
<option value="16"> 16ème siècle</option>
<option value="17"> 17ème siècle</option>
<option value="18"> 18ème siècle</option>
<option value="19"> 19ème siècle</option>
<option value="20"> 20ème siècle</option>
<option value="21"> 21ème siècle</option>
</select>
</td>
</tr>
<tr>
<td colspan="2">Trier par : auteur <input type="radio" name="tri"
value="nom" checked="checked" />
siècle : <input type="radio" name="tri" value="siecle" /></td>
</tr>
<tr>

```



```

$query . = " 1 " ; }
//si auteur différent de 'tous'
if($auteur != "tous")
{ $query . = "AND nom= '$auteur' "; }
else{ $query . = ""; }
//si siecle
if($siecle){ $query . = " AND siecle = '$siecle' "; }
$query . = " AND A.idauteur= C.idauteur";
$query . = " ORDER BY $stri ASC";
//*****
//connexion au serveur MySQL
//*****
include_once("connex.inc.php");
$idcom=connex("dico", "myparam");
//Envoi de la requête
$result = mysql_query($query, $idcom) or die("ERREUR RESULTAT");
//Affiche résultats
if(!$result) echo "<h2> Pas de réponse </h2>";
else
{
    echo "<h2> Liste des citations : $motcle</h2> <hr>";
    while($stab = mysql_fetch_assoc($result))
    {
        echo "<h4>", nl2br($stab["texte"]), "<br />";
        echo "<cite>", $stab["prenom"], " ", $stab["nom"], " ( ",
        $stab["siecle"], "<sup>ème</sup> siècle) </cite></h4><hr />";
    }
}
include_once('pied.htm');
?>

```

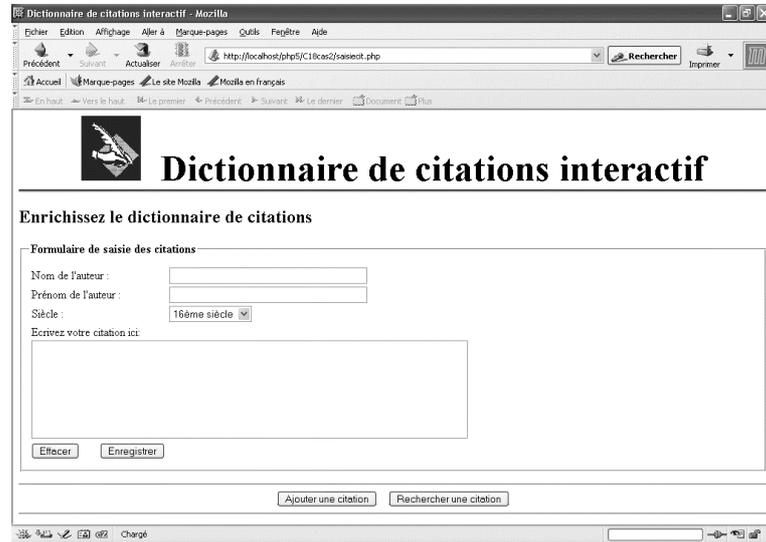
## La page d'insertion

Nommée `saisiecit.php` la page d'insertion comprend les éléments suivants :

- Formulaire contenant deux zones de saisie de texte pour le nom et le prénom de l'auteur, une liste de sélection du siècle, une zone de saisie multiligne pour le texte de la citation, ainsi que les habituels boutons d'effacement et d'envoi.
- Script de traitement des données situé dans le fichier lui-même, devant vérifier si l'auteur existe déjà dans la base puis insérer les données et afficher un avis d'insertion pour le visiteur.
- Lien vers la page d'accueil.

La figure 18-5 donne une idée de ce que pourrait être le formulaire d'insertion.

Figure 18-5  
*Le formulaire d'insertion*



La page d'insertion : le fichier `saisiecit.php`

```
<?php
include_once('entete.htm');
?>
<h2>Enrichissez le dictionnaire de citations</h2>
<form action="<?= $PHP_SELF?>" method="post">
<fieldset>
<legend><b>Formulaire de saisie des citations</b></legend>
<table>
<tbody>
<tr>
<td>Nom de l'auteur :</td>
<td><input type="text" name="nom" size="40" maxlength="256"><br
/></td>
</tr>
<tr>
<td>Prénom de l'auteur : </td>
<td><input type="text" name="prenom" size="40" maxlength="256"><br
/>
</td>
</tr>
<tr>
<td>Siècle : </td>
<td>
<select name="siecle" size="1">
<option value="16"> 16ème siècle</option>
<option value="17"> 17ème siècle</option>
<option value="18"> 18ème siècle</option>
<option value="19"> 19ème siècle</option>
<option value="20"> 20ème siècle</option>
<option value="null"> Autres</option>

```



```

mysql_query($queryins,$idcom)or die("ERREUR D'INSERTION
auteurs");
$idauteur= mysql_insert_id();
$query = "INSERT INTO citation (texte,idauteur) VALUES
('$texte',$idauteur) ";
mysql_query($query,$idcom)or die("ERREUR D'INSERTION
citation");
}
}
include_once('pied.htm');
?>

```

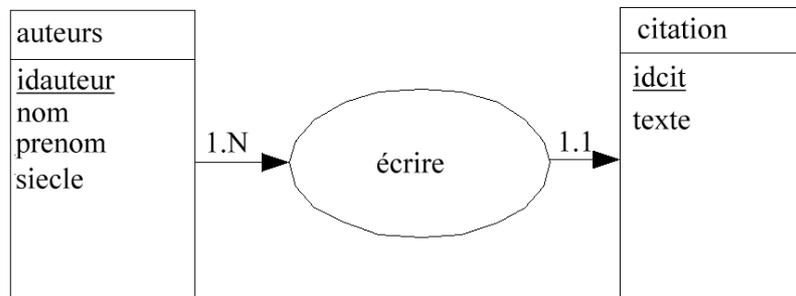
## La base de données MySQL

Nommée *dico*, la base de données doit répondre au modèle conceptuel de données représenté à la figure 18-6. Les contraintes sont les suivantes :

- Un auteur peut avoir écrit plusieurs citations.
- Une citation donnée ne peut être l'œuvre que d'un seul auteur.

Figure 18-6

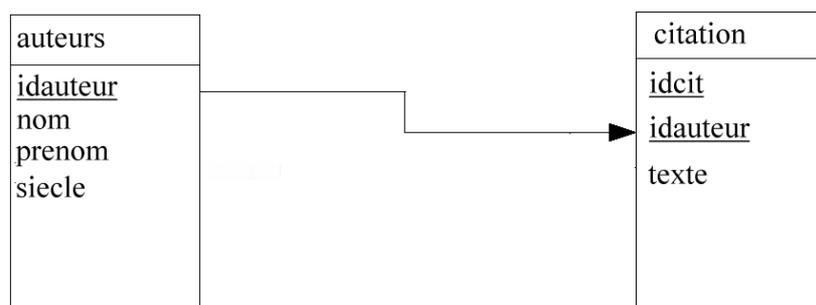
*Le modèle conceptuel de données de la base dico*



Le modèle logique de données réalisé en appliquant les règles du chapitre 13 est illustré à la figure 18-7. Il montre que la base *dico* ne contient que deux tables.

Figure 18-7

*Le modèle logique de données de la base dico*



### Structure de la table auteurs

```
CREATE TABLE `auteurs` (  
  `idauteur` int(11) NOT NULL auto_increment,  
  `nom` varchar(30) NOT NULL default '',  
  `prenom` varchar(30) default '',  
  `siecle` tinyint(2) default NULL,  
  PRIMARY KEY (`idauteur`)  
)
```

### Structure de la table citation

```
CREATE TABLE `citation` (  
  `idcit` int(11) NOT NULL auto_increment,  
  `texte` mediumtext NOT NULL,  
  `idauteur` int(11) NOT NULL default '0',  
  PRIMARY KEY (`idcit`)  
)
```

## TP n° 3. Commerce en ligne

Ce troisième et dernier thème constitue la partie la plus difficile de votre travail puisqu'il a pour but de vous placer dans une situation professionnelle.

Vous vous placerez en tant que concepteur indépendant vis-à-vis d'un client qui n'est pas un professionnel du Web mais un commerçant. Il ne peut donc exprimer ses besoins dans un langage technique propre à Internet et à la programmation PHP, dont il ignore tout.

### Les besoins du client

Le client dirige une PME qui vend des produits informatiques et vidéo. Vous devez lui créer un site de commerce en ligne pour vendre ses produits par correspondance. Son budget ne lui permettant pas de vous rémunérer en permanence pour maintenir le site, vous lui créez une interface Internet à accès privé lui permettant d'ajouter ou d'enlever des produits dans la base de données sans avoir à passer par un logiciel FTP.

### Fonctionnement du site

La solution proposée comporte sept fichiers :

- Le fichier `entete.htm` qui constitue l'en-tête commun à toutes les pages.



```

{
//Récupération des saisies
$motcle= $_POST['motcle'];
$categorie= $_POST['categorie'];
$tri= $_POST['tri'];
//Création de la requête
$requete = "SELECT id_article,design,prix FROM article WHERE ";
if($motcle) $requete.=" design LIKE '%$motcle%' ";
if($categorie != "") $requete.=" AND categorie='$categorie' ";
$requete.=" ORDER BY $tri";
$result=mysql_query($requete,$idcom);
$nbart=mysql_num_rows($result);
echo "<h3>Il y a $nbart articles répondant à votre recherche</h3>";
while($stab = mysql_fetch_assoc($result))
{
echo"<form action=\"panier.php\" method=\"post\">";
echo "<div class=\"bord\"><b>",$stab['design'] ,"</b><br /> Prix
unitaire : ",$stab['prix'] ," &euro; <br /> Référence :
",$stab['id_article']," <br /> Choisir la quantité : <input
type=\"text\" name=\"quantite\" size=\"2\" maxlength=\"2\"
value=\"0\"/> <input type=\"submit\" value=\"Commander\" /> <input
type=\"hidden\" name=\"id_article\"
value=\"",$stab['id_article'],\" /> <input type=\"hidden\"
name=\"prix_unit\" value=\"",$stab['prix'],\" /><input
type=\"hidden\" name=\"design\" value=\"",$stab['design'],\" /></div>";
echo"</form>";
}
}
?>
</div>
</div>
<?php
include_once('pied.htm');
?>

```

- Le fichier panier.php qui gère les articles mis en panier et affiche le contenu du panier.

```

<?php
session_start();
//Récupération des saisies
$id_article=$_POST['id_article'];
$quantite=$_POST['quantite'];
$prix_unit=$_POST['prix_unit'];
$design=$_POST['design'];
unset($_POST);
//SESSION
$_SESSION['id_article'][]=$id_article;
$_SESSION['quantite'][]=$quantite;
$_SESSION['prix_unit'][]=$prix_unit;
$_SESSION['design'][]=$design;
include_once('entete.htm');
include_once("connex.inc.php");
?>
<h3>Votre panier</h3>
<table border="1" >

```

```

<tr>
<th>Code Article</th>
<th>Désignation</th>
<th>Quantité</th>
<th>Prix unitaire</th>
<th>Prix total</th>
<?php
//Connexion
$idcom=connex("magasin","myparam");
//Affichage du panier
$total=0;
for($i=0;$i<count($_SESSION['id_article']);$i++)
{
    $requete = "SELECT id_article,design,prix FROM article WHERE
id_article='".$_SESSION['id_article'][$i].'";
    $result=mysql_query($requete,$idcom);
    while($stab = mysql_fetch_assoc($result))
    {
        echo
" <tr><td>",$stab['id_article'], "</td><td>",$stab['design'], "</td><td>
",$_SESSION['quantite'][$i], "</td><td
align=\"right\">",$stab['prix'], "</td><td align=\"right\"> ",
round($stab['prix']*$_SESSION['quantite'][$i],2), "</td></tr>";
$total+=$stab['prix']*$_SESSION['quantite'][$i];
    }
}
echo "<tr><td colspan=\"4\"><b>Prix total T.T.C</b></td><td
align=\"right\"><b>$total</b></td></tr>";
echo "</table>";
?>
<a href="saisieclient.php"><button type="button"> Terminer la
commande </button></a>
<?php
include_once('pied.htm');
?>

```

- Le fichier `saisieclient.php` qui permet la saisie des coordonnées du client déjà enregistré ou pas.

```

<?php
session_start();
include_once('entete.htm');
?>
<h2>Votre compte client</h2>
<h4>Vous êtes déjà enregistré</h4>
<form action="<?=$_SERVER['PHP_SELF']?>" method="post">
<fieldset>
<legend><b>Vos coordonnées </b></legend>
<table>
<tbody>
<tr>
<td>Mail : </td>
<td> <input type="text" name="mail" size="60" maxlength="60" /><br
/>
</td>
</tr>
<tr>
<td>Code personnel </td>

```

```

<td> <input type="password" name="code" size="10" maxlength="8"
:><br />
</td>
</tr>
<tr>
<td colspan="2">
<input type="submit" value="Vérifier" name="envoi" />
</td>
</tr>
</tbody>
</table>
</fieldset>
<h4>Vous n'êtes pas encore client</h4>
<fieldset>
<legend><b>Vos coordonnées </b></legend>
<table>
<tbody>
<tr>
<td>Nom :
</td>
<td> <input type="text" name="nom" size="40" maxlength="40" /><br
/>
</td>
</tr>
<tr>
<td>Prénom : </td>
<td> <input type="text" name="prenom" size="40" maxlength="40"
/><br />
</td>
</tr>
<tr>
<td>Age : </td>
<td> <input type="text" name="age" size="10" maxlength="2" /><br />
</td>
</tr>
<tr>
<td>Adresse : </td>
<td> <input type="text" name="adresse" /><br />
</td>
</tr>
<tr>
<td>Ville : </td>
<td> <input type="text" name="ville" />
<br />
</td>
</tr>
<tr>
<td>Mail : </td>
<td> <input type="text" name="mail2" size="60" maxlength="60" /><br
/>
</td>
</tr>
<tr>
<td>Code personnel <br>(8 caractères maxi) </td>

```



```

$mail= $_POST["mail"];
$code= $_POST["code"];
//Sessions
$_SESSION['nom']=$nom;
$_SESSION['prenom']=$prenom;
$_SESSION['age']=$age;
$_SESSION['adresse']=$adresse;
$_SESSION['ville']=$ville;
$_SESSION['mail']=$mail;
$_SESSION['code']=$code;
$_SESSION['nouveau']=TRUE;
//Redirection vers la page de paiement
echo "<script type=\"text/javascript\">
window.location='paiement.php' </script>";
}
else echo "<script type=\"text/javascript\">alert('Erreur de
saisie'); </script>";
}
include_once('pied.htm');
?>

```

- Le fichier `luhn.php` qui contient la fonction de vérification de la carte bancaire.

```

<?php
function luhn($carte)
{
    $verif=0;
    $N =array();
    for($i=0;$i<strlen($carte);$i++)
    {
        $N[$i]=(integer) $carte[$i];
        if($i%2==0)
        {
            $N[$i]= 2*$N[$i];
            if($N[$i] > 9) {$N[$i]=$N[$i]-9;}
        }
        $total+=$N[$i];
    }
    if($total%10 ==0) return TRUE;
    else return FALSE;
}
?>

```

- Le fichier `paiement.php` qui permet la saisie et la vérification du numéro de carte bancaire puis l'enregistrement des données dans la base MySQL et l'envoi des mails.

```

<?php
session_start();
include_once('luhn.php');
include_once('entete.htm');
include_once("connex.inc.php");
?>
<div class="bord">
<h2> Paiement </h2>

```

```

<form action="<?=$_SERVER['PHP_SELF'] ?>" method="post">
<fieldset>
Carte bancaire <input type="text" name="carte" size="16"
maxlength="16" />
<input type="submit" name="envoi" value=" Payer " />
</fieldset>
</form>
</div>
<?php
if(isset($_POST['envoi']))
{
//Récupération des saisies
$carte= $_POST['carte'];
//Vérification de la carte bancaire
if(luhn($carte))
{
$idcom=connex("magasin","myparam");
$nom= $_SESSION['nom'];
$prenom= $_SESSION['prenom'];
$age= $_SESSION['age'];
$adresse= $_SESSION['adresse'];
$ville= $_SESSION['ville'];
$mail= $_SESSION['mail'];
$code= $_POST["code"];
//Nouveau client
if($_SESSION['nouveau'])
{
$id_client="\N";
$requete = "INSERT INTO client VALUES('$id_client',
'$nom','$prenom', '$age','$adresse', '$ville','$mail', '$code)";
$result=mysql_query($requete,$idcom);
$id_client=mysql_insert_id($idcom);
}
//Déjà client
else
{
$id_client=$_SESSION['id_client'];
}
$date= date("Y-m-d");
//Insertion dans la table commande
$id_comm="\N";
$requete = "INSERT INTO commande VALUES('$id_comm',
'$id_client','$date)";
$result=mysql_query($requete,$idcom);
$id_comm=mysql_insert_id($idcom);
//boucle d'insertion dans la table ligne
for($i=0;$i<count($_SESSION['id_article']);$i++)
{
$requete = "INSERT INTO ligne
VALUES('$id_comm','" . $_SESSION['id_article'][$i]."', '" . $_SESSION['q
uantite'][$i]."', '" . $_SESSION['prix_unit'][$i]."'");
$result=mysql_query($requete,$idcom);
if($result) echo "ligne $i inserée <br />";
}
//Envoi des mails
$dest=$_SESSION['mail'].",depot@monecommerce.com";

```

```

$objet="Votre commande";
$texte="Bonjour ".$_SESSION['prenom']."
".$_SESSION['nom']."\n";
$texte.="$_SESSION['adresse']."\n";
$texte.="$_SESSION['ville']."\n";
$texte.="Nous vous confirmons la commande de :\n";
$total=0;
for($i=0;$i<count($_SESSION['id_article']);$i++)
{
    $texte.="$_SESSION['quantite'][$i]."
".$_SESSION['design'][$i]."au prix de
:".$_SESSION['prix_unit'][$i]." soit : ".
".$_SESSION['quantite'][$i]*$_SESSION['prix_unit'][$i]."\n";
    $quant=(double) $_SESSION['quantite'][$i];
    $prixunit=(double) $_SESSION['prix_unit'][$i];
    $total+=$quant*$prixunit;
}
$texte.="Montant total de la commande =". $total."\n";
echo "<h4>$texte</h4>";
mail($dest,$objet,$texte);
session_destroy();
}
else "Carte bancaire non valable";
}
include_once('pied.htm');
?>

```

Le site est visible en fonctionnement réel à l'adresse suivante :

<http://www.funhtml.com/php5/>

Toutes vos remarques peuvent être envoyées à l'adresse :

[php5@funhtml.com](mailto:php5@funhtml.com)

Merci et bonne lecture.