

Présentation de l'exercice.

Comme la question revient très fréquemment sur le forum et sur le chat. Je vais vous apprendre à créer simplement un affichage page par page à partir du résultat d'une requête SQL.
Cet exo est en quelque sorte une extension de l'exo n°15

```
<mode type="hors sujet" voix="voix d'hôtesse">
Ding ding dong ding !
Le capitaine vous souhaite un bon voyage sur notre ligne.
Le tuto est bien long... Alors respirez un bon coup, concentrez-vous! On y va!
</mode>
```

La table SQL.

Pour notre exercice, nous allons utiliser une table assez simple.

Requête de création et d'insertion des données

```
CREATE TABLE vaches (
  id int(11) NOT NULL auto_increment,
  surnom varchar(50) NOT NULL default '',
  prenom varchar(20) NOT NULL default '',
  PRIMARY KEY (id)
) TYPE=MyISAM;

INSERT INTO vaches VALUES (1, 'Qui aime les frites', 'Margueritte');
INSERT INTO vaches VALUES (2, 'Belle clochette', 'Roussette');
INSERT INTO vaches VALUES (3, 'La starlette', 'Blanchette');
INSERT INTO vaches VALUES (4, 'Petite Pataude', 'Noireau');
INSERT INTO vaches VALUES (5, 'Toujours chouette', 'Paquerette');
INSERT INTO vaches VALUES (6, 'Tête de linotte', 'Cowpilot');
INSERT INTO vaches VALUES (7, 'Qui saute partout', 'Kancowroo');
INSERT INTO vaches VALUES (8, 'La gentille', 'Cowchenille');
INSERT INTO vaches VALUES (9, 'Le plus bizarre des cows', 'Haricow');
```

Première étape : afficher tous les enregistrements dans une table HTML

Comme c'est une question qui revient aussi très souvent, je vais vous expliquer comment faire pour afficher des résultats dans une table HTML.

Normalement, avant d'avoir posé les yeux sur ce tuto, vous êtes censés avoir lu, compris et assimilé les tutos précédents. Si c'est le cas, le script ci-dessous ne devrait pas vous poser trop de problèmes.

Affichage des enregistrements dans une table HTML.

```
<html>
<body>
<?php
// information pour la connection à la DB
$host = 'localhost';
$user = 'root';
$pass = '';
$db = 'test';

// connection à la DB
```

```

$link = mysql_connect ($host,$user,$pass) or die ('Erreur : '.mysql_error() );
mysql_select_db($db) or die ('Erreur :'.mysql_error());

// requête SQL qui compte le nombre total d'enregistrements dans la table et qui
//récupère tous les enregistrements
$select = 'SELECT prenom,surnom FROM vaches';
$result = mysql_query($select,$link) or die ('Erreur : '.mysql_error() );
$total = mysql_num_rows($result);

// si on a récupéré un résultat on l'affiche.
if($total) {
    // début du tableau
    echo '<table bgcolor="#FFFFFF">'. "\n";
    // première ligne on affiche les titres prénom et surnom dans 2 colonnes
    echo '<tr>';
    echo '<td bgcolor="#669999"><b><u>Prénom</u></b></td>';
    echo '<td bgcolor="#669999"><b><u>Surnom</u></b></td>';
    echo '</tr>'. "\n";
    // lecture et affichage des résultats sur 2 colonnes, 1 résultat par ligne.
    while($row = mysql_fetch_array($result)) {
        echo '<tr>';
        echo '<td bgcolor="#CCCCC">'. $row['prenom']. '</td>';
        echo '<td bgcolor="#CCCCC">'. $row['surnom']. '</td>';
        echo '</tr>'. "\n";
    }
    echo '</table>'. "\n";
    // fin du tableau.
}
else echo 'Pas d\'enregistrements dans cette table...';

// on libère le résultat
mysql_free_result($result);

?>
</body>
</html>

```

Le résultat de ce code donnera ceci à l'écran :

<u>Prénom</u>	<u>Surnom</u>
Margueritte	Qui aime les frites
Roussette	Belle clochette
Blanchette	La starlette
Noireau	Petite Pataude
Paquerette	Toujours chouette
Cowpilot	Tête de linotte
Kancowroo	Qui saute partout
Cowchenille	La gentille
Haricow	Le plus bizarre des cows

Explications

Juste avant notre boucle, nous affichons une ligne de la table HTML.
Elle est constituée de 2 colonnes dans lesquelles nous mettons le titre de chacune d'elles sous cette forme :

Affichage des titres dans des colonnes d'une table HTML

```
<tr>
<td bgcolor="#669999"><b><u>Prénom</u></b></td>
<td bgcolor="#669999"><b><u>Surnom</u></b></td>
</tr>
```

Dans la boucle, nous allons afficher chaque enregistrement dans une ligne du tableau.
Le prénom dans une cellule et le surnom dans une autre, sous cette forme :

Affichage des données.

```
<tr>
<td bgcolor="#CCCCCC"><b><u>Roussette</u></b></td>
<td bgcolor="#CCCCCC"><b><u>Belle clochette</u></b></td>
</tr>
```

Explication d'une option d'un select SQL : Limit

Bien pratique, cette option nous permet de sélectionner une partie des enregistrements de la table en lui donnant le nombre d'enregistrements à récupérer à partir d'une position donnée.
Voici quelques exemples :

Syntaxe d'un SELECT avec une limite – Exemple 1

```
SELECT prenom,surnom FROM vache LIMIT 0,4;
```

Cette requête retournera 4 enregistrements à partir du premier de la liste (0 + 1).
Dans le cas de notre table :

<u>Prénom</u>	<u>Surnom</u>
Margueritte	Qui aime les frites
Roussette	Belle clochette
Blanchette	La starlette
Noireau	Petite Pataude

Syntaxe d'un SELECT avec une limite – Exemple 2

```
SELECT prenom,surnom FROM vache ORDER BY prenom ASC LIMIT 0,4;
```

Cette requête retournera 4 enregistrements à partir du premier de la liste dans l'ordre alphabétique.

Donc, la requête trie les enregistrements et ensuite en prend 4 à partir du premier (0 + 1).
Dans le cas de notre table :

<u>Prénom</u>	<u>Surnom</u>
Blanchette	La starlette
Cowchenille	La gentille
Cowpilot	Tête de linotte
Haricow	Le plus bizarre des cows

Syntaxe d'un SELECT avec une limite – Exemple 2

```
SELECT prenom,surnom FROM vache ORDER BY prenom ASC LIMIT 5,4;
```

Cette requête retournera 4 enregistrements à partir du 6ème (on en saute 5 , 5 +1) dans l'ordre alphabétique.

Voilà ce que donne le résultat :

<u>Prénom</u>	<u>Surnom</u>
Margueritte	Qui aime les frites
Noireau	Petite Pataude
Paquerette	Toujours chouette
Roussette	Belle clochette

Si vous voulez comparer avec ce que donnerait un select sans limit avec un tri alphabétique :

Comparaison de l'exemple 2 sans limit

```
SELECT prenom,surnom FROM vache ORDER BY prenom ASC;
```

Voilà le résultat :

<u>Prénom</u>	<u>Surnom</u>
Blanchette	La starlette
Cowchenille	La gentille
Cowpilot	Tête de linotte
Haricow	Le plus bizarre des cows
Kancowroo	Qui saute partout
Margueritte	Qui aime les frites
Noireau	Petite Pataude
Paquerette	Toujours chouette
Roussette	Belle clochette

Notre affichage page par page.

Je vais reprendre chaque partie du script de base pour le modifier afin de créer un affichage page par page.

Ouvrez bien vos yeux, c'est parti !

Première partie du script.

Première partie du script.

```
// information pour la connection à le DB
$host = 'localhost';
$user = 'root';
$pass = '';
$db = 'test';

// connection à la DB
$link = mysql_connect ($host,$user,$pass) or die ('Erreur : '.mysql_error() );
mysql_select_db($db) or die ('Erreur :'.mysql_error());
```

Explications

Cette partie-là ne change pas et ne comporte que des choses que vous êtes censés connaître.

- ♦ **\$host**, **\$pass**, **\$user** et **\$db** contiennent les informations pour se connecter à la base de données.
- ♦ Ensuite on se connecte à la base de données.

De quelles variables avons-nous besoin ?

Pour notre affichage page par page, nous avons besoin de connaître certaines données que l'on va stocker dans des variables.

- ♦ **\$nombre** : Nombre d'enregistrements que l'on veut afficher par page.
- ♦ **\$total** : Le nombre total d'enregistrements de la table .
- ♦ **\$limite** : Où en sommes-nous dans notre affichage.

Nous allons reprendre notre script de base et y ajouter ce dont nous avons besoin pour travailler. Au début du script nous allons initialiser les variables **\$nombre** et **\$limite**.

Initialisation des variables.

```
$nombre = 5; // on va afficher 5 résultats par page.
if (!isset($limite)) $limite = 0; // si on arrive sur la page pour la première fois
    // on met limite à 0.

$path_parts = pathinfo($_SERVER['PHP_SELF']);
$page = $path_parts['basename'];
```

Explications :

- ♦ **\$nombre** est initialisé à 5 (5 enregistrements affichés à la fois)
- ♦ Si **\$limite** n'existe pas on l'initialise à 0.
- ♦ Les 2 lignes de codes suivantes permettent de récupérer le nom de la page qui sera utilisée dans les liens.
- ♦ La variable **\$total** contient le nombre d'enregistrement à afficher au total.

Compte le nombre d'enregistrements total de la table.

```
$select = 'SELECT count(id) FROM vaches';

$result = mysql_query($select,$link) or die ('Erreur : '.mysql_error() );

$row = mysql_fetch_row($result);

$total = $row[0];
```

Ajout de la requête select avec le limit

Maintenant, nous avons besoin de faire notre requête de sélection qui va afficher **\$nombre** enregistrements à partir de **\$limite**.

Sélectionne la portion d'enregistrements à afficher.

```
// requête SQL qui ne prend que le nombre d'enregistrements nécessaire à l'affichage.

$select = 'select prenom,surnom FROM vaches ORDER BY prenom ASC limit '.$limite.','.$nombre;

$result = mysql_query($select,$link) or die ('Erreur : '.mysql_error() );
```

Explications

On utilise l'opérateur de concaténation pour compléter notre requête de base par le tri et la limite de sélection. Notre requête donnera donc au premier affichage de la page (**\$limite** étant initialisé à 0 la première fois):

Requête de sélection de la première page

```
SELECT prenom,surnom FROM vaches ORDER BY prenom ASC limit 0,5
```

Qui donnera ceci à l'écran :

<u>Prénom</u>	<u>Surnom</u>
Blanchette	La starlette
Cowchenille	La gentille
Cowpilot	Tête de linotte
Haricow	Le plus bizarre des cows

Calcul de la nouvelle limite d'affichage

Imaginons que nous avons affiché la page pour la première fois. Notre **\$limite** est à zéro, et notre script affiche donc 5 enregistrements à partir du début.

C'est pas mal, mais maintenant il faut savoir s'il reste d'autres enregistrements à afficher. Si oui il nous faut mettre un lien ou un bouton permettant de les afficher.

Et pour cela, il faut recalculer la limite d'affichage.

Nous voulons afficher les **\$nombre** enregistrements suivants et/ou les **\$nombre** enregistrements précédents:

Il s'agit d'un calcul très simple :

Calcul des limites précédente et suivante

```
$limitesuivante = $limite + $nombre;
$limiteprecedente = $limite - $nombre;
```

Voilà à quoi notre requête SQL ressemblera si on est sur la première page et que l'on clique sur le lien suivant.

Requête de sélection de la deuxième page

```
SELECT prenom,surnom FROM vaches ORDER BY prenom ASC limit 5,5
```

Qui donnera ceci à l'écran :

<u>Prénom</u>	<u>Surnom</u>
Margueritte	Qui aime les frites
Noireau	Petite Pataude
Paquerette	Toujours chouette
Roussette	Belle clochette

Pour avoir la limite d'affichage de la portion suivante, on ajoute **\$nombre** à **\$limite**

Pour avoir la limite d'affichage de la portion précédente, on retire **\$nombre** à **\$limite**

Affichage des boutons ou liens page précédente , page suivante

Maintenant, il va nous falloir afficher les liens ou les boutons pour afficher le reste des enregistrements.

Pour cela nous allons devoir réfléchir à 2 choses.

- ◆ Sommes nous sur la première page de notre affichage ? C'est à dire est ce que **\$limite** vaut 0 .
 - ◇ oui : il ne faut donc pas afficher de bouton ou de lien 'page précédente'.
 - ◇ non : il faut afficher un bouton ou un lien 'page précédente'.

Affichage du lien précédent si besoin

```
if($limite != 0) {
    echo '<a href="'. $page.'?limite='.$limiteprecedente.'">Page précédente</a>';
}
```

Ou bien affichage d'un bouton formulaire

```
if($limite != 0) {
    echo '<form action="'. $page.'" method="post">';
    echo '<input type="submit" value="précédents">';
    echo '<input type="hidden" value="'. $limiteprecedente.'" name="limite">';
    echo '</form>';
}
```

- ◆ Y a-t-il encore une page à afficher après la page courante? C'est à dire : est-ce que notre limite recalculée (**\$limitesuivante**) est plus petite que notre **\$total** ?
 - ◇ oui : il faut donc afficher un bouton ou un lien 'page suivante'.
 - ◇ non : il ne faut pas afficher de bouton ou de lien 'page suivante' car on est arrivé à la fin de notre affichage.

Affichage du lien suivant si besoin

```
if($limitesuivante < $total) {
    echo '<a href="'. $page.'?limite='.$limitesuivante.'">Page Suivante</a>';
}
```

Ou bien affichage d'un bouton formulaire

```
//Ou bien affichage d'un bouton formulaire
if($limitesuivante < $total) {
    echo '<form action="'. $page.'" method="post">';
    echo '<input type="submit" value="suivants">';
    echo '<input type="hidden" value="'. $limitesuivante.'" name="limite">';
    echo '</form>';
}
```

Et voilà le tour est joué !

Maintenant, Si vous voulez afficher + de 5 enregistrements par page , il vous suffit de changer la valeur de **\$nombre**.

```
<mode type="hors sujet" voix="voix d'hôtesse">
Ding ding dong ding !
Nos hôtesse vont passer vous proposer des pizzas et du coca ou des jus de fruits.
C'est le moment de faire une petite pause avant d'aller plus loin.
</mode>
```

Comment utiliser ce que nous venons d'apprendre de façon intelligente.

Il serait particulièrement malin de créer des fonctions pour notre affichage page/page pour pouvoir les réutiliser dans tous nos scripts.

```
<mode type="hors sujet" voix="voix d'hôtesse">
Ding ding dong ding !
Le capitaine vous prie de bien vouloir attacher vos ceintures, nous passons à la
vitesse supérieure !.
</mode>
```

Valider la \$limite passée par l'url.

Comme pour tous les scripts, il est important de penser à la sécurité, particulièrement quand des variables sont passées par URL ou par un formulaire.

Nous devons récupérer la variable puis tester si la valeur de \$limite :

- est bien numérique.
- se trouve dans une fourchette de valeurs possibles (de 0 à \$max)
- est un multiple de \$nombre.

Pourquoi faire tout ces tests ? Parce qu'il ne faut jamais faire confiance à l'utilisateur, ni aux données qu'il peut entrer dans un formulaire ou via l'URL !

Fonction qui vérifie la validité de \$limite

```
//récupération de $limite

if(isset($_GET['limite']))

    $limite=$_GET['limite'];
else    $limite=0;

function verifLimite($limite,$total,$nombre) {

    // je verifie si limite est un nombre.

    if(is_numeric($limite)) {

// si $limite est entre 0 et $total, $limite est ok

        // sinon $limite n'est pas valide.

        if(($limite >=0) && ($limite <= $total) &(($limite%$nombre)==0)) {

            // j'assigne 1 à $valide si $limite est entre 0 et $max

            $valide = 1;

        }

        else {

            // sinon j'assigne 0 à $valide

            $valide = 0;

        }

    }
```

```

    }

    else {

        // si $limite n'est pas numérique j'assigne 0 à $valide

        $valide = 0;

    }

// je renvoie $valide

return $valide;

}

```

Explication des arguments passés en paramètres à la fonction `verifLimite`

- ♦ **\$limite** : la limite passée en paramètre.
- ♦ **\$total** : le nombre total d'enregistrements à afficher.
- ♦ **\$nombre** : nombre total d'enregistrements à afficher par page.

Fonction qui affiche les boutons précédent/suivant

```

<?php
function displayNextPreviousButtons($limite,$total,$nb,$page) {
$limiteSuivante = $limite + $nb;
$limitePrecedente = $limite - $nb;
echo '<table><tr>'. "\n";
if($limite != 0) {
    echo '<td valign="top">'. "\n";
    echo '<form action="'. $page. '" method="post">'. "\n";
    echo '<input type="submit" value="précédents">'. "\n";
    echo '<input type="hidden" value="'. $limitePrecedente. '" name="limite">'. "\n";
    echo '</form>'. "\n";
    echo '</td>'. "\n";
}
if($limiteSuivante < $total) {
    echo '<td valign="top">'. "\n";
    echo '<form action="'. $page. '" method="post">'. "\n";
    echo '<input type="submit" value="suivants">'. "\n";
    echo '<input type="hidden" value="'. $limiteSuivante. '" name="limite">'. "\n";
    echo '</form>'. "\n";
    echo '</td>'. "\n";
}
echo '</tr></table>'. "\n";
}

```

Explication des arguments passés en paramètres à la fonction `displayNextPreviousButtons`.

- ♦ **\$limite** : la limite d'affichage courante.
- ♦ **\$total** : nombre total d'enregistrements retournés par le query
- ♦ **\$nb** : nombre d'enregistrements à afficher par page.
- ♦ **\$page** : nom de la page php.

Explication de la fonction `displayNextPreviousButtons`.

Elle se comporte exactement comme les bouts de script expliqués ci-dessus.

On calcule la limite précédente et suivante, et, selon la page sur laquelle on se trouve ainsi que le nombre d'enregistrements total à afficher, on affiche ou pas les boutons next/previous.

Comment afficher les numéros de page.

Vous vous demandez sûrement comment faire pour afficher des liens vers les différents pages à afficher.

Cette petite fonction va vous aider.

Fonction qui affiche les liens vers les pages

```
function affichePages($nb,$page,$total) {
    $nbpages=ceil($total/$nb);
    $numeroPages = 1;
    $compteurPages = 1;
    $limite = 0;
    echo '<table border = "0" ><tr>'. "\n";
    while($numeroPages <= $nbpages) {
        echo '<td ><a href = "'. $page. '?limite='. $limite. '">'. $numeroPages. '</a></td>'. "\n";
        $limite = $limite + $nb;
        $numeroPages = $numeroPages + 1;
        $compteurPages = $compteurPages + 1;
        if($compteurPages == 10) {
            $compteurPages = 1;
            echo '<br>'. "\n";
        }
    }
    echo '</tr></table>'. "\n";
}
```

Explication des arguments passés en paramètres à la fonction affichePages

- ♦ **\$nb** : \$nb : nombre d'enregistrements à afficher par page.
- ♦ **\$page** : nom de la page php.
- ♦ **\$total** : nombre total d'enregistrements retournés par le query

Explication de la fonction affichePages.

1. On calcule le nombre de pages en divisant le nombre total d'enregistrements (retournés par le query) par le nombre d'enregistrements à afficher par page.

Si nous avons 20 enregistrements à afficher il nous faudra $20/5 = 4$ pages. Si nous avons 21 enregistrements à afficher, comme $21/5 = 4,2$, cinq pages seront nécessaires. On doit donc arrondir à l'entier supérieur grâce à la fonction ceil();

2. On initialise **\$numeroPages** à 1 (première page)
3. On fait une boucle qui va faire varier **\$numeroPages** tant qu'il est inférieur ou égal à **\$nbpages**.
4. A chaque passage dans la boucle, on incrémente **\$numeroPages**.
5. Pour ne pas se retrouver avec trop de numéros de pages sur la même ligne, on met en place un compteur **\$compteurPages** qui va varier dans la boucle lui aussi.
6. Une fois que sa valeur arrivera à 10, on ajoute un retour à la ligne et on remet notre compteur à 1.

Modification de notre script de base afin qu'il utilise nos 2 nouvelles fonctions.

Si le nombre total d'enregistrements à afficher est plus grand que le nombre d'enregistrements à afficher par page, on appelle la fonction qui affiche les numéros de page ainsi que celle qui affiche les boutons

Modification du script de base pour utiliser nos 2 fonctions

```
<html>

<body>

<?php

//=====

// includes du fichier fonctions

//=====

require 'fonctions.php';

//=====

// information pour la connection à le DB

//=====

$host = 'localhost';

$user = 'root';

$pass = '';

$db = 'test';

//=====

// initialisation des variables

//=====

// on va afficher 5 résultats par page.

$nombre = 5;

// si limite n'existe pas on l'initialise à zéro

if (!$limite) $limite = 0;

// on cherche le nom de la page.

$path_parts = pathinfo($_SERVER['PHP_SELF']);

$page = $path_parts["basename"];

//=====

// connection à la DB

//=====
```

```

$link = mysql_connect ($host,$user,$pass) or die ('Erreur : '.mysql_error() );

mysql_select_db($db) or die ('Erreur : '.mysql_error());


//=====

// requête SQL qui compte le nombre total

// d'enregistrements dans la table.

//=====

$select = 'SELECT count(id) FROM vaches';

$result = mysql_query($select,$link) or die ('Erreur : '.mysql_error() );

$row = mysql_fetch_row($result);

$total = $row[0];


//=====

// vérifier la validité de notre variable

// $limite;

//=====

$verifLimite= verifLimite($limite,$total,$nombre);

// si la limite passée n'est pas valide on la remet à zéro

if(!$verifLimite) {

    $limite = 0;

}

//=====

// requête SQL qui ne prend que le nombre

// d'enregistrement necessaire à l'affichage.

//=====

$select = 'select prenom,surnom FROM vaches ORDER BY prenom ASC limit '.$limite.','.$nombre;

$result = mysql_query($select,$link) or die ('Erreur : '.mysql_error() );


//=====

// si on a récupéré un resultat on l'affiche.

//=====

if($total) {

    // début du tableau

```

```

echo '<table bgcolor="#FFFFFF">'. "\n";

// première ligne on affiche les titres prénom et surnom dans 2 colonnes

echo '<tr>';

echo '<td bgcolor="#669999"><b><u>Prénom</u></b></td>';

echo '<td bgcolor="#669999"><b><u>Surnom</u></b></td>';

echo '</tr>'. "\n";

// lecture et affichage des résultats sur 2 colonnes

while($row = mysql_fetch_array($result)) {

    echo '<tr>';

    echo '<td bgcolor="#CCCCCC">'. $row['prenom']. '</td>';

    echo '<td bgcolor="#CCCCCC">'. $row['surnom']. '</td>';

    echo '</tr>'. "\n";

}

echo '</table>'. "\n";

}

else echo 'Pas d\'enregistrements dans cette table...';

mysql_free_result($result);

//=====

// si le nombre d'enregistrement à afficher

// est plus grand que $nombre

//=====

if($total > $nombre) {

    // affichage des liens vers les pages

    affichePages($nombre,$page,$total);

    // affichage des boutons

    displayNextPreviousButtons($limite,$total,$nombre,$page);

}

?>

</body>

</html>

```

Que pourriez-vous ajouter?

- ◆ Signaler sur quelle page l'utilisateur se trouve.
"vous êtes sur la page n°1", "vous êtes sur la page n°2",...
- ◆ Changer la couleur du numéro de page active.
- ◆ Afficher où on en est dans l'affichage des enregistrements :
Affichage des 5 premiers enregistrements sur / 32.
Affichage des 10 premiers enregistrements sur / 32.
etc...
<mode type="hors sujet" voix="voix d'hôtesse">
Ding ding dong ding !
J'espère que vous avez bon voyage en notre compagnie. L'équipe de PHPdébutant
espère vous revoir très bientôt sur ses lignes :p.
Le capitaine du tuto : [flyingcow](#).
</mode>

www.phpdebutant.org © 2006 – L'équipe de phpDebutant