**Extraindo uma listagem aleatória do MySQL**   
  
Bem, primeiro vamos ver como se extrai uma listagem aleatória no MySQL. De acordo com o manual do MySQL, existe uma função RAND(), que gera um número aleatório. A partir da versão 3.23, você pode fazer:

SELECT \* FROM nome\_tabela ORDER BY RAND()

Pronto; até aí já podemos ter uma listagem aleatória. Se eu quiser apenas três registros, por exemplo, escolhidos aleatoriamente, da tabela, basta fazer:

SELECT \* FROM nome\_tabela ORDER BY RAND() LIMIT 3

**2. Um script de paginação**   
  
Agora, vou mostrar aqui um algoritmo de paginação que, se não me engano, peguei no phpbrasil.com quando comecei a programar em PHP. Aí eu mexi, aperfeiçoei, converti pra outras liguagens, converti pra DB\_dataobject do PEAR e por aí vai; e utilizo até hoje. A tabela que vou utilizar utiliza a estrutura abaixo:

--

-- Estrutura da tabela `produtos`   
--

CREATE TABLE `produtos` (   
  `ID` int(11) NOT NULL auto\_increment,   
  `PRODUTO` varchar(255) default '',   
  `DESCRICAO` varchar(255) default '',   
  `EXIBIR` char(1) NOT NULL default '1',   
  PRIMARY KEY  (`ID`)   
) ;

--    
-- Extraindo dados da tabela `produtos`   
--

INSERT INTO `produtos` (`ID`, `PRODUTO`, `DESCRICAO`, `EXIBIR`) VALUES (1, 'Bola', 'Objeto esférico que visa entreterimento', '1');   
INSERT INTO `produtos` (`ID`, `PRODUTO`, `DESCRICAO`, `EXIBIR`) VALUES (2, 'Chuveiro', 'Objeto que serve para limpeza', '1');   
INSERT INTO `produtos` (`ID`, `PRODUTO`, `DESCRICAO`, `EXIBIR`) VALUES (3, 'CD', 'Objeto que serve para ouvir música', '1');   
INSERT INTO `produtos` (`ID`, `PRODUTO`, `DESCRICAO`, `EXIBIR`) VALUES (4, 'prod3', 'Produto 3', '1');   
INSERT INTO `produtos` (`ID`, `PRODUTO`, `DESCRICAO`, `EXIBIR`) VALUES (5, 'pp2pp2p', 'ipip', '1');   
INSERT INTO `produtos` (`ID`, `PRODUTO`, `DESCRICAO`, `EXIBIR`) VALUES (6, 'ggg', '', '1');   
INSERT INTO `produtos` (`ID`, `PRODUTO`, `DESCRICAO`, `EXIBIR`) VALUES (7, 'dasdA', 'ggsdf', '1');   
INSERT INTO `produtos` (`ID`, `PRODUTO`, `DESCRICAO`, `EXIBIR`) VALUES (8, 'DQWDQ', 'dqqdw', '1');   
INSERT INTO `produtos` (`ID`, `PRODUTO`, `DESCRICAO`, `EXIBIR`) VALUES (9, 'qDWQ', 'agfagraw', '1');   
INSERT INTO `produtos` (`ID`, `PRODUTO`, `DESCRICAO`, `EXIBIR`) VALUES (10, 'efrfefwe', '', '1');   
INSERT INTO `produtos` (`ID`, `PRODUTO`, `DESCRICAO`, `EXIBIR`) VALUES (11, 'Mais produto', '+++', '1');   
INSERT INTO `produtos` (`ID`, `PRODUTO`, `DESCRICAO`, `EXIBIR`) VALUES (12, 'M&Ms', '', '1');

E abaixo, o código de paginação em PHP:

<?php  
//Conexão com o banco:   
mysql\_connect("localhost","desenvolvimento","12345");   
mysql\_select\_db("test");

// Informações da query. No caso, "SELECT \* FROM produtos WHERE EXIBIR=1"   
$campos\_query = "\*";   
$final\_query  = "FROM produtos WHERE EXIBIR=1";

// Declaração da pagina inicial   
$pagina = $\_GET["pagina"];   
if($pagina == "") {   
    $pagina = "1";   
}

// Maximo de registros por pagina   
$maximo = 5;

// Calculando o registro inicial   
$inicio = $pagina - 1;   
$inicio = $maximo \* $inicio;

// Conta os resultados no total da minha query   
$strCount = "SELECT COUNT(\*) AS 'num\_registros' $final\_query";   
$query    = mysql\_query($strCount);   
$row      = mysql\_fetch\_array($query);   
$total    = $row["num\_registros"];

if($total<=0) {   
    echo "<center>Nenhum registro encontrado.</center>";   
} else {   
    $strQuery = "SELECT $campos\_query $final\_query LIMIT $inicio,$maximo";   
    $query    = mysql\_query($strQuery);

while($row = mysql\_fetch\_array($query)) {   
        echo "Produto: ".$row["PRODUTO"]."<BR>";   
    }

// Calculando pagina anterior   
    $menos = $pagina - 1;

// Calculando pagina posterior   
    $mais = $pagina + 1;

$pgs = ceil($total / $maximo);   
    if($pgs > 1 ) {   
        // Mostragem de pagina   
        if($menos > 0) {   
           echo "<a href=\"?pagina=$menos&\" class='texto\_paginacao'>anterior</a> ";   
        }   
        // Listando as paginas   
        for($i=1;$i <= $pgs;$i++) {   
            if($i != $pagina) {   
               echo "  <a href=\"?pagina=".($i)."\" class='texto\_paginacao'>$i</a>";   
            } else {   
                echo "  <strong lass='texto\_paginacao\_pgatual'>".$i."</strong>";   
            }   
        }   
        if($mais <= $pgs) {   
           echo "   <a href=\"?pagina=$mais\" class='texto\_paginacao'>próxima</a>";   
        }   
    }   
}

?>

Ok, até aí já temos os resultados sendo exibidos em páginas:   
  
Produto: Bola   
Produto: Chuveiro   
Produto: CD   
Produto: prod3   
Produto: pp2pp2p   
1 2 3 próxima   
  
(Peraí, mas os resultados não estão aleatórios!...) Calma, vamos chegar lá...   
  
**3. A paginação aleatória**   
  
Agora, vamos fazer uma pequena alteração no script para exibirmos os resultados de forma aleatória. Para isso, basta fazer um alteração na query:

<?php

//Conexão com o banco:   
mysql\_connect("localhost","desenvolvimento","12345");   
mysql\_select\_db("test");

// Informações da query. No caso, "SELECT \* FROM produtos WHERE EXIBIR=1"   
$campos\_query = "\*";   
$final\_query  = "FROM produtos WHERE EXIBIR=1";

// Declaração da pagina inicial   
$pagina = $\_GET["pagina"];   
if($pagina == "") {   
    $pagina = "1";   
}

// Maximo de registros por pagina   
$maximo=5;

// Calculando o registro inicial   
$inicio = $pagina - 1;   
$inicio = $maximo \* $inicio;

// Conta os resultados no total da minha query   
$strCount   = "SELECT COUNT(\*) AS 'num\_registros' $final\_query ";   
$query      = mysql\_query($strCount);   
$row        = mysql\_fetch\_array($query);   
$total      = $row["num\_registros"];

if($total <= 0) {   
    echo "<center>Nenhum registro encontrado.</center>";   
} else {   
    $strQuery = "SELECT $campos\_query $final\_query ORDER BY RAND() LIMIT $inicio,$maximo"; // ALTEREI AQUI   
    $query    = mysql\_query($strQuery);

while($row = mysql\_fetch\_array($query)) {   
        echo "Produto: ".$row["PRODUTO"]."<BR>";   
    }

 // Calculando pagina anterior   
    $menos = $pagina - 1;

 // Calculando pagina posterior   
    $mais = $pagina + 1;

 $pgs = ceil($total / $maximo);   
    if($pgs > 1 ) {   
        // Mostragem de pagina   
        if($menos>0) {   
           echo "<a href=\"?pagina=$menos\" class='texto\_paginacao'>anterior</a> ";   
        }   
        // Listando as paginas   
        for($i=1;$i <= $pgs;$i++) {   
            if($i != $pagina) {   
                echo "  <a href=\"?pagina=".($i)."\" class='texto\_paginacao'>$i</a>";   
            } else {   
                echo "  <strong class='texto\_paginacao\_pgatual'>".$i."</strong>";   
            }   
        }   
        if($mais <= $pgs) {   
           echo "   <a href=\"?pagina=$mais\" class='texto\_paginacao'>próxima</a>";   
        }   
    }   
}

?>

Note que agora os resultados são exibidos totalmente desordenados. Cada vez que você entra na primeira página, ela aparece em uma seqüência diferente, e quando vai para a segunda, da mesma maneira, aparecendo inclusive registros que você já viu:   
  
Ex. de página 1:   
  
Produto: pp2pp2p   
Produto: prod3   
Produto: ggg   
Produto: CD   
Produto: M&Ms   
1 2 3 próxima   
  
Ex. de página 2 :   
  
Produto: dasdA   
Produto: pp2pp2p ->repetido   
Produto: DQWDQ   
Produto: ggg -> repetido   
Produto: M&Ms -> repetido   
anterior 1 2 3 próxima   
  
Para resolver isso, vamos utilizar um princípio de programação de números aleatórios: o uso da "semente" (seed, em Inglês). Resumindo, uma semente é um valor que serve para gerar uma seqüência de números aleatórios que será sempre a mesma, uma vez que a semente for a mesma. Por exemplo, suponhamos que, ao utilizar o número 7 como semente eu obtivesse, em uma seqüência de números aleatórios, o seguinte resultado:[ 8, 5, 6, 3]. Significa que sempre que eu chamar a mesma função de geração de números com a semente 7, vou obter o mesmo resultado: [8, 5, 6, 3]. Se eu usar semente 6, por exemplo, ou qualquer outra, talvez obtenha [6, 1, 2, 4] ou qualquer outra seqüência. Entendido?   
Bom, a função RAND() do MySQL aceita um parâmetro como semente (ah... estou começando a entender aonde você quer chegar...). Para isso, basta utilizar RAND(N), onde "N" é um número inteiro. Na query que utilizamos, significa dizer que, sempre que eu chamar "SELECT \* FROM produtos WHERE EXIBIR=1 ORDER BY RAND(7)", vou obter:   
  
Na página 1:   
  
Produto: CD   
Produto: DQWDQ   
Produto: M&Ms   
Produto: Chuveiro   
Produto: qDWQ   
1 2 3 próxima   
  
Na página 2:   
  
Produto: ggg   
Produto: dasdA   
Produto: efrfefwe   
Produto: Mais produto   
Produto: prod3   
  
Na página 3:   
  
Produto: pp2pp2p   
Produto: Bola   
anterior 1 2 3   
  
**Obs.:** A seqüência escolhida pelo computador depende da função implementada pela versão da linguagem, do sistema operacional e das condições que a função analisa pra gerar os números.   
  
E não vai haver mais aquela confusão. (Sim, mas aí perde a graça, porque você vai ter sempre a mesma ordem!) Sim, é verdade, então podemos dar uma melhorada: vamos fazer de forma que, a primeira vez que a pessoa entrar na página, o programa gere aleatoriamente, via PHP, uma semente e passe a utilizar ela para as outras páginas. Farei isso passando a semente via url. O script vai ficar assim:

<?php

//Conexão com o banco:   
mysql\_connect("localhost","desenvolvimento","12345");   
mysql\_select\_db("test");

// Informações da query. No caso, "SELECT \* FROM produtos WHERE EXIBIR=1 ORDER BY RAND()"   
$campos\_query = "\*";   
$final\_query  = "FROM produtos WHERE EXIBIR=1";

// Declaração da pagina inicial   
$pagina = $\_GET["pagina"];   
if($pagina == "") {   
    $pagina = "1";   
}

// Maximo de registros por pagina   
$maximo = 5;

// Calculando o registro inicial   
$inicio = $pagina - 1;   
$inicio = $maximo \* $inicio;

// Conta os resultados no total da minha query   
$strCount = "SELECT COUNT(\*) AS 'num\_registros' $final\_query";   
$query    = mysql\_query($strCount);   
$row      = mysql\_fetch\_array($query);   
$total    = $row["num\_registros"];

if($total <= 0) {   
    echo "<center>Nenhum registro encontrado.</center>";   
} else {   
    if(!isset($\_GET["seed"])) {   
        $seed = rand();   // Caso ainda não exista uma semente, cria a semente via PHP.   
    } else {   
        $seed = addslashes($\_GET["seed"]);  // Caso já exista uma semente, utiliza a que foi passada na url. (o addslashes é por questão de segurança)   
    }   
    $strQuery   = "SELECT $campos\_query $final\_query ORDER BY RAND($seed) LIMIT $inicio,$maximo";     
    $query      = mysql\_query($strQuery);

while($row = mysql\_fetch\_array($query)) {   
        echo "Produto: ".$row["PRODUTO"]."<BR>";   
    }

// Calculando pagina anterior   
    $menos = $pagina - 1;

// Calculando pagina posterior   
    $mais = $pagina + 1;

$pgs = ceil($total / $maximo);   
    if($pgs > 1 ) {   
        // Mostragem de pagina   
        if($menos>0) {   
           echo "<a href=\"?pagina=$menos&seed=$seed\" class='texto\_paginacao'>anterior</a> ";   
        }   
        // Listando as paginas   
        for($i=1;$i <= $pgs;$i++) {   
            if($i != $pagina) {   
                echo "  <a href=\"?pagina=".($i)."&seed=$seed\" class='texto\_paginacao'>$i</a>";   
            } else {   
                echo "  <strong class='texto\_paginacao\_pgatual'>".$i."</strong>";   
            }   
        }   
        if($mais <= $pgs) {   
           echo "   <a href=\"?pagina=$mais&seed=$seed\" class='texto\_paginacao'>próxima</a>";   
        }   
    }   
}

?>

Pode testar; o resultado é bem interessante. Além disso, se você quiser fazer com PostgreSQL também dá. A diferença é que a função de número aleatório no PostgreSQL é random(); e para inserir uma semente é necessário enviar antes o comando "SET SEED=N", aonde "N" é um valor float entre 0 e 1. Em algumas versões, pode ser utilizado ao invés disso, "SELECT setseed(N)". O LIMIT no PosgreSQL também é diferente. Utiliza-se LIMIT A OFFSET B, aonde A e B são números inteiros ([http://www.postgresql.org/docs/8.0/static/queries-limit.html](http://www.phpbrasil.com/click.php?send=http://www.postgresql.org/docs/8.0/static/queries-limit.html)). Aí vai ser necessário fazer umas alteraçõezinhas no código, mas agora que você já aprendeu, fica fácil...   
Estou aberto a comentários, sugestões e dúvidas.