

# Paginazione dei dati

di [Gianni Tomasicchio](#) - 6 gennaio 2008

[database](#) [MySQL](#) [Oracle](#) [SQL Server](#) [PostgreSQL](#) [SQL](#) [SQLite](#)

## Introduzione

In una applicazione web capita spesso di dover mostrare all'utente un elenco di dati provenienti dal database. A volte questi dati sono numerosi pertanto, per ragioni estetiche e di usabilità, è necessario suddividerli in diverse pagine web. L'accesso alle singole pagine avviene tramite dei link posti comunemente al disopra e al disotto dell'elenco parziale di dati. Questa tecnica di rappresentazione dei dati viene comunemente detta **paginazione** e gli esempi sul web sono numerosi, si pensi ad esempio ai risultati di un motore di ricerca o all'elenco dei messaggi di un forum o di un blog.

In alcuni casi la navigazione tra le pagine è garantita solo da due link che puntano alla pagina precedente e successiva, altre volte invece i link sono più numerosi e permettono di raggiungere direttamente una pagina in particolare, identificata dal suo numero.

Anche se la sorgente dei dati da paginare non influisce sull'estetica del sistema di navigazione (la *presentation*), l'impatto sul codice necessario alla realizzazione dello script è invece considerevole. Pertanto in questo articolo focalizzeremo la nostra attenzione sui dati provenienti da database, situazione di gran lunga più comune.

Come avremo modo di verificare nel corso di questo articolo, la realizzazione di un sistema di paginazione dei dati provenienti da database non è sempre un compito banale. Diversi fattori sono da tenere in considerazione (es.: usabilità, prestazioni) che richiedono soluzioni spesso in contrasto tra loro e tra le quali è necessario mediare.

## Paginazione con MySQL

Se in una pagina web dobbiamo mostrare solo una parte dei dati complessivi è conveniente che solo questa porzione venga prelevata dal database. Per fare ciò dobbiamo modificare la query che utilizziamo per recuperare i dati in modo che essa estragga solo i record necessari a costruire la pagina web richiesta dall'utente. Nello standard SQL però non è prevista una sintassi per limitare i dati provenienti da una SELECT.

I diversi database comunque hanno introdotto nel loro dialetto SQL particolari comandi proprietari per ottenere questo risultato. In particolare MySQL permette l'impiego delle clausole **LIMIT** e **OFFSET**, secondo una sintassi del tipo:

**SELECT ... LIMIT *n***

per prelevare solo le prime *n* righe, e

**SELECT ... LIMIT *m, n***

ovvero

**SELECT ... LIMIT *n* OFFSET *m***

per prelevare solo *n* righe, partendo dalla *m*-esima riga. Facciamo qualche esempio, con:

**SELECT \* FROM *nome\_tabella* WHERE ... LIMIT 10**

otteniamo i primi 10 record estratti dalla SELECT, oppure con

**SELECT \* FROM *nome\_tabella* WHERE ... LIMIT 20, 10**

ovvero

**SELECT \* FROM *nome\_tabella* WHERE ... LIMIT 10 OFFSET 20**

otteniamo 10 record a partire dal 21° (l'offset inizia a contare da zero), cioè i record dal 21° al 30°. E' proprio questa sintassi a permetterci di estrarre i record da mostrare in una particolare pagina web. Ad esempio, se ogni pagina deve contenere 10 record, le precedenti query permettono di estrarre i record necessari a costruire la terza pagina web dei risultati.

Ma non basta la clausola LIMIT a rendere efficace la SELECT. Infatti negli esempi precedenti abbiamo considerato una numerazione per i record (i primi 10 record, i record dal 21° al 30°) senza però specificare secondo quale ordinamento va considerata questa numerazione. Infatti nelle query appena viste l'estrazione di un numero parziale di record avviene in maniera casuale poiché nell'SQL non è indicato il criterio di ordinamento con cui vogliamo che ci vengano restituiti i risultati. Pertanto solo con un ordinamento esplicitamente definito ha senso utilizzare la clausola LIMIT.

Inoltre il criterio di ordinamento deve essere univoco, cioè non deve accadere che due o più record possano essere scambiati di posizione poiché equivalenti nei confronti dell'ordinamento indicato. Vediamo il motivo con un esempio, supponiamo di voler paginare l'elenco telefonico di Napoli. Se usassimo un ordinamento per cognome le pagine contenenti i dati dei signori "Esposito", più di

7.200 record, conterrebbero un elenco casuale di nomi. Ad ogni visualizzazione di una di queste pagine, l'elenco si mostrerebbe sempre diverso, poiché secondo un ordinamento per cognome, i signori Esposito sono tutti uguali e non è possibile decidere in maniera assoluta chi viene prima e chi dopo. Una soluzione potrebbe essere ordinarli per cognome e numero di telefono. In pratica quindi specificare un criterio di ordinamento non univoco equivale in certi casi a non specificare alcun ordinamento.

Per la creazione dei link di navigazione infine serve sapere il numero totale di pagine web in cui stiamo suddividendo l'elenco di record. Questo numero è facilmente ricavabile dal numero complessivo di record da mostrare, ottenibile con una query del tipo:

```
SELECT COUNT(*) FROM nome_tabella WHERE ...
```

in cui le condizioni utilizzate nella clausola WHERE devono essere le stesse utilizzate nelle SELECT di estrazione paginata dei dati, ovvero quelle con la clausola LIMIT.

## Applicazione di esempio

L'applicazione di esempio realizzata, scaricabile a [questo link](#), si occupa di mostrare l'elenco dei comuni italiani, ordinati per nome e codice ISTAT (codice numerico univoco). I dati vengono prelevati da un'unica tabella, che è possibile creare e popolare utilizzando lo script SQL contenuto nell'archivio ZIP dell'applicazione. Nel nostro esempio mostreremo tutti i dati presenti nella tabella dei comuni, pertanto non ci saranno clausole WHERE nelle SELECT.

Il programma è costituito da 3 file: il template HTML, lo script principale che estrae i dati dal database e uno script contenente tre funzioni utili alla creazione dei link di navigazione. Questi saranno nella forma:

« 1...n-2 n-1 [n] n+1 n+2 ... N »

ovvero conterranno sempre un link alla prima e all'ultima pagina (1 e N). Inoltre conterranno l'indicazione della pagina corrente racchiusa tra parentesi quadre [n] e dei link alle pagine che precedono e seguono la pagina corrente. Il numero di questi link a pagine limitrofe sarà configurabile. Infine ci saranno anche dei link alla pagina precedente e successiva, indicati dai simboli « e ».

Iniziamo ad analizzare il codice dell'applicazione, partendo dal template HTML, riportato di seguito:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Elenco Comuni Italiani</title>
</head>
<body>
<h1>Elenco Comuni Italiani</h1>
<p><?php echo $link_paginazione?></p>
<table border="1">
  <tr>
    <th>COMUNE</th>
    <th>PROVINCIA</th>
    <th>REGIONE</th>
    <th>CAP</th>
    <th>PREFISSO TEL.</th>
    <th>CODICE COMUNE</th>
    <th>CODICE ISTAT</th>
  </tr>
  <?php foreach($elenco_comuni as $riga):?>
  <tr>
    <td><?php echo htmlentities($riga['comune'])?></td>
    <td><?php echo $riga['provincia']?></td>
    <td><?php echo $riga['regione']?></td>
    <td><?php echo $riga['cap']?></td>
    <td><?php echo $riga['prefisso']?></td>
    <td><?php echo $riga['cod_comune']?></td>
    <td><?php echo $riga['cod_istat']?></td>
  </tr>
  <?php endforeach ?>
</table>
<p><?php echo $link_paginazione?></p>
</body>
```

</html>

Come si può notare, il template HTML impiegato è davvero molto semplice. Il riempimento della tabella dei dati è affidato ad un ciclo foreach mentre l'HTML che costituisce i link di paginazione, riportati al di sopra ed al disotto della tabella, viene incluso nel template leggendo la variabile `$link_paginazione`, creata nello script PHP principale, riportato nella pagina seguente.

## Lettura paginata dei dati

Di seguito viene riportato il listato di `index.php`, il principale script PHP dell'applicazione di esempio, responsabile tra l'altro dell'estrazione dei record dal database. Le prime righe servono a specificare tutti i settaggi del programma necessari alla connessione con il database, alla creazione delle pagine web e dei link di navigazione (modificateli per adattarli alla vostra configurazione). Si noti in particolare la variabile `$url_base` con la quale viene impostato l'URL necessario a richiamare lo script corrente. Nel nostro esempio viene settato a `"index.php"` ma potremmo trovarci in situazioni più generiche, con un URL contenente parametri, ad esempio `"index.php?nazione=italia"`. `$url_base` quindi è l'URL che utilizzeremo per costruire i link di navigazione, al quale pertanto dovremo poi aggiungere un ulteriore parametro `"pag"` necessario a specificare la pagina richiesta (es.: `index.php?nazione=italia&pag=2`). La creazione di questi URL con l'indicazione della pagina richiesta sarà effettuata dalla funzione `crea_url()` descritta in seguito.

Specificati i parametri dell'applicazione si prosegue poi con la connessione a MySQL e la selezione del DB.

Il calcolo del numero totale di pagine viene fatto attraverso la `SELECT COUNT(*)` che nel nostro esempio non richiede una clausola `WHERE` poiché mostreremo tutti i dati contenuti nella tabella. Si noti che per calcolare tale numero viene arrotondato per eccesso con la funzione `ceil()` il risultato della divisione tra `$tot_righe` e `$righe_per_pagina`.

Dall'URL recuperiamo il numero della pagina richiesta dall'utente (`$_GET['pag']`) e se questo assume un valore illecito, minore di uno o maggiore del numero dell'ultima pagina, eseguiamo un redirect rispettivamente alla prima o all'ultima pagina. Per effettuare il redirect viene utilizzata la funzione `crea_url()` che, come anticipato, serve a confezionare l'URL di destinazione contenente l'indicazione sulla pagina da mostrare.

Si procede infine con l'esecuzione della query di estrazione dei record e con la fetch dei risultati. La creazione dei link di navigazione viene affidata alla funzione `paginazione()`, che verrà illustrata nella prossima pagina.

```
<?php
include 'paginazione.php';

$DB_host      = 'localhost';
$DB_user      = 'root';
$DB_password  = 'secret';
$DB_name      = 'test';

$righe_per_pagina = 1;
$url_base     = "index.php";
$pagine_vicine = 1;

// -----
//          C O N N E S S I O N E      A L      D A T A B A S E

$link = mysql_connect($DB_host, $DB_user, $DB_password);
if (!$link) {
    die ('Non riesco a connettermi: ' . mysql_error());
}

$db_selected = mysql_select_db($DB_name, $link);
if (!$db_selected) {
    die ("Errore nella selezione del database: " . mysql_error());
}
```

```

}

// -----
//      C A L C O L O   D E L   N U M E R O   D I   P A G I N E

// ricavo il numero totale di record
$query = "SELECT COUNT(*) FROM comuni";
$result = mysql_query($query);
// record complessivi
$tot_righe = mysql_result($result,0);
// totale pagine
$tot_pagine = ceil($tot_righe / $righe_per_pagina);

// -----
//
//          P A G I N A   C O R R E N T E

$pagina_corrente = isset($_GET['pag']) ? (int)$_GET['pag'] : 1;

// se la pagina corrente è minore di 1
if($pagina_corrente < 1) {
    header('location: ' . $url_base);
    exit();
}

// se la pagina corrente è maggiore dell'ultima pagina
if($pagina_corrente > $tot_pagine) {
    header('location: ' . crea_url($url_base, $tot_pagine));
    exit();
}

// -----
//
//          E S T R A Z I O N E   D E I   R E C O R D

// calcolo la prima riga da estrarre con la query
$prima_riga = ($pagina_corrente - 1) * $righe_per_pagina;

$query = "SELECT *
          FROM comuni
          ORDER BY comune, cod_istat
          LIMIT $prima_riga, $righe_per_pagina";

$result = mysql_query($query);
if (!$result) {
    die("Errore nella query $query: " . mysql_error());
}

$elenco_comuni = array();
while ($row = mysql_fetch_assoc($result)) {
    $elenco_comuni[] = $row;
}

// creazione dei link di paginazione
$link_paginazione = paginazione($tot_pagine, $url_base, $pagina_corrente,
$pagine_vicine);

// carico il template HTML
include 'index.html';
?>

```

## Link di navigazione

Lo script PHP che segue, `paginazione.php`, è costituito da tre funzioni che concorrono alla realizzazione dei link di navigazione. Si è scelto di separare il codice per la creazione dei link da quello di estrazione dei dati per semplificarne l'utilizzo anche in altri contesti.

La funzione `crea_url()`, già introdotta in precedenza, ha il compito di accordare il parametro che specifica il numero di pagina all'`$url_base`. Per effettuare tale operazione è necessario controllare se nell'`$url_base` siano già presenti dei parametri. In caso positivo il parametro "pag" va accordato con una "&" altrimenti con un "?". Ovviamente la "&" va riportata nell'URL utilizzando la sua entità HTML "&amp;".

La funzione `crea_link()` invece serve a creare l'HTML che costituisce un link ad una pagina dei risultati. Se il link da mostrare è alla pagina corrente allora viene generato solo il numero di pagina racchiuso tra parentesi quadre (es.: [4]), altrimenti viene restituito un vero e proprio link (es.: <a href="index.php?pag=4">4</a>)

La funzione `paginazione()` è il cuore del sistema di creazione dei link di navigazione. Tale funzione realizza un set di link piuttosto elaborato, secondo la seguente logica:

- se la pagina corrente è diversa da 1 allora mostriamo un link alla pagina precedente contenente il simbolo "«"
- mostriamo sempre il link alla prima pagina ( « 1 )
- se il successivo link che mostreremo non è alla pagina 2 allora:
  - se è alla pagina 3 allora inseriamo anche il link alla pagina 2, così da non avere discontinuità ( « 1 2 3 )
  - altrimenti inseriamo dei punti di sospensione (...) per segnalare la discontinuità ( « 1 ... 4 )
- mostriamo i link alla pagina corrente e alle sue limitrofe ( « 1 ... 4 5 [6] 7 8 )
- Se l'ultimo link mostrato non è alla penultima pagina:
  - se è alla terzultima allora inseriamo anche un link alla penultima in modo da evitare la discontinuità ( 24 25 [26] 27 28 29 30 )
  - altrimenti inseriamo dei punti di sospensione (...) per segnalare la discontinuità ( 24 25 [26] 27 28 ... 31 )
- mostriamo un link all'ultima pagina, se questa non coincide con la prima
- se la pagina corrente non è l'ultima allora mostriamo un link alla pagina successiva contenente il simbolo "»"

Tutti i link realizzati vengono di volta in volta accordati alla stringa `$link_paginazione` che al termine dell'esecuzione della funzione `paginazione()` viene restituita al chiamante.

```
<?php
function crea_url($url_base, $pagina) {
    if(strpos($url_base, '?') === false) {
        return $url_base . '?pag=' . $pagina;
    } else {
        return $url_base . '&pag=' . $pagina;
    }
}

function crea_link($url_base, $pagina_corrente, $numero_pagina) {
    if($pagina_corrente == $numero_pagina) {
        return "[$numero_pagina]";
    } else {
```

```

        return '<a href="' . crea_url($url_base, $numero_pagina) . '">' .
$numero_pagina . '</a>';
    }
}

// funzione che crea i link alle pagine dei risultati
function paginazione($tot_pagine, $url_base, $pagina_corrente, $pagine_vicine) {
    $link_paginazione = "Pagine: ";

    // link alla pagina precedente
    if($pagina_corrente != 1) {
        $link_paginazione .= '<a href="' . crea_url($url_base, $pagina_corrente
- 1) . '">&laquo;</a> ';
    }

    // mostriamo sempre il link alla prima pagina
    $link_paginazione .= crea_link($url_base, $pagina_corrente, 1);

    // se il prossimo link non è alla seconda pagina aggiungo dei puntini ...
    // oppure la sola pagina mancante
    if($pagina_corrente - $pagine_vicine > 2) {
        if($pagina_corrente - $pagine_vicine == 3) {
            $link_paginazione .= " " . crea_link($url_base, $pagina_corrente,
2);
        } else {
            $link_paginazione .= " ... ";
        }
    }

    // creo i link alla pagina corrente ed a quelle ad essa vicine
    for($i = $pagina_corrente - $pagine_vicine; $i <= $pagina_corrente +
$pagine_vicine; $i++) {
        // se tra quelle vicine c'è la prima pagina (già riportata)
        if($i < 2) continue;

        // se tra quelle vicine c'è l'ultima pagina (che mostrerò con le
prossime istruzioni)
        if($i > $tot_pagine - 1) continue;

        $link_paginazione .= " " . crea_link($url_base, $pagina_corrente, $i);
    }

    // se il precedente link non era alla penultima pagina aggiungo dei puntini
...
    // oppure la sola pagina mancante
    if($pagina_corrente + $pagine_vicine < $tot_pagine - 1) {
        if($pagina_corrente + $pagine_vicine == $tot_pagine - 2) {
            $link_paginazione .= " " . crea_link($url_base, $pagina_corrente,
$tot_pagine - 1) . " ";
        } else {
            $link_paginazione .= " ... ";
        }
    }

    // mostriamo il link all'ultima pagina se questa non coincide con la prima
    if($tot_pagine != 1) {
        $link_paginazione .= " " . crea_link($url_base, $pagina_corrente,
$tot_pagine);
    }

    // link alla pagina successiva
    if($pagina_corrente != $tot_pagine) {

```

```
        $link_paginazione .= ' <a href="' . crea_url($url_base, $pagina_corrente
+ 1) . '">&raquo;</a>';
    }

    return $link_paginazione;
}
?>
```