Stránkování

Pokud je v databázi více záznamů, než je únosné zobrazit na jedné stránce, je potřeba nějak vyřešit stránkování. Osobně mám raději delší stránky – pokud se data čitelně zobrazují už během načítání (tedy pokud není použit tabulkový design, obrázky mají uvedené rozměry a tak dále), tak delší stránky ničemu nevadí, ale určitá hranice se v mnoha situacích udělat musí.

O tom, jak zjistit celkový počet řádek na stránkách se stránkováním jsem už psal, teď bych rád rozebral, jaké parametry se pro stránkování dají předávat. Řešení, které se nabízí samo, je předávat počet přeskočených záznamů nebo číslo stránky:

```
<?php
// předávání počtu přeskočených záznamů
$result = mysql_query("SELECT SQL_CALC_FOUND_ROWS * FROM tabulka ORDER BY
datum DESC, id DESC LIMIT $limit OFFSET " . intval($_GET["offset"]));
$pocet = mysql_result(mysql_query(" SELECT FOUND_ROWS()"), 0);
if ($_GET["offset"]) {
   echo ' <a href="' . htmlspecialchars($_SERVER["PHP_SELF"]) . ($_GET["offset"] !=
$limit ? "?offset=" . ($_GET["offset"] - $limit) : "") . '">zpět</a>';
}
if ($pocet > $_GET["offset"] + $limit) {
   echo ' <a href="' . htmlspecialchars($_SERVER["PHP_SELF"]) . "?offset=" .
($_GET["offset"] + $limit) . '">vpřed</a>';
}

// předávání čísla stránky
$result = mysql_query("SELECT SQL_CALC_FOUND_ROWS * FROM tabulka ORDER BY
datum DESC, id DESC LIMIT $limit OFFSET " . ($limit * $_GET["strana"]));
$pocet = mysql_result(mysql_query(" SELECT FOUND_ROWS()"), 0);
if ($_GET["strana"]) {
   echo ' <a href="' . htmlspecialchars($_SERVER["PHP_SELF"]) . ($_GET["strana"] != 1 ?
"?strana=" . ($_GET["strana"] - 1) : "") . '">zpět</a>';
}
if ($pocet > $limit * ($_GET["strana"] + 1)) {
   echo ' <a href="' . htmlspecialchars($_SERVER["PHP_SELF"]) . "?strana=" .
($_GET["strana"] + 1) . '">vpřed</a>';
}
?>
```

Způsoby jsou si podobné jako vejce vejci, druhý mi je ale o něco sympatičtější, protože se předávají nižší hodnoty a uživatel nám nemůže podstrčit neočekávaný offset (pokud by s tím počítala jiná část skriptu). Všimněte si, že záznamy se třídí nejprve podle data a potom ještě podle id. Pokud totiž datum není unikátní (např. vychází dva články denně), nemusely by se některé záznamy zobrazit, zatímco jiné by se zobrazily na dvou stránkách. Databáze totiž nezaručuje, že řádky bude vracet v nějakém konkrétním pořadí (např. v pořadí vložení, jak by se někdo mohl domnívat). Také si všimněte toho, že u listování zpět se parametr předává jen tehdy, pokud je nenulový – to je důležité kvůli vyhledávačům a cachím – stejný obsah by neměl být dostupný na různých URL (na které navíc vede odkaz).

Chování vyhledávačů a cachí mě ostatně vedlo i k napsání tohoto článku. Jisté je, že tyto tradiční způsoby nejsou vhodné pro cache – když přidáme nový záznam, obsah stránek na původních URL se změní. Způsoby nejsou vhodné ani pro uživatele – nemůžou si konkrétní stránku uložit do oblíbených položek, protože už za týden na ní bude něco jiného. Chování vyhledávačů je jako vždy spekulace – vyhledávače sice mají rády, když se stránky mění – častěji na ně pak chodí, ale pokud se všechen čas vyhrazený pro náš web spotřebuje na tyto pseudozměny, nemusí zbýt čas na skutečně nově přidaný obsah.

Ze všech úhlů pohledu tedy tradiční řešení představují problém, který je potřeba řešit. Asi nejjednodušší bude, když offset nebudeme předávat od začátku, ale od konce:

```php
<?php
// předávání počtu záznamů zbývajících do konce
$pocet = mysql_result(mysql_query("SELECT COUNT(*) FROM tabulka"), 0);
$offset = ($_GET["offset"] ? $_GET["offset"] : $pocet); // offset se předává od konce, aby stránky zůstaly trvale platné
$result = mysql_query("SELECT * FROM tabulka ORDER BY datum DESC, id DESC LIMIT $limit OFFSET " . ($pocet - $offset));
if ($offset < $pocet) {
    echo ' <a href="' . htmlspecialchars($_SERVER["PHP_SELF"]) . ($offset + $limit < $pocet ? "?offset=" . ($offset + $limit) : "") . '">zpět</a>';
}
if ($offset > $limit) {
    echo ' <a href="' . htmlspecialchars($_SERVER["PHP_SELF"]) . "?offset=" . ($offset - $limit) . '">vpřed</a>';
}
?>
```

Kvůli vícerému přičítání a odčítání kód vyžaduje pro pochopení trochu více soustředění, ale nepřehledný myslím není (pokud si jsme vědomi faktu, že offset počítáme od konce). Spolu se stránkováním tento způsob pochopitelně použít nelze – pokud je v databázi např. 55 záznamů a `$limit` je 10, musely by se na stránce č. 5 (počítáno od konce) zobrazit záznamy 41-50 (opět počítáno od konce), ale uživatel tam očekává záznamy 36-45 (protože na úvodní stránce jsou záznamy 46-55).

Tento způsob má ještě jednu výhodu – pokud do databáze přibude záznam v době mezi načtením stránky a přechodem na další stránku (typické u diskusních fór, kde příspěvky přibývají rychle), nezobrazí se již zobrazený záznam znovu. (Pokud je v době načtení stránky v databázi 55 záznamů a `$limit` je 10, následně do databáze přibude 56. záznam a uživatel prvním popsaným způsobem přejde na `?offset=10`, zobrazí se záznamy 37-46 – záznam č. 46 ale uživatel už viděl. U třetího způsobu tento problém nevznikne.) Daní za popsané výhody je to, že do cachí a vyhledávačů se postupně dostanou stránky se všemi offsety (1, 2, 3, …) a ne jen násobky limitu.

V mnoha situacích je lepší se stránkování úplně vyhnout a data rozdělit po logických celcích (např. dnech nebo měsících), jindy to ale možné není.

Přijďte si o tomto tématu popovídat na školení Návrh a používání MySQL databáze.