
4

Zobrazování dat pomocí tabulek

Nyní už umíte využít spojené síly PHP a MySQL ke tvorbě dynamických stránek. Co však dělat v případě, chcete-li zobrazit opravdu velké množství řádků? Musíte mít nějaký mechanismus, který by vašim návštěvníkům umožnil snadno si tato data pročitat. Takový mechanismus by měl navíc působit elegantním, čistým a uhlazeným dojmem. Nejsnáze tohoto výsledku dosáhnete pomocí tabulek jazyka HTML.

V této kapitole se naučíte:

- tvorbě tabulek HTML pro data načtená z databáze,
- automatické tvorbě záhlaví sloupců,
- naplnění tabulky HTML výsledky základních dotazů MySQL,
- naplnění tabulky HTML výsledky složitých dotazů MySQL,
- tvorbě uživatelsky příjemného výstupu.

Vytváříme tabulku

Ještě předtím, než data zobrazíte, je třeba připravit si strukturu, záhlaví jednotlivých sloupců a formát celé tabulky HTML, abyste svá data vůbec měli kam umístit. Kostra takové tabulky vám poskytne návod pro rozmístění dat načtených z databáze.

Vyzkoušejte

Tvorba tabulky

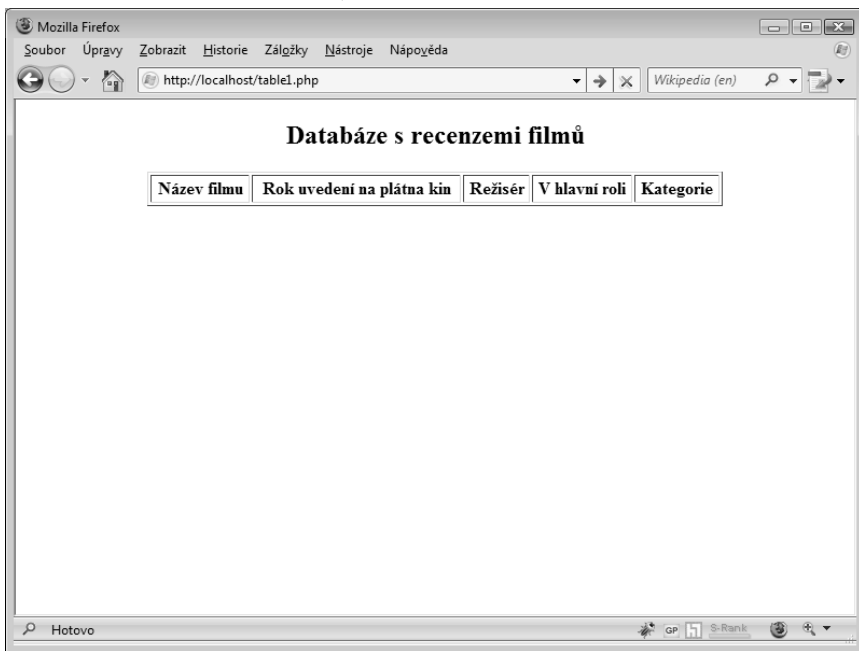
V tomto cvičení definujete záhlaví sloupců tabulky HTML, kterou poté naplníte daty.

1. Otevřete svůj oblíbený textový či HTML editor a napište následující kód:

```
<div style="text-align: center;">  
<h2>Databáze s recenzemi filmů</h2>
```

```
<table border="1" cellpadding="2" cellspacing="2"
  style="width: 70%; margin-left: auto; margin-right: auto;"
  <tr>
    <th>Název filmu</th>
    <th>Rok uvedení na plátna kin</th>
    <th>Režisér</th>
    <th>V hlavní roli</th>
    <th>Kategorie</th>
  </tr>
</table>
</div>
```

2. Soubor uložte jako `table1.php` a nahrajte jej na svůj webový server.
3. Otevřete svůj oblíbený webový prohlížeč a pohlédněte si stránku, kterou jste právě nahráli. Vaše tabulka by měla vypadat jako na obrázku 4.1.



Obrázek 4.1

4. Otevřete znovu soubor `table1.php` ve svém editoru a přidejte kód pro připojení k databázi. Pro účely našeho příkladu použijeme databázi vytvořenou ve 3. kapitole. Nezapomeňte v případě nutnosti nahradit uvedené hodnoty pro název serveru, uživatelské jméno, heslo a název databáze vlastními hodnotami.

```
<?php
// připoj se k MySQL
$db = mysql_connect('localhost', 'uzivatel', 'heslo') or
```

```

die('Nemohu se připojit. Zkontrolujte připojovací parametry.');
```

```

// zajisti, abychom používali správnou databázi
mysql_select_db('moviesite', $db) or die(mysql_error($db));
?>
```

```

<div style="text-align: center;">
  <h2>Databáze s recenzemi filmů</h2>
  <table border="1" cellpadding="2" cellspacing="2"
    style="width: 70%; margin-left: auto; margin-right: auto;">
    <tr>
      <th>Název filmu</th>
      <th>Rok uvedení na plátna kin</th>
      <th>Režisér</th>
      <th>V hlavní roli</th>
      <th>Kategorie</th>
    </tr>
  </table>
</div>
```

5. Nyní spusťte dotaz a načtěte výsledky. Přitom můžete spočítat, kolik řádků provedený dotaz vrátil.

```

<?php
// připoj se k MySQL
$db = mysql_connect('localhost', 'uzivatel', 'heslo') or
  die('Nemohu se připojit. Zkontrolujte připojovací parametry.');
```

```

// zajisti, abychom používali správnou databázi
mysql_select_db('moviesite', $db) or die(mysql_error($db));
// načti informace
$dotaz = 'SELECT movie_name, movie_year, movie_director,
  movie_leadactor, movie_type
  FROM movie
  ORDER BY movie_name ASC,
  movie_year DESC';
$výsledky = mysql_query($dotaz, $db) or die(mysql_error($db));

// zjistí počet řádků ve výsledku
$počet_filmů = mysql_num_rows($výsledky);
?>
```

```

<div style="text-align: center;">
  <h2>Databáze s recenzemi filmů</h2>
  <table border="1" cellpadding="2" cellspacing="2"
    style="width: 70%; margin-left: auto; margin-right: auto;">
    <tr>
      <th>Název filmu</th>
      <th>Rok uvedení na plátna kin</th>
      <th>Režisér</th>
```

```
        <th>V hlavní roli</th>
        <th>Kategorie</th>
    </tr>
</table>
</div>
```

6. Mezi uzavírací značky `</tr>` a `</table>` v původním kódu HTML vložte cyklus `while`, který bude procházet záznamy. Poté vypište za uzavírací značku `</table>` celkový počet načtených filmových záznamů.

```
<?php
// připoj se k MySQL
$db = mysql_connect('localhost', 'uzivatel', 'heslo') or
    die('Nemohu se připojit. Zkontrolujte připojovací parametry.');
```

```
// zajisti, abychom používali správnou databázi
mysql_select_db('moviesite', $db) or die(mysql_error($db));

// načti informace
$dotaz = 'SELECT movie_name, movie_year, movie_director,
            movie_leadactor, movie_type
        FROM movie
        ORDER BY movie_name ASC,
            movie_year DESC';
$výsledky = mysql_query($dotaz, $db) or die(mysql_error($db));

// zjistí počet řádků ve výsledku
$počet_filmů = mysql_num_rows($výsledky);
?>
```

```
<div style="text-align: center;">
    <h2>Databáze s recenzemi filmů</h2>
    <table border="1" cellpadding="2" cellspacing="2"
        style="width: 70%; margin-left: auto; margin-right: auto;">
    <tr>
        <th>Název filmu</th>
        <th>Rok uvedení na plátna kin</th>
        <th>Režisér</th>
        <th>V hlavní roli</th>
        <th>Kategorie</th>
    </tr>
```

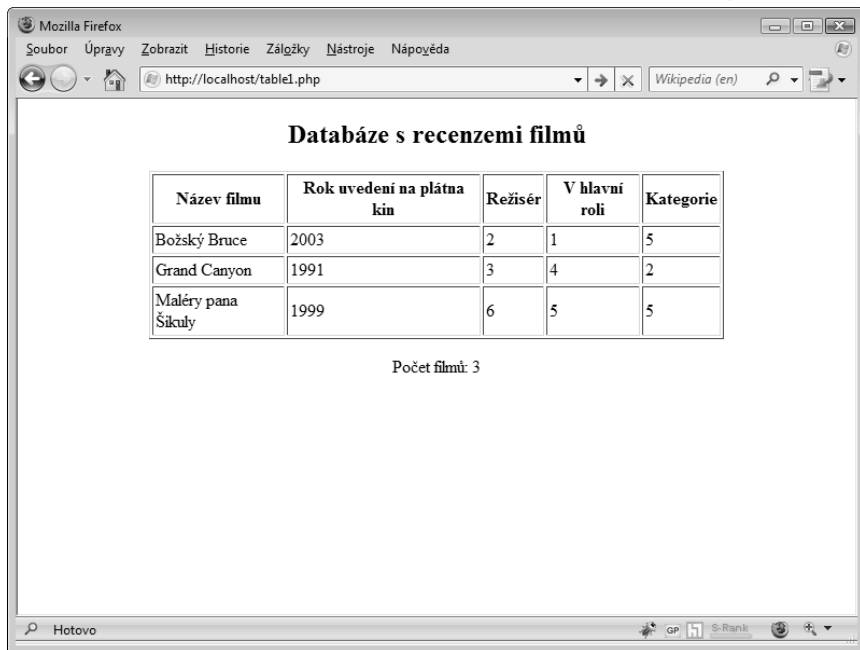
```
<?php
// projdi výsledky
while ($řádek = mysql_fetch_assoc($výsledky)) {
    extract($řádek);
    echo '<tr>';
    echo '<td>'. $movie_name. '</td>';
    echo '<td>'. $movie_year. '</td>';
```

```

echo '<td>'. $movie_director. '</td>';
echo '<td>'. $movie_leadactor. '</td>';
echo '<td>'. $movie_type. '</td>';
echo '</tr>';
}
?>
</table>
<p>Počet filmů: <?php echo $počet_filmů; ?></p>
</div>

```

7. Po otevření ve webovém prohlížeči by tato stránka měla vypadat jako na obrázku 4.2.



Obrázek 4.2

Jak to funguje

Uvedený kód pro vás vykonává spoustu práce. Rozeberme si jej tedy malinko podrobněji.

Nejdříve je ustaveno připojení k databázi a poté vybíráte databázi s filmy. Nad ní dále provádíte dotaz pro získání názvu, data vydání a obsazení hlavní role jistých filmů. Funkce `mysql_num_rows()` přijímá odkaz na výsledky a vrací celkový počet záznamů nalezených v databázi MySQL.

Načtené záznamy prochází příkaz `while`. Pro každý záznam provádí blok kódu, který se nachází mezi složenými závorkami. Nemusíte se vůbec strachovat, protože v tomto případě je jazyk PHP dost chytrý na to, aby věděl, s kolika záznamy se pracuje a na jakém z nich se právě nachází. Nehrozí tedy žádné nebezpečí, že by se nějakému záznamu přiřadily nesprávné hodnoty.

První řádek v cyklu `while` používá funkce `extract()` pro vytvoření proměnných se stejnými názvy, jako mají názvy polí. Tyto proměnné zároveň naplní hodnotami z aktuálního záznamu. Dalších sedm řádků pak jednoduše vypisuje hodnoty s malou příměsí kódu jazyka HTML.

Vylepšení výstupu

Dosud jsme pro výstup obsahu z kódu jazyka PHP (mezi značkami `<?php a ?>`) používali příkaz `echo`. Větší kusy kódu jazyka HTML stojí mimo tyto značky, a jsou proto prohlížeči odeslány okamžitě, aniž by je musel analyzovat interpret jazyka PHP. Skript tedy vstupuje do režimu PHP a vystupuje z něj a přeskakuje tak mezi kódem jazyka HTML a PHP. Někteří z vás mohou přitakat, že se jedná o optimální způsob realizace podobných řešení, zatímco jiní mohou namítnout, že je to zmatené a ztěžuje to údržbu kódu. Pojďme se tedy znovu podívat na syntaxi heredoc.

Vyzkoušejte

Jak to vše dát dohromady

Zkopírujte soubor `table1.php` do souboru `table2.php` a postupujte podle uvedených kroků.

1. Nahradte kód HTML odpovědný za záhlaví tabulky syntaxí heredoc uloženou do proměnné `$tabulka`:

```
$tabulka = <<<ENDHTML
<div style="text-align: center;">
  <h2>Databáze s recenzemi filmů</h2>
  <table border="1" cellpadding="2" cellspacing="2"
    style="width: 70%; margin-left: auto; margin-right: auto;">
    <tr>
      <th>Název filmu</th>
      <th>Rok uvedení na plátna kin</th>
      <th>Režisér</th>
      <th>V hlavní roli</th>
      <th>Kategorie</th>
    </tr>
ENDHTML;
```

2. Nahradte příkazy `echo` uvnitř cyklu `while` syntaxí heredoc a její výstup připojte k proměnné `$tabulka`.

```
$tabulka .= <<<ENDHTML
<tr>
  <td>$movie_name</td>
  <td>$movie_year</td>
  <td>$movie_director</td>
  <td>$movie_leadactor</td>
  <td>$movie_type</td>
</tr>
ENDHTML;
```

Všimněte si, že jsme místo operátoru = použili operátor .=. To je velice důležité, protože jsme tím připojili blok řetězce zadaného pomocí syntaxe heredoc ke stávajícímu obsahu proměnné \$tabulka. Pokud bychom použili jen operátor =, původní obsah by se nahradil (což bychom jistě nechtěli).

- Nahradte kód jazyka HTML pro uzavření tabulky a příkaz echo, který vypisuje počet vrácených filmů, syntaxí heredoc, kterou opět připojte k proměnné \$tabulka.

```
$tabulka .= <<<ENDHTML
</table>
<p>Počet filmů: $počet_filmů</p>
</div>
ENDHTML;
```

Kód v souboru table2.php by nyní měl vypadat takto:

```
<?php
// připoj se k MySQL
$db = mysql_connect('localhost', 'uzivatel', 'heslo') or
    die('Nemohu se připojit. Zkontrolujte připojovací parametry.');
```

```
// zajisti, abychom používali správnou databázi
mysql_select_db('moviesite', $db) or die(mysql_error($db));
```

```
// načti informace
$dotaz = 'SELECT movie_name, movie_year, movie_director,
          movie_leadactor, movie_type
          FROM movie
          ORDER BY movie_name ASC,
                  movie_year DESC';
$výsledky = mysql_query($dotaz, $db) or die(mysql_error($db));
```

```
// zjistí počet řádků ve výsledku
$počet_filmů = mysql_num_rows($výsledky);
```

```
$tabulka = <<<ENDHTML
<div style="text-align: center;">
  <h2>Databáze s recenzemi filmů</h2>
  <table border="1" cellpadding="2" cellspacing="2"
    style="width: 70%; margin-left: auto; margin-right: auto;">
    <tr>
      <th>Název filmu</th>
      <th>Rok uvedení na plátna kin</th>
      <th>Režisér</th>
      <th>V hlavní roli</th>
      <th>Kategorie</th>
    </tr>
  </div>
ENDHTML;
```

```
// projdi výsledky
while ($řádek = mysql_fetch_assoc($výsledky)) {
    extract($řádek);
    $tabulka .= <<<ENDHTML
    <tr>
        <td>$movie_name</td>
        <td>$movie_year</td>
        <td>$movie_director</td>
        <td>$movie_leadactor</td>
        <td>$movie_type</td>
    </tr>
ENDHTML;
}
```

```
$tabulka .= <<<ENDHTML
</table>
<p>Počet filmů: $počet_filmů</p>
</div>
ENDHTML;
?>
```

- Uložte soubor `table2.php` a otevřete jej ve svém webovém prohlížeči. Možná vás překvapí, že nevidíte žádný výstup! Je to dáno faktem, že jste interpretu jazyka PHP dosud neřekli, aby obsah proměnné `$tabulka` odeslal prohlížeči.
- Na předposlední řádek souboru přidejte příkaz `echo`. Soubor uložte a znovu zobrazte. Nyní by výstup již měl vypadat jako předtím (na obrázku 4.2).

```
echo $tabulka;
?>
```

Jak to funguje

Nejdříve jste ve svém prohlížeči žádný výstup neviděli, protože informace byly shromážděny v proměnné `$tabulka` a nebyly odeslány prohlížeči. O to se postaral až příkaz `echo`, který jste přidali na konec souboru. Na stránce se tedy znovu objevila známá tabulka!

Při přidávání dalšího textu do proměnné `$tabulka` musíte dbát na to, abyste místo operátoru `=` používali operátor `.=`, který připojí text za původní obsah proměnné, zatímco operátor `=` stávající hodnotu jen přepíše.

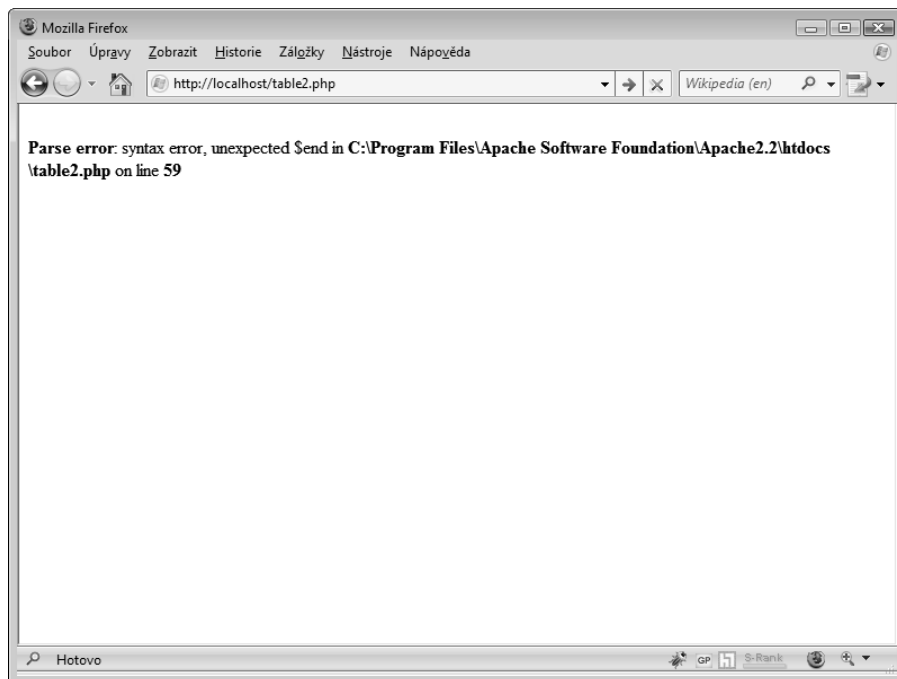
Možná si vzpomenete z dřívějšího výkladu ve 2. kapitole ohledně použití syntaxe heredoc, že identifikátor `ENDHTML` lze zaměnit prakticky za cokoliv jiného, podstatné je pouze to, aby se počáteční a koncové značky shodovaly. To znamená, že i dále uvedený kód bude fungovat bez problémů:

```
$tabulka =<<<HAHAHA
    // sem přijde kód HTML
HAHAHA;
```


Avšak toto již fungovat nebude:

```
$tabulka =<<<HAHAHA
    // sem přijde kód HTML
BOOH00;
```

V takovém případě totiž obdržíte chybu jako na obrázku 4.3.



Obrázek 4.3

Všimněte si, že po značkách <<<ENDHTML a ENDHTML; nesmí následovat žádná mezera. Kromě toho na řádku s uzavírací značkou syntaxe heredoc nesmějí být žádné úvodní mezery, odsazení nebo jiné znaky (kromě středníků). Kdyby tam byla byl jen jediná mezera, obdržíte chybové hlášení. (Můžete tak strávit hodiny a hodiny opravou chyby, která je výsledkem zapomenuté mezery před těmito značkami!) Nikdy nezapomeňte všechny mezery před těmito značkami smazat.

Syntaxi heredoc lze také použít i v jiných situacích, než je odeslání výstupu prohlížeči. Používá se třeba pro přiřazení velkých bloků obsahu proměnné, takže ji lze použít také pro přiřazení dotazu SQL proměnné, což je možné provést například takto:

```
$dotaz = <<<ENDSQL
SELECT movie_name, movie_year, movie_director,
       movie_leadactor, movie_type
FROM movie
ORDER BY movie_name ASC, movie_year DESC
ENDSQL;
```

Tabulka možná vypadá pěkně, stejně jako ve 3. kapitole, ale není uživatelům nic platná, nemá-li zrovna po ruce svůj tajný dešifrovací prsten, který by jim dešifroval, o které režiséry a herce vlastně jde. Pro získání těchto informací musíte svoje tabulky propojit.

Vyzkoušejte

Vylepšení tabulky

V tomto cvičení propojíte tabulky dohromady, takže budete konečně moci zobrazit smysluplná data.

1. Upravte soubor `table2.php` podle zvýrazněného textu:

```
<?php
// vezme identifikátor režiséra a vrátí jeho celé jméno
function načti_režiséra($id_režiséra) {
    global $db;

    $dotaz = 'SELECT people_fullname
              FROM people
              WHERE people_id = '. $id_režiséra;
    $výsledky = mysql_query($dotaz, $db) or die(mysql_error($db));

    $řádek = mysql_fetch_assoc($výsledky);
    extract($řádek);

    return $people_fullname;
}

// vezme identifikátor hlavního aktéra a vrátí jeho celé jméno
function načti_hlaktéra($id_hlaktéra) {
    global $db;

    $dotaz = 'SELECT people_fullname
              FROM people
              WHERE people_id = '. $id_hlaktéra;
    $výsledky = mysql_query($dotaz, $db) or die(mysql_error($db));

    $řádek = mysql_fetch_assoc($výsledky);
    extract($řádek);

    return $people_fullname;
}

// vezme identifikátor žánru a vrátí jeho smysluplný textový popis
function načti_žánr($id_žánru) {
    global $db;

    $dotaz = 'SELECT movietype_label
```

```

        FROM movietype
        WHERE movietype_id = '. $id_žánru;
    $výsledky = mysql_query($dotaz, $db) or die(mysql_error($db));

    $řádek = mysql_fetch_assoc($výsledky);
    extract($řádek);

    return $movietype_label;
}

// připoj se k MySQL
$db = mysql_connect('localhost', 'uzivatel', 'heslo') or
    die('Nemohu se připojit. Zkontrolujte připojovací parametry.');
```

```

// zajisti, abychom používali správnou databázi
mysql_select_db('moviesite', $db) or die(mysql_error($db));

// načti informace
$dotaz = 'SELECT movie_name, movie_year, movie_director,
        movie_leadactor, movie_type
        FROM movie
        ORDER BY movie_name ASC,
        movie_year DESC';
$výsledky = mysql_query($dotaz, $db) or die(mysql_error($db));

// zjistí počet řádků ve výsledku
$počet_filmů = mysql_num_rows($výsledky);

$tabulka = <<<ENDHTML
<div style="text-align: center;">
  <h2>Databáze s recenzemi filmů</h2>
  <table border="1" cellpadding="2" cellspacing="2"
    style="width: 70%; margin-left: auto; margin-right: auto;">
    <tr>
      <th>Název filmu</th>
      <th>Rok uvedení na plátna kin</th>
      <th>Režisér</th>
      <th>V hlavní roli</th>
      <th>Kategorie</th>
    </tr>
ENDHTML;

// projdi výsledky
while ($řádek = mysql_fetch_assoc($výsledky)) {
    extract($řádek);
```

```
$režisér = načti_režiséra($movie_director);  
$hlaktér = načti_hlaktéra($movie_leadactor);  
$žánr = načti_žánr($movie_type);
```

```
$tabulka .= <<<ENDHTML  
<tr>  
  <td>$movie_name</td>  
  <td>$movie_year</td>  
  <td>$režisér</td>  
  <td>$hlaktér</td>  
  <td>$žánr</td>
```

```
</tr>  
ENDHTML;  
}
```

```
$tabulka .= <<<ENDHTML  
</table>  
<p>Počet filmů: $počet_filmů</p>  
</div>  
ENDHTML;
```

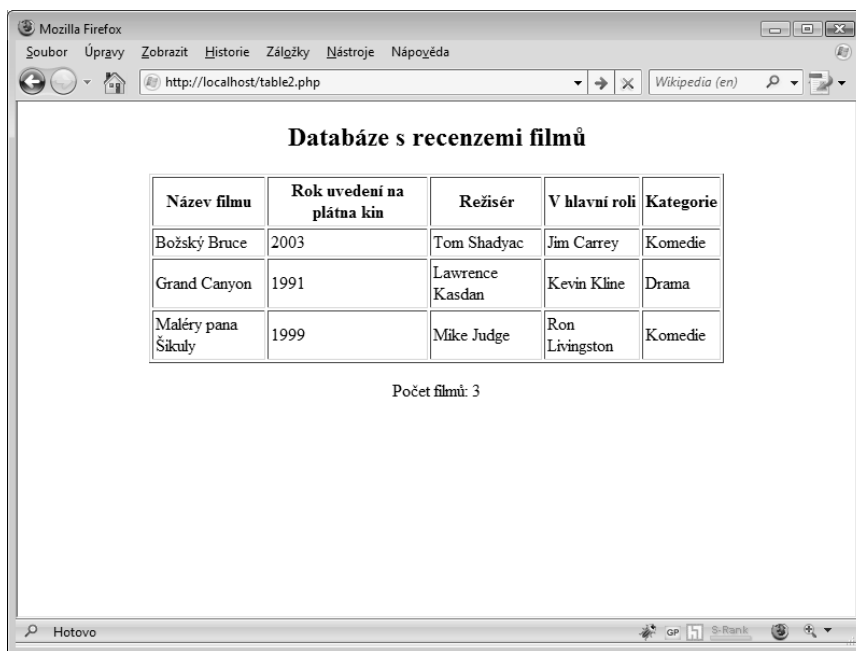
```
echo $tabulka;  
?>
```

2. Svůj soubor uložte a v prohlížeči jej znovu načtěte. Vaše obrazovka by nyní měla vypadat jako na obrázku 4.4.

Jak to funguje

Pomocí našich vlastních funkcí `načti_režiséra()`, `načti_hlaktéra()` a `načti_žánr()` si skript vyžádá načtení těchto údajů ze serveru pro každý jednotlivý řádek v tabulce. Díky tomu získáme informace, které potřebujeme, aniž bychom komplikovali původní dotaz klauzulí `LEFT JOIN`.

Gratulujeme! Právě jste úspěšně vytvořili výkonný skript, který odešle dotaz do databáze a výsledky vloží do tabulky HTML. Teď se můžete spokojeně poplácat po rameni. Ale jak už to chodí u všech dobrých badatelů, musíme vyrazit dále.



Obrázek 4.4

Hlavní a podřízené údaje

Pokusme se nyní rozšířit nabyté vědomosti a přidejme do tabulky dodatečné informace a funkce. Implementace vztahu hlavní/podřízený umožní uživatelům klepnutím na název filmu získat podrobnější informace o vybraném titulu. Je jasné, že takové stránky je třeba generovat dynamicky. Pokusme se nyní implementovat právě vztah mezi hlavním a podřízenými údaji.

Vyzkoušejte

Přidání odkazů do tabulky

Kroky popisované v tomto cvičení umožní načíst do okna prohlížeče další informace v závislosti na filmu, na který uživatel klepne. Postupujte následujícím způsobem:

1. Otevřete soubor `table2.php` a upravte dotaz, který načítá informace o filmu, tak, aby načítal také pole `movie_id`.

```
// načti informace
$dotaz = 'SELECT movie_id, movie_name, movie_year, movie_director,
          movie_leadactor, movie_type
FROM movie
ORDER BY movie_name ASC,
         movie_year DESC';
```

2. Upravte syntaxi heredoc, jež generuje řádky tabulky, a to tak, aby se z názvu filmu stal hypertextový odkaz.

```
$tabulka .= <<<ENDHTML
<tr>
  <td>
    <a href="movie_details.php?movie_id=$movie_id">$movie_name</a>
  </td>
  <td>$movie_year</td>
  <td>$režisér</td>
  <td>$hlaktér</td>
  <td>$žánr</td>
</tr>
ENDHTML;
```

3. Soubor uložte jako `table3.php` a otevřete jej ve svém webovém prohlížeči. Jeho obrazovka by měla vypadat jako na obrázku 4.5.



Obrázek 4.5

Jak to funguje

Jistě jste si všimli drobného rozdílu oproti obrázku 4.4 (`table2.php`) a 4.5 (`table3.php`). Vaši návštěvníci mohou nyní klepnout na odkazy vedoucí k podrobnějším informacím o každém z filmů.

První změna provedená v předchozí části upravila dotaz MySQL, aby obsahoval také pole `movie_id`. Druhá změna vytvořila kód HTML, jenž z názvů filmů vytváří hypertextové odkazy.

Chcete-li, můžete k odkazům doplnit popisky. K tomu slouží atribut `title`. Naneštěstí jej však některé webové prohlížeče nepodporují (omlouváme se všem, kteří takové prohlížeče používají).

Informace o používání atributů `alt` a `title` pro popisky najdete na adrese http://developer.mozilla.org/en/docs/Defining_Cross-Browser_Tooltips.

```
<a href="movie_details.php?movie_id=$movie_id"
  title="Více informací o filmu $movie_name">
  $movie_name
</a>
```

Změny jsou tedy hotovy, ale co vlastně dělají? Umístíte kurzor myši nad některý z odkazů a v případě, že máte zobrazen stavový pruh, uvidíte, že každý odkaz je jedinečný a je vygenerován dynamicky. Této stránce se říká též *hlavní stránka* (master page) a stránce, na kterou odkaz ukazuje, se říká *podřízená stránka* (child page).

Než budeme pokračovat, je nutné do stávající databáze přidat ještě nějaká data, která lze použít jako podrobné informace o filmu. Možná si z textu ve 3. kapitole pamatujete, že pro každý film uchováváme název, režiséra, obsazení hlavní role, žánr a rok uvedení filmu. Nyní k těmto informacím přidáme údaje o délce filmu, zisku a nákladech na výrobu filmu.

Vyzkoušejte Přidání dat do tabulky

V tomto cvičení přidáte do databáze dodatečná data o každém z filmů.

1. Otevřete svůj textový editor a napište následující kód:

```
<?php
$db = mysql_connect('localhost', 'uzivatel', 'heslo') or
    die('Nemohu se připojit. Zkontrolujte připojovací parametry.');
```

```
mysql_select_db('moviesite', $db) or die(mysql_error($db));

// uprav tabulku movie tak, aby obsahovala navíc délku,
// náklady a výnosy filmu
$dotaz = 'ALTER TABLE movie ADD COLUMN (
    movie_running_time TINYINT UNSIGNED NULL,
    movie_cost          DECIMAL(4,1)     NULL,
    movie_takings       DECIMAL(4,1)     NULL)';
mysql_query($dotaz, $db) or die (mysql_error($db));

// vlož nová data pro každý film do tabulky movie
$dotaz = 'UPDATE movie
    SET movie_running_time = 101,
        movie_cost = 81,
        movie_takings = 242.6
    WHERE
        movie_id = 1';
mysql_query($dotaz, $db) or die(mysql_error($db));
```

```
$dotaz = 'UPDATE movie
        SET movie_running_time = 89,
            movie_cost = 10,
            movie_takings = 10.8
        WHERE
            movie_id = 2';
mysql_query($dotaz, $db) or die(mysql_error($db));

$dotaz = 'UPDATE movie
        SET movie_running_time = 134,
            movie_cost = NULL,
            movie_takings = 33.2
        WHERE
            movie_id = 3';
mysql_query($dotaz, $db) or die(mysql_error($db));

echo 'Databáze filmů byla úspěšně aktualizována!';
?>
```

2. Soubor uložte jako db_ch04-1.php a poté jej otevřete ve svém prohlížeči. Po vložení informací do databáze by se vám měla zobrazit zpráva o úspěšné aktualizaci.

Jak to funguje

Ve skriptu jsme nejdříve pomocí příkazu ALTER TABLE přidali do stávající tabulky movie příslušná pole, do nichž jsme poté vložili nová data pomocí příkazu UPDATE. Pokud tyto příkazy neznáte, měli byste si znovu prostudovat 3. kapitolu.

Nyní již máte data na místě, můžete tedy vytvořit novou stránku, která bude sloužit pro zobrazení dodatečných informací o filmu (movie_details.php).

Vyzkoušejte

Zobrazení podrobných údajů o filmu

V tomto cvičení vytvoříte novou stránku, která zobrazí data přidaná v předchozím cvičení.

1. Otevřete textový editor a napište následující program:

```
<?php
// vezme identifikátor režiséra a vrátí jeho celé jméno
function načti_režiséra($id_režiséra) {
    global $db;

    $dotaz = 'SELECT people_fullname
            FROM people
            WHERE people_id = '. $id_režiséra;
    $výsledky = mysql_query($dotaz, $db) or die(mysql_error($db));

    $řádek = mysql_fetch_assoc($výsledky);
```



```

extract($řádek);

return $people_fullname;
}

// vezme identifikátor hlavního aktéra a vrátí jeho celé jméno
function načti_hlaktéra($id_hlaktéra) {
    global $db;

    $dotaz = 'SELECT people_fullname
              FROM people
              WHERE people_id = '. $id_hlaktéra;
    $výsledky = mysql_query($dotaz, $db) or die(mysql_error($db));

    $řádek = mysql_fetch_assoc($výsledky);
    extract($řádek);

    return $people_fullname;
}

// vezme identifikátor žánru a vrátí jeho smysluplný textový popis
function načti_žánr($id_žánru) {
    global $db;

    $dotaz = 'SELECT movietype_label
              FROM movietype
              WHERE movietype_id = '. $id_žánru;
    $výsledky = mysql_query($dotaz, $db) or die(mysql_error($db));

    $řádek = mysql_fetch_assoc($výsledky);
    extract($řádek);

    return $movietype_label;
}

// funkce pro výpočet, zda film byl film ziskový,
// ztrátový nebo zda zůstal na nule
function spočítej_rozdíl($výnosy, $náklady) {
    $rozdíl = $výnosy - $náklady;

    if ($rozdíl < 0) {
        $barva = 'red';
        $rozdíl = abs($rozdíl). ' mil. dolarů';
    } elseif ($rozdíl > 0) {
        $barva = 'green';
        $rozdíl = $rozdíl. ' mil. dolarů';
    }
}

```

```
    } else {
        $barva = 'blue';
        $rozdíl = 'ústal na nule';
    }

    return '<span style="color:'. $barva. ';">'. $rozdíl. '</span>';
}

// připoj se k MySQL
$db = mysql_connect('localhost', 'uzivatel', 'heslo') or
    die('Nemohu se připojit. Zkontrolujte připojovací parametry.');
```

```
mysql_select_db('moviesite', $db) or die(mysql_error($db));

// načti údaje
$dotaz = '
SELECT movie_name, movie_year, movie_director, movie_leadactor,
       movie_type, movie_running_time, movie_cost, movie_takings
FROM movie
WHERE movie_id = '. $_GET['movie_id'];
$výsledky = mysql_query($dotaz, $db) or die(mysql_error($db));

$řádek = mysql_fetch_assoc($výsledky);
$film_název      = $řádek['movie_name'];
$film_režisér    = načti_režiséra($řádek['movie_director']);
$film_hlaketér   = načti_hlaktéra($řádek['movie_leadactor']);
$film_rok        = $řádek['movie_year'];
$movie_running_time = $řádek['movie_running_time']. ' minut';
$film_výnosy     = $řádek['movie_takings']. ' mil. dolarů';
$film_náklady    = $řádek['movie_cost']. ' mil. dolarů';
$film_bilance    = spočítej_rozdíl($řádek['movie_takings'],
                                   $řádek['movie_cost']);

// zobraz informace
echo <<<ENDHTML
<html>
<head>
  <title>Podrobné údaje o filmu: $film_název</title>
</head>
<body>
  <div style="text-align: center;">
    <h2>$film_název</h2>
    <h3><em>Podrobnosti</em></h3>
    <table cellpadding="2" cellspacing="2"
      style="width: 70%; margin-left: auto; margin-right: auto;">
    <tr>
      <td><strong>Název</strong></strong></td>
```

```

        <td>${film_název}</td>
        <td><strong>Rok uvedení</strong></td>
        <td>${film_rok}</td>
    </tr><tr>
        <td><strong>Režie</strong></td>
        <td>${film_režisér}</td>
        <td><strong>Náklady</strong></td>
        <td>${film_náklady}</td>
    </tr><tr>
        <td><strong>V hlavní roli</strong></td>
        <td>${film_hlaketér}</td>
        <td><strong>Výnosy</strong></td>
        <td>${film_výnosy}</td>
    </tr><tr>
        <td><strong>Délka</strong></td>
        <td>${movie_running_time}</td>
        <td><strong>Balance</strong></td>
        <td>${film_balance}</td>
    </tr>
</table></div>
</body>
</html>
ENDHTML;
?>

```

2. Uložte soubor jako `movie_details.php` a nahrajte jej na svůj webový server.
3. Otevřete soubor `table3.php` ve svém prohlížeči a klepněte na jeden z odkazů. Otevře se stránka `movie_details.php` a ve vašem prohlížeči se zobrazí výsledek jako na obrázku 6.4.

Jak to funguje

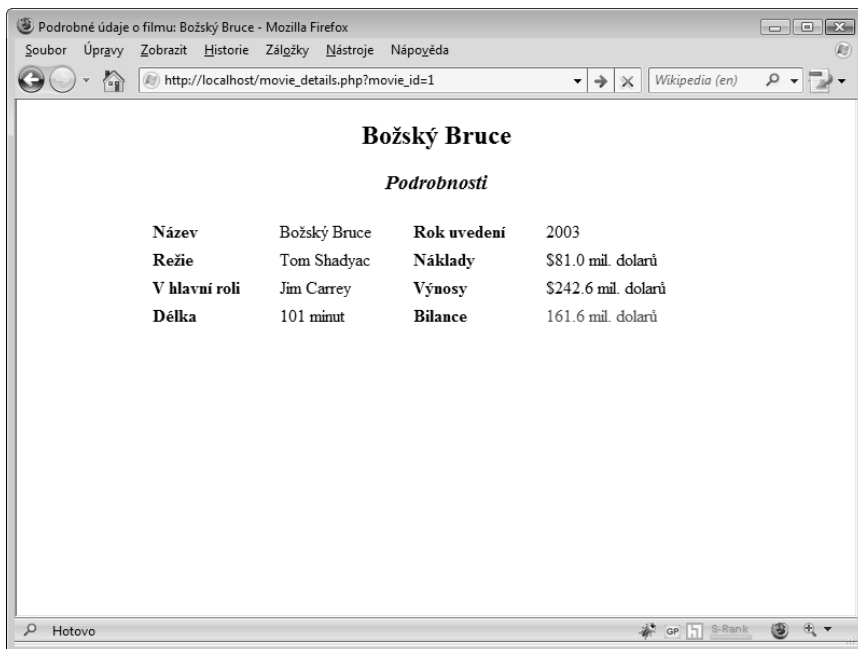
Tři funkce na začátku skriptu by vám měly být povědomé: `načti_režiséra()`, `načti_hlaktéra()` a `načti_žánr()`. Každá přijímá identifikátor a pomocí vyhledání v příslušné databázové tabulce jej překládá na odpovídající hodnotu čitelnou pro člověka. Ve skutečnosti si funkce můžete představit jako vlastní malinké programky, které existují uvnitř většího skriptu a které přijímají určité údaje, jež zpracují, načež vrátí nějaký výsledek.

Čtvrtá funkce, pojmenovaná `spočítej_rozdíl()`, generuje řetězec s kódem jazyka HTML, který zobrazuje informaci o tom, zda byl daný film výtěžný, prodělečný nebo zda skončil na nule. Přijímá výnosy a náklady související s filmem. Nejdříve vypočítá rozdíl odečtením nákladů od výnosů. Pomocí příkazu `if` pak dále upravuje výstup. Pokud film prodělal, pak je rozdíl záporný, takže první blok kódu nastaví barvu na červenou a převodem funkcí `abs()` na absolutní hodnotu ořízne záporné znaménko. Je-li rozdíl kladný, znamená to, že film vydělal, a proto se zisk zobrazí v zelené barvě. Poslední klauzule nastavuje modrou barvu pro případ, že film skončil finančně neutrálně.

Skript se připojí k databázi a z tabulky `movie` načte informace o filmu. Klauzule `WHERE` v dotazu zajistí, že se načtou údaje o správném filmu, protože porovnává pole `movie_id` s hodnotou, která byla skriptu předána prostřednictvím adresy URL. Jak jste si jistě všimli, tentokrát jsme pro získání informací z polí po provedení dotazu nepoužili funkci `extract()`. Místo toho jsme je do přísluš-

ných proměnných přiřadili přímo z pole \$řádek. Tyto hodnoty totiž nebudeme používat v jejich původním stavu, ale k délce připojíme řetězec „minut“ a k částkám řetězec „mil. dolarů“.

Poté voláme funkci `spočítej_rozdíl()` a vrácený kód HTML uložíme do proměnné `$film_bilance`. Pak pomocí příkazu `echo` a syntaxe `heredoc` zobrazíme uživateli tyto informace v tabulce naformátované pomocí jazyka HTML.



Obrázek 4.6

Trvalá relace

Co dělat v případě, kdy budete chtít vyhledat všechny recenze určitého filmu? V současném stavu musíte v souboru `movies_details.php` vytvořit nový dotaz SQL, který spustíte při načtení stránky do okna prohlížeče. Pak to ovšem znamená, že na jedné stránce budou vykonány dva dotazy SQL. Kód sice bude fungovat, ale nebude příliš efektivní. (Chceme být přece efektivními programátory, že?). Výsledkem by byl zbytečný kód.

Je tedy načase položit si zásadní otázku: Co jsou relace?

Relace je takový způsob spojování databázových tabulek, aby bylo možné přistupovat k datům ze všech tabulek. Výhodou MySQL je skutečnost, že se jedná o relační databázi, která jako taková tvorbu relací mezi tabulkami podporuje. Ve správných rukou (asi chvíli potrvá, než se vše naučíte) mohou být relace opravdu velmi výkonným nástrojem, který lze používat k získávání dat z velkého množství tabulek v jediném dotazu SQL.

Nejpěšším způsobem, jak si ukázat relace, je využít dosavadní práce. Pustíme se tedy do díla.

Vyzkoušejte

Tvorba a naplnění tabulky recenzí

Dříve, než budete moci přistupovat k recenzím ve své tabulce recenzí filmů, musíte tuto tabulku nejdříve vytvořit a poté ji naplnit daty.

1. Otevřete svůj textový editor a napište následující kód:

```
<?php
$db = mysql_connect('localhost', 'uzivatel', 'heslo') or
    die('Nemohu se připojit. Zkontrolujte připojovací parametry.');
```

```
mysql_select_db('moviesite', $db) or die(mysql_error($db));

// vytvoř tabulku reviews
$dotaz = 'CREATE TABLE reviews (
    review_movie_id INTEGER UNSIGNED NOT NULL,
    review_date DATE NOT NULL,
    reviewer_name VARCHAR(255) NOT NULL,
    review_comment VARCHAR(255) NOT NULL,
    review_rating TINYINT UNSIGNED NOT NULL DEFAULT 0,

    KEY (review_movie_id)
)
ENGINE=MyISAM';
mysql_query($dotaz, $db) or die (mysql_error($db));

// vlož nová data do tabulky reviews
$dotaz = <<<ENDSQL
INSERT INTO reviews (review_movie_id, review_date,
    reviewer_name, review_comment,
    review_rating)
VALUES
    (1, "2008-09-23", "Marek Skála",
        "Věděl jsem, že to bude dobrý. I když moje přítelkyně
        mě donutila, abych se na něj podíval.", 4),
    (1, "2008-09-23", "Jakub Slonek",
        "No nevím, Popelka se mi líbila víc.", 2),
    (1, "2008-09-28", "Páťa", "Naprostá bomba!", 5),
    (2, "2008-09-23", "Marvin",
        "Tak tohle je můj oblíbený film. Moc jsem tomu nedával,
        ale nakonec jsme tehle kousek zamyloval.", 5),
    (3, "2008-09-23", "Václav K.",
        "Film se mi docela líbil, i když to nejdříve vypadalo jako
        nepovedené video od agenta nějaké cestovky.", 3)
ENDSQL;
mysql_query($dotaz, $db) or die(mysql_error($db));
```

```
echo 'Databáze filmů byla úspěšně aktualizována!';  
?>
```

2. Soubor uložte jako db_ch04-2.php a otevřete jej ve svém prohlížeči. Tabulka recenzí byla vytvořena a naplněna daty.

Jak to funguje

Nyní byste již měli vědět, jak se pomocí MySQL a PHP vytvářejí tabulky, takže uvedený kód snad ani nepotřebuje další komentář. Pokud jste měli s něčím potíže, nalistujte 3. kapitolu a znovu prostudujte související oddíly.

Vyzkoušejte

Zobrazení recenzí

V tomto příkladu propojíte dvě tabulky (movies a reviews) a zobrazíte recenze pro konkrétní film. K tomu je nutné provést spoustu změn v souboru movie_details.php, takže nejlepší bude, když si uděláte jeho záložní kopii, protože opatrnosti není nikdy dost. Pokud byste udělali nějakou chybu, můžete se vždy vrátit zpět k původní verzi. Pro zobrazení recenzí postupujte podle následujících kroků:

1. Přidejte uvedený kód do horní části souboru movie_details.php:

```
// funkce pro vygenerování hodnocení  
function vygeneruj_hodnocení($rating) {  
    $hodnocení_filmu = '';  
    for ($i = 0; $i < $rating; $i++) {  
        $hodnocení_filmu .= '';  
    }  
    return $hodnocení_filmu;  
}
```

2. Nyní rozdělte konec bloku heredoc, který vypisuje informace o filmu, na dvě části:

```
<td><strong>Balance</strong></td>  
<td>$film_balance<td />  
</tr>  
</table>  
ENDHTML;  
  
echo <<<ENDHTML  
</div>  
</body>  
</html>  
ENDHTML;
```

3. Mezi tyto dva bloky heredoc přidejte tento kód:

```
// načti recenze pro tento film  
$dotaz = '  
    SELECT review_movie_id, review_date, reviewer_name,
```

```

        review_comment, review_rating
    FROM reviews
    WHERE review_movie_id = '. $_GET['movie_id']. '
    ORDER BY review_date DESC';

$výsledky = mysql_query($dotaz, $db) or die(mysql_error($db));

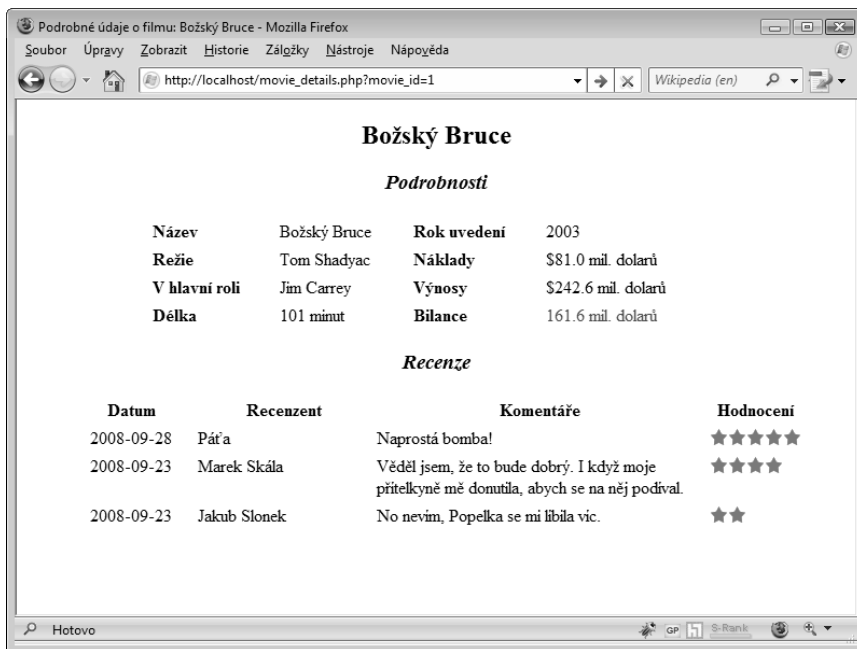
// zobraz recenze
echo <<< ENDHTML
<h3><em>Recenze</em></h3>
<table cellpadding="2" cellspacing="2"
    style="width: 90%; margin-left: auto; margin-right: auto;"
    <tr>
        <th style="width: 7em;">Datum</th>
        <th style="width: 10em;">Recenzent</th>
        <th>Komentáře</th>
        <th style="width: 5em;">Hodnocení</th>
    </tr>
ENDHTML;

while ($řádek = mysql_fetch_assoc($výsledky)) {
    $datum = $řádek['review_date'];
    $jméno = $řádek['reviewer_name'];
    $komentář = $řádek['review_comment'];
    $hodnocení = vygeneruj_hodnocení($řádek['review_rating']);

    echo <<<ENDHTML
    <tr>
        <td style="vertical-align:top; text-align: center;">$datum</td>
        <td style="vertical-align:top;">$jméno</td>
        <td style="vertical-align:top;">$komentář</td>
        <td style="vertical-align:top;">$hodnocení</td>
    </tr>
ENDHTML;
}

```

4. Soubor uložte jako `movie_details.php` (původní soubor přepište – snad jste si udělali záložní kopii).
5. Soubor nahrajte na svůj webový server, v prohlížeči načtete stránku `table3.php` a klepněte na libovolný film. Uvidíte výsledek podobný na obrázku 4.7.



Obrázek 4.7

Jak to funguje

Funkce `vygeneruj_hodnoceni()` je docela jednoduchá. Předáte jí hodnotu představující hodnocení filmu a ona vám vytvoří a vrátí řetězec s kódem jazyka HTML obsahující hodnotící obrázky k danému filmu. Všimněte si, že používáme operátor `.`, abychom u filmů s hodnocením větším než 1 zajistili, že se k jedinému hodnotícímu obrázku přidají také další obrázky.

Rozdělením bloku `heredoc` na dvě části jsme vytvořili místo pro vložení kódu HTML, který zobrazí recenze, aniž by došlo k porušení rozvržení stránky. První část zobrazí otevírací značky HTML a tabulku s detaily, zatímco druhá část zobrazí uzavírací značky pro celou stránku.

Pomocí dotazu jazyka MySQL načteme všechny recenze filmu s odpovídající hodnotou pole `review_movie_id`. Klauzule `ORDER BY` nařídí databázi MySQL, aby výsledky seřadila chronologicky v sestupném pořadí. Poté z výsledné sady extrahujeme hodnoty polí, které zobrazíme jako řádek v tabulce.

V této části jste provedli jen pár změn. Jak ale vidíte, stály opravdu za to. Nyní již tedy víte, jak pomocí MySQL vytvářet vztahy mezi tabulkami. Úspěšně jste načtli všechny recenze z tabulky `review` v závislosti na proměnné `movie_id`. Víte také, jak se pomocí superglobálního pole `$_GET` předávají hodnoty z jedné stránky na další.

Shrnutí

Naučili jste se, jak pomocí tabulek HTML zobrazovat data, jak data načítat z více než jedné databázové tabulky a hladce je zobrazit s daty z jiné tabulky a jak vytvářet dynamické stránky, jež zobrazují podrobné informace o řádcích z databázové tabulky. Svá data můžete návštěvníkům zobrazit také pomocí obrázků, podobně jako v případě hodnotících hvězdiček v příkladu v této kapitole.

Zatím jste všechny přírůstky do databáze psali napevno do skriptů, což není příliš dynamické. V 6. kapitole se naučíte, jak umožnit uživateli, aby mohl přidávat položky do databáze a upravovat je. Nejdříve se však musíte seznámit se způsobem, jakým se v prostředí PHP používají formuláře, na což se zaměříme v následující kapitole.

Cvičení

1. Přidejte záznam do horní tabulky ve svém souboru `movie_details.php`, který bude zobrazovat průměrné hodnocení udělené všemi recenzenty.
2. Všechny sloupce v záhlaví tabulky recenzí v souboru `movie_details.php` změňte na odkazy, které uživatelům umožní seřadit daný sloupec (tj. chce-li uživatel seřadit všechny recenze podle data, klepne na „Datum“).
3. Vytvořte střídavou barvu pozadí řádků v tabulce recenzí v souboru `movie_details.php`. Recenze se pak budou lépe číst. Náповěda: Liché řádky budou mít pozadí v jedné barvě, zatímco sudé řádky budou mít pozadí v jiné.