helion.pl
KSIĘGARNIA INTERNETOWA · WYDAWNICTWO

IDŹ DO
PRZYKŁADOWY ROZDZIAŁ
SPIS TREŚCI

KATALOG KSIĄŻEK
KATALOG ONLINE
ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK
DODAJ DO KOSZYKA

CENNIK I INFORMACJE
ZAMÓW INFORMACJE O NOWOŚCIACH
ZAMÓW CENNIK

CZYTELNIA
FRAGMENTY KSIĄŻEK ONLINE

PHP, Microsoft IIS i SQL Server. Projektowanie i programowanie baz danych

Autor: Andrzej Szeląg
ISBN: 978-83-246-1513-1
Format: 158x235, stron: 368



Poznaj i wykorzystaj narzędzia do projektowania baz danych

- Jak konfigurować repozytorium PEAR?
- Jak przetwarzać formularze za pomocą języka PHP?
- Jak wprowadzać dane do tabeli bazy danych, a później je usuwać?

Umiejętność tworzenia interaktywnych stron www typu e-commerce i zarządzania nimi jest dzisiaj właściwie niezbędna. Przydaje się nie tylko administratorom sieci, ale także wszystkim informatykom i właścicielom firm. Relacyjne bazy danych stanowią podstawę nowoczesnej aplikacji internetowej, więc przede wszystkim należy poznać najpopularniejsze języki wykorzystywane podczas pracy na tych bazach: język skryptowy PHP oraz język zapytań SQL, a także internetowe usługi informacyjne (IIS) występujące w najnowszych serwerowych systemach operacyjnych Microsoft Windows.

Książka „PHP, Microsoft IIS i SQL Server. Projektowanie i programowanie baz danych" w przystępny sposób i na konkretnym przykładzie przedstawia zagadnienia dotyczące projektowania, programowania i implementowania relacyjnej bazy danych. Mimo że podręcznik przeznaczony jest dla osób, które przynajmniej w niewielkim stopniu orientują się w tej tematyce, szeroka skala prezentowanych informacji (od prostych zagadnień teoretycznych do problemów zaawansowanych technicznie) sprawia, że każdy może z niego skorzystać – aby na przykład nauczyć się podstawowych instrukcji języka SQL i PHP, instalowania i konfigurowania ich w systemie Windows, korzystania z usług IIS i wreszcie zarządzania własną bazą danych.

- System zarządzania bazą danych
- Nazwy i typy danych
- Indeksy, klucze i schematy
- Architektura baz danych
- Normalizacja i projektowanie baz danych
- Środowisko języka SQL
- Podstawowe klauzule języka SQL
- Podstawowe właściwości języka PHP
- Architektura usług IIS
- Obsługa baz danych Microsoft SQL z poziomu PHP
- PEAR

Znajdziesz tu wszystko, czego potrzebujesz, aby stworzyć własną profesjonalną bazę danych!

Wydawnictwo Helion
ul. Kościuszki 1c
44-100 Gliwice
tel. 032 230 98 63
e-mail: helion@helion.pl

Spis treści

Wprowadzenie 9
Kto powinien przeczytać tę książkę? 10
O czym jest ta książka? 11
Informacje o testowym środowisku sieciowym 13
Przykłady do książki 24

Rozdział 1. Podstawy baz danych 25
Czym jest baza danych? 25
System Zarządzania Bazą Danych 27
Podstawowe pojęcia związane z bazami danych 30
Tabele, kolumny i wiersze danych 31
Nazwy i typy danych 36
Indeksy, klucze i schematy 38
Architektura baz danych 43
Typy baz danych 46
Normalizacja relacyjnej bazy danych 48
Projektowanie relacyjnej bazy danych 51
Tworzenie relacyjnej bazy danych 56
Dodatkowe źródła informacji 68

Rozdział 2. Język SQL 69
Czym jest SQL? 69
Środowisko języka SQL 73
Podstawowe instrukcje języka SQL 77
Data Definition Language (DDL) 78
Instrukcja CREATE DATABASE 79
Instrukcja CREATE TABLE 81
Instrukcja CREATE VIEW 83
Instrukcja CREATE INDEX 84
Data Manipulation Language (DML) 85
Instrukcja INSERT 86
Instrukcja SELECT 88
Instrukcja UPDATE 93
Instrukcja DELETE 95
Data Control Language (DCL) 96

Podstawowe klauzule języka SQL ... 97
Klauzula FROM ... 97
Klauzula WHERE ... 98
Klauzula ORDER BY ... 99
Klauzula GROUP BY ... 100
Klauzula HAVING ... 101
Dodatkowe źródła informacji ... 102

Rozdział 3. PHP ... 103
Czym jest PHP? ... 104
Podstawowe elementy języka PHP ... 114
Zmienne i stałe ... 114
Wyrażenia i operatory ... 117
Struktury kontrolujące ... 124
Funkcje ... 129
Tablice ... 131
Podstawowe właściwości języka PHP ... 133
Przetwarzanie formularzy ... 133
Zapis i odczyt plików ... 138
Data i czas ... 141
Instalowanie PHP w systemie Windows Server 2003 i Windows Vista ... 143
Pobieranie PHP ... 144
Instalowanie i konfigurowanie PHP w systemie Windows ... 146
Dodatkowe źródła informacji ... 156

Rozdział 4. Microsoft IIS ... 157
Podstawowe informacje o usługach IIS ... 158
Architektura usług IIS ... 160
Instalowanie serwera IIS 6.0 w systemie Windows Server 2003 ... 165
Instalowanie serwera IIS 7.0 w systemie Windows Vista ... 180
Rozszerzenia ISAPI i CGI/FastCGI ... 184
Dodawanie obsługi PHP do serwera IIS 6.0 jako rozszerzenia ISAPI ... 185
Dodawanie obsługi PHP do serwera IIS 7.0 jako rozszerzenia ISAPI ... 204
Dodawanie obsługi PHP do serwera IIS 6.0 jako rozszerzenia CGI ... 210
Dodatkowe źródła informacji ... 218

Rozdział 5. Microsoft SQL Server ... 221
Microsoft SQL Server 2000 ... 222
Wprowadzenie ... 222
Przygotowanie do instalacji ... 223
Wymagania sprzętowe i programowe ... 224
Instalowanie bazy danych programu Microsoft SQL Server 2000 ... 225
Instalowanie dodatku Service Pack 4 dla Microsoft SQL Server 2000 ... 233
Microsoft SQL Server 2005 ... 238
Wprowadzenie ... 238
Przygotowanie do instalacji ... 240
Wymagania sprzętowe i programowe ... 241
Instalowanie bazy danych programu Microsoft SQL Server 2005 ... 241
Instalowanie dodatku Service Pack 2 dla Microsoft SQL Server 2005 ... 257
Logowanie do serwera bazy danych programu Microsoft SQL Server 2005 ... 264
Implementacja relacyjnej bazy danych w programie Microsoft SQL Server 2005 ... 265
Dodatkowe źródła informacji ... 270

Rozdział 6. Obsługa bazy danych Microsoft SQL Server z poziomu PHP 271
Aplikacja Obsługa bazy danych programu Microsoft SQL Server z poziomu skryptów PHP 272
Standardy sieciowe W3C 276
Oddzielenie struktury, prezentacji i zachowania 276
Deklaracja typu dokumentu i typu przestrzeni nazw 278
Deklaracja typu kodowania znaków 279
Deklaracja typu arkusza stylów 279
Konfigurowanie obsługi bazy danych w PHP 281
Obsługa bazy danych z poziomu skryptów PHP 286
Podstawowe funkcje PHP obsługujące bazę danych Microsoft SQL Server 291
Funkcja mssql_connect() 292
Funkcja mssql_select_db() 293
Funkcja mssql_query() 293
Funkcja mssql_num_rows() 294
Funkcja mssql_fetch_array() 295
Funkcja mssql_free_result() 296
Funkcja mssql_close() 296
Pozostałe funkcje PHP 296
Dostęp do bazy danych Microsoft SQL Server 2005 z poziomu skryptów PHP 297
Połączenie z serwerem bazy danych 298
Połączenie z relacyjną bazą danych 299
Wprowadzanie nowych danych do tabeli bazy danych 300
Wyświetlanie zawartości tabeli bazy danych 303
Uaktualnianie zawartości tabeli bazy danych 305
Usuwanie danych z tabeli bazy danych 309
Wyszukiwanie danych w tabeli bazy danych 311
Dodatkowe źródła informacji 314

Rozdział 7. PEAR 315
Czym jest PEAR? 315
Instalowanie PEAR DB w systemie Windows 317
Pobieranie repozytorium PEAR 317
Instalowanie repozytorium PEAR 317
Instalowanie modułu DB 321
Obsługa bazy danych Microsoft SQL Server 2005 z wykorzystaniem modułu PEAR DB 324
Połączenie z serwerem bazy 325
Połączenie z relacyjną bazą 327
Wprowadzanie nowych danych do tabeli bazy danych 328
Wyświetlanie zawartości tabeli bazy danych 332
Uaktualnianie zawartości tabeli bazy danych 334
Usuwanie danych z tabeli bazy danych 336
Wyszukiwanie danych w tabeli bazy danych 338
Dodatkowe źródła informacji 339

Dodatek A **Struktura relacyjnej bazy danych WirtualnySklep 341**
Dodatek B **Operatory stosowane w klauzuli WHERE 345**
Dodatek C **Funkcje PHP do kontroli typów zmiennych 347**
Dodatek D **Zmienne superglobalne występujące w PHP 349**
Dodatek E **Zawartość folderu skompresowanego phiisq.zip 351**
Skorowidz 353

Rozdział 6.

Obsługa bazy danych Microsoft SQL Server z poziomu PHP

Jedną z najważniejszych zalet języka PHP jest możliwość pozyskiwania różnych informacji z relacyjnych baz danych znajdujących się na serwerach baz danych firmy Microsoft: SQL Server 2000 lub SQL Server 2005 za pomocą skryptów PHP. Stąd też język PHP jest logicznym wyborem dla dynamicznego tworzenia zawartości stron czy aplikacji internetowych, za pomocą których możliwa jest obsługa relacyjnej bazy danych.

Poza wspomnianymi wyżej Systemami Zarządzania Bazami Danych język PHP standardowo udostępnia wiele wbudowanych oraz wyspecjalizowanych funkcji umożliwiających dostęp do większości popularnych Systemów Zarządzania Bazami Danych, do których można zaliczyć:

- Database firmy Oracle,
- MySQL firmy MySQL AB,
- PostgreSQL firmy PostgreSQL Global Development Group,
- DB2 firmy IBM.

Język PHP umożliwia również współpracę z relacyjnymi bazami danych obsługującymi uniwersalny standard ODBC (ang. *Open DataBase Connectivity*)[1], którym nie będziemy się zajmowali w tej książce. Skupimy się tu przede wszystkim na przedstawieniu obsługi bazy danych programu Microsoft SQL Server 2005 za pomocą wbudowanych w język PHP 5.2.5 funkcji do obsługi tego typu bazy danych.

[1] ODBC to niezależny od języka programowania, systemu operacyjnego i bazy danych interfejs API (ang. *Application Programming Interface*), który pozwala programom łączyć się z systemami zarządzania bazą danych.

Dostęp do przykładowej relacyjnej bazy danych o nazwie *WirtualnySklep* zostanie przedstawiony na przykładzie aplikacji internetowej o nazwie *Obsługa bazy danych Microsoft SQL Server z poziomu skryptów PHP*, na którą składa się kilkanaście skryptów PHP. Aplikacja ta będzie uruchamiana z poziomu przeglądarki internetowej klienta WWW. Z kolei za pomocą serwera internetowego IIS firmy Microsoft i odpowiednio skonfigurowanego interpretera skryptów PHP będzie realizowana obsługa relacyjnej bazy danych.

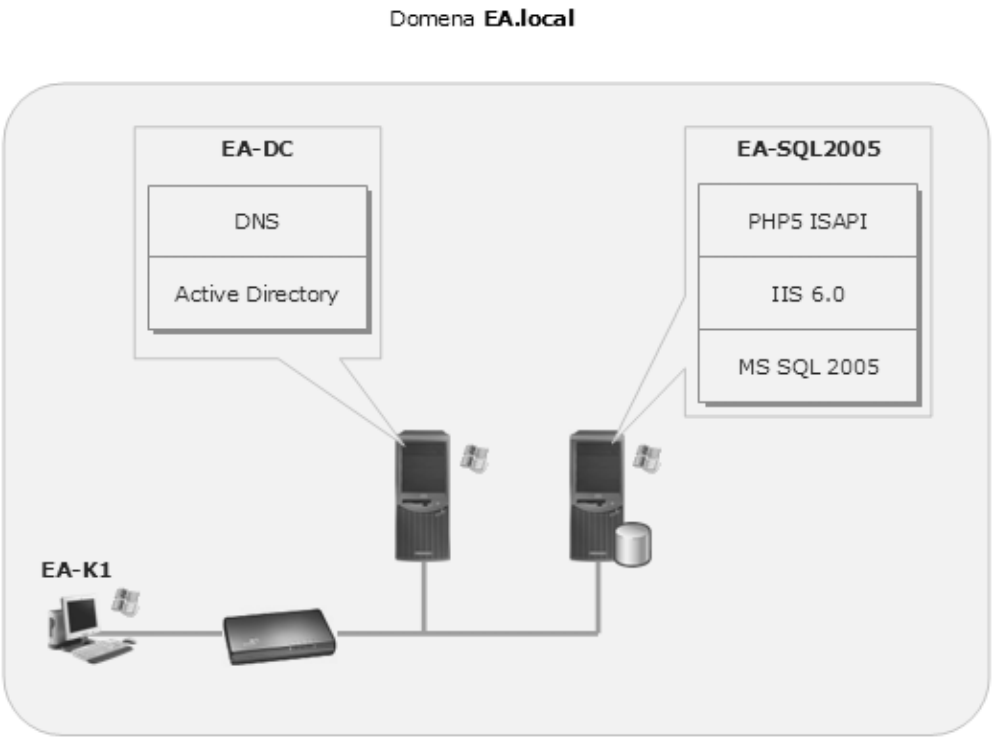
Aplikacja Obsługa bazy danych programu Microsoft SQL Server z poziomu skryptów PHP

W poprzednich rozdziałach zostały opisane różne składniki i zastosowania języka strukturalnego PHP oraz baz danych firmy Microsoft: SQL Server 2000 oraz SQL Server 2005. Z kolei dalsze podrozdziały koncentrują się na podstawowych elementach i metodach dostępu do relacyjnych baz danych przy wykorzystaniu przeglądarki internetowej oraz skryptów PHP. Przykłady te z pewnością nie mówią wszystkiego o dostępie do tego typu baz danych. Należy je raczej traktować jako wstęp do projektowania czy programowania na większą skalę, tj. tworzenia w pełni profesjonalnych i rozbudowanych aplikacji internetowych, za pomocą których możliwa będzie obsługa baz danych dowolnych typów.

Podane dalej przykłady dają solidne podstawy do zrozumienia kolejnych czynności, jakie mają miejsce podczas obsługi relacyjnej bazy danych programu Microsoft SQL Server 2005 z poziomu skryptów języka PHP. Do tych czynności można zaliczyć:

- połączenie z serwerem bazy danych o nazwie *EA-SQL2005*,
- wybranie relacyjnej bazy danych o nazwie *WirtualnySklep*,
- wykonanie zapytania SQL (lub kilku zapytań) na tabeli bazowej o nazwie *tKlienci* relacyjnej bazy danych,
- obejrzenie wyników przygotowanych przez interpreter PHP, przesłanych przez serwer internetowy i wyświetlonych w przeglądarce internetowej klienta,
- zwolnienie zajmowanych zasobów systemu przez wynik zapytania oraz połączenie z serwerem bazy danych.

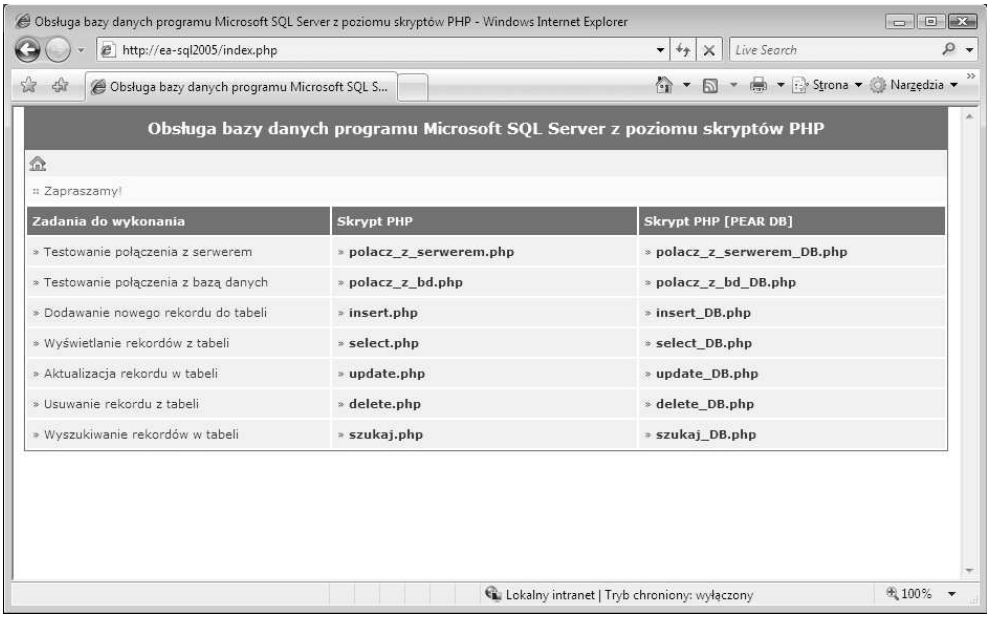
Wszystkie przykłady zawarte w tym i w następnym rozdziale będą uruchamiane w uproszczonym środowisku testowym, które zostało przedstawione na rysunku 6.1. Za pomocą przeglądarki internetowej Internet Explorer 7.0, która jest częścią systemu klienckiego Microsoft Windows Vista Business zainstalowanego na komputerze o nazwie *EA-K1*, będzie uzyskiwany dostęp do relacyjnej bazy danych o nazwie *WirtualnySklep*, która została zainstalowana na serwerze członkowskim *EA-SQL2005*. Na tym serwerze zostały również zainstalowane trzy aplikacje: interpreter PHP w wersji 5.2.5,



Rysunek 6.1. *Uproszczone środowisko testowe do obsługi bazy danych Microsoft SQL Server 2005 z poziomu skryptów PHP*

usługi IIS w wersji 6.0 (serwer internetowy) i baza danych programu Microsoft SQL Server 2005 w wersji Enterprise Evaluation Edition. Serwer członkowski *EA-SQL2005*, jak już wiemy z wcześniejszych rozdziałów, będzie pracował pod kontrolą systemu operacyjnego Microsoft Windows Server 2003 w wersji Enterprise Edition z dodatkiem Service Pack 2. Dostęp do serwera internetowego firmy Microsoft realizowany będzie za pomocą lokalnej sieci komputerowej, która pracuje w domenie sieciowej o nazwie *EA.local*.

Na serwerze internetowym IIS w wersji 6.0 (w jego domyślnym katalogu *%SystemDrive%\Inetpub\wwwroot*) powinny znajdować się skrypty PHP, które składają się na prostą aplikację internetową o nazwie *Obsługa bazy danych programu Microsoft SQL Server z poziomu skryptów PHP*. Wszystkie skrypty są dostępne pod adresem *ftp://ftp.helion.pl/przyklady/phiisq.zip* w folderze *Skrypty PHP*. Po ściągnięciu i rozpakowaniu tego katalogu należy je skopiować do powyższej lokalizacji na serwerze internetowym. Okno główne tej aplikacji zostało przedstawione na rysunku 6.2, a jej kod na listingu 6.1. Aplikacja ta składa się z kilkunastu przykładowych skryptów PHP zawierających kod XHTML, CSS, PHP i tzw. osadzony SQL, który przeznaczony jest głównie do manipulowania danymi zawartymi w relacyjnej bazie danych *WirtualnySklep* pracującej pod kontrolą programu Microsoft SQL Server 2005. Aplikacja internetowa, która stanowi część tej książki, została w skrócie przedstawiona szerzej w następnym podrozdziale.



Rysunek 6.2. *Okno główne aplikacji internetowej Obsługa bazy danych programu Microsoft SQL Server z poziomu skryptów PHP*

Listing 6.1. *Kod aplikacji internetowej Obsługa bazy danych programu Microsoft SQL Server z poziomu skryptów PHP (index.php)*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/tr/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
  <title>Obsługa bazy danych programu Microsoft SQL Server z poziomu skryptów
  PHP</title>
  <meta http-equiv="content-type" content="text/html; charset=iso-8859-2" />
  <link href="style.css" type="text/css" rel="stylesheet" />
</head>

<body>
  <div id="wyrownanie_do_srodka">
    <h1>
      <?php
        require_once('nazwa_programu.php');
      ?></h1>

    <h2> <a href="index.php"><img style="border:none;width:16px;height:16px"
    src="home.gif" alt="" title="Strona główna" /></a></h2>
    <h3> <img src="kropka.jpg" alt="" title="Strona główna" /> Zapraszamy!</h3>

      <table>
        <tbody>
          <tr>
            <td class="td_kolor_glowna_nagl"> Zadania do wykonania</td>
            <td class="td_kolor_glowna_nagl"> Skrypt PHP</td>
            <td class="td_kolor_glowna_nagl"> Skrypt PHP [PEAR DB]</td></tr>
```

```
<tr>
  <td class="td_kolor_glowna"> &#187; Testowanie połączenia
  z serwerem</td>
  <td class="td_kolor_glowna"> &#187; <a href="polacz_z_serwerem.php">
  polacz_z_serwerem.php</a></td>
  <td class="td_kolor_glowna"> &#187; <a href="polacz_z_
  ➥serwerem_DB.php">
  polacz_z_serwerem_DB.php</a></td></tr>

<tr>
  <td class="td_kolor_glowna"> &#187; Testowanie połączenia
  z bazą danych</td>
  <td class="td_kolor_glowna"> &#187; <a href="polacz_z_bd.php">
  polacz_z_bd.php</a></td>
  <td class="td_kolor_glowna"> &#187; <a href="polacz_z_bd_DB.php">
  polacz_z_bd_DB.php</a></td></tr>

<tr>
  <td class="td_kolor_glowna"> &#187; Dodawanie nowego rekordu
  do tabeli</td>
  <td class="td_kolor_glowna"> &#187; <a href="insert.php">
  insert.php</a></td>
  <td class="td_kolor_glowna"> &#187; <a href="insert_DB.php">
  insert_DB.php</a></td></tr>

<tr>
  <td class="td_kolor_glowna"> &#187; Wyświetlanie rekordów
  ➥z tabeli</td>
  <td class="td_kolor_glowna"> &#187; <a href="select.php">
  select.php</a></td>
  <td class="td_kolor_glowna"> &#187; <a href="select_DB.php">
  select_DB.php</a></td></tr>

<tr>
  <td class="td_kolor_glowna"> &#187; Aktualizacja rekordu w tabeli</td>
  <td class="td_kolor_glowna"> &#187; <a href="update.php">
  update.php</a></td>
  <td class="td_kolor_glowna"> &#187; <a href="update_DB.php">
  update_DB.php</a></td></tr>

<tr>
  <td class="td_kolor_glowna"> &#187; Usuwanie rekordu z tabeli</td>
  <td class="td_kolor_glowna"> &#187; <a href="delete.php">
  delete.php</a></td>
  <td class="td_kolor_glowna"> &#187; <a href="delete_DB.php">
  delete_DB.php</a></td></tr>

<tr>
  <td class="td_kolor_glowna"> &#187; Wyszukiwanie rekordów
  ➥w tabeli</td>
  <td class="td_kolor_glowna"> &#187; <a href="szukaj.php">
  szukaj.php</a></td>
  <td class="td_kolor_glowna"> &#187; <a href="szukaj_DB.php">
  szukaj_DB.php</a></td></tr></tbody></table></div></body></html>
```

Powyższy kod zostanie dość szczegółowo opisany w kolejnych podrozdziałach — szczególnie te jego elementy, które są związane ze standardami sieciowymi organizacji W3C, czyli między innymi z technologiami CSS i XHTML.

Aplikacja internetowa *Obsługa bazy danych programu Microsoft SQL Server z poziomu skryptów PHP* jest niezwykle prostym projektem zawierającym kilkanaście skryptów PHP, za pomocą których realizowana jest obsługa bazy danych programu Microsoft SQL Server 2005 przy wykorzystaniu funkcji bazodanowych dostępnych w PHP. Skrypty te ilustrują zastosowanie języków strukturalnych: PHP i SQL oraz bazy danych programu Microsoft SQL Server 2005. Oczywiście kod tej aplikacji można zoptymalizować tak, by w jeszcze mniejszym stopniu obciążał zasoby systemu i lepiej realizował swoje funkcje.

Standardy sieciowe W3C

W przypadku budowania witryn i aplikacji internetowych opartych na nowoczesnych standardach sieciowych należy pamiętać o tym, aby możliwe było ich szybkie przebudowanie w przyszłości, jeżeli kiedykolwiek zajdzie taka potrzeba. Pozwalają na to technologie rekomendowane przez międzynarodowe konsorcjum o nazwie W3C (ang. *World Wide Web Consortium*). Generalnie, aplikacja internetowa o nazwie *Obsługa bazy danych programu Microsoft SQL Server z poziomu skryptów PHP* jest prostym projektem, którego struktura została zbudowana przy użyciu tzw. standardów sieciowych[2].

Co daje projektowanie i budowanie witryn oraz aplikacji internetowych z użyciem standardów sieciowych? Odpowiedź jest prosta: obniżenie kosztów produkcji i utrzymania przy zapewnieniu dostępności dla klientów czy użytkowników, a także zgodność z przeglądarkami internetowymi i działanie na różnych platformach systemowych (Windows, Unix itp.). Generalnie mówiąc, standardy sieciowe są potężnym i profesjonalnym narzędziem, dzięki któremu można projektować oraz budować zaawansowane i profesjonalnie wyglądające witryny czy aplikacje internetowe. Są one kluczem do projektowania w niezwykle efektywny, efektowny, a zarazem dostępny dla wielu osób sposób.

Oddzielenie struktury, prezentacji i zachowania

Powszechnie znana wśród projektantów i programistów witryn czy aplikacji internetowych jest idea stosowania kodu XHTML[3] do opisywania struktury dokumentu internetowego (np. strony internetowej), stylów CSS[4] do opisywania jego wyglądu (czyli prezentacji), a języków PHP czy JavaScript — jego zachowania. Koncepcja oddzielenia sposobu prezentacji (za pomocą plików z rozszerzeniem *.css*) od zawartości (pliki z rozszerzeniem *.html*) może być z powodzeniem rozszerzona na skrypty PHP, tj. pliki

[2] Standardy sieciowe stanowią kontynuację istniejących dotąd technik internetowych. Są to więc języki strukturalne (takie, jak XHTML czy XML), języki warstwy aplikacji (takie, jak CSS) czy języki skryptowe (takie, jak ECMAScript).

[3] Więcej informacji na temat XHTML-a można znaleźć na stronie pod adresem *http://www.w3.org/MarkUp/*.

[4] Więcej informacji na temat CSS można znaleźć na stronie pod adresem *http://www.w3.org/Style/CSS/*.

z rozszerzeniem *.php*. Przykładem takiego rozszerzenia jest właśnie przedstawiona tutaj aplikacja internetowa *Obsługa bazy danych programu Microsoft SQL Server z poziomu skryptów PHP*. Dlaczego warto stosować tego typu rozwiązanie? Otóż dlatego, że dużo łatwiej utrzymywać, używać czy też modyfikować poszczególne pliki składające się na aplikację internetową przez dłuższy czas, jeżeli logika (pliki *.php*) będzie oddzielona od zawartości (pliki *.html*), a ta z kolei od sposobu prezentacji (pliki *.css*). Generalnie mówiąc, sprowadza się to do rozdzielenia kodu PHP od kodu XHTML i CSS. Bez tego kod aplikacji internetowej stanie się znacznie trudniejszy w utrzymaniu, użytkowaniu czy modyfikowaniu. Szczególnie ważne jest to np. w przypadku większych projektów. Otóż może się np. okazać, że w przypadku większych zmian w kodzie witryny czy aplikacji internetowej niemożliwe będzie przystosowanie jej do nowej rzeczywistości. Problemem może okazać się np. duża część kodu XHTML, który zawiera znaczniki HTML decydujące o wyglądzie poszczególnych jej elementów. Ich usunięcie czy zmodyfikowanie może zająć naprawdę bardzo dużo czasu oraz stworzyć wiele problemów.

Język znaczników XHTML może zawierać sformatowane dane tekstowe, obiekty osadzone (jak rysunki, filmy, prezentacje itp.). *Język prezentacyjny CSS* z kolei formatuje poszczególne witryny czy aplikacje internetowe oraz kontroluje rozmieszczenie, ułożenie czy kolor poszczególnych ich elementów. Zaś *język strukturalny PHP* czy *JavaScript* umożliwia tworzenie zaawansowanych funkcji witryny czy aplikacji internetowej, a także efektów działających w różnych przeglądarkach internetowych czy platformach systemowych (Windows, Unix, MacOS itp).


Uwaga

Odseparowanie struktury dokumentu od prezentacji i zachowania jest oznaką profesjonalnego podejścia do projektowania oraz programowania witryn i aplikacji internetowych. Mówiąc jeszcze inaczej, jest to metoda, według której będą projektowane witryny i aplikacje internetowe w przyszłości.

Autor niniejszej książki jest zdania, że strony czy aplikacje internetowe powinny być tworzone w oparciu o kaskadowe arkusze stylów (jak język prezentacji CSS) i standardy sieciowe (jak język znaczników XHTML). Dlaczego? Otóż powyższe mechanizmy rozwiązują liczne problemy nie tylko projektantów czy programistów, ale również zwykłych użytkowników, którzy korzystają z witryn i aplikacji internetowych.

Technologie takie, jak CSS czy XHTML, pozwalają:

- zmniejszać nakład pracy projektantów i programistów oraz skrócić czas projektowania,
- projektować zaawansowane funkcje interaktywne witryn lub aplikacji internetowych, które działają w różnych przeglądarkach internetowych i na różnych platformach systemowych (Windows, Unix, MacOS itp.),
- kontrolować rozmieszczenie w przeglądarkach internetowych poszczególnych elementów czy części witryn lub aplikacji internetowych,
- łatwo i szybko modyfikować część czy całość witryn lub aplikacji internetowych,
- odciążać serwery internetowe i łącza internetowe,

- oddzielać styl witryn lub aplikacji internetowych od ich struktury czy zachowania,
- skrócić czas wczytywania witryn internetowych,
- zapewnić wzrost dostępności witryny internetowej,
- obsługiwać urządzenia przenośne, takie jak np. telefony komórkowe czy inne urządzenia bezprzewodowe.

Deklaracja typu dokumentu i typu przestrzeni nazw

Aplikacja internetowa *Obsługa bazy danych programu Microsoft SQL Server z poziomu skryptów PHP* jest projektem, którego struktura została zbudowana przy użyciu jednego z trzech aktualnie wykorzystywanych standardów sieciowych — XHTML 1.0 Transtional, który rekomendowany jest przez konsorcjum W3C. Można się o tym przekonać, edytując kod dowolnego ze skryptów PHP, które składają się na tę aplikację. Mianowicie każdy dokument zaczyna się od deklaracji typu dokumentu internetowego wraz z deklaracją typu przestrzeni nazw, które zostały pokazane na listingu 6.2.

Listing 6.2. *Deklaracja typu dokumentu internetowego wraz z deklaracją typu przestrzeni nazw (index.php)*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/tr/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

Dokumenty internetowe, które zostały stworzone przy użyciu języka znaczników XHTML, rozpoczynają się od elementów informujących dowolną przeglądarkę internetową o tym, w jaki sposób mają być one interpretowane i sprawdzane. Stąd też pierwszy z tych elementów (`DOCTYPE`) jest *deklaracją typu dokumentu internetowego*. Generalnie język XHTML pozwala projektantom i programistom tworzyć klika różnych typów dokumentów, każdy z zestawem innych reguł, które są wymieniane w specyfikacji XHTML jako *definicja typu dokumentu* (w skrócie *DTD*). Wspomniana deklaracja typu dokumentu internetowego (`DOCTYPE`) informuje przeglądarkę internetową o tym, jakiej definicji typu dokumentu trzymał się programista tworzący daną stronę internetową. Deklaracja ta jest niezwykle ważna w dokumencie internetowym stworzonym w języku znaczników XHTML. Bez poprawnie zdefiniowanej deklaracji `DOCTYPE` żadna przeglądarka internetowa nie zweryfikuje kodu znaczników XHTML i stylu CSS. W efekcie pojawią się problemy, gdyż przeglądarka internetowa nie będzie potrafiła obsłużyć, tj. sprawdzić i poprawnie wyświetlić, strony internetowej. Po deklaracji `DOCTYPE` znajduje się *deklaracja typu przestrzeni nazw* XHTML. Przestrzeń ta jest kolekcją typów elementów i nazw atrybutów skojarzonych z definicją typu dokumentu internetowego. Deklaracja przestrzeni nazw pozwala zidentyfikować swoją przestrzeń nazw przez wskazanie jej lokalizacji (*http://www.w3.org/1999/xhtml*).

Deklaracja typu kodowania znaków

Aby dokumenty internetowe utworzone za pomocą języka znaczników o nazwie XHTML mogły być poprawnie zinterpretowane przez dowolną przeglądarkę internetową i przejść pomyślnie wszystkie testy sprawdzające, muszą deklarować tzw. typ kodowania znaków użyty do ich stworzenia. W przypadku aplikacji *Obsługa bazy danych programu Microsoft SQL Server z poziomu skryptów PHP* jest to `charset=iso-8859-2`, zgodnie z tym, co zostało przedstawione na listingu 6.3.

Listing 6.3. *Deklaracja typu kodowania znaków w dokumencie internetowym (index.php)*

```
<head>
  <title>Obsługa bazy danych programu Microsoft SQL Server z poziomu skryptów
  PHP</title>
  <meta http-equiv="content-type" content="text/html; charset=iso-8859-2" />
  <link href="style.css" type="text/css" rel="stylesheet" />
</head>
```

Jak widać na powyższym listingu, typ kodowania znaków, który zapewnia poprawne wyświetlanie znaków narodowych, został zadeklarowany za pomocą elementu `http-equiv="content-type"` umieszczonego w elemencie `<meta… />` części nagłówkowej dokumentu internetowego, czyli pomiędzy znacznikami `<head>…</head>`. Zaznaczony na listingu 6.3 czcionką pogrubioną wiersz pokazuje zapis dla kodowania `iso-8859-2`. Takie oznaczenie nosi polska strona kodowa, tj. dla polskich czcionek.



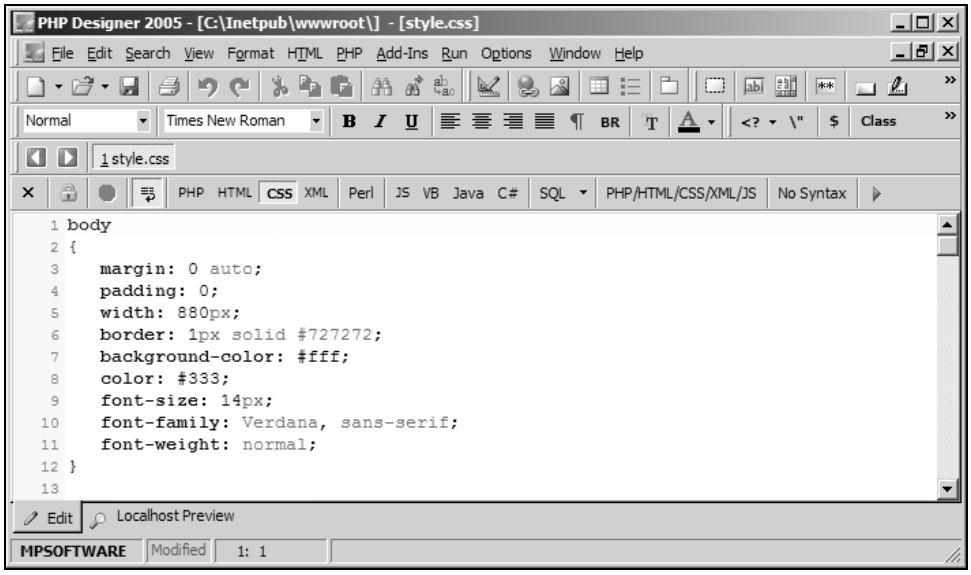
W przypadku tworzenia międzynarodowej witryny czy aplikacji internetowej, która będzie zawierać znaki niewystępujące w kodzie ASCI, należy użyć kodowania utf-8 zamiast iso-8859-2.

Standard ISO 8859 stworzony został w połowie lat 80. XX w. przez Europejskie Stowarzyszenie Producentów Komputerów. Jest on serią standaryzowanych wielojęzycznych zestawów znaków graficznych, które są kodowane na jednym bajcie (1 B), czyli 8 bitach (1 B = 8 bitów). Jednym z tych zestawów jest wspomniany już wyżej standard iso-8859-2.

Deklaracja typu arkusza stylów

Kaskadowe arkusze stylów mogą występować jako zewnętrzne, być osadzone w kodzie XHTML danej witryny internetowej lub występować w tzw. miejscu (ang. *inline*). W niniejszej książce zostanie omówione jedynie pierwsze z tych wystąpień arkuszy stylów. Dlaczego? Otóż dlatego, że przy zastosowaniu zewnętrznych arkuszy stylów otrzymujemy szereg wymiernych korzyści, które wynikają choćby ze znacznego zmniejszenia obciążenia serwera internetowego czy łączy internetowych. Poza tym są one obsługiwane przez większość popularnych przeglądarek internetowych (w tym również Internet Explorer, która jest składnikiem systemów operacyjnych z rodziny Microsoft Windows).

Zewnętrzny arkusz stylów, czyli plik z rozszerzeniem *.css* (np. *style.css*, który jest dostępny pod adresem *ftp://ftp.helion.pl/przyklady/phiisq.zip* w folderze *Skrypty PHP*, a jego przykładowa zawartość została pokazana na rysunku 6.3), jest dokumentem tekstowym zawierającym kod CSS w postaci określonych reguł CSS. Może on być wykorzystany z powodzeniem na stronie internetowej zgodnej ze standardami sieciowymi. Inaczej mówiąc, strona internetowa XHTML może używać pliku z rozszerzeniem *.css* (pliku arkusza stylów), odwołując się do niego poprzez element o nazwie `link`, który został umieszczony w nagłówku dokumentu internetowego, tj. w sekcji `<head>…</head>`.



Rysunek 6.3. *Przykładowa zawartość pliku style.css*

Przykładowy element `link` może wyglądać podobnie to tego z listingu 6.4.

Listing 6.4. *Deklaracja arkusza stylów w dokumencie internetowym (index.php)*

```
<head>
  <title>Obsługa bazy danych programu Microsoft SQL Server z poziomu skryptów
  PHP</title>
  <meta http-equiv="content-type" content="text/html; charset=iso-8859-2" />
  <link href="style.css" type="text/css" rel="stylesheet" />
</head>
```



> Reguła w kodzie CSS jest to kompletne określenie stylu składającego się z selektora oraz bloku deklaracji umieszczonych pomiędzy nawiasami klamrowymi, np. w postaci: `p {font-family: Verdana, sans-serif;}`.

Takie podejście powoduje, że plik ten jest podłączany w trakcie pierwszego uruchomienia skryptu PHP o nazwie *index.php*, w którym zadeklarowany został arkusz stylów CSS. Zmniejsza to nie tylko obciążenie samego serwera internetowego, ale i łącz, za pomocą których plik ten jest wywoływany.

Konfigurowanie obsługi bazy danych w PHP

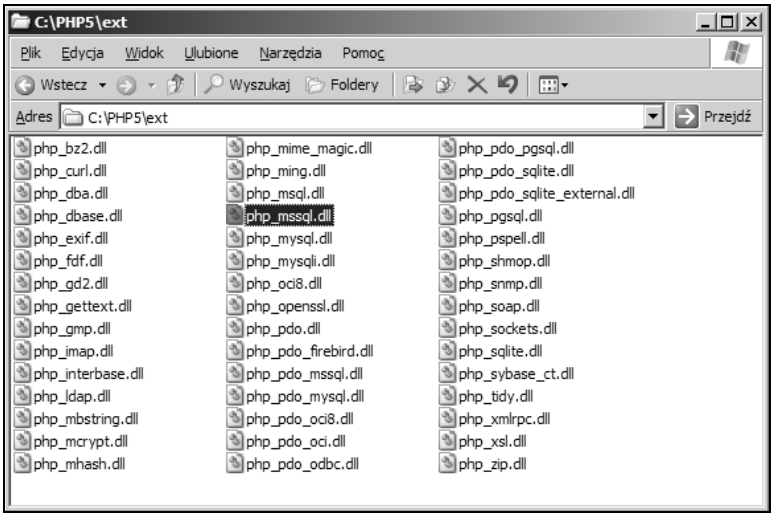
Gdy już poznaliśmy podstawowe elementy języków SQL, PHP czy XHTML, a także utworzyliśmy relacyjną bazę danych o nazwie *WirtualnySklep* wraz z poszczególnymi jej tabelami bazowymi w programie Microsoft SQL Server 2005, możemy przystąpić do skonfigurowania obsługi tejże bazy z poziomu skryptów PHP.

W niniejszym podrozdziale skupimy się głównie na klientach wirtualnego sklepu, tj. na obsłudze tabeli bazowej o nazwie *tKlienci*, która znajduje się w relacyjnej bazie danych o nazwie *WirtualnySklep*. To właśnie na jej przykładzie zostanie zaprezentowany dostęp do serwera bazy danych Microsoft SQL Server 2005 z poziomu skryptów PHP (za pomocą wbudowanych funkcji obsługujących tego typu bazę danych), do relacyjnej bazy danych *WirtualnySklep* oraz jej tabeli bazowej *tKlienci*, a także wszelkiego rodzaju operacje związane z dodawaniem, odczytywaniem, modyfikacją, usuwaniem czy wyszukiwaniem danych.

Aby w ogóle możliwa była jakakolwiek komunikacja pomiędzy bazą danych programu Microsoft SQL Server 2005 a interpreterem PHP, należy wykonać pewne wstępne kroki konfiguracyjne, które zostaną opisane w tym podrozdziale. W przypadku języka PHP w wersji 5.2.5 każdemu serwerowi baz danych odpowiada dynamiczna biblioteka, czyli zazwyczaj jakiś plik z rozszerzeniem *.dll*, który znajduje się w katalogu instalacyjnym PHP z rozszerzeniami, tj. *C:\PHP5\ext*. Plik ten zawiera zbiór określonych z góry funkcji, które są wykorzystywane w komunikacji z określoną relacyjną bazą danych.

Przykładowa zawartość katalogu rozszerzeń języka PHP w wersji 5.2.5 została przedstawiona na rysunku 6.4. Warto tu dodać, że w przypadku bazy danych programu Microsoft SQL Server 2005 bardzo ważne są dwa pliki: *php_mssql.dll* (znajdujący się w lokalizacji *C:\PHP5\ext*) oraz *ntwdblib.dll* (znajdujący się w lokalizacji *C:\PHP5*), który do działania wymaga wspomnianej biblioteki *php_mssql.dll*. W przypadku innych baz danych, np. Oracle Database, niezwykle ważny jest plik *php_oci8.dll*, dla MySQL AB MySQL — *php_mysql.dll*, z kolei dla bazy danych PostgreSQL Global Development Group PostgreSQL — *php_pgsql.dll*.

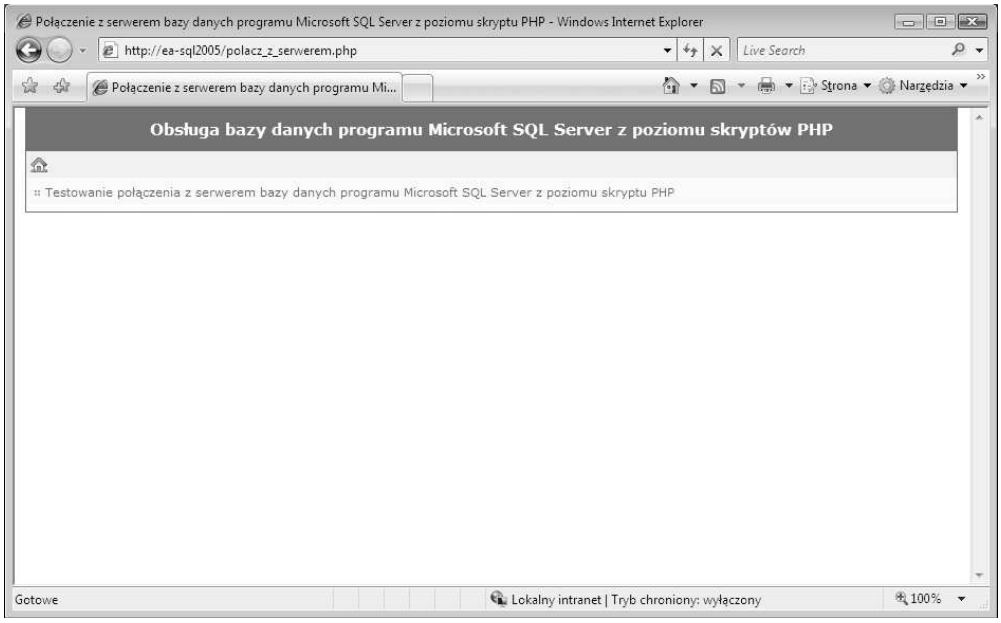
Po czystej instalacji bazy danych programu Microsoft SQL Server 2005 oraz dodatku Service Pack 2 dla tego programu, która została przedstawiona szczegółowo w poprzednim rozdziale, należy uruchomić na komputerze klienta *EA-K1* przeglądarkę internetową i wpisać jako adres URL *http://ea-sql2005*. Po wybraniu klawisza *Enter* zostanie wyświetlona testowa aplikacja internetowa *Obsługa bazy danych programu Microsoft SQL Server z poziomu skryptów PHP*, której okno główne zostało przedstawione wcześniej, tj. na rysunku 6.2. Po kliknięciu odnośnika *polacz_z_serwerem.php*, za pomocą którego można przetestować połączenie z serwerem bazy danych o nazwie *EA-SQL2005* z poziomu skryptu PHP, pojawi się „pusta strona internetowa” — zgodnie z tym, co zostało pokazane na rysunku 6.5. Termin „pusta strona internetowa” należy tutaj rozumieć jako „strona bez przetworzonego kodu PHP”, tj. kodu znajdującego się pomiędzy znacznikami języka PHP `<?php //… ?>`. Czyli inaczej mówiąc, bez kodu



Rysunek 6.4. *Zawartość katalogu C:\PHP5\ext z plikami do obsługi różnych typów relacyjnych baz danych*

pokazanego na listingu 6.5, który pochodzi z pliku *polacz_z_serwerem.php*. Plik ten, jak już wiemy, znajduje się w katalogu głównym serwera internetowego *EA-SQL2005*, tj. *C:\Inetpub\wwwroot*. Wyświetlany jest jedynie statyczny kod strony internetowej, tj. kod HTML. Dlaczego tak się dzieje? Otóż dzieje się tak z kilku przedstawionych poniżej powodów.

- Brak w katalogu *%SystemRoot%\system32* serwera członkowskiego *EA-SQL2005* (na którym działa interpreter PHP 5.2.5 i usługi IIS 6.0) bardzo ważnego pliku o nazwie *ntwdblib.dll*, który dostarczany jest wraz z interpreterem PHP 5.2.5, czyli z repozytorium o nazwie *php-5.2.5-Win32.zip*. W przypadku programu Microsoft SQL Server 2000 plik ten jest domyślnie instalowany w powyższym katalogu.
- Wyłączone jest rozszerzenie *php_mssql.dll* w sekcji *Windows Extensions* pliku konfiguracyjnego *%SystemRoot%\php.ini*, które po domyślnej instalacji interpretera PHP 5.2.5 nie jest aktywne (znak średnika w linii `;extension=` ↪`php_mssql.dll`, który oznacza komentarz). Jak pamiętamy, polecenia ze znakiem komentarza nie są wczytywane do pamięci operacyjnej serwera członkowskiego podczas uruchamiania interpretera PHP.
- Nie została poprawnie zdefiniowana ścieżka w dyrektywie `extension_dir = "C:\PHP5\ext"` pliku konfiguracyjnego *%SystemRoot%\php.ini*, która po domyślnej instalacji interpretera PHP 5.2.5 ma postać `extension_dir = "./"`. Powoduje to, że nie może być wczytany plik *php_mssql.dll*, który znajduje się w katalogu rozszerzeń PHP.
- Dodatkowo — w przypadku gdy PHP wykorzystuje rozszerzenie CGI (plik wykonywalny *php-cgi.exe* zamiast *php5isapi.dll*) — nie została włączona (odpowiednio skonfigurowana) w pliku konfiguracyjnym *%SystemRoot%\php.ini* linijka tekstu `;cgi.force_redirect = 1`. Należy ją włączyć, usuwając znak średnika, który, jak wiemy, jest komentarzem, i ustawić na wartość `0`. Poprawna linijka kodu powinna wyglądać tak: `cgi.force_redirect = 0`.



Rysunek 6.5. *Negatywny wynik testowania połączenia z serwerem bazy danych programu Microsoft SQL Server 2005 z poziomu skryptu polacz_z_serwerem.php*

Listing 6.5. *Kod PHP realizujący połączenie z serwerem bazy danych programu Microsoft SQL Server 2005 (polacz_z_serwerem.php)*

```
<?php
  require_once('funkcje.php');
  polacz_z_serwerem_bd();
?>
```

Jak widać na powyższym listingu, wykorzystana została instrukcja `require_once()` pozwalająca na załadowanie pliku *funkcje.php* do skryptu PHP i wywołana funkcja `polacz_z_serwerem_bd()`, która znajduje się w tym pliku. Funkcja ta zawiera parametry potrzebne do zestawienia połączenia z serwerem bazy danych Microsoft SQL Server 2005, który działa na serwerze członkowskim o nazwie *EA-SQL2005*, oraz obsługi błędów, jakie mogą wystąpić podczas zestawiania tego połączenia.

Biblioteka dynamiczna *ntwdblib.dll*, która do działania wymaga innej biblioteki — *php_mssql.dll*, znajduje się w katalogu instalacji interpretera PHP, czyli *C:\PHP5* (po rozpakowaniu repozytorium *php-5.2.5-Win32.zip*), oraz jest wykorzystywana podczas dostępu do relacyjnej bazy danych programu Microsoft SQL Server 2005 z poziomu skryptów PHP. Plik ten wykorzystywany jest w komunikacji z serwerem bazy danych Microsoft SQL Server 2005.

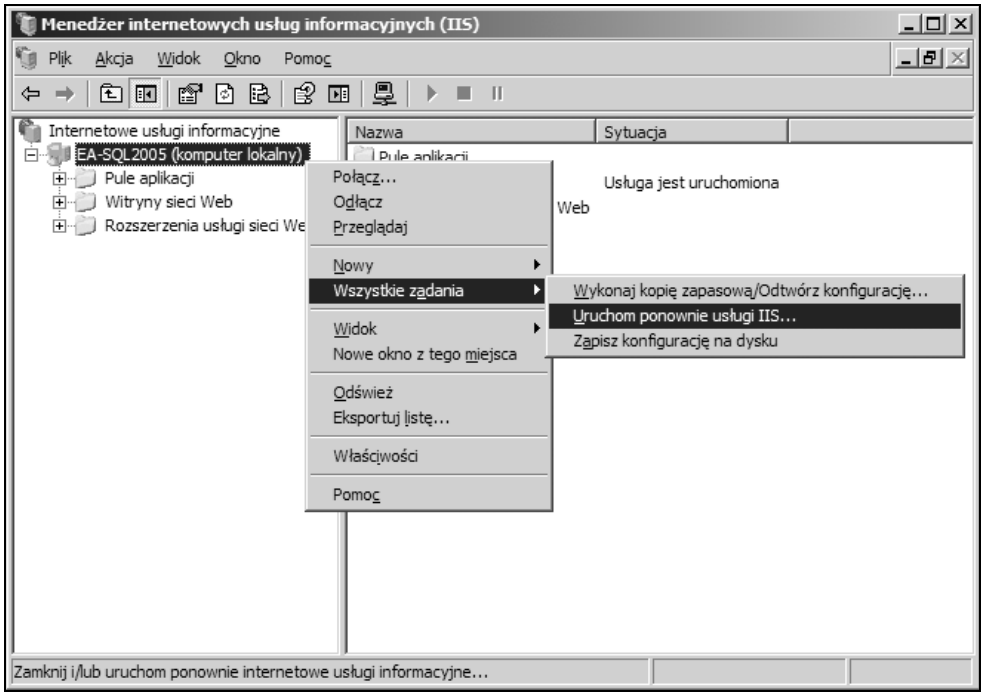


Ze względu na szybkość wykonywania skryptów PHP najlepiej jest, kiedy zarówno PHP, jak i Microsoft SQL Server 2005 działają na tym samym komputerze.

Aby istniała możliwość odczytywania plików z rozszerzeniami *.php* (skryptów PHP) i wyświetlania wyników ich działania z poziomu przeglądarki internetowej klienta *EA-K1*, należy wykonać poniższe kroki na serwerze członkowskim *EA-SQL2005*, na którym zostały zainstalowane:

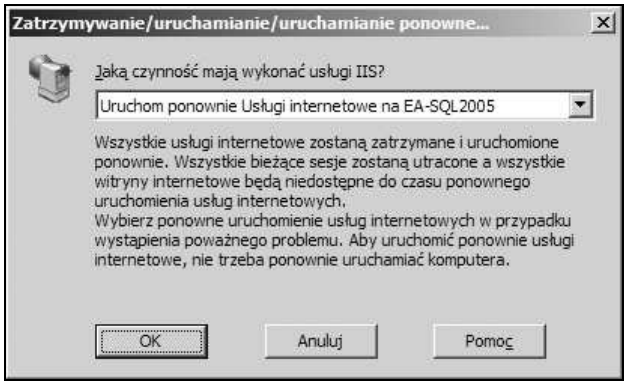
- interpreter skryptów PHP w wersji 5.2.5,
- serwer internetowy IIS w wersji 6.0,
- serwer bazy danych Microsoft SQL Server 2005.

1. Skopiować plik *ntwdblib.dll* z katalogu *C:\PHP5* do katalogu *%SystemRoot%\system32*. Można to wykonać bardzo szybko np. za pomocą komendy wydanej z okna *Wiersz polecenia*: `copy C:\PHP5\ntwdblib.dll %SystemRoot%\system32`.
2. Zatrzymać i ponownie uruchomić usługi IIS 6.0 na serwerze członkowskim *EA-SQL2005* — zgodnie z tym, co przedstawia rysunek 6.6 — czyli z poziomu konsoli Menedżer internetowych usług informacyjnych (IIS), którą można uruchomić za pomocą komendy `inetmgr` (wydanej z poziomu okna dialogowego *Uruchamianie*) lub `start inetmgr` (wydanej z poziomu okna *Wiersz polecenia*).



Rysunek 6.6. *Ponowne uruchamianie usługi IIS na serwerze członkowskim EA-SQL2005*

3. W oknie dialogowym o nazwie *Zatrzymywanie/uruchamianie/uruchamianie ponowne...*, które zostało przedstawione na rysunku 6.7, należy sprawdzić, czy wybrana jest czynność *Uruchom ponownie Usługi internetowe na EA-SQL2005*, i kliknąć przycisk *OK*.



Rysunek 6.7. *Okno dialogowe o nazwie Zatrzymywanie/uruchamianie/uruchamianie ponowne...*



Zatrzymanie i ponowne uruchomienie usług IIS 6.0 na serwerze członkowskim *EA-SQL2005* można szybko wykonać za pomocą następującej komendy wiersza poleceń: `iisreset EA-SQL2005`.

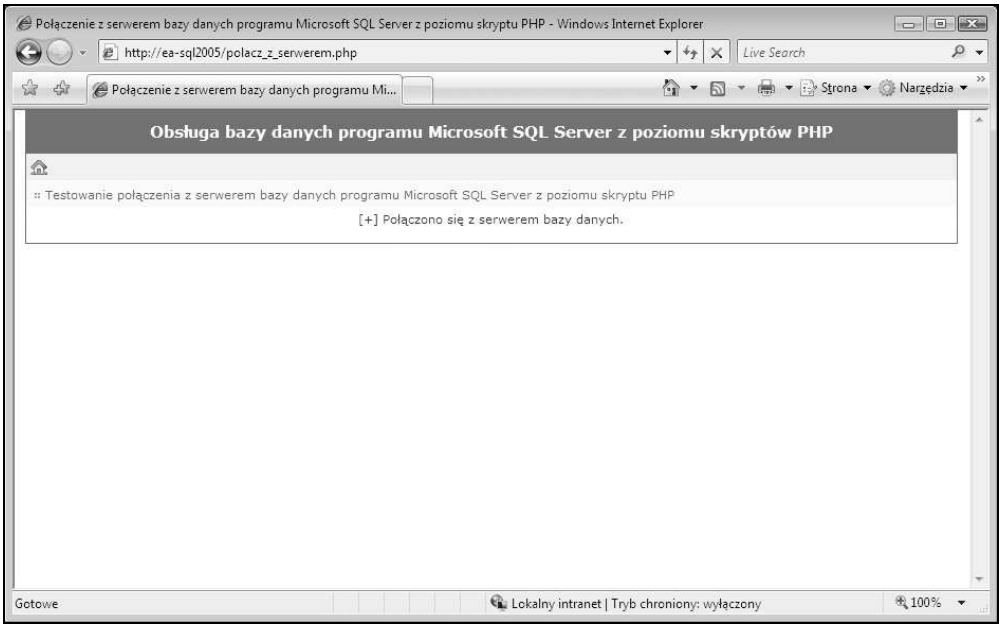
4. Sprawdzić w skryptach PHP, czy zostały zdefiniowane poprawnie takie zmienne, jak: nazwa serwera bazy danych Microsoft SQL Server 2005 (`EA-SQL2005`), nazwa użytkownika (`sa`) i hasło (`27aB.#cd75`) — zgodnie z tym, co przedstawia listing 6.6. Dodatkowo należy też sprawdzić zmienną `$bazaDanych`, która powinna mieć przypisaną nazwę `WirtualnySklep`, czyli `$bazaDanych = 'WirtualnySklep'`. Zmienna ta nie jest potrzebna w przypadku nawiązywania połączenia z samym serwerem bazy danych Microsoft SQL Server 2005. Jest za to wręcz niezbędna w pozostałych skryptach PHP, które np. operują na danych zgromadzonych w tabeli bazowej *tKlienci*.

Listing 6.6. *Zmienne wymagane do połączenia z serwerem bazy danych Microsoft SQL Server 2005 i bazą danych WirtualnySklep*

```
<?php
  $nazwaSerwera = 'EA-SQL2005';
  $bazaDanych = 'WirtualnySklep';
  $nazwaUzytkownika = 'sa';
  $hasloUzytkownika = '27aB.#cd75';

  /*
    blok kodu skryptu PHP
  */
?>
```

Po wykonaniu powyższych kroków próba połączenia za pomocą przeglądarki internetowej klienta *EA-K1* z serwerem bazy danych Microsoft SQL Server 2005 z poziomu skryptu PHP (za pomocą adresu *http://ea-sql2005/polacz_z_serwerem.php*) powinna się powieść, co przedstawia rysunek 6.8, czyli komunikat o treści *[+] Połączono się z serwerem bazy danych*. Jeżeli tak się stanie, to możemy założyć, że odtąd możliwa jest komunikacja nie tylko pomiędzy relacyjna bazą danych programu Microsoft SQL Server 2005 a interpreterem PHP 5.2.5, ale także pomiędzy serwerem IIS 6.0



Rysunek 6.8. *Pozytywny wynik testowania połączenia z serwerem bazy danych programu Microsoft SQL Server 2005 z poziomu skryptu polacz_z_serwerem.php*

i przeglądarką internetową klienta *EA-K1*. Inaczej mówiąc, możliwa jest obsługa bazy danych programu Microsoft SQL Server 2005 z poziomu aplikacji internetowej *Obsługa bazy danych programu Microsoft SQL Server z poziomu skryptów PHP*, która znajduje się na serwerze internetowym IIS 6.0 (w katalogu głównym tego serwera).

W razie problemów z podłączeniem się do serwera bazy danych Microsoft SQL Server 2005 z poziomu skryptu PHP o nazwie *polacz_z_serwerem.php* należy sprawdzić ustawienia w pliku konfiguracyjnym w katalogu *%SystemRoot%\php.ini*. Najważniejsze ustawienia zostały zaprezentowane w podrozdziale „Instalowanie i konfigurowanie PHP w systemie Windows" niniejszej książki.

Obsługa bazy danych z poziomu skryptów PHP

Obsługę relacyjnej bazy danych o nazwie *WirtualnySklep* znajdującej się na serwerze bazy danych Microsoft SQL Server 2005 o nazwie *EA-SQL2005* (z poziomu skryptu PHP, np. *select.php*, który jest dostępny pod adresem *ftp://ftp.helion.pl/przyklady/*
↪phiisq.zip w folderze *Skrypty PHP*) można podzielić na pięć poniżej przedstawionych kroków.

Krok 1. Nawiązanie połączenia z serwerem bazy danych programu Microsoft SQL Server 2005 o nazwie *EA-SQL2005*.

Przykładowy kod PHP został przedstawiony na listingu 6.7. Istotne elementy kodu PHP zostały zaznaczone tekstem pogrubionym.

Listing 6.7. *Nawiązanie połączenia z serwerem bazy danych programu Microsoft SQL Server 2005 o nazwie EA-SQL2005*

```
<?php
  $nazwaSerwera = 'EA-SQL2005';
  $bazaDanych = 'WirtualnySklep';
  $nazwaUzytkownika = 'sa';
  $hasloUzytkownika = '27aB.#cd75';

  $polacz_z_serwerem = @mssql_connect($nazwaSerwera,
                                      $nazwaUzytkownika,
                                      $hasloUzytkownika)
                                      or die("[-] Nie można połączyć się z serwerem
                                      bazy danych!<br /><br />");

  /*
    kod skryptu PHP
  */
?>
```

Krok 2. Nawiązanie połączenia z relacyjną bazą danych o nazwie *WirtualnySklep*, która znajduje się na serwerze bazy danych programu Microsoft SQL Server 2005 o nazwie *EA-SQL2005*.

Przykładowy kod PHP został przedstawiony na listingu 6.8. Jego istotne elementy zostały zaznaczone tekstem pogrubionym.

Listing 6.8. *Nawiązanie połączenia z relacyjną bazą danych o nazwie WirtualnySklep*

```
<?php
  $nazwaSerwera = 'EA-SQL2005';
  $bazaDanych = 'WirtualnySklep';
  $nazwaUzytkownika = 'sa';
  $hasloUzytkownika = '27aB.#cd75';

  $polacz_z_serwerem = @mssql_connect($nazwaSerwera,
                                      $nazwaUzytkownika,
                                      $hasloUzytkownika)
                                      or die("[-] Nie można połączyć się z serwerem
                                      bazy danych!<br /><br />");

  $polacz_z_baza = @mssql_select_db($bazaDanych,
                                    $polacz_z_serwerem)
                                    or die("[-] Nie można połączyć się z bazą
                                    danych!<br /><br />");

  /*
    kod skryptu PHP
  */
?>
```

Krok 3. Wysłanie zapytania SQL do tabeli bazowej *tKlienci* relacyjnej bazy danych *WirtualnySklep* programu Microsoft SQL Server 2005 o nazwie *EA-SQL2005*.

Przykładowy kod został przedstawiony na listingu 6.9. Jego istotne elementy zostały zaznaczone tekstem pogrubionym.

Listing 6.9. *Wysłanie zapytania SQL do tabeli bazowej tKlienci*

```
<?php
  /*
    kod skryptu PHP
  */

    $zapytanie = "SELECT *
                  FROM tKlienci";

  /*
    kod skryptu PHP
  */
?>
```

Krok 4. Pobranie i wykorzystanie otrzymanych wyników (np. wyświetlenie zawartości tabeli bazowej *tKlienci* relacyjnej bazy danych *WirtualnySklep*) w postaci statycznej tabeli HTML.

Przykładowy kod został pokazany na listingu 6.10. Jego istotne elementy zostały zaznaczone tekstem pogrubionym.

Listing 6.10. *Pobranie i wykorzystanie otrzymanych wyników w postaci statycznej tabeli HTML*

```
<?php
  /*
    kod skryptu PHP
  */

    $wynik = @mssql_query($zapytanie);

    if (!$wynik)
      {
        echo "[-] Wykonanie zapytania nie powiodło się!<br /><br />";
        exit;
      }
      else
        {
          echo "<table cellspacing='3' cellpadding='5'>";
          echo "<tr bgcolor='#dddddd' align='left'>";
          echo "<th>Nr ID</th>
                <th>Nazwisko</th>
                <th>Imię</th>
                <th>Ulica</th>
                <th>Kod poczt.</th>
                <th>Miejscowość</th>
                <th>Województwo</th>
                <th>Telefon</th>
                <th>Email</th>";
```

```
            echo "</tr>";
            while ($wiersz = @mssql_fetch_array($wynik))
              {
                echo "<tr align='left'>";
                echo "<td>".$wiersz['KlientID']."</td>";
                echo "<td>".$wiersz['Nazwisko']."</td>";
                echo "<td>".$wiersz["Imie"]."</td>";
                echo "<td>".$wiersz["Ulica"]."</td>";
                echo "<td>".$wiersz["KodPocztowy"]."</td>";
                echo "<td>".$wiersz["Miejscowosc"]."</td>";
                echo "<td>".$wiersz["Wojewodztwo"]."</td>";
                echo "<td>".$wiersz["Telefon"]."</td>";
                echo "<td>".$wiersz["Email"]."</td>";
                echo "</tr>";
              }
            echo "</table>";
          }

    /*
      kod skryptu PHP
    */
  ?>
```

Krok 5. Zwolnienie zajętych zasobów serwera bazy danych Microsoft SQL Server 2005 o nazwie *EA-SQL2005* (głównie jego pamięci operacyjnej RAM) przez zmienne `$wynik` i `$polacz_z_serwerem` oraz zamknięcie połączenia z tym serwerem bazy danych.

Przykładowy kod PHP został przedstawiony na listingu 6.11. Jego istotne elementy zostały zaznaczone tekstem pogrubionym.

Listing 6.11. *Zwolnienie zajętych zasobów serwera bazy danych Microsoft SQL Server 2005 o nazwie EA-SQL2005 i zamknięcie połączenia z tym serwerem*

```
<?php
  /*
    blok kodu skryptu PHP
  */

    @mssql_free_result($wynik);
    @mssql_close($polacz_z_serwerem);

  /*
    blok kodu skryptu PHP
  */
?>
```

Reasumując, spróbujmy określić, jakie czynności wykonuje skrypt PHP o nazwie *select.php*. W skrócie można przedstawić je w następujących krokach.

Krok 1. Na samym początku zestawiane jest połączenie z serwerem bazy danych Microsoft SQL Server 2005 o nazwie *EA-SQL2005*, na którym znajduje się relacyjna baza danych o nazwie *WirtualnySklep* — przy użyciu funkcji wewnętrznej PHP `mssql_connect()`, której podstawowymi parametrami są:

- nazwa serwera bazy danych Microsoft SQL Server 2005 (`EA-SQL2005`),
- nazwa użytkownika (`sa`), którego konto ma przyznane odpowiednie uprawnienia do relacyjnej bazy danych *WirtualnySklep*,
- hasło użytkownika (`27aB.#cd75`).

Krok 2. Kiedy połączenie z serwerem bazy danych Microsoft SQL Server 2005 o nazwie *EA-SQL2005* zostanie pomyślnie nawiązane, wynikiem działania funkcji o nazwie `mssql_connect()` będzie identyfikator połączenia (np. `Resource id #1`), który jest przechowywany w zmiennej o nazwie `$polacz_z_serwerem`. W przypadku jakichkolwiek problemów z połączeniem z serwerem bazy danych zostanie wyświetlony (przy wykorzystaniu funkcji o nazwie `die()`) komunikat błędu o następującej treści: *[-] Nie można połączyć się z serwerem bazy danych!*.

Krok 3. Po poprawnym zestawieniu połączenia z serwerem bazy danych Microsoft SQL Server 2005 o nazwie *EA-SQL2005* kolejnym krokiem jest wybranie (za pomocą funkcji o nazwie `mssql_select_db()`) relacyjnej bazy danych *WirtualnySklep*, która znajduje się na tym serwerze.

Krok 4. Jeżeli nie wystąpi żaden krytyczny błąd (identyfikator połączenia wynosi `1`), kolejnym krokiem będzie utworzenie zmiennej `$zapytanie`, która będzie przechowywała instrukcję SQL, tj. zdefiniowane zapytanie języka SQL.

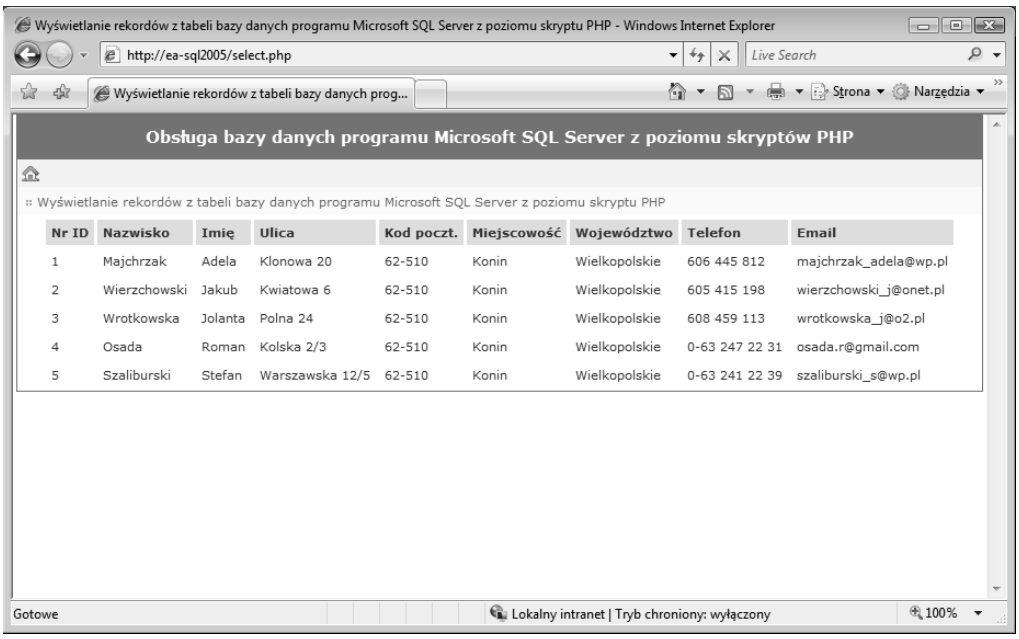
Krok 5. W następnym kroku tworzona jest zmienna `$wynik`, która przechowuje identyfikator wyniku zapytania przeprowadzonego przez funkcję o nazwie `mssql_query()`. Kiedy zapytanie zostanie poprawnie wykonane (nie pojawi się komunikat o treści *[-] Wykonanie zapytania nie powiodło się!*), wtedy wydzielane są wiersze wyniku za pomocą funkcji PHP o nazwie `mssql_fetch_array()`, które są przechowywane w zmiennej `$wynik`. Wcześniej jednak jest tworzona statyczna tabela HTML. Pętla `while` tworzy tablicę o nazwie `$wiersz` dla każdego rekordu wynikowego. Wiersze są podstawiane do tablicy dopóty, dopóki wartością całego wyrażenia nie będzie wartość `False`. Ma to miejsce wówczas, gdy nie ma już więcej wierszy do przetworzenia. Wtedy pętla `while` kończy swoje działanie.

Krok 6. Na końcu skryptu PHP następuje zwolnienie wykorzystywanych do przeprowadzenia zapytania SQL zasobów systemowych (głównie pamięci operacyjnej RAM serwera członkowskiego o nazwie *EA-SQL2005*) — przy wykorzystaniu funkcji wbudowanej PHP o nazwie `mssql_free_result()`, oraz zamknięcie połączenia z serwerem bazy danych Microsoft SQL Server 2005. Odpowiedzialna jest za to funkcja wewnętrzna języka PHP o nazwie `mssql_close()`.



Użycie funkcji wewnętrznej PHP o nazwie `mssql_close()` (na końcu skryptu PHP) nie jest wymagane, gdyż połączenia nietrwałe (ang. *nonpersistent*) są automatycznie zamykane po zakończeniu działania tego skryptu. Niemniej jednak dobrym nawykiem programistycznym jest, aby zwalniać zajmowane przez zmienne zasoby oraz zamykać otwarte połączenia z relacyjną bazą danych.

Wynik skryptu PHP *select.php*, który został wywołany z poziomu przeglądarki internetowej komputera klienta *EA-K1* pracującego pod kontrolą systemu operacyjnego Microsoft Windows Vista Business, został pokazany na rysunku 6.9. Jak widać na tym rysunku, w wyniku żądania o treści *http://ea-sql2005/select.php* zwróconych zostało pięć przykładowych rekordów z relacyjnej bazy danych *WirtualnySklep*, która znajduje się na serwerze członkowskim o nazwie *EA-SQL2005*.



Rysunek 6.9. *Wynik wykonania skryptu PHP o nazwie select.php*

Podstawowe funkcje PHP obsługujące bazę danych Microsoft SQL Server

Jak już wspomniano na początku niniejszego rozdziału, interpreter języka skryptowego PHP w wersji 5.2.5 posiada wbudowaną obsługę różnych typów baz danych (w tym bazy danych programu Microsoft SQL Server 2005) za pomocą wewnętrznych funkcji. Oznacza to tyle, że wiele popularnych typów serwerów baz danych jest obsługiwanych z poziomu skryptów PHP.

W niniejszej książce zostaną opisane wybrane funkcje języka PHP, za pomocą których możliwy jest dostęp z poziomu kodu PHP do bazy danych programu Microsoft SQL Server 2005.