

## 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

## Dreamweaver UltraDev 4. Vademecum profesjonalisty

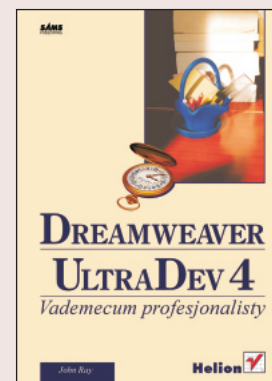
Autor: John Ray

Tłumaczenie: Szymon Drejewicz, Iwo Nowosielski,  
Jan Ostrowski, Tomasz Wrona

ISBN: 83-7197-431-0

Tytuł oryginału: [TY Dreamweaver UltraDev 4 in 21  
days](#)

Format: B5, stron: 650



Firma Macromedia jeszcze raz wywiązała się z zadania stworzenia środowiska do projektowania aplikacji sieciowych, które łączą siłę dynamicznych witryn z istniejącym interfejsem Dreamweavera. Program Dreamweaver UltraDev jest jedynym w swoim rodzaju środowiskiem projektowania wykorzystującym najnowsze pomysły i metody.

UltraDev udostępnia nie jedną, ale trzy różne technologie serwera (JSP, CFML, ASP), z możliwością rozszerzenia środowiska pracy o nowe technologie poprzez pluginy. Ponadto firma Macromedia stworzyła wersje programu dla Windows i dla Macintosha, umożliwiając użytkownikom komputerów Macintosh – zmuszonych wcześniej do pisania kodu na komputerach Windows – powrót do znajomych pulpitów.

Pakiet UltraDev może tworzyć witryny internetowe w pełni sterowane przez bazy danych poprzez interfejs „wskaż i kliknij” (point and click). W odróżnieniu od innych pakietów, oferuje on projektantowi możliwość podglądu danych w czasie rzeczywistym wewnątrz interfejsu UltraDeva. Program UltraDev czyni życie znacznie łatwiejszym, umożliwiając osobie projektującej mechanizmy aplikacji pracę tylko z bazą danych, podczas kiedy projektanci pracują nad układem strony. Skończyły się czasy ustawicznego przesyłania plików.

Po przeczytaniu tej książki będziesz umiał wykorzystać wszystkie możliwości programu UltraDev. Niezależnie od tego, czy kiedykolwiek pracowałeś z Dreamweaverem lub edytorem HTML, już wkrótce będziesz tworzył swoje własne witryny internetowe sterowane przez bazy danych. Jedynym warunkiem zrozumienia wiadomości zawartych w tej książce jest podstawowa znajomość HTML-a i dostęp do komputera pracującego pod kontrolą Windows lub MacOS.

Po zakończeniu lektury książki będziesz umiał szybko i z łatwością tworzyć różnorodne aplikacje sieciowe w trzech różnych technologiach serwerów. Staniesz się jednym z władców świata dynamicznych aplikacji.



# Spis treści

	Informacja o Autorze .....	15
	Wprowadzenie do programu Macromedia Dreamweaver UltraDev ....	17
<b>Część I</b>	<b>Podstawowe wiadomości o programie UltraDev i tworzeniu dynamicznych witryn .....</b>	<b>21</b>
<b>Rozdział 1.</b>	<b>Interfejs programu UltraDev .....</b>	<b>23</b>
	Interfejs programu UltraDev.....	23
	Struktura menu.....	24
	Okno dokumentu.....	26
	Edycja obiektów .....	27
	Rozmiar strony .....	27
	Wielkość strony a czas pobierania.....	28
	Konfigurowanie okna dokumentu .....	28
	Palety narzędziowe .....	30
	Obiekty HTML .....	30
	Paleta Properties (Properties Palette) .....	40
	Pozostałe okna i palety .....	41
	Bezpośredni dostęp do kodu HTML.....	44
	Zaznaczanie elementów w różnych widokach .....	45
	Opcje wyświetlania kodu.....	45
	Sprawdzanie informacji o znacznikach HTML.....	46
	Przeszukiwanie dokumentów .....	46
	Uruchamianie zewnętrznego edytora HTML .....	49
	Dostosowywanie ustawień programu UltraDev .....	50
	Podsumowanie .....	52
	Pytania i odpowiedzi.....	52
<b>Rozdział 2.</b>	<b>Praca z podstawowymi narzędziami programu UltraDev .....</b>	<b>53</b>
	Tworzenie pierwszej witryny.....	53
	Definiowanie witryny .....	54
	Uwagi o projektowaniu.....	56
	Do kogo kierujesz witrynę? .....	56
	Projektuj z myślą o różnych platformach .....	56
	Nie rób bałaganu! .....	57
	Podstawowe elementy strony.....	57
	Tworzenie łączy .....	58
	Elementy nagłówka strony — sekcja <HEAD> .....	59
	Podstawowe atrybuty strony.....	60
	Podgląd projektu strony.....	61
	Sprawdzanie pisowni .....	62

Tabele.....	62
Wstawianie tabeli .....	63
Układ tekstów i grafiki .....	66
Tryb „Table Layout” .....	70
Paski i przyciski nawigacyjne.....	71
Lista łączy.....	72
Aktywne obrazki i mapy aktywnych obszarów.....	72
Rollovery .....	74
Paski nawigacji.....	76
Warstwy .....	78
Precyzyjne pozycjonowanie elementów z wykorzystaniem warstw .....	78
Przekształcanie warstw w tabelę i odwrotnie .....	81
Ramki.....	82
Układ ramek .....	83
Paleta Frames.....	84
Znacznik <NOFRAMES>.....	84
Skomplikowane układy ramek .....	85
Ramki — argumenty za i przeciw .....	85
Formularze .....	86
Testowanie kodu HTML.....	86
Raporty.....	87
Podsumowanie .....	88
Pytania i odpowiedzi.....	88
<b>Rozdział 3. Planowanie oraz zarządzanie witryną WWW.....</b>	<b>89</b>
Planowanie witryny .....	89
Mapa witryny.....	90
Konwencje nazewnictwa .....	91
Mapa witryny a UltraDev .....	93
Zastosowanie mapy witryny w pracy nad istniejącymi witrynami WWW .....	93
Zastosowanie mapy witryny do tworzenia nowych stron WWW .....	96
Wprowadzanie istniejącej witryny WWW do środowiska UltraDev .....	98
Sposoby uzyskiwania dostępu do danych .....	98
Sposoby realizacji dostępu do maszyny zdalnej.....	99
Korzystanie z połączenia .....	104
Zarządzanie witryną.....	106
Weryfikacja łączy .....	106
Rozproszone środowisko edycyjne.....	108
Konfiguracja systemu Check In/Out .....	108
Użytkowanie systemu Check In/Out .....	109
Notatki projektu.....	110
Podsumowanie .....	111
Pytania i odpowiedzi.....	112
<b>Rozdział 4. Dynamiczne witryny WWW — behawiory.....</b>	<b>113</b>
Czym jest DHTML? .....	113
Kaskadowe arkusze stylów.....	114
JavaScript.....	114
Problemy związane z DHTML-em.....	114
DHTML — stosować czy nie? .....	115
Podstawowe wiadomości o behawiorach.....	116
Paleta Behaviors .....	116
Tworzenie behawiorów .....	117
Wykorzystanie behawiorów do tworzenia podpowiedzi wyświetlanych na pasku statusu przeglądarki .....	121

Wykorzystanie behawiorów do tworzenia odpowiedzi wyświetlanych bezpośrednio na stronie WWW .....	123
Wykorzystanie behawiorów do otwierania nowego okna przeglądarki .....	126
Technika „przeciągnij i upuść” na stronach WWW? .....	128
Tworzenie animacji za pomocą narzędzia Timelines .....	133
Paleta Timelines .....	133
Klatki kluczowe animacji .....	135
Ścieżki ruchu .....	136
Inne polecenia palety Timelines .....	138
Flash? Ależ nic prostszego! .....	138
Przyciski Flash .....	139
Tekst Flash .....	140
Inne techniki „ożywiania” tworzonych w UltraDev dokumentów WWW .....	141
Narzędzie programisty JavaScript .....	142
Podsumowanie .....	144
Pytania i odpowiedzi .....	145
<b>Rozdział 5. Zwiększanie wydajności pracy z programem UltraDev .....</b>	<b>147</b>
Wielokrotne wykorzystanie kodu .....	147
Formatowanie hurtowe, czyli kaskadowe arkusze stylów .....	148
Wspólne elementy witryny .....	150
Szablony .....	151
Style HTML .....	151
Kaskadowe arkusze stylów .....	153
Praca z kaskadowymi arkuszami stylów .....	154
Biblioteki .....	161
Tworzenie elementów biblioteki z wykorzystaniem istniejącego kodu HTML .....	162
Paleta Assets — widok Library .....	162
Tworzenie elementów biblioteki od podstaw .....	163
Szablony .....	164
Tworzenie szablonów z wykorzystaniem istniejących stron HTML .....	165
Paleta Assets — widok Templates .....	166
Tworzenie szablonów od podstaw .....	167
Korzystanie z szablonów .....	167
Paleta Assets .....	168
Listy Site i Favorites .....	169
Rozszerzanie możliwości środowiska UltraDev .....	169
Rejestrowanie poleceń .....	169
Rejestrowanie poleceń do użytku tymczasowego .....	170
Pakiety rozszerzeń programu UltraDev .....	171
Podsumowanie .....	172
Pytania i odpowiedzi .....	172
<b>Rozdział 6. Podstawy dynamicznych aplikacji WWW oraz projektowania baz danych .....</b>	<b>175</b>
Dynamiczne witryny WWW .....	175
CGI .....	176
Języki osadzone w HTML-u .....	177
Działanie dynamicznych witryn WWW .....	178
Przekazywanie zmiennych .....	178
Cookies .....	181
Tworzenie dynamicznych witryn WWW .....	183
Typowe rodzaje witryn WWW realizowanych jako witryny dynamiczne .....	183
Projektowanie dynamicznych witryn WWW .....	184

Projektowanie baz danych .....	185
Słownictwo związane z modelowaniem struktur baz danych .....	186
Normalizacja w praktyce .....	188
Od modelu danych do bazy danych.....	194
Podsumowanie .....	198
Pytania i odpowiedzi.....	199
<b>Rozdział 7. Podstawy języków skryptowych wykorzystywanych w UltraDev ....</b>	<b>201</b>
Platformy skryptowe obsługiwane przez UltraDev .....	201
Active Server Pages.....	202
Java Server Pages .....	203
ColdFusion.....	205
Platforma systemowa dla serwera WWW .....	206
Technologia przyjazna dla użytkownika .....	207
Liczba i rodzaj oferowanych usług.....	207
Stabilność.....	208
Współczynnik wydajność — koszt.....	208
Pomoc techniczna .....	209
Podjęmujemy ostateczną decyzję.....	209
Języki skryptowe obsługiwane przez UltraDev .....	210
Elementy podstawowe.....	210
Zmienne .....	211
Pętle i instrukcje warunkowe.....	214
Charakterystyczne elementy aplikacji WWW .....	216
Podsumowanie .....	218
Pytania i odpowiedzi.....	218
<b>Część II Ożywianie witryn internetowych .....</b>	<b>219</b>
<b>Rozdział 8. Konfiguracja systemów Mac OS i MS Windows jako otoczeń baz danych .....</b>	<b>221</b>
Połączenia z bazami danych .....	221
Konfiguracja połączeń z bazami danych .....	222
ODBC w MacOS? Naturalnie — połączenie JDBC-ODBC .....	231
Konfiguracja klienta RmiJdbc w systemie MacOS .....	232
Konfiguracja serwera RmiJdbc w systemie Windows NT .....	233
Instalacja i uruchamianie serwera MySQL.....	236
Instalacja i uruchamianie serwera MySQL w systemach MS Windows.....	237
Instalacja i uruchamianie serwera MySQL w systemach z rodziny Unix.....	238
Testowanie funkcjonowania serwera MySQL .....	239
Konfiguracja źródeł danych ODBC dla baz danych MySQL .....	240
Konfiguracja źródeł danych ODBC dla baz danych Microsoft Access.....	242
Podsumowanie .....	243
Pytania i odpowiedzi.....	244
<b>Rozdział 9. Podstawowe narzędzia programu UltraDev służące do tworzenia dynamicznych witryn WWW .....</b>	<b>245</b>
Bazy danych w sieci WWW .....	245
Połączenia .....	246
Behawiory wykonywane po stronie serwera .....	247
Podstawowe behawiory po stronie serwera.....	249
Różnice w funkcjonowaniu tradycyjnych baz danych i baz pracujących w sieci.....	255
Dodatkowe behawiory wykonywane po stronie serwera .....	256

Data Bindings .....	260
Dane pozyskiwane dynamicznie .....	260
Powiązania danych z elementami dokumentów .....	261
Uwierzytelnianie użytkowników .....	263
Log In User .....	264
Restrict Acces To Page .....	265
Log Out User .....	266
Check New Username .....	267
Tworzenie i modyfikowanie behawiorów serwera .....	267
Podsumowanie .....	270
Pytania i odpowiedzi .....	270
<b>Rozdział 10. Tworzenie stron generowanych na podstawie bazy danych .....</b>	<b>271</b>
Strony generowane na podstawie bazy danych .....	271
Prosta tabela .....	272
Tworzenie procesów drugoplanowych .....	272
Tworzenie aplikacji UltraDev .....	273
Dodawanie grafiki do zestawu rekordów .....	280
Procesy drugoplanowe bazy danych umożliwiające wykorzystanie grafiki .....	281
Dodawanie dynamicznego obrazka do projektu .....	282
Upiększanie strony .....	284
Dynamiczne atrybuty i złożone strony .....	285
Rozbudowane procesy drugoplanowe bazy danych .....	286
Dodatkowe funkcje stron dynamicznych .....	290
Sortowanie .....	290
Podsumowanie .....	291
Pytania i odpowiedzi .....	292
<b>Rozdział 11. Zapisywanie, edycja i uaktualnianie danych z poziomu aplikacji UltraDev .....</b>	<b>293</b>
Manipulacja danymi w sieci .....	293
Tworzenie formularzy .....	294
Tworzenie prostego formularza w języku HTML .....	295
Pola tekstowe .....	297
Przyciski Submit (Wyślij) .....	300
Dodawanie akcji Insert .....	305
Ekran szczegółowy i usuwanie rekordów .....	307
Ekran szczegółowy .....	307
Usuwanie rekordów .....	311
Łączenie z ekranem szczegółowym .....	312
Ekran aktualizacji .....	314
Tworzenie formularzy dynamicznych .....	316
Tworzenie formularzy dynamicznych za pomocą behawiorów serwera .....	316
Tworzenie elementów dynamicznego formularza za pomocą palety Properties .....	317
Formularze i ekrany generowane automatycznie .....	319
Obiekt Master-Detail .....	319
Obiekt Record Insert Form .....	321
Obiekt Record Update Form .....	322
Tworzenie ostatniego połączenia .....	325
Podsumowanie .....	326
Pytania i odpowiedzi .....	326

<b>Rozdział 12. Pisanie zaawansowanych zapytań bazy danych.....</b>	<b>327</b>
Po co nam SQL? .....	327
Obiekty baz danych.....	328
Tworzenie baz danych i tabel .....	329
Rodzaje danych.....	331
Dodawanie i zmienianie danych.....	332
Aktualizacja danych .....	333
Usuwanie danych.....	335
Zapytania tabel baz danych.....	336
Sortowanie informacji .....	337
Łączenie tabel.....	337
Narzędzia sumowania.....	340
Inne zaawansowane techniki.....	342
Widoki .....	342
Procedury składowane.....	343
Zapytania w zapytaniu.....	343
Zaawansowane zapytania a UltraDev.....	344
Podsumowanie .....	345
Pytania i odpowiedzi.....	346
<b>Rozdział 13. Składniki zaawansowane — katalogi, systemy wyszukiwania .....</b>	<b>347</b>
Katalog produktów .....	347
Kategorie produktu .....	349
Pozycje katalogu.....	349
Pozycje pokrewne.....	350
Kod SQL.....	351
Zapytania SQL.....	352
Projektowanie katalogu.....	356
Formularz wyszukiwania.....	356
Wyniki wyszukiwania .....	359
Automatyzacja paska nawigacyjnego .....	367
Status nawigacji.....	368
Paski nawigacji .....	368
Podsumowanie .....	369
Pytania i odpowiedzi.....	369
<b>Rozdział 14. Składniki zaawansowane — logowanie się użytkownika .....</b>	<b>371</b>
Systemy logowania .....	371
Uwierzytelnianie HTTP.....	371
Rejestrowanie poprzez sieć .....	372
SQL.....	373
Proste logowanie .....	375
Przetwarzanie informacji logowania .....	377
Śledzenie przebiegu sesji .....	380
Przekazywanie zmiennej .....	382
Ciąg dalszy przekazywania zmiennej.....	385
Przechodzenie do strony pokrewnej.....	386
Czas na cookies.....	387
Definiowanie zmiennej sesji.....	388
Używanie zmiennych sesji .....	390
Zmienne sesji w zapytaniach.....	390
System uwierzytelniania oferowany przez program UltraDev.....	391
Prawdziwe bezpieczeństwo witryn internetowych .....	391
Podsumowanie .....	392
Pytania i odpowiedzi.....	392

**Część III Tworzenie rozbudowanych aplikacji internetowych..... 395****Rozdział 15. Zaawansowane techniki — przekształcanie istniejących witryn internetowych w witryny dynamiczne..... 397**

Przejście od stron statycznych do dynamicznych.....	397
Losowe wybieranie banerów .....	399
Definicja tabeli .....	400
Przykładowe dane.....	400
Zapytanie SQL.....	401
Implementacja .....	402
Obrazki okresowe i czasowe.....	405
Definicja tabeli .....	405
Zapytanie SQL.....	406
Wiele obrazków okresowych i czasowych.....	408
Pozyskiwanie opinii od użytkowników .....	410
Definicja tabeli .....	411
Zapytanie SQL.....	411
Tworzenie dokumentu HTML .....	413
Podsumowanie .....	418
Pytania i odpowiedzi.....	418

**Rozdział 16. Zaawansowane techniki — zbieranie danych i tworzenie raportów ..... 419**

Wejście i wyjście — przegląd systemu zarządzania wiadomościami .....	419
Definiowanie mapy serwisu .....	420
Tworzenie kodu SQL.....	421
Budowanie głównego widoku wiadomości .....	424
Tworzenie listy wiadomości.....	425
Sortowanie listy wiadomości.....	427
Dodanie parametru do behawiora serwera Go To Related Page .....	429
Przeszukiwanie wiadomości.....	430
Czytanie wiadomości.....	431
Tworzenie zestawów rekordów .....	433
Tworzenie układu strony .....	434
Usuwanie wiadomości.....	435
Ulepszanie okna przeglądania wiadomości.....	436
Projektowanie strony służącej do pisania nowej wiadomości .....	438
Wstawianie ukrytych pól.....	439
Zachowywanie wiadomości .....	440
Podsumowanie .....	441
Pytania i odpowiedzi.....	441

**Rozdział 17. Techniki zaawansowane — pozyskiwanie i wykorzystanie danych użytkownika ..... 443**

Czy to wciąż ta sama witryna?.....	443
Struktura konfigurowalnej witryny WWW .....	445
Definiowanie tabel bazy danych.....	446
Użytkownicy.....	447
Preferencje .....	447
Właściwości i opcje.....	448
Poprawki w strukturze bazy danych.....	449
Już bardziej praktycznie... ..	450
Inicjalizacja właściwości i dostępnych dla nich opcji .....	451
Domyślne preferencje.....	454



System logowania i rejestracji .....	455
Strony związane z rejestracją .....	456
Strony związane z logowaniem .....	460
Zawartość witryny .....	463
Preferencje .....	469
Podsumowanie .....	473
Pytania i odpowiedzi .....	473
<b>Rozdział 18. Złożone systemy wyszukiwania .....</b>	<b>475</b>
Złożone metody wyszukiwania .....	475
Ograniczanie rezultatów wyszukiwania .....	480
Określanie wagi poszczególnych właściwości .....	483
Autostopowicz .....	484
Definiowanie tabel bazy danych .....	484
Tabela tblUzytkownicy .....	485
Tabela tblOpcje .....	486
Wprowadzanie do bazy danych przykładowych informacji .....	487
System logowania i rejestracji .....	489
Rejestracja i ustalenie domyślnego profilu .....	489
Logowanie .....	492
Profil użytkownika .....	493
Wyszukiwanie .....	495
Ograniczenie dostępu do stron witryny .....	498
Podsumowanie .....	500
Pytania i odpowiedzi .....	500
<b>Rozdział 19. Techniki zaawansowane — testy i edukacja online .....</b>	<b>501</b>
System ankiet i testów .....	501
Mapa witryny .....	501
Tworzenie zapytań SQL .....	502
Tworzenie strony rejestracji .....	507
Tworzenie formularza quizu .....	510
Pomijanie pytań .....	512
Dynamiczne odpowiedzi .....	512
Zapisywanie odpowiedzi .....	514
Zliczanie poprawnych odpowiedzi (opcjonalne) .....	517
Kończenie quizu .....	520
Zabezpieczanie quizu .....	523
Dostęp administracyjny .....	524
Tworzenie ekranu wyników .....	525
Posumowanie .....	527
Pytania i odpowiedzi .....	527
<b>Rozdział 20. Techniki zaawansowane — rozwiązania e-commerce .....</b>	<b>529</b>
Zawiła prostota .....	529
Zakupy bazujące na ID sesji .....	530
Konta użytkowników .....	531
Nasze rozwiązanie .....	531
Mapa witryny e-commerce .....	532
Definicje tabel .....	532
Kategorie produktów .....	533
Elementy katalogu .....	533
Elementy pokrewne .....	534
Informacja na temat użytkownika .....	534
Informacje na temat koszyka .....	535

System katalogu.....	538
Formularz wyszukiwania.....	538
Wyniki wyszukiwania.....	539
Strony szczegółowe.....	540
Elementy pokrewne.....	542
Czyszczenie strony.....	542
System uwierzytelniania użytkownika.....	543
Strony zgłoszenia.....	544
Strony logowania.....	545
Koszyk.....	547
System sprawdzania.....	549
Widok koszyka.....	549
Końcowe sprawdzanie.....	553
Strona zamykająca.....	556
Podsumowanie.....	557
Pytania i odpowiedzi.....	557
<b>Rozdział 21. Testowanie i wdrażanie aplikacji dynamicznych.....</b>	<b>559</b>
Problemy występujące podczas kodowania.....	559
Błędy dotyczące kodu HTML.....	560
Formularze.....	560
Warstwy.....	565
Problemy związane z behawiorami serwera.....	568
Pozostałości.....	569
Zaawansowane zapytania.....	569
Utrzymuj prostotę kodu.....	572
Testowanie.....	573
Sprawdzanie wartości NULL.....	574
Zatwierdzanie wykorzystania powtarzających się danych.....	575
Testowanie maksymalnych długości.....	575
Dodatkowe zasoby informacji dotyczących rozwiązywania problemów.....	576
Podsumowanie.....	577
Pytania i odpowiedzi.....	577
<b>Dodatki.....</b>	<b>579</b>
<b>Dodatek A Dostosowywanie środowiska UltraDev do własnych potrzeb.....</b>	<b>581</b>
Menu.....	581
Okna dialogowe i okienka.....	582
<b>Dodatek B Krótki przewodnik po funkcjach MySQL.....</b>	<b>587</b>
Najczęściej wykorzystywane typy danych.....	587
Operacje arytmetyczne.....	588
Operacje logiczne.....	589
Porównania liczbowe.....	589
Funkcje porównywania łańcuchów.....	589
Funkcje matematyczne.....	590
Funkcje łańcuchowe.....	591
Funkcje daty i czasu.....	593
Funkcje sumowania używane z warunkiem GROUP BY.....	594

<b>Dodatek C</b>	<b>Kody źródłowe aplikacji z książki.....</b>	<b>595</b>
	Rozdział 15 — Księga gości (listagosci.asp).....	595
	Rozdział 16 — Lista wiadomości (index.asp).....	598
	Rozdział 16 — Czytanie wiadomości (wiadomosc.asp).....	604
	Rozdział 17 — Strona rejestrowania się (logowanie2.asp).....	608
	Rozdział 17 — Strona zawartości (zawartosc.asp).....	610
	Rozdział 19 — Strona pytania quizu (quiz.asp).....	615
	Rozdział 20 — Dodaj do koszyka (szczegoly.asp).....	626
	Rozdział 20 — Usuwanie elementów z koszyka (sprawdzanie.asp).....	635
	Rozdział 20 — Końcowe sprawdzanie (sprawdzanie2.asp).....	639
	<b>Skorowidz .....</b>	<b>647</b>

Rozdział 15.

# Zaawansowane techniki — przekształcanie istniejących witryn internetowych w witryny dynamiczne

Jeśli tworzyłeś witryny internetowe, zanim zacząłeś czytać tę książkę, doświadczyłeś zapewne wielu sytuacji, w których mógłbyś użyć dynamicznych narzędzi w celu poprawienia funkcjonalności tych witryn, uatrakcyjnienia ich czy też sprawienia, by były ciągle atrakcyjne. W tym rozdziale przyjrzymy się kilku krótkim aplikacjom stworzonym za pomocą programu UltraDev, których możesz użyć w celu dodania lub zastąpienia istniejących, statycznych części Twojej strony. Starałem się wybrać takie aplikacje, w których wykorzystanie baz danych nie jest widoczne na pierwszy rzut oka. W rozdziale zostaną omówione następujące zagadnienia:

- uruchamianie i wykorzystanie projektów z poprzedniego rozdziału w sposób najbardziej efektywny,
- dodawanie losowo wybranego banera reklamowego do strony,
- automatyczne wyświetlanie obrazków na stronie w zależności od pory roku czy dnia,
- tworzenie personalizowanej książki gości.

## Przejście od stron statycznych do dynamicznych

Najtrudniejszą rzeczą w programowaniu stron dynamicznych jest konieczność całkowitego zerwania z zasadami, które przyswoiłeś sobie wcześniej. Zamiast używać na stronach statycznych łącz lub obiektów, masz możliwość zmieniania wszystkich stron w serwisie za każdym razem, gdy użytkownik je odwiedza.

Najbardziej złożone aplikacje internetowe, takie jak rozwiązania *e-commerce* czy przeszukiwanie baz danych, są omówione w dalszej części tego rozdziału. Przekonasz się też, że większość spośród najistotniejszych stron internetowych używa bardzo prostych narzędzi dynamicznych, co sprawia, że są one bardziej interesujące. Na przykład, ile razy odwiedziłeś witryny Dilberta? (Jeśli odpowiedź brzmi: „Ani razu!”, zastanawiam się, co masz wspólnego z komputerami.) Wielu ludzi odwiedza codziennie tę stronę. To samo dotyczy witryny CNN lub stron poświęconych informatyce. Przypuszczam, że odwiedzasz przynajmniej jedną taką stronę dziennie. Pytam więc, co sprawia, że odwiedzasz je tak często?

Zastanów się nad tym przez chwilę. Odpowiedź brzmi: „Ponieważ ich zawartość często się zmienia”. Każdego dnia na stronie Dilberta pojawia się nowy odcinek komiksu, natomiast w witrynie CNN można znaleźć aktualne informacje. Zapewnienie aktualnej zawartości stron jest doskonałym sposobem przyciągania użytkowników.

Co zatem zrobić, jeśli nie chcesz uaktualniać codziennie Twojej witryny? Zawsze możesz zapewnić użytkownikom nowe informacje i elementy graficzne. Wiele stron przechowuje swoją zawartość w bazie danych, która jest udostępniana we właściwym czasie. Mimo że takie rozwiązanie nie sprawdza się na stronach informacyjnych (chyba że wierzysz w to, że media tworzą zdarzenie, które nie miało jeszcze miejsca), działa jednak dobrze w przypadku zawartości, która może być generowana, zanim jest użyta (np. obrazki, miesięczne biuletyny informacyjne).

Poznasz kilka niewielkich projektów, które możesz wykorzystać na swoich stronach w celu zastąpienia standardowych, statycznych elementów. Są to ogólne moduły stron internetowych, które mogą być użyte jako uzupełnienie stron już istniejących. Krótki przegląd tych projektów przedstawiam niżej.

- *Losowe banery reklamowe lub obrazki*. Banery reklamowe są sposobem na reklamę innych stron. Za każdym razem, gdy użytkownik odwiedza stronę, obrazek na niej zmienia się. Tę metodę możesz wykorzystać zarówno wobec banerów reklamowych, jak i obrazków.
- *Okresowe lub czasowe obrazki*. Czy chciałbyś, aby Twoja strona była aktualizowana w zależności od pory roku? Czy chciałbyś zmieniać szatę graficzną tak, by odzwierciedlała kolor liści na drzewach? Zmianie grafiki na stronie czyni ją zróżnicowaną i interesującą, przy czym nie wymaga żadnej dodatkowej pracy, z wyjątkiem zainicjowania sposobu zmian.
- *Księga gości*. Często występującym elementem witryn internetowych jest księga gości. Możesz łatwo włączyć ją do swojej strony. Możesz również dodać filtr „grzecznościowy”, polegający na eliminacji wulgarnych wiadomości lub języka, który uważasz za nieodpowiedni.

Tworząc te małe projekty, będziesz zmagał się z różnorodnymi problemami, dzięki czemu nabierzesz wprawy potrzebnej do zrozumienia dużych projektów, którymi będziemy zajmować się w dalszej części tej książki. Być może będziesz również chciał przejrzeć swoje strony i sprawdzić, czy można w nich zastosować poznane funkcje.

Zanim rozpoczniemy, przedstawię krótko zagadnienia omówione w ostatniej części tej książki. Moim zdaniem najlepszą nauką jest praktyka. Dreamweaver UltraDev sam w sobie nie jest programem skomplikowanym. Do tej pory zapoznałeś się ze wszystkimi funkcjami programu, wiesz także, gdzie można je znaleźć. Bardziej skomplikowane jest utworzenie układu dokumentu HTML w UltraDev niż wykorzystanie dynamicznych narzędzi do utworzenia połączenia z bazą danych. Z tego powodu ostatnia część książki jest poświęcona praktycznym przykładom, które mają zademonstrować możliwości programu UltraDev w połączeniu z serwerem SQL. Wszystkie przykłady z tego rozdziału są zgodne z serwerem MySQL, więc możesz je napisać i uruchomić bez dodatkowych kosztów.

Wszystkie pozostałe rozdziały są podzielone na trzy części:

1. *Projekt bazy danych.* Najważniejszą częścią każdego projektu jest właściwe zaprojektowanie procesów drugoplanowych bazy danych. Pewne ograniczenia UltraDev można ominąć, pisząc odpowiednie zapytania. Nawet jeśli masz doświadczenie w pracy z bazami danych, powinieneś zwrócić szczególną uwagę na tę część.
2. *Projektowanie witryny.* Zanim zaczniesz programować czy też pisać kod HTML, musisz zdecydować, jak strona ma wyglądać. Nie zajmuje to zwykle dużo czasu, ale nie może zostać pominięte. Nie możesz zacząć tworzenia stron, dopóki nie będziesz miał całkowitej pewności, jak ma wyglądać Twoja witryna. Brak ustalonej wcześniej struktury może być przyczyną wielu problemów.
3. *Implementacja.* Każdy rozdział zawiera w sobie część implementacji całego projektu, czyli kod HTML, wykorzystanie behawiorów serwera i inne czynności. Czasem będzie to wymagać ręcznej edycji kodu źródłowego. Gdy implementacja zostanie zakończona, będziesz miał w pełni funkcjonalną witrynę internetową, gotową do umieszczenia na serwerze. Pozostaje Ci jedynie stworzenie zawartości odpowiadającej Twoim potrzebom, lecz struktura serwisu będzie już gotowa.

Przedstawiony plan działania stanowi kompletny obraz przebiegu prac nad projektem. Lepiej zrozumiesz materiał zawarty w tej książce, jeśli będziesz czytał ją rozdział po rozdziale, a nie skacząc pomiędzy nimi. Czasem będziemy wykonywać w UltraDev rzeczy, które nie wchodzi w skład końcowego projektu, jednak okażą się one pomocne, gdy przystąpisz do tworzenia własnych projektów.

Do dzieła!

## Losowe wybieranie banerów

Wiem, wiem, są one denerwujące, lecz bardzo popularne. Paski reklamowe mogą zostać sprzedane za bardzo wysoką cenę, o ile Twoja strona jest często odwiedzana. Wiesz już, jak w sposób dynamiczny wyświetlić serię obrazków, ale nie stworzyłeś do tej pory pojedynczego obrazka, który byłby dynamiczny. Użycie tabeli zawierającej zbiór obrazków jest już połową sukcesu, ale fakt ten nie wpływa na możliwość zmieniania ich przy każdorazowym ładowaniu strony.

## Definicja tabeli

Kluczem do utworzenia zmieniających się obrazków jest napisanie zapytania, które będzie zwracać za każdym razem różne rekordy. Jest to łatwiejsze, niż myślisz, lecz musisz najpierw stworzyć bazę danych, do której zapytanie będzie się odwoływać.

W każdym z projektów założyłem, że nim rozpoczniesz ćwiczenie, będziesz miał ustaloną połączenie z bazą. Zrób to, jeśli dotąd tego nie uczyniłeś. Czas zająć się językiem SQL.

Musisz mieć zbiór obrazków, aby losowo wybrać jeden z nich. Tę informację będziesz przechowywał w tabeli w bazie. Większość pasków reklamowych to łącza do innych stron, więc musisz przyporządkować każdemu obrazkowi właściwy adres internetowy.

Definicja tabeli musi zawierać tylko te dwa pola, a ponieważ zakładamy, że każdy z banerów wykorzystuje obrazki o innych nazwach, użyjemy nazwy pliku jako klucza głównego:

```
create table tblBanner (
  obrazek      varchar(80) not null,
  url          varchar(80) not null,
  primary key  (obrazek)
);
```

Dowiedziałeś się już, że nie możemy przechowywać obrazków w tabeli, ale możemy przechowywać w niej nazwy ich plików, a następnie wstawić te nazwy w znaczniki <IMG>. Otaczając obrazki znacznikami, tworzymy łącza, które wypełnione jest polem url.

Rysunek 15.1 przedstawia schemat tabeli użytej w aplikacji wyświetlającej baner.

### Rysunek 15.1.

Tabela Banner przechowuje lokalizację obrazków oraz adres URL

tblBanner
obrazek*
url

## Przykładowe dane

Rozpocznię od umieszczenia trzech pasków reklamowych w moim systemie, które posłużą jako przykład — są one przedstawione na rysunku 15.2.

### Rysunek 15.2.

Dla potrzeb przykładowego programu stworzyłem kilka pasków reklamowych. Mam nadzieję, że Twoje będą lepsze



Oprócz tych trzech pasków potrzebujemy nazwę pliku i folder, w którym są one przechowywane, oraz adres URL dostępny dla użytkownika, który kliknie wybrany baner:

- *banners/fade.gif* — *http://fade.poisonooth.com/*
- *banners/buddy.gif* — *http://buddy.poisonooth.com/*
- *banners/detailing.gif* — *http://detailing.poisonooth.com/*



Powyższe adresy URL nie są prawdziwe, ale domena *poisonooth* istnieje i należy do mnie. Zamiast zgadywać nazwę nieistniejącej domeny, wolę użyć własnej. Nie chciałbym, by prawdziwa firma „Salon samochodowy Carla” zasypała mnie stosem e-maili w związku z nieautoryzowanym użyciem ich nazwy.

Wprowadźmy przykładowe dane do tabeli w bazie:

```
insert into tblBanner values ('banners/fade.gif', 'http://fade.poisonooth.com/');
insert into tblBanner values ('banners/buddy.gif', 'http://buddy.poisonooth.com/');
insert into tblBanner values ('banners/detailing.gif', 'http://detailing.
poisonooth.com/');
```

## Zapytanie SQL

Mimo że mamy już dane w tabeli, wciąż nie zostało wyjaśnione, w jaki sposób będziemy losowo wybierać obrazek. Oczywiście, możemy wziąć wszystkie dane, tworząc bardzo proste zapytanie:

```
mysql> SELECT * FROM tblBanner;
+-----+-----+
| obrazek          | url                               |
+-----+-----+
| banners/fade.gif | http://fade.poisonooth.com/      |
| banners/buddy.gif| http://buddy.poisonooth.com/     |
| banners/detailing.gif| http://dateiling.poisonooth.com/ |
+-----+-----+
3 rows in set (0.04 sec)
```

Niestety, za każdym razem, gdy wykonywane jest zapytanie, informacje zostają zwrócone w tej samej kolejności. Idealnym rozwiązaniem byłaby zamiana kolejności zwracanych rekordów. Dodając do przeszukiwania komendę `ORDER BY` połączoną z funkcją MySQL `rand()`, otrzymamy rekordy w porządku losowym.

```
mysql> SELECT * FROM tblBanner ORDER BY RAND();
+-----+-----+
| obrazek          | url                               |
+-----+-----+
| banners/detailing.gif | http://detailing.poisonooth.com/ |
| banners/buddy.gif    | http://buddy.poisonooth.com/     |
| banners/fade.gif     | http://fade.poisonooth.com/      |
+-----+-----+
```

Ten rezultat już jest dużo lepszy od poprzedniego. Pozostaje jednak jeszcze jeden problem, polegający na tym, że zwracane są wszystkie rekordy, a nie tylko jeden. Ponieważ jesteśmy zainteresowani wyświetlaniem pojedynczego obrazka, reszta rekordów nas nie interesuje. Najprostszym sposobem ograniczenia liczby zwracanych rekordów jest zastosowanie funkcji `LIMIT`. Użycie `LIMIT 1` spowoduje wyświetlenie pojedynczego rekordu:



```
mysql> SELECT * FROM tblBanner ORDER BY RAND() LIMIT 1;
+-----+-----+
| obrazek          | url          |
+-----+-----+
| banners/detailing.gif | http://detailing.poison.tooth.com/ |
+-----+-----+
1 row in set (0.04 sec)
```

Jeśli nie chcesz, nie musisz tego robić. W UltraDev *Recordset* działa w taki sposób, że masz bezpośredni dostęp tylko do pierwszego zwracanego rekordu, chyba że zaznaczysz wcześniej w opcjach programu, by przeglądał wszystkie rekordy w bazie — na czym nam teraz zależy.

Kod SQL mamy już gotowy. Teraz powinniśmy zająć się tworzeniem mapy serwisu, jednak ten projekt różni się od innych tym, że zajmujemy się funkcjonowaniem obrazków, które można osadzić na dowolnej stronie. Z tego powodu pominiemy mapę, ponieważ nie jest ona tutaj konieczna.

## Implementacja

Aby zaimplementować losowe wybieranie obrazków, otwórz nowe okno projektu dokumentu w UltraDev. Możesz to uczynić, używając istniejącego pliku. Stworzenie dynamicznego paska reklamowego wymaga użycia na stronie trzech elementów UltraDev:

1. Zestawu rekordów zawierającego obrazki, które mają być wyświetlone — będzie on zawierał obrazek, który chcemy wyświetlić. Ponieważ kod SQL mamy już gotowy, wystarczy na jego podstawie stworzyć zapytanie w UltraDev.
2. Obrazka, którego plik źródłowy będzie określany przez zestaw rekordów.
3. Łącza otaczającego obrazek i związanego z polem URL.

Rozpoczniemy od wstawienia na stronę zestawu rekordów, ponieważ wszystkie inne elementy będą korzystać z jego pól.

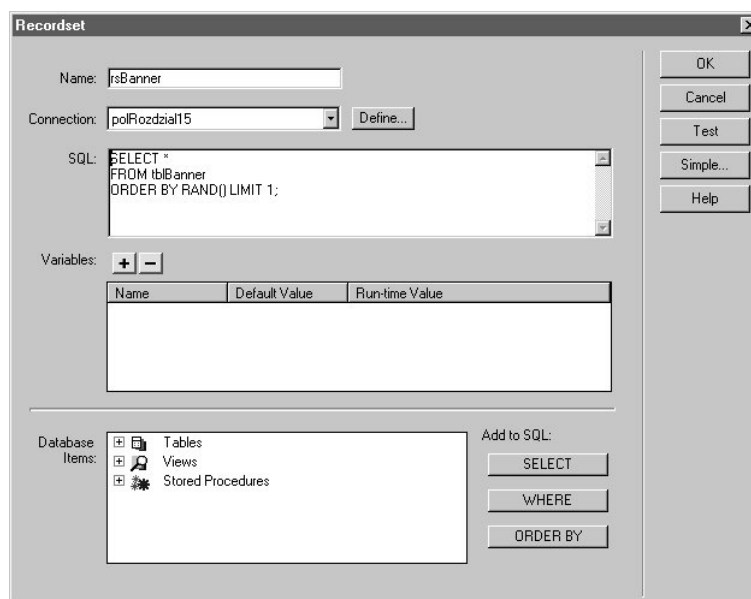
Otwórz paletę *Server Behavior* i kliknij przycisk +, następnie wybierz opcję *Recordset* z menu rozwijanego. Ponieważ to zapytanie wymaga użycia specjalnych funkcji, nie będziesz mógł losowo wybierać zwracanego rekordu w trybie prostego tworzenia zapytania. Kliknij przycisk *Advanced*, aby przełączyć się w tryb zaawansowany.

W widoku zaawansowanego zapytania, przedstawionym na rysunku 15.3, możesz ręcznie wskazać zapytanie, którego chcesz użyć na stronie. Łatwiej jest pracować w trybie zaawansowanym, ponieważ jest on bardziej elastyczny, jednak wymaga to wprawy.

Skonfiguruj zaawansowane zapytanie tak, jak poprzednio:

1. Stwórz zapytanie o nazwie *rsBanner*.
2. Wybierz bieżące połączenie, które stworzyłeś dla projektu.
3. Wprowadź poniższe zdanie jako treść zapytania: "Select \* FROM tblBanner ORDER BY RAND() LIMIT 1;".

**Rysunek 15.3.**  
 Będziemy ręcznie  
 tworzyć zapytanie  
 w trybie  
 zaawansowanym.  
 Użycie funkcji  
 RAND jest  
 niedozwolone  
 w trybie prostym  
 konfiguracji  
 zapytania



Kliknij kilka razy przycisk *Test*, aby sprawdzić działanie zapytania. Nie powinieneś za-  
 uważać żadnej prawidłowości w otrzymywaniu rekordów. Mimo że żaden komputer tak  
 naprawdę nie potrafi działać losowo, otrzymany rezultat powinien być wystarczający  
 w większości wypadków. Jeśli nie używasz MySQL, sprawdź, jaki jest odpowiednik  
 funkcji `Rand()` w Twojej bazie.

Wstawimy teraz obrazek do naszego dokumentu. Otwórz paletę *Object* i kliknij ikonę  
*Image* lub wybierz *Insert Image* z menu *Insert*. Program zapyta o nazwę pliku. Zamiast  
 wybierać jeden z plików graficznych, które stworzyłeś, kliknij *Select File* z przycisku  
*Datasource*. Pojawi się wtedy okno, przedstawione na rysunku 15.4. W nim będziesz  
 mógł połączyć pole z tabeli `tblBanner` z atrybutem `SRC` w znaczniku `<IMG>`.

Kliknij znak `+` obok zestawu rekordów *rsBanner*, aby zostały wyświetlone jego pola.  
 Wybierz pole *obrazek* i kliknij *OK*, aby zakończyć połączenie.

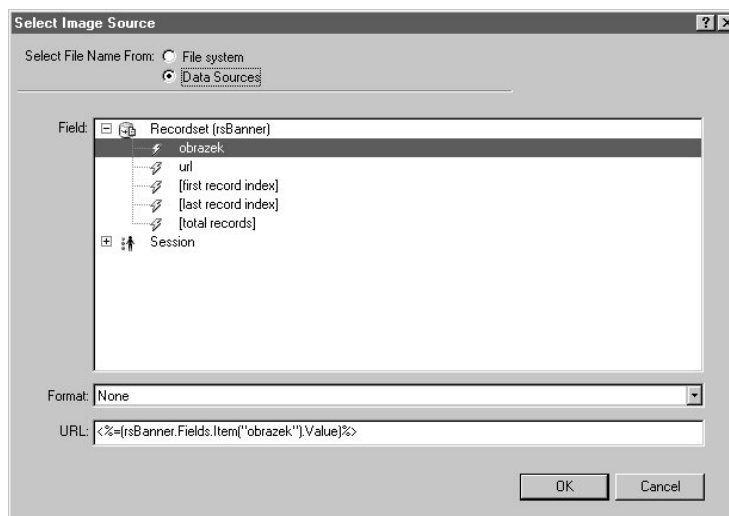
Ponieważ UltraDev nie operuje w kodzie nazwą obrazka, będziesz widział tylko ikonę  
 przedstawiającą obrazek. Możesz oczywiście włączyć opcję *Live Data* z menu *View*,  
 aby wybrać bazę danych i wyświetlić losowo wybrany obrazek. Kliknięcie przycisku  
*Refresh*, umieszczonego na górze okna, spowoduje wybranie nowego obrazka. Pamię-  
 taj, że wybór jest losowy i jeden obrazek może się pojawić kilka razy pod rząd. Jeśli  
 poprawnie stworzyłeś zapytanie, powinieneś zobaczyć rezultat podobny do przedsta-  
 wionego na rysunku 15.5.

#### Zdarzenia losowe

Jeśli zdarzy się, że zobaczysz ten sam rekord pojawiający się kolejny raz, nie znaczy to, że coś  
 poszło nie tak. Po prostu prawdopodobieństwo wybrania jednego z naszych rekordów wynosi 1/3.

**Rysunek 15.4.**

Połącz Twój obrazek z polem obrazek w zestawie rekordów rsBanner

**Rysunek 15.5.**

Powinieneś zobaczyć baner reklamowy. Kliknij przycisk refresh, by zobaczyć losowo wygenerowany obrazek



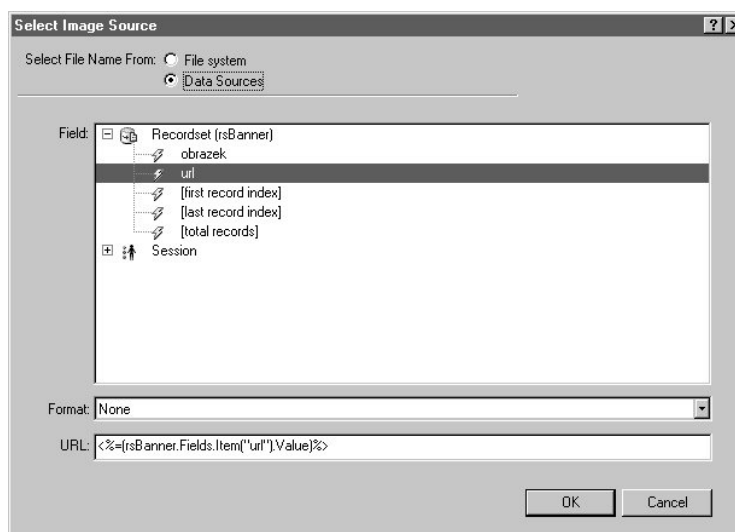
Dodamy teraz łącze do banera. Jest to proces podobny do tego, który właśnie zakończyliśmy. Zaznacz cały baner reklamowy (możesz to zrobić w trybie *Live Data*, jeśli chcesz), po czym wybierz *Make Link* z menu *Modify*. Będziesz ponownie zapytany o plik, do którego chcesz utworzyć łącze. Tak jak to zrobiłeś ze źródłem pliku, kliknij *Select File* z menu *Data Source*. Tym razem jednak, zamiast wybierać pole *obrazek*, wybierz pole *url* i kliknij *OK*, jak pokazano na rysunku 15.6. W ten sposób wstawiliśmy do łącza URL z bazy.

Tak oto ukończyliśmy podstawowy program losowego generowania paska reklamowego. Obejrzyj teraz swoją stronę w przeglądarce i sprawdź, co się dzieje. Masz stworzoną kompletną aplikację do losowej zamiany obrazków na stronie.



Zestaw rekordów jest losowo tworzony za każdym razem, gdy wykonujemy zapytanie w bazie. Nie oznacza to jednak, że możesz skopiować i wkleić dynamiczny obrazek w inne miejsce danej strony, aby mieć kilka miejsc, w których są losowo generowane obrazki. W tym celu musiałbyś zdefiniować kilka zestawów rekordów na podstawie tego samego zapytania, a każdy z nich zwróciłby Ci pojedynczy obrazek.

**Rysunek 15.6.**  
Ustaw połączenie  
do pola URL



Taki projekt, mimo jego prostoty, dołączałem do prawie każdej komercyjnej strony, jaką tworzyłem. Standardowo jest to robione przez oddzielny skrypt CGI, lecz UltraDev pozwala tworzyć i zarządzać tą właściwością w łatwy sposób, poprzez intuicyjny interfejs.

Brakuje w tym projekcie pewnej rzeczy — możliwości pamiętania, jaki obrazek użytkownik widział poprzednim razem, gdy odwiedzał stronę, oraz możliwości niewyświetlania tego obrazka przy ponownym wejściu. Można tego dokonać poprzez wysłanie *cookie*, w którym będzie przechowana nazwa pliku wyświetlonego wcześniej i użycie go w celu przefiltrowania zbioru rekordów. To jest zaawansowana funkcja, która wymaga ręcznej edycji kodu HTML — nauczysz się tego w rozdziale 17.

## Obrazki okresowe i czasowe

Innym popularnym typem dynamicznego obrazka, który możesz dodać do swojej strony, jest obrazek zmieniający się w zależności od pory roku lub dnia. Jest to stosowane wtedy, gdy masz zdefiniowaną zawartość strony i chcesz ją automatycznie uaktualnić, gdy minie określona data. Zmianie szaty graficznej całego serwisu co miesiąc lub częściej sprawia wrażenie, jakby strona była ciągle aktualizowana, nawet gdy Ty nic przy niej nie robisz.

Ten projekt jest bardzo podobny do poprzedniego, lecz wymaga drobnej modyfikacji zapytania i tabeli.

### Definicja tabeli

Tak, jak w poprzednim przykładzie, tabela dla dynamicznych obrazków będzie prosta. Musimy zapamiętać nazwę obrazka oraz datę i czas. Rozwiązanie, które przedstawię poniżej będzie zmieniać obrazek w zależności od określonej daty. Po przeczytaniu tego rozdziału będziesz mógł zmienić pola tabeli tak, aby zmiana obrazka następowała co dzień, miesiąc, rok czy też z innych powodów.

Definicja tabeli opartej na zmianie obrazka względem daty jest następująca:

```
create table      tblObrazkiCzasowe(
  obrazek        varchar(80) not null,
  dataAktywacji  date,
  Primary key    (obrazek)
);
```

Rysunek 15.7 przedstawia stosowną tabelę.

### Rysunek 15.7.

*Tabela podobna do tej, która była wykorzystana w poprzedniej aplikacji, będzie użyteczna przy zamianie obrazków opartych na określonej dacie*

tblObrazkiCzasowe
obrazek*
data aktywacji

## Zapytanie SQL

MySQL udostępnia zbiór funkcji, które pozwalają szybko operować na danych zależnych od daty. Możesz łączyć ze sobą poniższe funkcje, aby utworzyć zapytanie, które będzie spełniać nasze wymagania:

- DAYOFWEEK (<date>). Zwraca wartość numeryczną reprezentującą dzień tygodnia. Tutaj 1=niedziela, 2=poniedziałek itd.
- DAYOFYEAR (<date>). Zwraca dzień roku, na przykład 1 stycznia jest pierwszym dniem roku, natomiast 31 grudnia jest 365 dniem roku (lub 366, jeśli rok jest przestępny).
- DYNNAME (<date>). Zwraca nazwę dnia tygodnia, a nie numer dnia.
- MONTHNAME (<date>). Zwraca nazwę miesiąca w danej dacie.
- WEEK (<date>). Zwraca wartość numeryczną (1-52) odpowiadającą numerowi tygodnia w roku.
- CURDATE(). Zwraca bieżącą datę.
- CURTIME(). Zwraca bieżącą godzinę.
- NOW(). Zwraca bieżącą datę i godzinę.



Aby zapoznać się z pełną listą funkcji MySQL, zajrzyj do dodatku B, *Krótki przewodnik po funkcjach MySQL*.

W celu zaoszczędzenia czasu użyjemy tych samych obrazków, których używaliśmy poprzednio. Tym razem jednak nie będą pojawiać się losowo, ale będą zależne od konkretnej daty. Jeśli daty w tym przykładzie ulegną przedawnieniu, zmień je tak, by były aktualne:

```
insert into tblObrazkiCzasowe values ('banners/fade.gif', '2000-11-30');
insert into tblObrazkiCzasowe values ('banners/buddy.gif', '2001-02-28');
insert into tblObrazkiCzasowe values ('banners/detailing.gif', '2001-05-30');
```

Najważniejszym zadaniem jest właściwe rozpoczęcie tworzenia zapytania. Oczywiście jest fakt, że musimy znać bieżącą datę — zwracaną przez funkcję `CURDATE()` lub `NOW()` — nie możemy jednak po prostu porównać daty dzisiejszej z datą przechowywaną w tabeli. Dałoby to nam w rezultacie rekord z pewną datą. My próbujemy zrealizować zbudowanie systemu, który wybiera obrazek, gdy minęła określona data. W tym celu wybierz rekordy z bazy, w których data z pola *dataAktywacji* jest większa niż data bieżąca:

```
mysql> SELECT * FROM tb1ObrazkiCzasowe WHERE dataAktywacji > CURDATE() ORDER BY
➤dataAktywacji;
+-----+-----+
| obrazek          | dataAktywacji |
+-----+-----+
| banners/fade.gif | 2000-11-30   |
| banners.buddy.gif | 2001-02-28   |
| banners/detailing.gif | 2001-05-30   |
+-----+-----+
3 rows in set (1.19 sec)
```

Funkcja `ORDER BY` jest potrzebna po to, by data najbliższa dzisiejszej była na początku tabeli. Jeśli ją pominiesz, otrzymasz wynik, w którym daty zostaną wyświetlone tak, jak były wstawiane do tabeli.

Podobnie jak w projekcie z banerem, interesuje nas tylko pierwszy rekord z zapytania. Możesz albo używać tylko jednego rekordu w zbiorze rekordów, albo użyć funkcji `LIMIT`:

```
mysql> SELECT * FROM tb1ObrazkiCzasowe WHERE dataAktywacji > CURDATE() ORDER BY
➤dataAktywacji LIMIT 1;
+-----+-----+
| obrazek          | dataAktywacji |
+-----+-----+
| banners/fade.gif | 2000-11-30   |
+-----+-----+
1 row in set (0.71 sec)
```

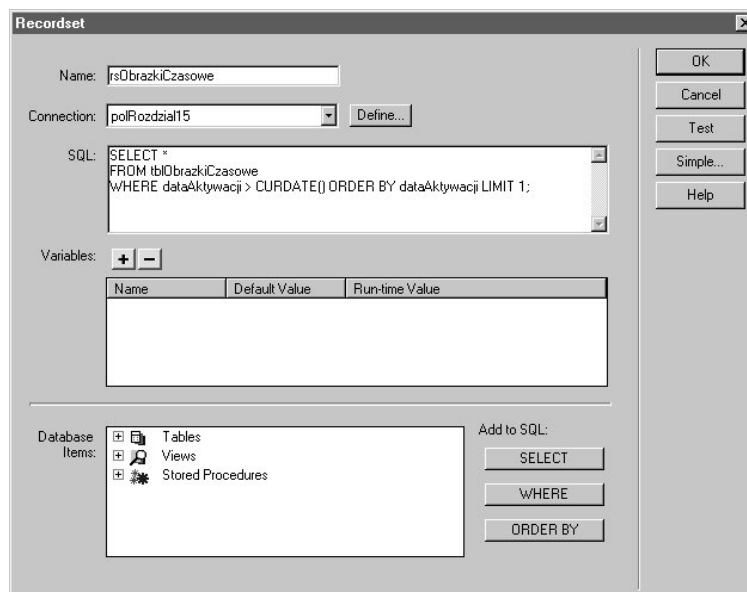
Otrzymujesz w ten sposób konkretny obrazek, który zostanie użyty, gdy minie określona data.

Wykonamy teraz takie same czynności jak w poprzednim przykładzie, aby wstawić nasz obrazek na stronę:

1. Dodaj zestaw rekordów `rsObrazkiCzasowe` do dokumentu, który zawiera stworzone wcześniej zapytanie. Rysunek 15.8 przedstawia poprawnie zdefiniowany zestaw rekordów.
2. Wstaw obrazek na stronę, która jest połączona z polem *obrazek* Twojego nowego zestawu rekordów.

Jak widzisz, jest to prostsze niż losowe wybieranie obrazków, ponieważ pracujemy tylko na jednym polu w tabeli. Niestety, mimo że ten i poprzedni przykład są bardzo proste i przejrzyste, nie nadają się do użytku na szerszą skalę.

**Rysunek 15.8.**  
Stwórz  
zaawansowane  
zapytanie  
w taki sposób,  
jak w przypadku  
banera



## Wiele obrazków okresowych i czasowych

Najbardziej podoba mi się to, że tworząc system zarządzania obrazkami w zależności od danej daty, nie musimy troszczyć się o zachodzące zmiany. Możesz zostać porwany przez kosmitów i znaleźć się na stacji kosmicznej, a w tym czasie serwer będzie się troszczył o zmianę wyglądu Twojej strony.

Tak sprecyzowany sposób zarządzania obrazkami w zależności od daty działa tylko w odniesieniu do pojedynczego obrazka, co jest ewidentnym problemem. Wszystkie obrazki znajdujące się w tabeli `tblObrazkiCzasowe` są używane jako wariacje dotyczące pojedynczego obrazka. Idealnym rozwiązaniem byłoby stworzenie systemu tematycznego, w którym różne obrazki byłyby przechowywane w tabeli i dotyczyły całej grupy danych. Jedynym sposobem w obecnym systemie byłoby tworzenie nowej tabeli dla każdego obrazka. Potrzebujemy innego rozwiązania.

Przykładowo, mógłbyś mieć schemat strony na Halloween, który połączony z odpowiednimi zapytaniami zastąpiłby obrazki wyświetlane na stronie jakimiś strasznymi zdjęciami. Zamiast zastępować pojedynczy obrazek, można zastąpić wszystkie elementy graficzne, korzystając z jednej tabeli.

Aby to zrobić, musimy przedefiniować istniejącą tabelę i dodać jedną nową:

```
create table tblObrazkiCzasowe (
    obrazek      varchar (80) not null,
    nazwa        varchar(80) not null,
    IDSchematu   varchar(80) not null,
    primary key  (obrazek, nazwa, IDSchematu)
);
```

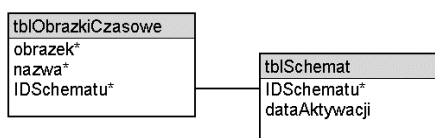
Nowa definicja tabeli `tblObrazkiCzasowe` zawiera pole `nazwa` oraz pole `IDSchematu`, które będzie zdefiniowane w dodatkowej tabeli. Pole `nazwa` będzie użyte do identyfikacji obrazka przez nazwę abstrakcyjną, a nie przez rzeczywistą nazwę pliku. Taki zabieg pozwoli wywołać obrazek nazwany *Mój Tytuł* czy też podobnie, niezależnie od rodzaju schematu, nawet wtedy, gdy rzeczywiste nazwy plików będą różne.

```
create table tblSchemat (
  IDSchematu      varchar(80) not null,
  dataAktywacji   date,
  primary key (IDSchematu)
);
```

W tablicy tematów pole `IDSchematu` związane jest z polem `dataAktywacji`. Na przykład, możesz wstawić schemat dotyczący Świąt Bożego Narodzenia, związanego z datą 25 XII 20XX. Nowa baza danych przedstawiona jest na rysunku 15.9.

### Rysunek 15.9.

*Nowy schemat zmian oparty na danych używa dwóch tabel — jedną na przechowywanie obrazków, drugą na schematy tematyczne przyporządkowane datom aktywacji*



## Zapytanie SQL

Najlepszym sposobem zrozumienia sposobu działania tworzonego systemu jest wprowadzenie kilku przykładowych danych i przetestowanie działania zapytania.

```
insert into tblObrazkiCzasowe values ('mainimage1.jpg', 'naglowek', 'lato');
insert into tblObrazkiCzasowe values ('mainimage23.jpg', 'naglowek', 'zima');
insert into tblObrazkiCzasowe values ('mainimage65.jpg', 'naglowek', 'jesien');
insert into tblObrazkiCzasowe values ('mainimage2.jpg', 'naglowek', 'wiosna');

insert into tblSchemat values ('lato', '2000-06-21');
insert into tblSchemat values ('zima', '2000-12-21');
insert into tblSchemat values ('jesien', '2000-09-21');
insert into tblSchemat values ('wiosna', '2000-03-21');
```

Taki zbiór danych tworzy cztery różne schematy (wiosna, lato, jesień, zima) oraz cztery różne obrazki wstawiane w nagłówek strony. Tym, co czyni ten system lepszym od poprzedniego, jest fakt, iż umożliwia on przechowywanie w jednej tabeli różnych obrazków (dotyczących nie tylko nagłówka).

Aby zastosować nowy układ, musimy zmodyfikować poprzednie zapytanie SQL tak, by utworzyć relację pomiędzy tabelami i wybrać właściwy obrazek określony za pomocą abstrakcyjnej nazwy:

```
mysql> SELECT * FROM tblObrazkiCzasowe WHERE dataAktywacji > CURDATE() ORDER BY
➤dataAktywacji LIMIT 1;
```



Powyższy wpis musimy zmienić na następujący:

```
mysql> SELECT * FROM tb1ObrazkiCzasowe, tb1Schemat WHERE tb1ObrazkiCzasowe.IDSchematu
↳=tb1Schemat.IDSchematu AND IDSchematu.dataAktywacji > CURDATE() AND tb1Obrazki
↳Czasowe.nazwa='naglowek' ORDER BY tb1Schemat.dataAktywacji LIMIT 1;
```

Ponieważ nie mieliśmy jeszcze do czynienia z tak skomplikowanym zapytaniem, przyjrzyjmy się mu dokładnie:

- `tb1ObrazkiCzasowe.IDSchematu=tb1Schemat.IDSchematu`. Tworzy relację pomiędzy dwoma tabelami (tabelą obrazków i tematów), łącząc je polem `IDSchematu`.
- `IDSchematu.dataAktywacji > CURDATE()`. Porównuje bieżącą datę z datą aktywacji zapisaną w tabeli `tb1Schemat`.
- `tb1ObrazkiCzasowe.nazwa='naglowek'`. Wybiera obrazek nie tylko na podstawie daty aktywacji, lecz również na podstawie abstrakcyjnej nazwy nadanej każdemu polu.

Użycie nowego systemu wymaga uaktualnienia bazy danych i zastąpienia zapytania. Wciąż jednak musisz zdefiniować osobne zapytania dla każdego z obrazków, które chcesz wyświetlać na stronie. Ten system jest lepszy od poprzedniego. Wystarczy bowiem tylko jedna tabela, więc nie ma potrzeby tworzenia nowej dla każdego obrazka umieszczanego na stronie.



Jak wspominałem wcześniej, UltraDev póki co nie ma możliwości przechowywania obrazków ani żadnych innych obiektów binarnych bezpośrednio w bazie danych. Mam nadzieję, że możliwość taka pojawi się w następnej wersji. Jeśli tak się stanie, aplikacje będą mogły ściągać nowe obrazki bezpośrednio do bazy danych. Możesz równie dobrze napisać to dziś, ale będzie to wymagać dodatkowego oprogramowania w celu otrzymania danych, zapisania ich na dysku, a następnie odwołania się do nich z poziomu bazy danych.

## Pozyskiwanie opinii od użytkowników

Ostatnim zadaniem w tym rozdziale będzie stworzenie księgi gości. Warto taką księgę wdrożyć w witrynie — osoby dokonujące w niej wpisów mają uzasadnioną pewność, że ich uwagi będą czytane. Ile razy, odwiedzając witrynę internetową, klikałeś łącze po to, by zostawić swoją opinię? Okazywało się wtedy, że otwierało ono jedynie Twojego klienta pocztowego po to, byś mógł wysłać e-maila do bezimiennej osoby, która zapewne i tak nie odpisze. Księga gości umożliwia zostawienie na stronie znaku swojej obecności i nie może być tak łatwo ignorowana.

Jestem przekonany, że masz pewne obawy co do wiadomości, jakie użytkownik może zostawić w księdze gości. Niestety, to są słuszne obawy. Jest wiele niemiłych osób, które wykorzystają każdą szansę, aby wypowiedzieć się na forum publicznym i pokazać w ten sposób swoją niedojrzałość. Nieodpowiedni język, osadzanie obrazków, nieodpowiednie znaczniki HTML w tekście — to jedne z możliwych problemów, które się będą pojawiać. Zobaczmy, czy ta księga gości będzie mogła sobie poradzić z niektórymi z nich.

## Definicja tabeli

Zawartość tabeli dotyczącej księgi gości zależy wyłącznie od Ciebie — od tego, ile informacji chcesz w niej przechowywać. Przykładowa księga gości stworzona w tym ćwiczeniu będzie zawierać nazwę użytkownika, e-mail oraz komentarz.

Tabelę można zdefiniować następująco:

```
create table tblKomentarz (  
    IDWiadomosci int auto_increment not null,  
    nazwa varchar (50),  
    email varchar (80),  
    wiadomosc varchar(250),  
    Primary key (IDWiadomosci)  
);
```

Użyliśmy tutaj po raz pierwszy typu `auto_increment` dla pola `IDWiadomosci`. Funkcja ta automatycznie zwiększa numer tego pola, gdy nowy rekord jest dodawany. Nie musisz się zatem martwić o samodzielne generowanie kolejnego numeru. Niestety, typ ten różni się w zależności od konkretnej bazy danych i być może będziesz musiał użyć innej funkcji. Jeśli nie używasz MySQL, możesz użyć treści wiadomości jako klucza głównego, ponieważ jest raczej niemożliwe, by różni ludzie mogli napisać identyczną wiadomość.

Ponieważ nie będziemy pamiętać już żadnej innej informacji, tabela będzie wyglądać jak na rysunku 15.10.

### Rysunek 15.10.

*W tabeli `tblKomentarz` możesz przechowywać wiadomości oraz inne dane, które mogą być Ci potrzebne*

tblKomentarz
IDWiadomosci
nazwa
email
wiadomosc

## Zapytanie SQL

Zapytanie SQL, które wybierze i wyświetli całą zawartość księgi gości, ma następującą postać:

```
SELECT * FROM tblKomentarz;
```

Można zdefiniować zestaw rekordów zawierający wszystkie wiadomości, po czym uporządkować je i stworzyć pełną listę wiadomości. Nie rozwiąże to jednak kilku problemów — kolejności, w jakiej mają być ustawione wiadomości, oraz tego, że pewne wiadomości mogą nie być warte pokazania.

Najlepiej sprawdzić ten mechanizm, wprowadzając pewne przykładowe dane do naszej tabeli. Chcę zachować kulturę języka, dlatego będę wstawiał do bazy komentarze ocenzone. Gdy będziesz tworzył własną witrynę, możesz rozszerzyć zakres słów dyskwalifikujących daną wiadomość.

```

insert into tblKomentarz values ('', 'John Ray', 'jray@poisontooth.com', 'Najlepsza
➤strona jaką widziałem! Oby tak dalej.');
```

```

insert into tblKomentarz values ('', 'Robin Nest', 'robyn@nospammingme.com'
➤'Nieprawda, jest strasznie durna.');
```

```

insert into tblKomentarz values ('', 'Bad Dude', 'ads@adsrus.com', 'Zapomnij o tej
➤stronie! Odwiedź <a href="http://www.badplacesforyou.com/"> moją </a>');
```

Z tych trzech przykładowych wiadomości tylko pierwsza powinna być wyświetlona.

Druga wiadomość zawiera niepożądane słowo „durna”, chociaż bywają komentarze znacznie bardziej wulgarne. Z kolei trzecia wiadomość zawiera łącze do innej strony HTML spoza Twojego serwisu — nie chcemy, by użytkownik mógł wyświetlić wiadomość z odnośnikiem do swojej strony internetowej.

Filtrowanie wiadomości odbywa się poprzez zablokowanie w zapytaniu słów, których nie życzysz sobie oglądać w wiadomościach.

Wypróbujmy dla tego przykładu następujące zapytanie:

```

mysql> SELECT * FROM tblKomentarz WHERE wiadomosc NOT LIKE "%durn%" \G
***** 1. row *****
IDWiadomosci: 1
nazwa: John Ray
email: jray@poisontooth.com
wiadomosc: Najlepsza strona jaką widziałem! Oby tak dalej.
***** 2. row *****
IDWiadomosci: 3
Name: Bad Dude
Email: ads@adsrus.com
Message: Zapomnij o tej stronie! Odwiedź <a href=
➤"http://www.badplacesforyou.com/"> moją </a>
2 rows in set (0,01 sec)
```

Wiadomość została przefiltrowana dzięki użyciu konstrukcji WHERE wiadomosc NOT LIKE "%durn%". Filtr tego zapytania bazuje na wzorcu wziętym z pola wiadomosc. W tym przypadku wzorcem jest słowo durn otoczone czymkolwiek, co jest reprezentowane przez symbol %. Możesz połączyć ze sobą tyle warunków, ile w danym momencie potrzebujesz.

Przykładowo, aby pozbyć się słów takich jak *durne*, *lampa* czy *krzesło*, użyj następującej konstrukcji:

```

mysql> SELECT * FROM tblKomentarz WHERE wiadomosc NOT LIKE "%durn%" AND wiadomosc NOT
➤LIKE "%lampa%" AND wiadomosc NOT LIKE "krzeslo";
```

Jak widzisz, nasze zapytanie trzeba rozszerzyć o nowy warunek. Ponieważ nie chcemy pozwolić na to, by wiadomość zawierała w sobie znaczniki HTML, ostateczna postać zapytania powinna wyglądać następująco:

```

mysql> SELECT * FROM tblKomentarz WHERE wiadomosc NOT LIKE "%durn%" AND wiadomosc NOT
➤LIKE "%<%>" \G
***** 1. row *****
IDWiadomosci: 1
Name: John Ray
Email: jray@poisontooth.com
Message: Najlepsza strona jaką widziałem! Oby tak dalej.
1 row in set (0,07 sec)
```

I to wszystko, co chcieliśmy tutaj osiągnąć. Wiadomość, która będzie zawierać kod HTML, zostanie zablokowana, jeśli będzie zawierać znaki `<` oraz jakiegokolwiek inne znaki pomiędzy nimi i na ich końcach.

Ostatnią czynnością, jaką możesz wykonać, jest dodanie napisu `ORDER BY IDwiadomosci DESC` na końcu kwerendy, żeby najnowsze wiadomości były pokazane jako pierwsze.



Zapytanie zdefiniowane powyżej będzie pasować do czegokolwiek, co przypomina znacznik HTML. Właściwie będzie pasować do wszystkiego, co będzie miało znak `<` na początku oraz `>` na końcu. Nie jest możliwe użycie wszystkich znaczników jako wzorca, możesz jednak użyć niektórych z nich, jak na przykład `<a href>` lub `<img>`.

Pora na stworzenie strony w HTML-u, która umożliwi wykonanie wpisu w księdze gości.

## Tworzenie dokumentu HTML

Podobnie jak w przypadku poprzednich dwóch projektów, pominiemy tworzenie mapy serwisu, ponieważ lista gości będzie zajmować tylko jedną stronę. Nasza strona będzie miała krótki formularz, dzięki któremu będzie można wysłać uwagi oraz same komentarze bezpośrednio pod tym formularzem.

Zacznijmy od stworzenia formularza, do którego będzie można wpisywać dane. Otwórz paletę *Object* i przełącz na *Forms*. Tak jak w każdej stronie zawierającej formularz, musisz najpierw wstawić na stronę obiekt `form` — bez znacznika `form` elementy formularza nie będą dostępne. Dokument będzie wyglądał dobrze w oknie projektu, lecz nie pojawi się w oknie przeglądarki. Wstaw nowy formularz do dokumentu i nazwij go `frmKomentarz` — zawartość tego formularza będziesz zachowywał w bazie danych. Aby nazwać formularz, wykonaj następujące czynności:

1. Otwórz paletę *Properties*.
2. Kliknij znacznik `form`, znajdujący się na dole dokumentu w oknie projektu, aby wybrać formularz.
3. Zmień wartość w polu *Form Name* na `frmKomentarz`.

Następnie wstaw takie elementy formularza, jakie są Ci potrzebne. Najpierw wstaw tabelę, aby mieć właściwy układ strony. Wstawimy trzy pola z następującymi nazwami:

- nazwa
- email
- wiadomosc

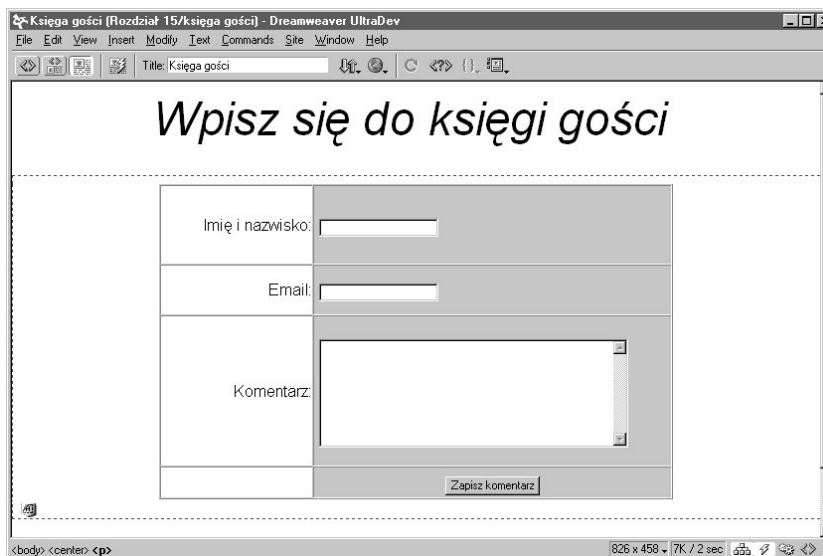
Pola te dodajemy przy użyciu pól tekstowych dostępnych w sekcji *Forms* na palecie *Object*. Pola `nazwa` oraz `email` są zwykłymi polami tekstowymi (UltraDev wstawia je jako standard), natomiast pole `wiadomosc` musisz dostosować tak, by można było wpisywać więcej niż jedną linię tekstu. W tym celu wybierz pole `wiadomosc` w projekcie dokumentu i otwórz paletę *Properties*. Kliknij przycisk *Multiline*, by zmienić typ pola.



Dla zaspokojenia Twojej ciekawości dodam, iż wieloliniowe pole tekstowe nie jest w rzeczywistości związane z polem tekstowym jednoliniowym czy też polem z hasłem. Jest to zupełnie inny obiekt formularza, zwany *textarea*, który może zawierać wiele linii tekstu.

Na koniec wstawimy do formularza przycisk *Submit*. Kliknięcie tego przycisku zachowa w bazie danych nowy komentarz. Ostateczna wersja formularza przedstawiona jest na rysunku 15.11.

**Rysunek 15.11.**  
Jest to przykład formularza pozwalającego zamieszczać uwagi



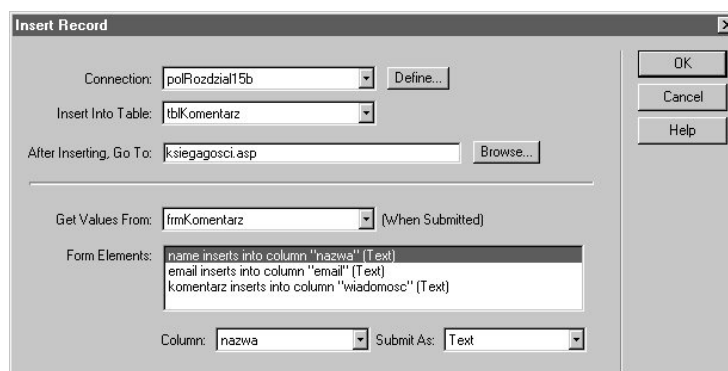
Ostatnim czynnością jest dodanie behawiora *Insert Record* z palety *Server Behavior*. W ten sposób dodamy nowy komentarz do bazy danych.

1. Otwórz paletę *Server Behavior*.
2. Kliknij przycisk + i wybierz *Insert Record*.
3. Z listy rozwijanej *Connection* wybierz połączenie zdefiniowane dla tego projektu.
4. Z listy rozwijanej *Table to Update* wybierz tabelę *tblKomentarz*.
5. W części *Get Values From* wybierz formularz *frmKomentarz*.
6. Każdy element formularza musi mieć swój odpowiednik w bazie danych. Jeśli nie będą pasować, wybierz jeden z elementów formularza i dobrać do niego odpowiednią kolumnę w bazie oraz typ danych.
7. Na koniec, gdy zakończysz wstawianie, wróć do swojej strony. Nazwij ją *listagosci.asp*.

Odpowiednio zdefiniowane parametry behawiora są przedstawione na rysunku 15.12.

Pozostaje nam jedynie dodanie zestawu rekordów w celu wyświetlenia dostępnych komentarzy oraz samo ich wyświetlenie. Kod zapytania SQL mamy już gotowy, wystarczy więc tylko umieścić go na stronie.

**Rysunek 15.12.**  
Połącz elementy formularza z odpowiednimi polami w bazie danych



1. Otwórz paletę *Server Behavior*.
2. Kliknij przycisk + i wybierz *Recordset*.
3. Kliknij przycisk *Advanced*, aby wybrać tryb zaawansowany, o ile aktualnie w nim nie pracujesz.
4. Wybierz odpowiednie połączenie z bazą danych.
5. Nazwij nowy zestaw rekordów `rsComment`.
6. Wpisz następujący kod SQL w oknie zapytania: `SELECT * FROM tblKomentarz WHERE wiadomosc NOT LIKE '%durn%' AND wiadomosc NOT LIKE '%<>%' ORDER BY IDwiadomosci DESC`.
7. Kliknij przycisk *OK*.

Zauważyłeś jakieś zmiany? Jeśli tak, to dobrze, bo kod się zmienił. Zamiast napisu `%<>%` użyliśmy `%<>`. Możesz pozostawić pierwszą wersję zapytania, o ile nie używasz ASP. Problem jest taki, że zapis `<%` oraz `>%` reprezentuje osadzone polecenie ASP i jeśli wstawisz zapytanie w poprzedniej jego postaci, system się pogubi, ponieważ w tym momencie sam złamiesz regułę, którą właśnie implementujesz w dokumencie.

Po wstawieniu zmodyfikowanego behaviora musisz ręcznie edytować kod tak, aby był poprawny. Nie możesz wstawić znaku `%` tam, gdzie byś chciał, ponieważ UltraDev rozpozna go jako znacznik ASP. Musisz zachować go w oddzielnej zmiennej, a później wstawić tę zmienną do zapytania.

Otwórz swój dokument i znajdź fragment kodu, który wygląda następująco:

```
<%
set rsComment = Server.CreateObject("ADODB.Recordset")
rsComment.ActiveConnection = MM_connChapter15_String
rsComment.Source = "SELECT * FROM tblKomentarz WHERE wiadomosc NOT LIKE '%durn%' AND
➤wiadomosc NOT LIKE '%<>%' ORDER BY IDwiadomosci DESC"
rsComment.CursorType = 0
rsComment.CursorLocation = 2
rsComment.LockType = 3
rsComment.Open
rsComment_numRows = 0
%>
```



Pamiętaj, że jeśli masz kłopoty ze znalezieniem kodu w dokumencie, możesz użyć funkcji *Find* z menu *Edit*.

To jest fragment tworzący zestaw rekordów i zapytanie. Zmień ten kod, używając nowej zmiennej *pc*, którą ręcznie włączysz do treści zapytania:

```
<%
set rsComment = Server.CreateObject("ADODB.Recordset")
rsComment.ActiveConnection = MM_connChapter15_String
pc = ""
rsComment.Source = "SELECT * FROM tblKomentarz WHERE wiadomosc NOT LIKE '%durn%' AND
➤wiadomosc NOT LIKE '%<"+pc+">%' ORDER BY IDWiadomosci DESC"
rsComment.CursorType = 0
rsComment.CursorLocation = 2
rsComment.LockType = 3
rsComment.Open
rsComment_numRows = 0
%>
```

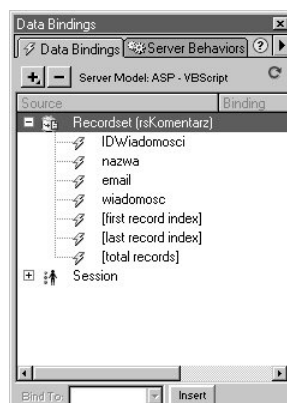
W wielu projektach będziesz musiał ręcznie edytować kod programu — przyzwyczaj się więc do sprawnego poruszania się po nim w UltraDev. Czasem tylko w ten sposób będziesz mógł sprawić, by projekt zaczął działać.

Po dokonaniu zmian zamknij okno HTML. To wszystko, co trzeba wykonać poza interfejsem UltraDev.

W dalszej kolejności zdefiniujemy układ, który będzie zastosowany przy wyświetlaniu komentarzy. Mój jest zwykłą tabelą zawierającą dwa wiersze — pierwszy to treść komentarza, drugi zawiera dane autora. Adres e-mail (w mojej wersji) jest przechowywany tylko dla administratora i nie jest widoczny w książce gości.

Otwórz okno *Data Bindings* i wyświetl zestaw rekordów *rsKomentarz*. Jest on przedstawiony na rysunku 15.13.

**Rysunek 15.13.**  
Twoja paleta  
*Data Bindings*  
powinna wyglądać  
tak jak ta

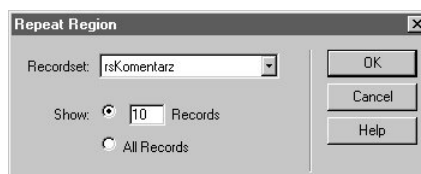


Przecignij pole *wiadomosc* oraz *nazwa* z okna *Data Bindings* do naszej strony. W ten sposób będziemy mogli wyświetlić tylko pierwszy rekord z otrzymanego zestawu rekordów, a co z pozostałymi?

Aby wyświetlić wszystkie rekordy, musisz dodać behavior serwera *Repeat Region*.

1. Wybierz dane, do których chcesz zastosować tę funkcję.
2. Otwórz paletę *Server Behavior*.
3. Kliknij przycisk + i wybierz *Repeat Region*.
4. W oknie konfiguracyjnym *Repeat Region*, widocznym na rysunku 15.14, wybierz zestaw rekordów *rsComment*.

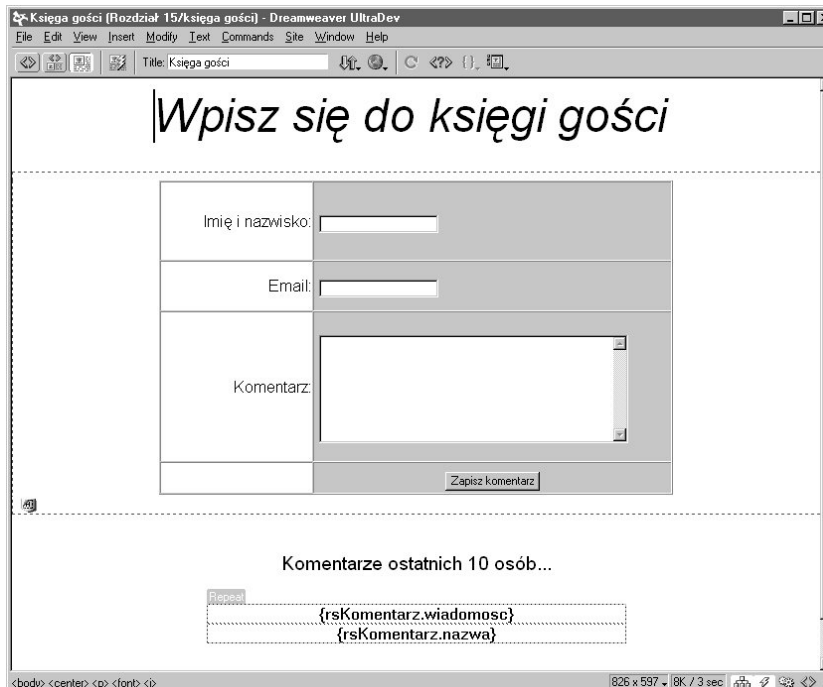
**Rysunek 15.14.**  
Wybierz liczbę rekordów do wyświetlenia



5. Ustaw maksymalną liczbę wiadomości (powtórzeń), które chcesz wyświetlić.
6. Kliknij przycisk *OK*.

W ten sposób zakończyliśmy wszystkie projekty przewidziane w tym rozdziale. Ostateczny wygląd projektu naszego dokumentu jest przedstawiony na rysunku 15.15. Przetestuj nowy system — spróbuj dodać kilka komentarzy do bazy danych. Równocześnie sprawdź opcje filtrowania wiadomości. Właściwie teraz należałoby wstawić słowa, które rzeczywiście chciałbyś usunąć z otrzymywanych wiadomości.

**Rysunek 15.15.**  
Książka gości powinna wyglądać następująco





## Podsumowanie

Projekty zamieszczone w tym rozdziale mogą być umieszczone na istniejących statycznych stronach internetowych, podnosząc ich funkcjonalność i atrakcyjność.

Zastosowanie funkcji SQL powoduje znaczny wzrost elastyczności pracy w programie UltraDev. Póki co, UltraDev jest narzędziem, które pozwala wstawiać informacje do bazy danych i wyświetlać te, które są w niej przechowywane. Od Ciebie zależy, w jaki sposób połączysz możliwości oferowane przez środowisko UltraDev oraz serwer bazy danych.

Niestety, dowiedziałeś się również, że UltraDev ma swoje wady. Ręczna edycja kodu źródłowego po zdefiniowaniu złożonego zapytania jest powszechna, ale zwykle nie jest trudna. Umiejętność znalezienia miejsca, w którym znajduje się kod źródłowy dostępu do bazy danych, jest kluczem do znajdowania i rozwiązywania problemów.

W następnym rozdziale zajmiemy się definiowaniem i tworzeniem kompletnej aplikacji internetowej. Nie będzie to proste zadanie, lecz będę starał się wytłumaczyć Ci to jak najlepiej.

## Pytania i odpowiedzi

- P: Czy banery reklamowe można skonfigurować tak, by ten sam nie był pokazywany dwa razy z rzędu?**
- O:** Dzięki użyciu *cookies* przeglądarka będzie pamiętać (i przekazywać do serwera), który pasek wyświetlany był jako ostatni. W rozdziale 17. znajdziesz więcej informacji na temat wykorzystania *cookies*.
- P: Czy można automatycznie wyświetlać obrazki okresowe w kolejnych latach?**
- O:** W przedstawionym przykładzie dana tematyka traci ważność po upływie określonej daty i nie można jej użyć ponownie. Jednakże, jak powiedziano w tym rozdziale, MySQL udostępnia funkcje do porównywania danych w oparciu o miesiąc bądź dzień. Każda z nich może działać przez wiele lat.
- P: Czy muszę modyfikować zapytanie `%<%>%` na platformach serwera innych niż ASP?**
- O:** ASP używa znacznika `<%>` na oznaczenie kodu wbudowanego — i stąd problem z programem UltraDev. Jeśli używasz wbudowanego języka, który w odmienny sposób przedstawia swoje znaczniki, nie powinieneś mieć problemów.
- P: Czy mogę rozszerzyć listę dyskusyjną w pełny system komunikacyjny?**
- O:** Poczekaj do następnego rozdziału, gdyż właśnie tym będziemy się zajmować.