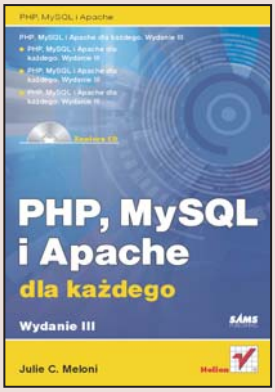
PHP, MySQL i Apache dla każdego. Wydanie III



Autor: Julie C. Meloni
Tłumaczenie: Jarosław Dobrzański na podstawie PHP, MySQL i Apache dla każdego. Wydanie II w tłumaczeniu Adama Byrtka i Jarosława Dobrzańskiego
ISBN: 83-246-0773-0
Tytuł oryginału: Sams Teach Yourself PHP, MySQL and Apache All in One (3rd Edition)
Format: B5, stron: 600

Witryny WWW korzystające z baz danych i wyświetlające dynamicznie generowaną treść stają się coraz bardziej popularne. Technologie skryptowe działające po stronie serwera i mechanizmy bazodanowe przestały być cechą charakterystyczną portali i sklepów – dziś wykorzystywane są również w firmowych i prywatnych stronach WWW, galeriach fotografii oraz blogach. Na rynku istnieje wiele technologii stosowanych do budowania dynamicznych witryn WWW, lecz największym uznaniem cieszy się duet PHP i MySQL uruchamiany na serwerze WWW Apache.

Książka „PHP, MySQL i Apache dla każdego. Wydanie III" to wprowadzenie do tworzenia witryn WWW z wykorzystaniem tej właśnie technologii i najnowszych wersji wchodzących w jej skład narzędzi PHP 5 i MySQL 5. Przedstawiono w niej proces instalacji serwera Apache, bazy danych MySQL, interpretera PHP na serwerze i stacji roboczej oraz zasady programowania w PHP i pobierania danych z tabel. Znajdziesz tu informacje o operacjach na plikach, generowaniu grafiki, zarządzaniu sesjami, tworzeniu zapytań do bazy MySQL i optymalizowaniu wydajności aplikacji. Dzięki zawartym w książce i doskonale omówionym gotowym projektom poznasz technologie PHP i MySQL w praktyce.

- Instalacja i konfiguracja PHP, MySQL i Apache w Windows i Linux
- Struktura języka PHP
- Programowanie obiektowe
- Przetwarzanie danych z formularzy HTML
- Operacje na plikach i katalogach
- Projektowanie baz danych i tabel
- Połączenie skryptów PHP z bazą danych MySQL
- Uwierzytelnianie użytkowników
- Korzystanie z plików XML
- Zabezpieczanie aplikacji sieciowych

Poznaj najnowsze wersje najpopularniejszych narzędzi do tworzenia dynamicznych witryn WWW

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

Spis treści

O autorach **17**

Wprowadzenie **19**

Część I Podstawy **23**

Rozdział 1. Szybka instalacja **25**

Instalacja w systemie Linux/Unix 25
Instalacja MySQL 26
Instalacja Apache 26
Instalacja PHP 28
Instalacja w systemie Windows 29
Instalacja MySQL 29
Instalacja Apache 30
Instalacja PHP 32
Rozwiązywanie problemów 34

Rozdział 2. Instalacja i konfiguracja MySQL **35**

Wersja aktualna i przyszłe wersje MySQL 35
Jak zdobyć MySQL? 36
Instalacja MySQL w systemie Linux/Unix 36
Instalacja MySQL w systemie Windows 37
Rozwiązywanie problemów instalacji 44
Podstawy bezpieczeństwa 44
Uruchamianie MySQL 45
Zabezpieczanie połączenia MySQL 45
Wprowadzenie do systemu uprawnień MySQL 46
Dwustopniowa autoryzacja 47
Korzystanie z systemu uprawnień 48
Dodawanie użytkowników 48
Usuwanie uprawnień 50
Podsumowanie 50
Pytania i odpowiedzi 51
Warsztaty 51
Test 52
Odpowiedzi 52
Ćwiczenia 52

Rozdział 3. Instalacja i konfiguracja Apache 53
Wersja aktualna i przyszłe wersje Apache 53
Wybór sposobu instalacji 54
Kompilacja kodu źródłowego 54
Instalacja dystrybucji binarnej 54
Instalacja Apache w systemie Linux/Unix 55
Pobieranie kodu źródłowego 55
Rozpakowanie kodu źródłowego 55
Przygotowania do kompilacji Apache 56
Budowanie i instalacja Apache 56
Instalacja Apache w systemie Windows 57
Format pliku konfiguracyjnego Apache 60
Dyrektywy 60
Pojemniki 61
Instrukcje warunkowe 62
Dyrektywa ServerRoot 63
Pliki konfiguracyjne dla katalogów 64
Pliki dziennika Apache 65
access_log 65
error_log 65
Pozostałe pliki 66
Polecenia związane z Apache 66
Serwer Apache 66
Skrypt kontrolny Apache 67
Pierwsze uruchomienie Apache 68
Sprawdzanie pliku konfiguracyjnego 68
Uruchamianie Apache 69
Rozwiązywanie problemów 70
Inny serwer WWW 70
Brak uprawnień do portu 70
Dostęp zabroniony 70
Złe parametry grupy 70
Podsumowanie 71
Pytania i odpowiedzi 71
Warsztaty 71
Test 72
Odpowiedzi 72
Ćwiczenia 72

Rozdział 4. Instalacja i konfiguracja PHP 73
Wersja aktualna i przyszłe wersje PHP 73
Kompilacja PHP w systemie Linux/Unix 74
Dodatkowe opcje konfiguracyjne w systemie Linux/Unix 76
Integracja PHP z Apache w systemie Linux/Unix 76
Instalacja PHP w systemie Windows 77
Integracja PHP z Apache w systemie Windows 77
Plik php.ini 78
Testowanie 79
Gdzie znaleźć pomoc 80
Podstawy skryptów PHP 81
Początek i koniec bloku instrukcji PHP 81
Instrukcja echo i funkcja print() 83
Łączenie HTML i PHP 84
Komentarze w kodzie PHP 85

Podsumowanie ... 86
Pytania i odpowiedzi ... 86
Warsztaty ... 87
Test ... 87
Odpowiedzi ... 87
Ćwiczenia ... 87

Część II Struktura języka PHP ... 89

Rozdział 5. Podstawowe elementy języka PHP ... 91
Zmienne ... 92
Zmienne globalne i superglobalne ... 93
Typy danych ... 94
Zmiana typu za pomocą settype() ... 96
Zmiana typu poprzez rzutowanie ... 97
Po co sprawdzać typ? ... 99
Operatory i wyrażenia ... 99
Operator przypisania ... 100
Operatory arytmetyczne ... 101
Operator konkatenacji ... 101
Złożony operator przypisania ... 102
Inkrementacja i dekrementacja wartości zmiennej całkowitej ... 103
Operatory porównania ... 104
Tworzenie złożonych wyrażeń za pomocą operatorów logicznych ... 105
Kolejność operatorów ... 106
Stałe ... 107
Stałe predefiniowane ... 108
Podsumowanie ... 108
Pytania i odpowiedzi ... 108
Warsztaty ... 109
Test ... 109
Odpowiedzi ... 110
Ćwiczenia ... 110

Rozdział 6. Sterowanie przepływem w PHP ... 111
Zmiana przepływu ... 112
Instrukcja if ... 112
Użycie klauzuli else w instrukcji if ... 113
Użycie klauzuli elseif w instrukcji if ... 113
Instrukcja switch ... 115
Operator ? ... 116
Pętle ... 117
Instrukcja while ... 117
Instrukcja do…while ... 118
Instrukcja for ... 119
Przerywanie pętli za pomocą instrukcji break ... 120
Pomijanie iteracji za pomocą instrukcji continue ... 122
Zagnieżdżone pętle ... 123
Bloki kodu PHP ... 125
Podsumowanie ... 125
Pytania i odpowiedzi ... 127
Warsztaty ... 127
Test ... 127
Odpowiedzi ... 127
Ćwiczenie ... 128

Rozdział 7. Funkcje ... 129
Czym jest funkcja? ... 129
Wywoływanie funkcji ... 130
Definiowanie funkcji ... 131
Zwracanie wartości przez funkcje użytkownika ... 134
Zasięg zmiennych ... 135
Wywoływanie zmiennych za pomocą instrukcji global ... 136
Przechowywanie wartości pomiędzy wywołaniami funkcji za pomocą instrukcji static ... 138
Więcej o argumentach ... 140
Przypisywanie zmiennym wartości domyślnych ... 140
Przekazywanie zmiennych przez referencję ... 142
Sprawdzanie istnienia funkcji ... 143
Podsumowanie ... 145
Pytania i odpowiedzi ... 145
Warsztaty ... 146
Test ... 146
Odpowiedzi ... 146
Ćwiczenie ... 147

Rozdział 8. Tablice ... 149
Czym jest tablica? ... 149
Tworzenie tablic ... 150
Tworzenie tablic asocjacyjnych ... 151
Tworzenie tablic wielowymiarowych ... 151
Niektóre funkcje operujące na tablicach ... 153
Podsumowanie ... 155
Pytania i odpowiedzi ... 155
Warsztaty ... 155
Test ... 155
Odpowiedzi ... 155
Ćwiczenia ... 156

Rozdział 9. Obiekty ... 157
Tworzenie obiektu ... 157
Własności obiektów ... 159
Metody obiektów ... 160
Konstruktor ... 162
Dziedziczenie ... 162
Podsumowanie ... 164
Pytania i odpowiedzi ... 164
Warsztaty ... 164
Test ... 164
Odpowiedzi ... 165
Ćwiczenia ... 165

Część III Pierwsze kroki z kodem ... 167

Rozdział 10. Ciągi znaków, data i czas ... 169
Formatowanie ciągów znaków ... 170
Funkcja printf() ... 170
Zamiana argumentów ... 176
Przechowywanie sformatowanych ciągów znaków ... 178

Analizowanie ciągów znaków ... 178
Uwaga na temat indeksowania ciągów znaków ... 179
Sprawdzanie długości ciągu za pomocą funkcji strlen() ... 179
Znajdowanie podciągu za pomocą strstr() ... 179
Określanie pozycji podciągu za pomocą funkcji strpos() ... 180
Pobieranie części ciągu za pomocą funkcji substr() ... 181
Podział ciągu na słowa za pomocą funkcji strtok() ... 181
Operacje na ciągach znaków ... 183
Oczyszczanie ciągu za pomocą funkcji trim(), ltrim(), rtrim() i strip_tags() ... 183
Zmiana fragmentu ciągu za pomocą funkcji substr_replace() ... 184
Zamiana podciągów za pomocą funkcji str_replace() ... 185
Zmiana wielkości liter ... 185
Zawijanie tekstu za pomocą funkcji wordwrap() i nl2br() ... 187
Dzielenie ciągów za pomocą funkcji explode() ... 188
Funkcje operujące na dacie i czasie ... 189
Pobieranie bieżącej daty za pomocą funkcji time() ... 189
Konwersja znacznika czasu za pomocą funkcji getdate() ... 189
Formatowanie znacznika czasu za pomocą funkcji date() ... 190
Tworzenie znacznika czasu za pomocą funkcji mktime() ... 193
Weryfikacja daty za pomocą funkcji checkdate() ... 194
Inne funkcje operujące na ciągach znaków, datach i czasie ... 194
Podsumowanie ... 194
Warsztaty ... 195
Pytania i odpowiedzi ... 195
Test ... 195
Odpowiedzi ... 196
Ćwiczenia ... 196

Rozdział 11. Formularze ... 197

Tworzenie prostego formularza ... 197
Przekazywanie informacji w tablicach ... 199
Łączenie kodu HTML i PHP w jednym skrypcie ... 202
Zapisywanie informacji o stanie w ukrytym polu ... 204
Przekierowania ... 206
Wysyłanie poczty elektronicznej ... 207
Konfiguracja systemu ... 207
Tworzenie formularza ... 208
Skrypt wysyłający wiadomość ... 209
Formatowanie wiadomości za pomocą HTML ... 211
Przesyłanie plików ... 213
Tworzenie formularza wysyłającego plik ... 213
Skrypt obsługujący przesłany plik ... 214
Podsumowanie ... 217
Warsztaty ... 217
Test ... 217
Odpowiedzi ... 217
Ćwiczenia ... 218

Rozdział 12. Cookies i sesje ... 219

Wprowadzenie do cookies ... 219
Anatomia cookie ... 220
Tworzenie cookie ... 221
Usuwanie cookie ... 222

Wprowadzenie do sesji ... 223
Otwieranie sesji ... 223
Zmienne sesyjne ... 224
Przekazywanie identyfikatora sesji w adresie ... 228
Niszczenie sesji i usuwanie zmiennych ... 229
Zastosowania sesji ... 229
Obsługa zarejestrowanych użytkowników ... 230
Obsługa ustawień użytkownika ... 230
Podsumowanie ... 230
Pytania i odpowiedzi ... 231
Warsztaty ... 231
Test ... 231
Odpowiedzi ... 231
Ćwiczenie ... 232

Rozdział 13. Pliki i katalogi ... 233

Dołączanie plików za pomocą funkcji include() ... 233
Zwracanie wartości z dołączonego dokumentu ... 235
Instrukcja include() wewnątrz struktur sterujących ... 235
Użycie include_once() ... 236
Dyrektywa include_path ... 237
Weryfikacja plików ... 238
Sprawdzanie, czy dany plik istnieje, za pomocą funkcji file_exists() ... 238
Plik czy katalog? ... 238
Sprawdzanie uprawnień pliku ... 238
Sprawdzanie rozmiaru pliku za pomocą funkcji filesize() ... 239
Pobieranie informacji o datach związanych z plikiem ... 239
Funkcja wyświetlająca informacje o pliku ... 240
Tworzenie i usuwanie plików ... 242
Otwieranie plików do zapisu, odczytu i dopisywania ... 242
Odczytywanie danych z pliku ... 243
Odczytywanie wierszy za pomocą funkcji fgets() i feof() ... 243
Odczytywanie określonej ilości danych za pomocą funkcji fread() ... 245
Odczytywanie kolejnych znaków za pomocą funkcji fgetc() ... 247
Zapisywanie i dopisywanie danych do pliku ... 247
Zapisywanie danych do pliku za pomocą funkcji fwrite() i fputs() ... 248
Blokowanie plików za pomocą funkcji flock() ... 249
Operacje na katalogach ... 250
Tworzenie katalogów za pomocą funkcji mkdir() ... 250
Usuwanie katalogu za pomocą rmdir() ... 250
Otwieranie katalogu za pomocą funkcji opendir() ... 251
Odczytywanie zawartości katalogu za pomocą funkcji readdir() ... 251
Otwieranie potoków do i z procesów za pomocą funkcji popen() ... 253
Uruchamianie poleceń za pomocą funkcji exec() ... 255
Uruchamianie poleceń funkcjami system() i passthru() ... 256
Podsumowanie ... 258
Pytania i odpowiedzi ... 259
Warsztaty ... 259
Test ... 259
Odpowiedzi ... 260
Ćwiczenia ... 260

Rozdział 14. Obrazki 261
Proces powstawania obrazka 261
Kilka słów o kolorze 261
Konieczne zmiany w PHP 262
Pobieranie dodatkowych bibliotek 262
Rysowanie nowego obrazka 263
Rysowanie kształtów i linii 263
Wypełnianie kształtów kolorem 265
Rysowanie wykresów 266
Modyfikacja istniejących obrazków 270
Tworzenie obrazków na podstawie danych przesłanych przez użytkownika 273
Podsumowanie 276
Pytania i odpowiedzi 277
Warsztaty 277
Test 278
Odpowiedzi 278
Ćwiczenie 278

Część IV Integracja PHP i MySQL 279

Rozdział 15. Tajniki procesu projektowania bazy danych 281
Rola dobrego projektu bazy danych 281
Typy relacji między tabelami 282
Relacje jeden do jednego 283
Relacje jeden do wielu 283
Relacje wiele do wielu 284
Normalizacja 285
Problemy z tabelą prostą 286
Pierwsza postać normalna 286
Druga postać normalna 287
Trzecia postać normalna 287
Postępowanie zgodnie z procesem projektowania 288
Podsumowanie 289
Pytania i odpowiedzi 290
Warsztaty 290
Test 290
Odpowiedzi 290
Ćwiczenie 290

Rozdział 16. Podstawowe polecenia SQL 291
Typy danych w MySQL 292
Liczbowe typy danych 292
Typy czasu i daty 293
Typy łańcuchowe 294
Składnia tworzenia tabel 295
Używanie polecenia INSERT 296
Bliższe spojrzenie na INSERT 296
Stosowanie polecenia SELECT 298
Porządkowanie wyników zwracanych przez SELECT 299
Ograniczanie wyników 300
Używanie WHERE w zapytaniach 300
Stosowanie operatorów w klauzuli WHERE 301
Porównywanie łańcuchów za pomocą LIKE 302

Selekcja z kilku tabel ... 302
Używanie JOIN ... 304
Stosowanie podselekcji ... 306
Modyfikowanie rekordów za pomocą polecenia UPDATE ... 307
Warunkowe instrukcje UPDATE ... 309
Stosowanie bieżących wartości kolumn z UPDATE ... 309
Używanie polecenia REPLACE ... 310
Stosowanie polecenia DELETE ... 311
Warunkowa instrukcja DELETE ... 312
Często stosowane funkcje MySQL operujące na ciągach tekstowych ... 313
Funkcje długości i konkatenacji ... 314
Funkcje przycinające i dopełniające ... 316
Funkcje lokalizacji i pozycji ... 317
Funkcje operujące na podciągach ... 318
Funkcje modyfikujące ciągi ... 319
Korzystanie z funkcji daty i czasu w MySQL ... 320
Operowanie na dniach ... 320
Operowanie na miesiącach i latach ... 323
Operowanie na tygodniach ... 324
Operowanie na godzinach, minutach i sekundach ... 325
Formatowanie daty i czasu w MySQL ... 326
Działania arytmetyczne na datach w MySQL ... 328
Funkcje specjalne i możliwości w zakresie konwersji ... 330
Podsumowanie ... 332
Pytania i odpowiedzi ... 333
Warsztaty ... 334
Test ... 334
Odpowiedzi ... 335
Ćwiczenie ... 335

Rozdział 17. Transakcje i procedury składowane w MySQL ... 337
Czym są transakcje? ... 337
Podstawowe elementy składni ... 338
Praktyczny przykład zastosowania transakcji ... 339
Czym są procedury składowane? ... 341
Podstawowa składnia procedur składowanych ... 341
Podsumowanie ... 343
Warsztaty ... 343
Test ... 343
Odpowiedzi ... 343

Rozdział 18. Interakcja z MySQL z poziomu PHP ... 345
Funkcje MySQL a funkcje MySQLi ... 345
Łączenie się z MySQL poprzez PHP ... 345
Nawiązywanie połączenia ... 346
Wykonywanie zapytań ... 347
Odbieranie komunikatów o błędach ... 348
Operowanie na danych z bazy MySQL ... 349
Wstawianie danych z poziomu PHP ... 349
Pobieranie danych z bazy w PHP ... 353
Pozostałe funkcje MySQL w PHP ... 354
Podsumowanie ... 355
Warsztaty ... 355
Test ... 356
Odpowiedzi ... 356

Część V Proste projekty ... 357

Rozdział 19. Zarządzanie prostą listą mailingową ... 359
Opracowywanie mechanizmu subskrypcji ... 359
Tworzenie tabeli subskrybentów ... 360
Tworzenie formularza subskrypcji ... 360
Budowa mechanizmu mailingu ... 367
Podsumowanie ... 371
Pytania i odpowiedzi ... 371
Warsztaty ... 372
Test ... 372
Odpowiedzi ... 372

Rozdział 20. Tworzenie internetowej książki adresowej ... 373
Planowanie i tworzenie tabel w bazie danych ... 373
Tworzenie menu ... 376
Tworzenie mechanizmu dodawania rekordów ... 376
Przeglądanie rekordów ... 381
Tworzenie mechanizmu usuwania rekordów ... 389
Uzupełnianie istniejących rekordów ... 391
Podsumowanie ... 396
Warsztaty ... 396
Test ... 397
Odpowiedzi ... 397
Ćwiczenia ... 398

Rozdział 21. Tworzenie prostego forum dyskusyjnego ... 399
Projektowanie tabel w bazie danych ... 399
Tworzenie formularzy wprowadzania danych i skryptów ... 400
Wyświetlanie listy tematów ... 404
Wyświetlanie postów w temacie ... 407
Dodawanie postu w wybranym temacie ... 412
Podsumowanie ... 415
Pytania i odpowiedzi ... 415
Warsztaty ... 416
Test ... 416
Odpowiedzi ... 416
Ćwiczenie ... 416

Rozdział 22. Tworzenie witryny sklepu internetowego ... 417
Planowanie i tworzenie tabel w bazie danych ... 417
Wstawianie rekordów do tabeli sklep_kategorie ... 419
Wstawianie rekordów do tabeli sklep_artykuly ... 420
Wstawianie rekordów do tabeli sklep_art_rozmiar ... 420
Wstawianie rekordów do tabeli sklep_art_kolor ... 421
Wyświetlanie kategorii artykułów ... 421
Wyświetlanie artykułów ... 424
Podsumowanie ... 427
Warsztaty ... 428
Test ... 428
Odpowiedzi ... 428

Rozdział 23. Tworzenie mechanizmu koszyka z zakupami 429
Planowanie i tworzenie tabel 429
Integracja koszyka z witryną sklepową 431
Dodawanie artykułów do koszyka 434
Przeglądanie zawartości koszyka 436
Usuwanie artykułów z koszyka 438
Sposoby dokonywania płatności i sekwencja kasowa 439
Tworzenie formularza kasowego 440
Realizowanie czynności kasowych 440
Podsumowanie 441
Warsztaty 441
Test 442
Odpowiedzi 442

Rozdział 24. Tworzenie prostego kalendarza 443
Tworzenie prostego kalendarza wyświetlanego na ekranie 443
Sprawdzenie danych przesłanych przez użytkownika 444
Tworzenie formularza HTML 445
Tworzenie tabeli kalendarza 446
Dodawanie terminów do kalendarza 450
Tworzenie biblioteki kalendarza 458
Podsumowanie 463
Pytania i odpowiedzi 464
Warsztaty 464
Test 464
Odpowiedzi 464
Ćwiczenie 464

Rozdział 25. Ograniczanie dostępu do aplikacji 465
Istota uwierzytelniania 465
Uwierzytelnianie klienta 466
Możliwości funkcjonalne modułu uwierzytelniającego serwera Apache 467
Uwierzytelnianie bazujące na plikach 468
Kontrola dostępu bazująca na pliku bazy danych 470
Apache jako narzędzie kontroli dostępu 471
Wprowadzanie reguł dostępu 471
Interpretacja reguł dostępu 473
Wiązane zastosowanie metod kontroli dostępu 474
Ograniczenie dostępu na podstawie metod HTTP 475
Ograniczenie dostępu na podstawie wartości cookies 476
Tworzenie tabeli uprawnionych użytkowników 476
Tworzenie formularza logowania i skryptu 477
Sprawdzanie cookie uwierzytelniającego 479
Podsumowanie 481
Pytania i odpowiedzi 481
Warsztaty 482
Test 482
Odpowiedzi 482
Ćwiczenie 483

Rozdział 26. Monitorowanie i prowadzenie dzienników aktywności serwera 485
Standardowe odnotowywanie dostępu do serwera 485
Ustalanie treści dzienników 486
Odnotowywanie dostępów w plikach 489
Odnotowywanie dostępów w programie 490

Standardowy tryb odnotowywania błędów serwera Apache ... 491
Odnotowywanie błędów w pliku ... 491
Odnotowywanie błędów w programie ... 491
Demon syslog jako argument ... 491
Dyrektywa LogLevel ... 492
Zarządzanie dziennikami serwera Apache ... 492
Ustalanie nazw hostów ... 493
Rotacja dzienników ... 493
Łączenie i podział dzienników ... 494
Analiza dzienników ... 494
Monitorowanie dzienników błędów ... 494
Odnotowywanie informacji w bazie danych ... 495
Tworzenie tabeli w bazie danych ... 495
Tworzenie skryptu PHP odnotowującego dane ... 495
Tworzenie przykładowych raportów ... 496
Podsumowanie ... 499
Pytania i odpowiedzi ... 500
Warsztaty ... 500
Test ... 500
Odpowiedzi ... 500

Rozdział 27. Lokalizacja aplikacji ... 501
Internacjonalizacja i lokalizacja ... 501
Zestawy znaków ... 502
Modyfikacje środowiska ... 503
Zmiany w konfiguracji serwera Apache ... 503
Zmiany w konfiguracji PHP ... 504
Zmiany w konfiguracji MySQL ... 504
Tworzenie zlokalizowanej struktury strony ... 505
Podsumowanie ... 510
Pytania i odpowiedzi ... 511
Warsztaty ... 511
Test ... 511

Rozdział 28. Korzystanie z XML ... 513
Co to jest XML? ... 513
Podstawowa struktura dokumentu XML ... 513
Dostęp do dokumentów XML z poziomu PHP za pomocą funkcji modelu DOM ... 515
Dostęp do danych XML z poziomu PHP za pomocą funkcji SimpleXML ... 517
Podsumowanie ... 520
Warsztaty ... 521
Test ... 521
Odpowiedzi ... 521

Część VI Administrowanie i dostrajanie ... 523

Rozdział 29. Poprawianie wydajności i wirtualny hosting na serwerze Apache ... 525
Kwestie skalowalności ... 526
Ograniczenia systemu operacyjnego ... 526
Ustawienia serwera Apache związane z wydajnością ... 528
Testowanie serwera pod obciążeniem przy użyciu ApacheBench ... 529
Aktywne dostrajanie wydajności ... 531
Odwzorowywanie plików w pamięci ... 531
Rozkład obciążenia ... 532

Buforowanie ... 532
Redukcja ilości transmitowanych danych ... 532
Ustawienia sieci ... 533
Zapobieganie nadużyciom ... 533
Roboty ... 533
Implementacja wirtualnego hostingu ... 534
Wirtualny hosting bazujący na adresach IP ... 535
Wirtualny hosting bazujący na nazwach ... 535
Masowy hosting wirtualny ... 537
Podsumowanie ... 538
Pytania i odpowiedzi ... 539
Warsztaty ... 540
Test ... 540
Odpowiedzi ... 540

Rozdział 30. Bezpieczny serwer WWW ... 541
Potrzeba bezpieczeństwa ... 541
Protokół SSL ... 542
Rozwiązanie kwestii poufności ... 542
Rozwiązanie kwestii nienaruszalności ... 544
Rozwiązanie kwestii uwierzytelniania ... 544
Uzyskiwanie i instalacja narzędzi SSL ... 547
OpenSSL ... 547
Moduł mod_ssl serwera Apache ... 548
Zarządzanie certyfikatami ... 549
Tworzenie pary kluczy ... 550
Tworzenie prośby o podpisanie certyfikatu ... 551
Tworzenie certyfikatu podpisanego przez nas samych ... 552
Konfiguracja SSL ... 552
Uruchamianie serwera ... 553
Podsumowanie ... 553
Pytania i odpowiedzi ... 553
Warsztaty ... 554
Test ... 554
Odpowiedzi ... 554

Rozdział 31. Optymalizacja i dostrajanie MySQL ... 555
Tworzenie zoptymalizowanej platformy ... 556
Stosowanie funkcji benchmark() ... 556
Opcje inicjalizacyjne MySQL ... 557
Kluczowe parametry startowe ... 558
Optymalizacja struktury tabel ... 559
Optymalizacja zapytań ... 560
Korzystanie z polecenia FLUSH ... 561
Korzystanie z polecenia SHOW ... 562
Pobieranie informacji o bazach danych i tabelach ... 563
Pobieranie informacji o strukturze tabel ... 564
Pobieranie statusu systemu ... 565
Podsumowanie ... 567
Pytania i odpowiedzi ... 568
Warsztaty ... 568
Test ... 568
Odpowiedzi ... 569
Ćwiczenia ... 569

Rozdział 32. Aktualizacja oprogramowania 571
Trzymanie ręki na pulsie 571
Kiedy aktualizować? 572
Aktualizacja MySQL 573
Aktualizacja Apache 573
Modyfikowanie Apache bez dokonywania aktualizacji 574
Aktualizacja PHP 575
Rozszerzanie PHP za pomocą PECL i PEAR 575
Podsumowanie 575
Warsztaty 576
Test 576
Odpowiedzi 576

Skorowidz 577

Rozdział 15.

Tajniki procesu projektowania bazy danych

W tym rozdziale poznamy tok rozumowania prowadzący do stworzenia relacyjnej bazy danych. Po tym skoncentrowanym na teorii rozdziale od razu zagłębimy się w naukę podstawowych poleceń MySQL w ramach przygotowań do zintegrowania bazy MySQL z naszymi aplikacjami.

Zagadnienia omówione w tym rozdziale to:

- Niektóre zalety dobrego projektu bazy danych
- Trzy typy relacji między tabelami
- Jak znormalizować bazę danych
- Jak wdrożyć proces właściwego projektowania baz danych

Rola dobrego projektu bazy danych

Dobry projekt bazy danych jest kluczowym składnikiem wydajnej aplikacji, tak jak opływowa karoseria jest ważną częścią samochodu wyścigowego. Jeżeli samochód nie ma opływowych kształtów, będzie stawiał opór i wolniej jechał. Jeżeli nie zoptymalizujemy relacji, nasza baza nie będzie działać tak wydajnie, jak by mogła. Myślenie o relacjach i wydajności bazy jest elementem *normalizacji*.

Na marginesie

Normalizacja to proces tworzenia takiej struktury danych, w której zminimalizowano powtarzanie i niespójności.

Poza wydajnością istnieje jeszcze kwestia utrzymania — nasza baza powinna być łatwa w utrzymaniu. Sprowadza się to do przechowywania jak najmniejszej ilości powtarzanych danych (albo całkowitej eliminacji powtórzeń). Jeżeli mamy dużo powtarzających się danych i nagle zachodzi zmiana w jednym z wystąpień takich danych (na przykład zmiana nazwiska), to trzeba zmienić wszystkie wystąpienia tej danej. Aby wyeliminować powtarzanie i ułatwić utrzymanie danych, można stworzyć tabelę możliwych wartości i odnosić się do tych wartości poprzez klucz. W ten sposób, jeżeli wartość ulegnie zmianie, to zmianę tę wystarczy wprowadzić tylko raz — w głównej tabeli. Odwołania występujące w innych tabelach pozostają bez zmian.

Przypuśćmy, że jesteśmy odpowiedzialni za utrzymanie bazy danych studentów oraz przedmiotów, na które się zapisali. Jeżeli 35 studentów zapisało się na ten sam przedmiot, niech to będzie „wyższa matematyka", to nazwa tego przedmiotu wystąpi w tabeli 35 razy. Jeżeli wykładowca stwierdzi, że trzeba zmienić nazwę przedmiotu na „matematyka zaawansowana", to musimy wprowadzić zmiany w 35 rekordach. Gdyby baza zbudowana została tak, że nazwy przedmiotów występowałyby w jednej tabeli, a w rekordach studentów występowałby tylko identyfikator przedmiotu, to aby zmienić nazwę przedmiotu, wystarczyłoby dokonać modyfikacji w jednym rekordzie, a nie w 35.

Korzyści z posiadania dobrze zaplanowanej i zaprojektowanej bazy danych są niezliczone. Im więcej pracy włożymy w bazę na początku, tym mniej będziemy mieli do zrobienia później. Przeprojektowywanie bazy danych po publicznym wdrożeniu aplikacji nie jest zbyt dobrym pomysłem (chociaż to się zdarza), a rezultaty są kosztowne.

Tak więc zanim w ogóle zabierzemy się do pisania kodu aplikacji, poświęćmy stosowny czas na zaprojektowanie bazy danych. Reszta tego rozdziału poświęcona jest relacjom i normalizacji — dwóm ważnym elementom bazodanowej układanki.

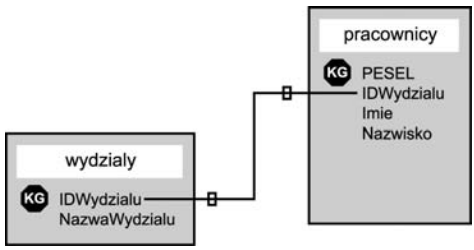
Typy relacji między tabelami

Relacje między tabelami można podzielić na:

- Relacje jeden do jednego
- Relacje jeden do wielu
- Relacje wiele do wielu

Przypuśćmy, że mamy tabelę o nazwie `pracownicy`, która zawiera numer PESEL każdej osoby, jej nazwisko oraz wydział, w którym pracuje. Powiedzmy, że mamy też odrębną tabelę zwaną `wydzialy`, która zawiera listę wszystkich wydziałów składającą się z identyfikatora wydziału i jego nazwy. Pole zawierające identyfikator wydziału w tabeli `pracownicy` odpowiada identyfikatorowi z tabeli `wydzialy`. Tego rodzaju relację przedstawia rysunek 15.1. Oznaczenie KG tuż przy nazwie pola oznacza klucz główny.

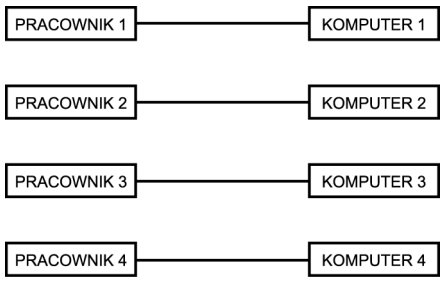
W kolejnych punktach opisano dokładniej każdy z typów relacji.



Rysunek 15.1. *Tabele „pracownicy" i „wydzialy" powiązane poprzez klucz „IDwydzialu"*

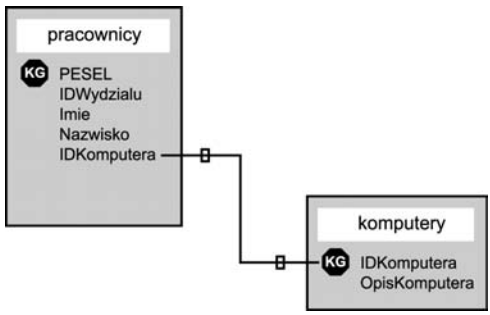
Relacje jeden do jednego

W relacji jeden do jednego klucz pojawia się tylko raz w powiązanej tabeli. Tabel `pracownicy` i `wydzialy` nie łączy relacja jeden do jednego, ponieważ do jednego wydziału niewątpliwie należy wielu pracowników. Relacja jeden do jednego istnieje na przykład wtedy, gdy każdemu pracownikowi jest przypisany jeden komputer. Taką relację pokazuje rysunek 15.2.



Rysunek 15.2. *Każdemu pracownikowi przypisany jest jeden komputer*

Tabele `pracownicy` i `komputery` w naszej bazie wyglądałyby tak, jak przedstawia to rysunek 15.3, który reprezentuje relację jeden do jednego.

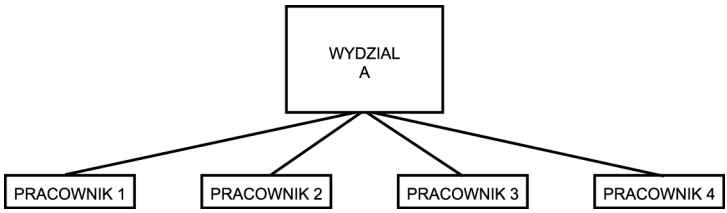


Rysunek 15.3. *Relacja jeden do jednego w modelu danych*

Relacje jeden do wielu

W relacji jeden do wielu klucze z jednej tabeli występują wielokrotnie w tabeli powiązanej. Relację jeden do wielu ilustruje przykład z rysunku 15.1, gdzie powiązani są pracownicy z wydziałami. Praktycznym przykładem takiej relacji jest struktura organizacyjna wydziałów (rysunek 15.4).

Rysunek 15.4. *Jeden wydział zawiera wielu pracowników*



Relacja jeden do wielu jest relacją najczęściej występującą. Innym praktycznym przykładem jest zastosowanie skrótów nazw krajów. Każdy kraj na świecie ma swój niepowtarzalny identyfikator (Polska ma PL, Irlandia IE itd.).

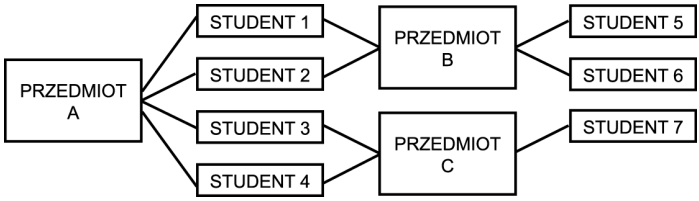
Jeżeli mamy ośmiu klientów w Polsce i pięciu w Irlandii, to w tabeli użyjemy jedynie dwóch różnych skrótów. Jeden skrót (PL) reprezentuje relację jeden do ośmiu, a drugi (IE), relację jeden do pięciu.

Relacje wiele do wielu

Relacje wiele do wielu często są przyczyną problemów w praktycznych przykładach znormalizowanych baz danych — powszechne jest rozbijanie relacji wiele do wielu na kilka relacji jeden do wielu. W relacjach wiele do wielu wartość klucza z jednej tabeli może występować wielokrotnie w powiązanej tabeli. Na razie brzmi to jak opis relacji jeden do wielu. Sęk w tym, że działa to w obie strony, co oznacza, że klucz główny z drugiej tabeli może również wielokrotnie występować w pierwszej tabeli.

Relację tego typu spróbujemy wyjaśnić, posługując się wcześniejszym przykładem ze studentami i przedmiotami. Student ma identyfikator i nazwisko. Przedmiot ma identyfikator i nazwę. Student zwykle zapisuje się na więcej niż jeden przedmiot, a na dany przedmiot jest zapisanych więcej studentów niż jeden, co widać na rysunku 15.5.

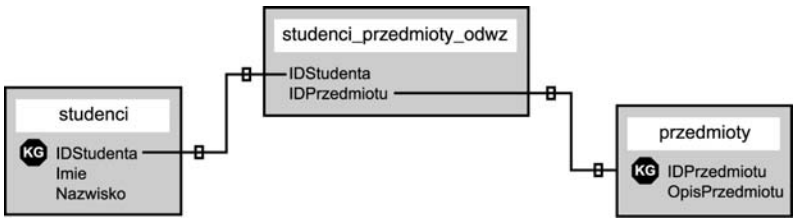
Rysunek 15.5. *Student zapisuje się na przedmioty, a na przedmiot zapisani są studenci*



Jak widać, tego rodzaju relacja nie jest łatwą metodą kojarzenia tabel. Nasze tabele wyglądają tak jak na rysunku 15.6 — na pozór nie są ze sobą w relacji.

Rysunek 15.6. *Tabela studentów i tabela przedmiotów, niepowiązane z sobą*

Aby stworzyć teoretyczną relację wiele do wielu, konieczne byłoby stworzenie tabeli przejściowej, która stanowi pomost między dwiema tabelami i opisuje ich wzajemne odwzorowania. Może ona wyglądać tak jak na rysunku 15.7.



Rysunek 15.7. *Tabela studenci_przedmioty_odwz służy jako element pośredniczący*

Jeżeli wykorzystamy informacje z rysunku 15.5 i wstawimy je do tabeli przejściowej, otrzymamy tabelę widoczną na rysunku 15.8.

Rysunek 15.8. *Tabela studenci_przedmioty_odwz wypełniona danymi*

IDSTUDENTA	IDPRZEDMIOTU
STUDENT 1	PRZEDMIOT A
STUDENT 2	PRZEDMIOT A
STUDENT 3	PRZEDMIOT A
STUDENT 4	PRZEDMIOT A
STUDENT 5	PRZEDMIOT B
STUDENT 6	PRZEDMIOT B
STUDENT 7	PRZEDMIOT C
STUDENT 1	PRZEDMIOT B
STUDENT 2	PRZEDMIOT B
STUDENT 3	PRZEDMIOT C
STUDENT 4	PRZEDMIOT C

Jak widać, wielu studentów i wiele przedmiotów może z powodzeniem współistnieć z sobą w tabeli `studenci_przedmioty_odwz`.

Po takim wprowadzeniu do typów relacji normalizacja powinna być błahostką.

Normalizacja

Normalizacja jest po prostu zbiorem reguł, które ostatecznie mają ułatwić nam życie, jeżeli pełnimy funkcję administratora bazy. Jest to sztuka organizacji bazy danych w taki sposób, by tabele były powiązane tam, gdzie jest to stosowne, i by była możliwość ich łatwej rozbudowy.

Zbiory reguł stosowane w normalizacji są nazywane *postaciami normalnymi*. Jeżeli nasz projekt bazy jest zgodny z pierwszym zbiorem reguł, jest uważany za *pierwszą postać normalną* bazy. Jeżeli nasz projekt jest zgodny z pierwszymi trzema zbiorami reguł normalizacyjnych, to bazę możemy uznać za *trzecią postać normalną*.

W tym rozdziale poznamy wszystkie reguły pierwszej, drugiej i trzeciej postaci normalnej, aby móc się ich trzymać przy tworzeniu własnych aplikacji. Zastosujemy przykładowy zbiór tabel z bazy studentów i przedmiotów i doprowadzimy go do trzeciej postaci normalnej.

Problemy z tabelą prostą

Zanim przejdziemy do pierwszej postaci normalnej, musimy wyjść od czegoś, co można poddać normalizacji. W przypadku baz danych jest to *tabela prosta*. Tabela prosta jest jak arkusz kalkulacyjny — ma nieokreśloną liczbę kolumn. Nie istnieją relacje między odrębnymi tabelami — wszystkie potrzebne dane zostały zgromadzone w jednej dużej tabeli. Takie rozwiązanie jest nieefektywne i zajmuje więcej miejsca na dysku niż baza znormalizowana.

Przyjmijmy, że w bazie studentów i przedmiotów występują następujące pola:

- `NazwiskoStudenta` — imię i nazwisko studenta.
- `IDprzedmiotu1` — identyfikator pierwszego przedmiotu wybranego przez studenta.
- `OpisPrzedmiotu1` — opis pierwszego przedmiotu wybranego przez studenta.
- `WykladowcaPrzedmiotu1` — wykładowca pierwszego przedmiotu wybranego przez studenta.
- `IDprzedmiotu2` — identyfikator drugiego przedmiotu wybranego przez studenta.
- `OpisPrzedmiotu2` — opis drugiego przedmiotu wybranego przez studenta.
- `WykladowcaPrzedmiotu2` — wykładowca drugiego przedmiotu wybranego przez studenta.
- Kolumny `IDPrzedmiotu`, `OpisPrzedmiotu` i `WykladowcaPrzedmiotu` powtarzają się wielokrotnie, aż ujęte zostaną wszystkie przedmioty wybrane przez studenta w trakcie jego studiów.

Wiedząc to, co już wiemy, powinniśmy być w stanie zidentyfikować pierwszy obszar problemów: kolumny `IDPrzedmiotu`, `OpisPrzedmiotu` i `WykladowcaPrzedmiotu` to powtarzające się grupy.

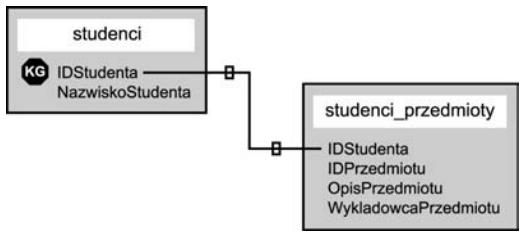
Eliminacja powtórzeń to pierwszy krok normalizacji, więc za chwilę doprowadzimy naszą tabelę do pierwszej postaci normalnej. Gdyby tabela pozostała w swojej płaskiej postaci, otrzymalibyśmy wiele pustej przestrzeni i wiele przestrzeni zajętej niepotrzebnie — nie jest to efektywna struktura tabel.

Pierwsza postać normalna

Reguły pierwszej postaci normalnej są następujące:

- Wyeliminować powtarzające się informacje.
- Stworzyć odrębne tabele dla powiązanych z sobą danych.

Zastanówmy się nad strukturą tabeli prostej z wieloma powtórzonymi grupami kolumn z bazy danych studentów i przedmiotów; może w niej zidentyfikować dwa odrębne tematy: studenci i zajęcia? Doprowadzenie naszej bazy do pierwszej postaci normalnej wymagałoby stworzenia dwóch tabel: jednej dla studentów, a drugiej dla przedmiotów, co widać na rysunku 15.9.



Rysunek 15.9. *Podział tabeli prostej na dwie tabele*

Dwie otrzymane tabele reprezentują teraz relację jeden do wielu jednego studenta z wieloma przedmiotami. Studenci mogą wybierać tyle przedmiotów, ile chcą, i nie są ograniczeni liczbą grup kolumn `IDPrzedmiotu`, `OpisPrzedmiotu` i `WykladowcaPrzedmiotu` występujących w tabeli prostej.

Następny krok to doprowadzenie tabel do drugiej postaci normalnej.

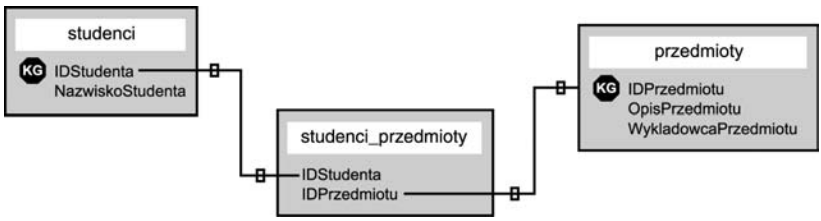
Druga postać normalna

Zasada drugiej postaci normalnej brzmi tak:

- Żaden atrybut niebędący kluczem nie zależy od części klucza głównego.

Mówiąc prościej — oznacza to, że jeżeli pola naszej tabeli nie są w całości powiązane z kluczem głównym, to mamy jeszcze coś do zrobienia. W przykładzie ze studentami i przedmiotami musimy zebrać przedmioty w oddzielnej tabeli i zmodyfikować tabelę `studenci_przedmioty`.

Kolumny `IDPrzedmiotu`, `OpisPrzedmiotu` i `WykladowcaPrzedmiotu` mogą stać się tabelą zwaną `przedmioty` z kolumną `IDprzedmiotu` w roli klucza głównego. Tabela `studenci_przedmioty` powinna wówczas zawierać tylko dwa pola: `IDStudenta` i `IDPrzedmiotu`. Nową strukturę przedstawiono na rysunku 15.10.



Rysunek 15.10. *Doprowadzenie tabel do drugiej postaci normalnej*

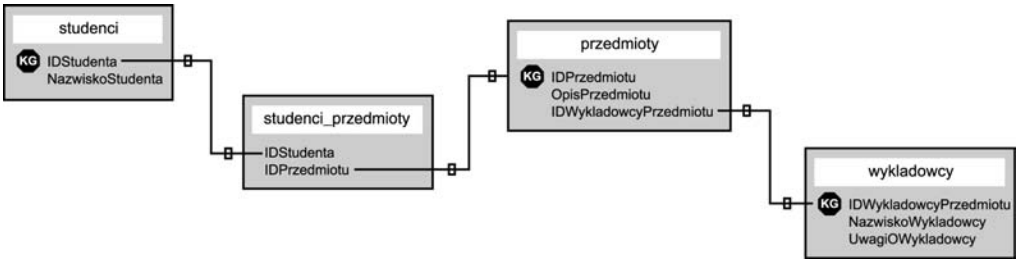
Struktura ta powinna wyglądać znajomo, jako relacja wiele do wielu z zastosowaniem pośredniczącej tabeli odwzorowującej. Trzecia postać normalna to ostatnia postać, jaka nas interesuje. Jak się zaraz okaże, zasada nią rządząca jest równie prosta, jak w przypadku pierwszych dwóch postaci.

Trzecia postać normalna

Zasada trzeciej postaci normalnej brzmi:

- Żaden atrybut nie zależy od innych atrybutów, które nie są kluczami.

Reguła ta oznacza po prostu, że musimy przyjrzeć się naszym tabelom i sprawdzić, czy istnieją jeszcze jakieś pola, które można rozbić i które nie będą zależne od klucza. Szukajmy powtarzających się danych, a szybko znajdziemy odpowiedź — wykładowcy. Jeden wykładowca z pewnością będzie wykładał więcej niż jeden przedmiot. Jednakże `WykladowcaPrzedmiotu` nie jest kluczem żadnego typu. Jeżeli więc rozbijemy te dane i stworzymy odrębną tabelę tylko po to, by zwiększyć efektywność i ułatwić utrzymanie bazy, to otrzymamy trzecią postać normalną (rysunek 15.11).



Rysunek 15.11. *Doprowadzanie tabel do trzeciej postaci normalnej*

Trzecia postać normalna zwykle wystarczy, by usunąć powtórzenia i zapewnić elastyczność oraz łatwość rozbudowy. W następnym podrozdziale otrzymamy kilka wskazówek co do procesu myślowego prowadzącego do zaprojektowania struktury bazy oraz jego powiązania z procesem tworzenia całej aplikacji.

Postępowanie zgodnie z procesem projektowania

Największym problemem przy projektowaniu aplikacji jest brak jej wcześniejszego przemyślenia. W odniesieniu do aplikacji bazodanowych proces projektowania musi uwzględniać wnikliwą analizę bazy danych — co powinna przechowywać, jakie relacje występują między danymi i, co najważniejsze, czy jest skalowalna.

Ogólne kroki procesu projektowania to:

- Definiowanie celów.
- Projektowanie struktur danych (tabele, pola).
- Rozpoznanie relacji.
- Zdefiniowanie i implementacja reguł obszaru zastosowania.
- Stworzenie aplikacji.

Tworzenie aplikacji jest krokiem ostatnim, a nie pierwszym! Wielu programistów wymyśla aplikację, po czym pisze ją, a następnie próbuje wpasować w nią zbiór pól bazy danych. Takie podejście to zaczynanie od końca, jest nieefektywne i będzie nas kosztować czas oraz pieniądze.

Zanim zaczniemy jakikolwiek proces projektowania aplikacji, warto usiąść i go przedyskutować. Jeżeli nie potrafimy opisać tworzonej aplikacji pod kątem celów, odbiorców i rynku docelowego, to znaczy, że nie jesteśmy gotowi do jej budowy, nie mówiąc już o modelowaniu bazy danych.

Po opisaniu innym, co będzie robić nasza aplikacja, i uzyskaniu ich aprobaty możemy zacząć myśleć o tabelach, jakie chcemy stworzyć. Zacznijmy od wielkich tabel prostych, ponieważ kiedy już je narysujemy, będziemy mogli zastosować właśnie zdobyte umiejętności normalizacyjne. W ten sposób znajdziemy powtórzenia i zwizualizujemy relacje.

Następnym krokiem jest normalizacja. Przejdźmy od tabeli prostej do pierwszej postaci normalnej, i po kolei aż do trzeciej, jeżeli to możliwe. Posługujmy się kartkami papieru, ołówkiem, fiszkami samoprzylepnymi i czymkolwiek, co pomoże zwizualizować tabele i relacje między nimi. To żaden wstyd modelować dane z pomocą fiszek, zanim będziemy gotowi do stworzenia samych tabel. Poza tym jest to o wiele tańsze niż kupowanie oprogramowania, które zrobi to za nas. Programy modelujące kosztują od pięciuset do kilku tysięcy złotych!

Kiedy już mamy wstępny model danych, spójrzmy na niego z perspektywy aplikacji. Albo spójrzmy na niego z perspektywy osoby, która będzie z aplikacji korzystać. W tym miejscu definiujemy reguły obszaru zastosowania i sprawdzamy, czy nasz model się sprawdza. Przykładem takiej reguły dla aplikacji rejestracji internetowej może być „każdy użytkownik musi mieć jeden adres e-mail, który nie może należeć do żadnego innego użytkownika". Jeżeli pole `AdresEmail` nie było polem z niepowtarzalnymi wartościami w naszym modelu, znaczy to, że model nie spełnia reguł obszaru zastosowania.

Dopiero po zastosowaniu wszystkich reguł obszaru zastosowania na naszym modelu danych może się rozpocząć programowanie aplikacji. Możemy być spokojni o to, że nasz model danych jest spójny i że programując, nie zamalujemy się w kącie pokoju. A to zdarza się dość często.

Podsumowanie

Trzymanie się prawidłowego procesu projektowania bazy danych to jedyny sposób na stworzenie efektywnej, elastycznej i łatwej do utrzymania aplikacji. Ważnym aspektem projektowania bazy danych jest wykorzystanie relacji między tabelami zamiast wrzucania wszystkich danych do jednej prostej tabeli. Relacje mogą być jeden do jednego, jeden do wielu i wiele do wielu.

Stosowanie relacji w celu prawidłowej organizacji danych jest nazywane *normalizacją*. Istnieje wiele poziomów normalizacji, ale podstawowe to pierwsza, druga i trzecia postać normalna. Z każdym poziomem wiąże się reguła lub reguły, którym trzeba sprostać. Trzymanie się tych reguł pomaga w stworzeniu dobrze zorganizowanej i elastycznej bazy danych.

Aby przeprowadzić pomysł od jego narodzin do realizacji, należy trzymać się procesu projektowania. Proces ten sprowadza się do tego, by pomyśleć, zanim zaczniemy działać. Przedyskutujmy reguły, wymogi i cele, a dopiero potem twórzmy ostateczną wersję znormalizowanych tabel.

Pytania i odpowiedzi

P: *Czy istnieją tylko trzy postaci normalne?*

O: Nie. Postaci normalnych jest więcej. Dodatkowe postacie to postać normalna Boyce-Codda, czwarta postać normalna, piąta postać normalna, zwana też postacią chroniącą złączenia. Dość rzadko doprowadza się bazę do tych postaci, ponieważ koszty pracy i utrata efektywności przeważają nad ewentualnymi korzyściami.

Warsztaty

Warsztaty mają na celu utrwalenie i sprawdzenie zdobytej wiedzy, powinny też pokazać, jak zastosować ją w praktyce.

Test

1. Wymień trzy typy relacji między danymi.
2. Jak należy radzić sobie z trudnościami w reprezentacji relacji wiele do wielu w efektywnej bazie danych?

Odpowiedzi

1. Jeden do jednego, jeden do wielu, wiele do wielu.
2. Zbudować szereg relacji jeden do wielu poprzez stworzenie pośredniczącej tabeli odwzorowującej.

Ćwiczenie

Objaśnij każdą z trzech postaci normalnych osobie, która pracuje z arkuszami kalkulacyjnymi i tabelami prostymi.