

Paul DuBois



Wydanie V

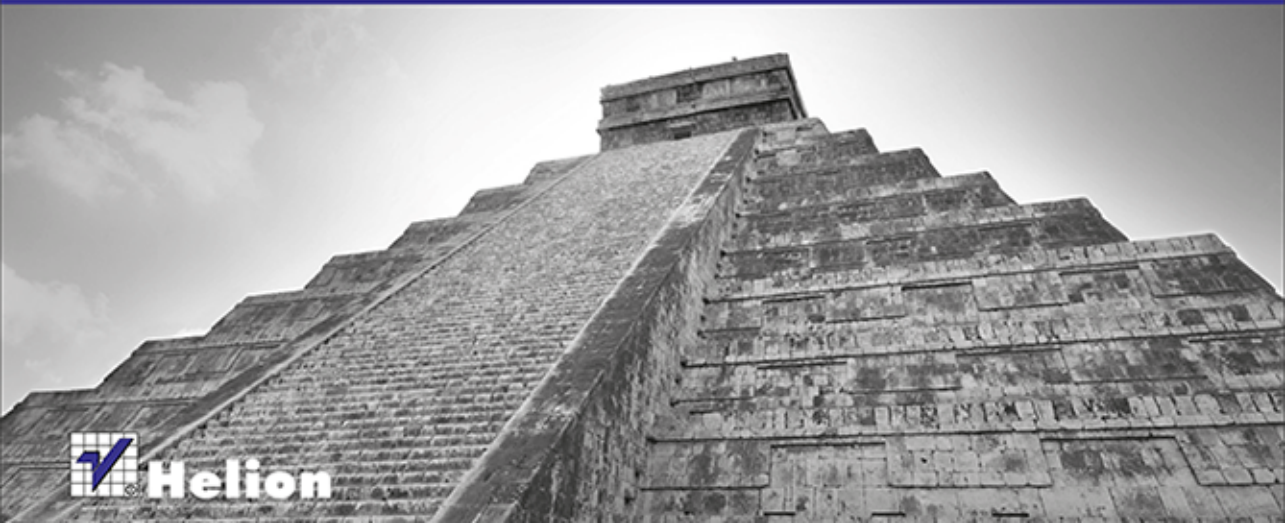
MySQL®

Vademecum profesjonalisty

Kompendium wiedzy o MySQL!



Helion



Tłumaczenie: Robert Górczyński

ISBN: 978-83-246-8146-4

Authorized translation from the English language edition, entitled: MYSQL, Fifth Edition; ISBN 0321833872; by Paul DuBois; published by Pearson Education, Inc, publishing as Addison Wesley. Copyright © 2013 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Polish language edition published by HELION S.A. Copyright © 2014.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie bierze jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Wydawnictwo HELION nie ponosi również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:

<ftp://ftp.helion.pl/przyklady/mysvp5.zip>

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/mysvp5>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

O autorze	17
Wprowadzenie	19
Część I Ogólne użycie MySQL	
Rozdział 1. Rozpoczęcie pracy z MySQL	33
1.1. W jaki sposób MySQL może Ci pomóc?	33
1.2. Przykładowa baza danych	37
1.2.1. Projekt „Liga Historyczna USA”	37
1.2.2. Projekt „Oceny uczniów”	40
1.2.3. Jak przykładowa baza danych może Ci pomóc?	41
1.3. Podstawowa terminologia bazy danych	41
1.3.1. Terminologia strukturalna	42
1.3.2. Terminologia języka zapytań	44
1.3.3. Terminologia architekuralna MySQL	45
1.4. Przewodnik po MySQL	46
1.4.1. Pobranie przykładowej bazy danych	47
1.4.2. Wymagania wstępne	47
1.4.3. Nawiązywanie i zrywanie połączenia z serwerem MySQL	49
1.4.4. Wykonywanie zapytań SQL	51
1.4.5. Tworzenie bazy danych	54
1.4.6. Tworzenie tabel	55
1.4.7. Dodawanie nowych rekordów	75
1.4.8. Przywrócenie bazy danych sampdb do znanego stanu	79
1.4.9. Pobieranie informacji	80
1.4.10. Usuwanie lub uaktualnianie istniejących rekordów	110
1.5. Wskazówki przydatne podczas pracy z klientem mysql	113
1.5.1. Uproszczenie procesu nawiązywania połączenia	113
1.5.2. Wpisywanie mniejszej ilości tekstu przy wykonywaniu zapytań	115
1.6. Co dalej?	120

Rozdział 2. Użycie MySQL do zarządzania danymi	123
2.1. Tryby SQL serwera	124
2.2. Identyfikatory składni MySQL i reguły nadawania nazw	126
2.3. Wielkość liter w zapytaniach SQL	128
2.4. Obsługa kodowania znaków	130
2.4.1. Określenie kodowania znaków	131
2.4.2. Określenie dostępności kodowania znaków i ustawień bieżących	132
2.4.3. Obsługa Unicode	133
2.5. Wybór, utworzenie, usunięcie i zmiana bazy danych	134
2.5.1. Wybór bazy danych	134
2.5.2. Utworzenie bazy danych	135
2.5.3. Usunięcie bazy danych	136
2.5.4. Zmiana bazy danych	136
2.6. Tworzenie, usuwanie, indeksowanie i modyfikowanie tabel	137
2.6.1. Cechy charakterystyczne silników bazy danych	137
2.6.2. Tworzenie tabel	143
2.6.3. Usuwanie tabel	152
2.6.4. Indeksowanie tabel	153
2.6.5. Zmiana struktury tabeli	158
2.7. Pobieranie metadanych bazy danych	161
2.7.1. Pobieranie metadanych za pomocą zapytania SHOW	162
2.7.2. Pobieranie metadanych z bazy danych INFORMATION_SCHEMA	164
2.7.3. Pobieranie metadanych z poziomu wiersza poleceń	166
2.8. Pobieranie danych z wielu tabel za pomocą złączeń	167
2.8.1. Złączenia wewnętrzne	169
2.8.2. Kwalifikowane odwołania do kolumn z poziomu złączonych tabel	170
2.8.3. Złączenia typu LEFT i RIGHT (OUTER)	171
2.9. Pobieranie informacji z wielu tabel za pomocą podzapytań	175
2.9.1. Podzapytania używające względnych operatorów porównania	176
2.9.2. Podzapytania IN i NOT IN	177
2.9.3. Podzapytania ALL, ANY i SOME	178
2.9.4. Podzapytania EXISTS i NOT EXISTS	179
2.9.5. Podzapytania skorelowane	180
2.9.6. Podzapytania w klauzuli FROM	181
2.9.7. Przepisywanie podzapytań na postać złączeń	181
2.10. Pobieranie informacji z wielu tabel za pomocą zapytań UNION	183
2.11. Usuwanie i uaktualnianie rekordów w wielu tabelach	186

2.12. Przeprowadzanie transakcji	188
2.12.1. Użycie transakcji do zapewnienia bezpiecznego środowiska wykonywania	190
2.12.2. Użycie punktów pośrednich transakcji	194
2.12.3. Izolacja transakcji	194
2.13. Klucze zewnętrzne i integralność odwołań	197
2.14. Wyszukiwanie pełnego tekstu	204
2.14.1. Wyszukiwanie pełnego tekstu w języku naturalnym	207
2.14.2. Wyszukiwanie pełnego tekstu w trybie boolowskim	208
2.14.3. Wyszukiwanie rozszerzone o wynik zapytania	210
2.14.4. Konfiguracja silnika wyszukiwania pełnego tekstu	211
Rozdział 3. Typy danych	213
3.1. Kategorie wartości danych	215
3.1.1. Wartości liczbowe	215
3.1.2. Wartości ciągu tekstowego	217
3.1.3. Wartości daty i godziny	226
3.1.4. Wartości przestrzenne	226
3.1.5. Wartości boolowskie	227
3.1.6. Wartość NULL	227
3.2. Typy danych w MySQL	227
3.2.1. Ogólny opis typów danych	228
3.2.2. Określenie typów kolumn w definicji tabeli	230
3.2.3. Definiowanie wartości domyślnych kolumn	231
3.2.4. Liczbowe typy danych	232
3.2.5. Typy danych w postaci ciągów tekstowych	240
3.2.6. Typy danych dla wartości daty i czasu	256
3.3. Jak MySQL obsługuje nieprawidłowe wartości danych?	267
3.4. Praca z sekwencjami	270
3.4.1. Ogólne właściwości AUTO_INCREMENT	270
3.4.2. Właściwości AUTO_INCREMENT charakterystyczne dla silnika bazy danych	272
3.4.3. Kwestie dotyczące kolumn AUTO_INCREMENT	275
3.4.4. Wskazówki pomocne podczas pracy z kolumnami AUTO_INCREMENT	276
3.4.5. Generowanie sekwencji bez AUTO_INCREMENT	278
3.5. Obliczanie wyrażeń i konwersja typu	280
3.5.1. Tworzenie wyrażeń	281
3.5.2. Konwersja typu	289
3.6. Wybór typu danych	297
3.6.1. Jakie wartości będą przechowywane w kolumnie?	299
3.6.2. Czy przechowywane wartości pochodzą z określonego zakresu?	302

Rozdział 4. Widoki i programy składowane	305
4.1. Używanie widoków	306
4.2. Używanie programów składowanych	309
4.2.1. Zapytania złożone i ograniczniki zapytań	310
4.2.2. Procedury i funkcje składowane	311
4.2.3. Wyzwalacze	316
4.2.4. Zdarzenia	318
4.3. Zapewnienie bezpieczeństwa widokom i programom składowanym	320
Rozdział 5. Optymalizacja zapytań	323
5.1. Użycie indeksowania	324
5.1.1. Zalety wynikające z indeksowania	324
5.1.2. Koszt indeksowania	327
5.1.3. Wybór indeksów	328
5.2. Optymalizator zapytań MySQL	333
5.2.1. Jak działa optymalizator zapytań?	334
5.2.2. Użycie zapytania EXPLAIN do sprawdzenia operacji optymalizatora	338
5.3. Wybór typu danych zapewniającego efektywne wykonywanie zapytań	344
5.4. Wybór formatu tabeli dla efektywnych zapytań	347
5.5. Efektywne wczytywanie danych	349
5.6. Harmonogram, blokady i współbieżność	352

Część II Użycie interfejsu programowania MySQL

Rozdział 6. Wprowadzenie do programowania MySQL	355
6.1. Dlaczego tworzyć własne aplikacje MySQL?	355
6.2. API dostępne dla MySQL	359
6.2.1. API C	360
6.2.2. API Perl DBI	360
6.2.3. API PHP	362
6.3. Wybór API	363
6.3.1. Przewidywane środowisko działania	364
6.3.2. Wydajność	365
6.3.3. Czas potrzebny na opracowanie aplikacji	366
6.4.3. Przenośność	367
Rozdział 7. Tworzenie programów MySQL przy użyciu języka C	369
7.1. Kompilacja i linkowanie programów klienckich	370
7.2. Nawiązanie połączenia z serwerem	373
7.3. Obsługa błędów i przetwarzanie opcji polecenia	377
7.3.1. Sprawdzanie pod kątem błędów	377

7.3.2. Pobieranie parametrów połączenia w trakcie działania programu	382
7.3.3. Implementacja przetwarzania opcji w programie klienta ...	395
7.4. Przetwarzanie zapytań SQL	399
7.4.1. Obsługa zapytań modyfikujących rekordy	401
7.4.2. Obsługa zapytań zwracających zbiór wyników	402
7.4.3. Procedury obsługi zapytań ogólnego przeznaczenia	405
7.4.4. Alternatywne podejścia do przetwarzania zapytań	407
7.4.5. Funkcja <code>mysql_store_result()</code> kontra <code>mysql_use_result()</code>	408
7.4.6. Używanie metadanych zbioru wynikowego	411
7.4.7. Kodowanie znaków specjalnych i danych binarnych	416
7.5. Program do interaktywnego wykonywania zapytań	420
7.6. Utworzenie klienta z obsługą SSL	421
7.7. Jednoczesne wykonywanie wielu zapytań	426
7.8. Używanie zapytań preinterpretowanych	428
7.9. Użycie preinterpretowanego zapytania CALL	440
Rozdział 8. Tworzenie programów MySQL przy użyciu Perl DBI	445
8.1. Cechy charakterystyczne skryptu Perl	446
8.2. Ogólny opis Perl DBI	447
8.2.1. Typy danych DBI	447
8.2.2. Prosty skrypt DBI	448
8.2.3. Obsługa błędów	453
8.2.4. Obsługa zapytań modyfikujących rekordy	456
8.2.5. Obsługa zapytań zwracających zbiór wyników	458
8.2.6. Cytowanie znaków specjalnych w ciągach tekstowych zapytań	467
8.2.7. Miejsca zarezerwowane i zapytania preinterpretowane ...	470
8.2.8. Dołączanie wyniku zapytania do zmiennych skryptu	474
8.2.9. Określenie parametrów połączenia	475
8.2.10. Usuwanie błędów	478
8.2.11. Używanie metadanych zbioru wynikowego	482
8.2.12. Przeprowadzanie transakcji	486
8.3. Praca z DBI	488
8.3.1. Generowanie katalogu Ligi Historycznej	488
8.3.2. Wysyłanie członkom Ligi przypomnień o konieczności przedłużenia członkostwa	495
8.3.3. Edycja rekordu członka Ligi Historycznej	500
8.3.4. Wyszukiwanie członków Ligi Historycznej o takich samych zainteresowaniach	506
8.3.5. Udostępnienie w internecie katalogu Ligi Historycznej ...	507
8.4. Użycie DBI w aplikacjach sieciowych	510
8.4.1. Konfiguracja Apache w celu obsługi skryptów CGI	512
8.4.2. Krótki opis modułu CGI.pm	513

8.4.3. Nawiązanie połączenia z serwerem MySQL z poziomu skryptu sieciowego	521
8.4.4. Przeglądarka bazy danych	523
8.4.5. Przeglądarka tabel projektu ocen uczniów	528
8.4.6. Wyszukiwanie wspólnych zainteresowań w projekcie Ligi Historycznej	531

Rozdział 9. Tworzenie programów MySQL przy użyciu języka PHP537

9.1. Ogólny opis PHP	539
9.1.1. Prosty skrypt PHP	541
9.1.2. Użycie biblioteki plików PHP w celu hermetyzacji kodu ...	545
9.1.3. Prosta strona pobierająca dane	550
9.1.4. Przetwarzanie wyników zapytania	554
9.1.5. Sprawdzanie pod kątem wartości NULL w wynikach zapytania	558
9.1.6. Używanie zapytań preinterpretowanych	559
9.1.7. Użycie miejsc zarezerwowanych do obsługi kwestii związanych z cytowaniem danych	559
9.1.8. Obsługa błędów	561
9.2. Praca z PHP	564
9.2.1. Aplikacja pozwalająca na wprowadzenie wyników uzyskanych przez uczniów	564
9.2.2. Tworzenie interaktywnego quizu w internecie	577
9.2.3. Edycja informacji o członkach Ligi Historycznej	582

Część III Administracja MySQL

Rozdział 10. Wprowadzenie do administracji bazą danych MySQL591

10.1. Komponenty MySQL	592
10.2. Ogólna administracja MySQL	593
10.3. Kontrola dostępu i zapewnianie bezpieczeństwa	594
10.4. Obsługa bazy danych, tworzenie kopii zapasowych i replikacja	595

Rozdział 11. Katalog danych w MySQL597

11.1. Położenie katalogu danych	598
11.2. Struktura katalogu danych	599
11.2.1. W jaki sposób serwer MySQL zapewnia dostęp do danych?	600
11.2.2. Przedstawienie baz danych w systemie plików	602
11.2.3. Przedstawienie tabel w systemie plików	603
11.2.4. Przedstawienie widoków i wyzwalaczy w systemie plików	604
11.2.5. Jak zapytania SQL są mapowane na operacje na pliku tabeli?	604
11.2.6. Ograniczenia systemu operacyjnego w zakresie nazw obiektów bazy danych	605

11.2.7. Czynniki wpływające na maksymalną wielkość tabeli	607
11.2.8. Wpływ struktury katalogu danych na wydajność systemu	609
11.2.9. Pliki dzienników zdarzeń i stanu MySQL	611
11.3. Przeniesienie zawartości katalogu danych	613
11.3.1. Metody przenoszenia katalogu danych lub jego elementów	614
11.3.2. Przygotowania do operacji przeniesienia	615
11.3.3. Uzyskanie dostępu do wyniku przeniesienia	615
11.3.4. Przeniesienie całego katalogu danych	617
11.3.5. Przeniesienie poszczególnych baz danych	617
11.3.6. Przeniesienie poszczególnych tabel	618
11.3.7. Przeniesienie systemowej przestrzeni tabel InnoDB	618
11.3.8. Przeniesienie plików dzienników zdarzeń i stanu	619
Rozdział 12. Ogólna administracja bazą danych MySQL	621
12.1. Zabezpieczenie nowej instalacji MySQL	622
12.1.1. Definiowanie haseł dla początkowych kont MySQL	622
12.1.2. Konfiguracja haseł dla serwerów dodatkowych	627
12.2. Konfiguracja uruchamiania i zamykania serwera MySQL	628
12.2.1. Uruchamianie serwera MySQL w systemach UNIX	628
12.2.2. Uruchamianie serwera MySQL w systemach Windows	634
12.2.3. Określanie opcji startowych serwera	637
12.2.4. W jaki sposób serwer nasłuchuje połączeń?	639
12.2.5. Zatrzymanie serwera	640
12.2.6. Odzyskanie kontroli nad serwerem, gdy nie można nawiązać z nim połączenia	641
12.3. Używanie zmiennych systemowych i stanu	644
12.3.1. Sprawdzanie i ustawienie wartości zmiennych systemowych	645
12.3.2. Sprawdzenie wartości zmiennej stanu	649
12.4. Interfejs wtyczek	651
12.5. Konfiguracja silnika bazy danych	654
12.5.1. Wybór silnika bazy danych	655
12.5.2. Wybór domyślnego silnika bazy danych	656
12.5.3. Konfiguracja silnika InnoDB	656
12.6. Kwestie związane z globalizacją	664
12.6.1. Konfiguracja obsługi stref czasowych	664
12.6.2. Ustawienie domyślnego kodowania znaków i kolejności sortowania	667
12.6.3. Wybór języka wyświetlania komunikatów błędów	667
12.6.4. Wybór ustawień językowych	668

12.7. Dostrajanie serwera	668
12.7.1. Zmienne systemowe ogólnego przeznaczenia do dostrajania serwera	670
12.7.2. Dostrajanie silnika bazy danych	673
12.7.3. Używanie bufora zapytań	680
12.7.4. Optymalizacje sprzętowe	682
12.8. Dzienniki zdarzeń serwera	684
12.8.1. Dziennik błędów	687
12.8.2. Ogólny dziennik zapytań	688
12.8.3. Dziennik wolno wykonywanych zapytań	689
12.8.4. Binarny dziennik zdarzeń	690
12.8.5. Dziennik przekazywania	692
12.8.6. Rejestracja zdarzeń w tabelach	692
12.8.7. Zarządzanie dziennikami zdarzeń	693
12.9. Uruchamianie wielu serwerów MySQL	701
12.9.1. Ogólne kwestie związane z uruchamianiem wielu serwerów	702
12.9.2. Konfiguracja i kompilacja różnych serwerów	705
12.9.3. Strategie podawania opcji startowych	707
12.9.4. Użycie skryptu mysqld_multi do zarządzania serwerem	708
12.9.5. Uruchamianie wielu serwerów w Windows	710
12.9.6. Uruchamianie klientów wielu serwerów	712
12.10. Uaktualnianie MySQL	712

Rozdział 13. Bezpieczeństwo i kontrola dostępu715

13.1. Zabezpieczenie dostępu do MySQL przez system plików	716
13.1.1. Jak ukraść dane?	717
13.1.2. Zabezpieczenie instalacji MySQL	718
13.2. Zarządzanie kontami użytkowników w MySQL	725
13.2.1. Zarządzanie kontem MySQL na wysokim poziomie	727
13.2.2. Nadawanie uprawnień	732
13.2.3. Wyświetlanie uprawnień użytkownika	744
13.2.4. Odbieranie uprawnień	744
13.2.5. Zmiana hasła lub zerowanie zapomnianego	745
13.2.6. Unikanie ryzyka związanego z kontrolą dostępu	746
13.2.7. Wtyczki metod uwierzytelniania i użytkownicy proxy ...	750
13.3. Struktura i zawartość tabel uprawnień	754
13.3.1. Istniejące w tabelach uprawnień kolumny dotyczące zasięgu udzielanego dostępu	757
13.3.2. Istniejące w tabelach uprawnień kolumny uprawnień	758
13.3.3. Istniejące w tabelach uprawnień kolumny uwierzytelnienia	759

13.3.4. Istniejące w tabelach uprawnień kolumny dotyczące SSL	759
13.3.5. Istniejące w tabelach uprawnień kolumny zarządzania zasobami	760
13.4. W jaki sposób serwer kontroluje dostęp uzyskiwany przez klientów?	761
13.4.1. Zawartość kolumn zasięgu	762
13.4.2. Weryfikacja uprawnień zapytania	764
13.4.3. Kolejność dopasowania kolumn zasięgu	766
13.4.4. Puzzle uprawnień	767
13.5. Konfiguracja bezpiecznych połączeń za pomocą SSL	770

Rozdział 14. Obsługa bazy danych, kopie zapasowe i replikacja775

14.1. Zasady obsługi profilaktycznej	775
14.2. Obsługa bazy danych w działającym serwerze	777
14.2.1. Blokowanie poszczególnych tabel w celu uzyskania dostępu tylko do odczytu lub odczytu i zapisu	779
14.2.2. Nałożenie na wszystkie bazy danych blokady pozwalającej jedynie na ich odczyt	782
14.3. Ogólne działania profilaktyczne	783
14.3.1. Używanie możliwości serwera w zakresie automatycznej naprawy	783
14.3.2. Harmonogram działań profilaktycznych	784
14.4. Tworzenie kopii zapasowej bazy danych	785
14.4.1. Cechy charakterystyczne przenośności silników bazy danych	788
14.4.2. Tworzenie kopii zapasowej za pomocą narzędzia mysqldump	790
14.4.3. Tworzenie binarnej kopii zapasowej	793
14.4.4. Tworzenie kopii zapasowej tabel InnoDB	795
14.5. Kopiowanie baz danych do innego serwera	796
14.5.1. Kopiowanie bazy danych za pomocą pliku kopii zapasowej	797
14.5.2. Kopiowanie baz danych z jednego serwera do innego	798
14.6. Sprawdzanie i naprawianie tabel bazy danych	798
14.6.1. Sprawdzanie tabel za pomocą zapytania CHECK TABLE	800
14.6.2. Naprawa tabel za pomocą zapytania REPAIR TABLE	801
14.6.3. Użycie narzędzia mysqlcheck do sprawdzania i naprawy tabel	801
14.7. Użycie kopii zapasowej do przywrócenia danych	803
14.7.1. Przywracanie całych baz danych	803
14.7.2. Przywracanie poszczególnych tabel	804

14.7.3. Ponowne wykonanie zapytań zapisanych w plikach binarnego dziennika zdarzeń	805
14.7.4. Rozwiązywanie problemów związanych z automatyczną naprawą w InnoDB	807
14.8. Konfiguracja serwerów replikacji	809
14.8.1. Jak działa replikacja?	809
14.8.2. Utworzenie relacji typu główny – podległy	811
14.8.3. Formaty rejestracji zdarzeń w dzienniku binarnym	815
14.8.4. Użycie serwera podległego replikacji do tworzenia kopii zapasowych	815

Dodatki

Dodatek A Oprogramowanie wymagane do użycia tej książki819

A.1. Pobranie dystrybucji sampdb zawierającej przykładową bazę danych	819
A.2. Pobieranie serwera MySQL i powiązanego z nim oprogramowania	820
A.3. Instalacja MySQL	822
A.3.1. Tworzenie konta logowania dla użytkownika MySQL	823
A.3.2. Instalacja MySQL	823
A.3.3. Konfiguracja zmiennej środowiskowej PATH	824
A.3.4. Inicjalizacja katalogu danych i tabel uprawnień	825
A.3.5. Uruchamianie serwera	826
A.3.6. Inicjalizacja innych tabel systemowych	827
A.4. Informacje dotyczące instalacji Perl DBI	828
A.5. Informacje dotyczące instalacji PHP i PDO	829

Dodatek B Przewodnik po typach danych831

B.1. Typy liczbowe	833
B.1.1. Typy liczb całkowitych	834
B.1.2. Typy liczb o stałej ilości cyfr	836
B.1.3. Typy liczb zmiennoprzecinkowych	836
B.1.4. Typ BIT	837
B.2. Typy tekstowe	838
B.2.1. Typy binarnych ciągów tekstowych	840
B.2.2. Typy niebinarnych ciągów tekstowych	842
B.2.3. Typy ENUM i SET	845
B.3. Typy daty i godziny	845

Dodatek C Przewodnik po operatorach i funkcjach851

C.1. Operatory	852
C.1.1. Kolejność operatorów	852
C.1.2. Operatory grupowania	853
C.1.3. Operatory arytmetyczne	854

C.1.4. Operatory porównania	856
C.1.5. Operatory bitowe	861
C.1.6. Operatory logiczne	862
C.1.7. Operatory rzutowania	864
C.1.8. Operatory dopasowania wzorca	865
C.2. Funkcje	869
C.2.1. Funkcje porównań	870
C.2.2. Funkcje rzutowania	872
C.2.3. Funkcje liczbowe	873
C.2.4. Funkcje ciągu tekstowego	878
C.2.5. Funkcje daty i godziny	892
C.2.6. Funkcje podsumowań	907
C.2.7. Funkcje zapewnienia bezpieczeństwa oraz związane z kompresją	911
C.2.8. Funkcje nakładania blokad doradczych	915
C.2.9. Funkcje związane z adresem IP	917
C.2.10. Funkcje XML	919
C.2.11. Funkcje przestrzenne	920
C.2.12. Różne funkcje	920
Dodatek D Przewodnik po zmiennych systemowych, stanu i użytkownika	927
D.1. Zmienne systemowe	927
D.1.1. Zmienne systemowe InnoDB	972
D.2. Zmienne stanu	987
D.2.1. Zmienne stanu InnoDB	994
D.2.2. Zmienne stanu bufora zapytań	998
D.2.3. Zmienne stanu SSL	999
D.3. Zmienne zdefiniowane przez użytkownika	1000
Dodatek E Przewodnik po składni SQL	1003
E.1. Składnia zapytań SQL (zapytania niezłożone)	1004
E.2. Składnia zapytań SQL (zapytania złożone)	1103
E.2.1. Polecenia struktury kontrolnej	1103
E.2.2. Zapytania obsługi deklaracji	1105
E.2.3. Zapytania obsługi kursora	1107
E.2.4. Zapytania obsługi warunków	1108
E.3. Składnia komentarzy	1112
Dodatek F Przewodnik po programie SQL	1115
F.1. Wyświetlanie komunikatu pomocy programu	1116
F.2. Określanie opcji programu	1117
F.2.1. Opcje standardowe programu MySQL	1119
F.2.2. Pliki opcji	1124
F.2.3. Zmienne środowiskowe	1129

F.3. Narzędzie myisamchk	1130
F.3.1. Opcje standardowe obsługiwane przez myisamchk	1132
F.3.2. Opcje charakterystyczne dla myisamchk	1132
F.3.3. Zmienne narzędzia myisamchk	1136
F.4. mysql	1137
F.4.1. Opcje standardowe obsługiwane przez mysql	1139
F.4.2. Opcje charakterystyczne dla mysql	1140
F.4.3. Zmienne programu mysql	1145
F.4.4. Polecenia programu mysql	1145
F.4.5. Sekwencje definiujące znak zachęty mysql	1148
F.5. mysql.server	1150
F.5.1. Opcje obsługiwane przez mysql.server	1150
F.6. mysql_config	1150
F.6.1. Opcje charakterystyczne dla mysql_config	1151
F.7. Skrypt mysql_install_db	1152
F.7.1. Opcje standardowe obsługiwane przez mysql_install_db	1152
F.7.2. Opcje charakterystyczne dla mysql_install_db	1152
F.8. mysql_upgrade	1153
F.8.1. Opcje standardowe obsługiwane przez mysql_upgrade	1153
F.8.2. Opcje charakterystyczne dla mysql_upgrade	1154
F.9. mysqladmin	1154
F.9.1. Opcje standardowe obsługiwane przez mysqladmin	1154
F.9.2. Opcje charakterystyczne dla mysqladmin	1155
F.9.3. Zmienne dla mysqladmin	1155
F.9.4. Polecenia mysqladmin	1156
F.10. mysqlbinlog	1158
F.10.1. Opcje standardowe obsługiwane przez mysqlbinlog	1159
F.10.2. Opcje charakterystyczne dla mysqlbinlog	1159
F.10.3. Zmienne dla mysqlbinlog	1162
F.11. mysqlcheck	1163
F.11.1. Opcje standardowe obsługiwane przez mysqlcheck	1163
F.11.2. Opcje charakterystyczne dla mysqlcheck	1163
F.12. mysqld	1167
F.12.1. Opcje standardowe obsługiwane przez mysqld	1168
F.12.2. Opcje charakterystyczne dla mysqld	1168
F.12.3. Zmienne dla mysqld	1180
F.13. mysqld_multi	1180
F.13.1. Opcje standardowe obsługiwane przez mysqld_multi	1180
F.13.2. Opcje charakterystyczne dla mysqld_multi	1181

F.14. mysqld_safe	1181
F.14.1. Opcje standardowe obsługiwane przez mysqld_safe	1182
F.14.2. Opcje charakterystyczne dla mysqld_safe	1182
F.15. mysqldump	1184
F.15.1. Opcje standardowe obsługiwane przez mysqldump	1184
F.15.2. Opcje charakterystyczne dla mysqldump	1185
F.15.3. Oferowane przez mysqldump opcje formatu danych	1193
F.15.4. Zmienne dla mysqldump	1193
F.16. mysqlimport	1194
F.16.1. Opcje standardowe obsługiwane przez mysqlimport ...	1194
F.16.2. Opcje charakterystyczne dla mysqlimport	1194
F.16.3. Oferowane przez mysqlimport opcje formatu danych	1196
F.17. mysqlshow	1196
F.17.1. Opcje standardowe obsługiwane przez mysqlshow	1197
F.17.2. Opcje charakterystyczne dla mysqlshow	1197
F.18. perror	1198
F.18.1. Opcje standardowe obsługiwane przez perror	1198
Skorowidz	1199

Katalog danych w MySQL

Pod względem koncepcyjnym różne systemy baz danych mają jedną wspólną cechę: zarządzają zestawem baz danych zawierających zestawy tabel. Jednak każdy system bazy danych charakteryzuje się własnym rozwiązaniem w zakresie zarządzania danymi i MySQL nie jest tutaj wyjątkiem. Domyślnie, serwer MySQL (`mysqld`) przechowuje wszystkie informacje w lokalizacji nazywanej katalogiem danych MySQL. We wspomnianym katalogu znajdują się wszystkie bazy danych, pliki stanu i pliki dzienników zdarzeń dostarczających informacje o działalności serwera. Jeżeli jesteś w jakimkolwiek stopniu odpowiedzialny za zadania administracyjne instalacji MySQL, powinieneś zapoznać się z projektem i sposobem używania katalogu danych, ponieważ ta wiedza będzie Ci potrzebna w wypełnianiu obowiązków administracyjnych. Nawet jeśli nie zamierzasz zajmować się administracją MySQL, to lektura niniejszego rozdziału może przynieść pewne korzyści, a wiedza o sposobie działania serwera na pewno nie zaszkodzi.

W tym rozdziale zostaną omówione wymienione poniżej zagadnienia:

- **Położenie katalogu danych.** Ponieważ katalog danych odgrywa ważną rolę w działalności serwera MySQL, powinieneś potrafić wskazać jego położenie, aby efektywnie administrować zawartością katalogu danych.
- **Sposób, w jaki serwer organizuje dostęp do baz danych i tabel oraz nim zarządza.** To bardzo ważne w celu konfiguracji harmonogramu obsługi serwera, a także przeprowadzania procesu odzyskiwania danych w przypadku uszkodzenia tabel.
- **Rodzaje generowanych plików stanu i dzienników zdarzeń oraz ich zawartość.** Wspomniane pliki zawierają użyteczne informacje na temat działania serwera i są wręcz nieocenione po wystąpieniu jakichkolwiek problemów.
- **Sposób zmiany domyślnego położenia i organizacji katalogu danych.** To może być ważne podczas zarządzania alokacją zasobów dyskowych w systemie, na przykład przez zrównoważenie aktywności dyskowej między dostępnymi napędami lub przez przeniesienie danych do systemów plików z większą ilością wolnej przestrzeni. Zdobyta tutaj wiedza jest także użyteczna podczas planowania miejsca położenia nowych baz danych.

W rozdziale przyjęto założenie, że w przypadku systemów UNIX istnieje konto logowania przeznaczone do wykonywania zadań administracyjnych oraz dla działającego serwera. W tej książce nazwa użytkownika i grupy wspomnianego konta to `mysql`. W podpunkcie 12.2.1.1, zatytułowanym „Działanie serwera w ramach pozbawionego uprawnień konta logowania”, przedstawiono powody, dla których warto używać oddzielnego konta logowania do wykonywania zadań administracyjnych MySQL.

11.1. Położenie katalogu danych

Domyślne położenie katalogu danych (*data*) jest skompilowane w serwerze. W systemach UNIX z reguły będzie to katalog `/usr/local/mysql/data` w przypadku instalacji MySQL z dystrybucji binarnej bądź źródłowej lub `/var/lib/mysql` po instalacji z pakietu RPM. Z kolei w systemach Windows najczęściej będzie to katalog `C:\ProgramData\MySQL` lub `C:\Documents and Settings\All Users\Application Data\MySQL`, w zależności od używanej wersji Windows.

Jeżeli kompilujesz MySQL ze źródeł, domyślne położenie katalogu danych możesz wskazać za pomocą opcji wiersza poleceń `-DMYSQL_DATADIR=nazwa_katalogu` podczas uruchamiania `CMake`.

Aby położenie katalogu danych wskazać w chwili uruchamiania serwera, należy użyć opcji `--datadir=nazwa_katalogu`. To użyteczne rozwiązanie w celu wskazania innego katalogu niż domyślny zdefiniowany w trakcie kompilacji. Jeszcze innym sposobem jest podanie katalogu w pliku opcji odczytywanym przez serwer w chwili jego uruchamiania. W takim przypadku nie ma konieczności podawania katalogu danych w wierszu poleceń w trakcie każdego uruchamiania serwera.

Jako administrator MySQL powinieneś znać miejsce położenia katalogu danych serwera, ale jeśli go nie znasz (prawdopodobnie po przejściu obowiązków po poprzednim administratorze, który nie pozostawił wystarczająco dokładnych informacji o instalacji MySQL), istnieje kilka sposobów na sprawdzenie wspomnianego położenia. Poniżej przedstawiono jedną z metod, stosowaną, gdy serwer nie jest uruchomiony. Kolejna metoda pozwala na sprawdzenie położenia katalogu danych serwera w przypadku działającego serwera.

Spójrz na plik opcji odczytywany w chwili uruchamiania serwera. Na przykład, w systemach UNIX po otwarciu pliku `/etc/my.cnf` w grupie `[mysqld]` możesz znaleźć wiersz `datadir`:

```
[mysqld]
datadir= /ścieżka/dostępu/do/katalogu/data
```

Ścieżka dostępu wskazuje położenie katalogu danych serwera.

Jeżeli nie jesteś pewien, czy serwer odczytuje pliki opcji, wywołaj go w przedstawiony poniżej sposób i sprawdź komunikat pomocy, który wyświetla położenia plików opcji:

```
% mysql --verbose --help
```

W przypadku działającego serwera nawiąż z nim połączenie i sprawdź położenie katalogu danych. Serwer zawiera pewną liczbę zmiennych systemowych dotyczących jego funkcjonowania i może wyświetlić ich wartości. Położenie katalogu danych wskazuje zmienna `datadir`, której wartość można wyświetlić za pomocą zapytania `SHOW VARIABLES` lub polecenia `mysqladmin variables`. Jeżeli zapytanie `SHOW VARIABLES` wykonasz w systemie UNIX, jego wynik może przedstawiać się następująco:

```
mysql> SHOW VARIABLES LIKE 'datadir';
+-----+-----+
| Variable_name | Value                |
+-----+-----+
| datadir       | /usr/local/mysql/data/ |
+-----+-----+
```

Z poziomu wiersza poleceń użyj narzędzia `mysqladmin`:

```
% mysqladmin variables
+-----+-----+
| Variable_name | Value                |
+-----+-----+
...
| datadir       | /usr/local/mysql/data/ |
...

```

W systemie Windows położeniem katalogu danych może być na przykład `C:\ProgramData\MySQL`.

Jeżeli masz kilka działających serwerów, będą one nasłuchiwały na różnych interfejsach sieciowych (porty TCP/IP, pliki gniazd systemu UNIX, nazwane potoki w Windows lub pamięć współdzielona). Informacje o położeniu katalogu danych możesz otrzymać po nawiązaniu połączenia z każdym serwerem po kolei za pomocą odpowiednich opcji parametrów połączenia.

W przypadku istnienia katalogu danych i konieczności jego przeniesienia do innego położenia zapoznaj się z podrozdziałem 11.3, zatytułowanym „Przeniesienie zawartości katalogu danych”, w którym omówiono techniki przenoszenia katalogu danych MySQL.

11.2. Struktura katalogu danych

Katalog danych MySQL zawiera wszystkie bazy danych, którymi zarządza serwer. Ogólnie rzecz biorąc, mają one postać struktury drzewa zaimplementowanego w bardzo prosty sposób, wykorzystujący hierarchiczną strukturę systemu plików UNIX lub Windows:

- Każda baza danych ma własny podkatalog w katalogu danych MySQL.
- Tabele, widoki i wyzwalacze w bazie danych odpowiadają plikom w podkatalogu danej bazy danych.

Poszczególne silniki bazy danych mogą stosować strukturę pamięci masowej różniącą się od ogólnej hierarchicznej implementacji bazy danych za pomocą katalogów i plików. Na przykład, silnik bazy danych InnoDB może we wspólnej przestrzeni tabel

przechowywać wszystkie tabele InnoDB ze wszystkich baz danych. Wspomniana przestrzeń tabel składa się z jednego lub więcej ogromnych plików traktowanych jako pojedyncza, jednolita struktura danych, w ramach której przedstawiane są tabele i indeksy. Domyślnie silnik InnoDB przechowuje pliki przestrzeni tabel w katalogu danych MySQL.

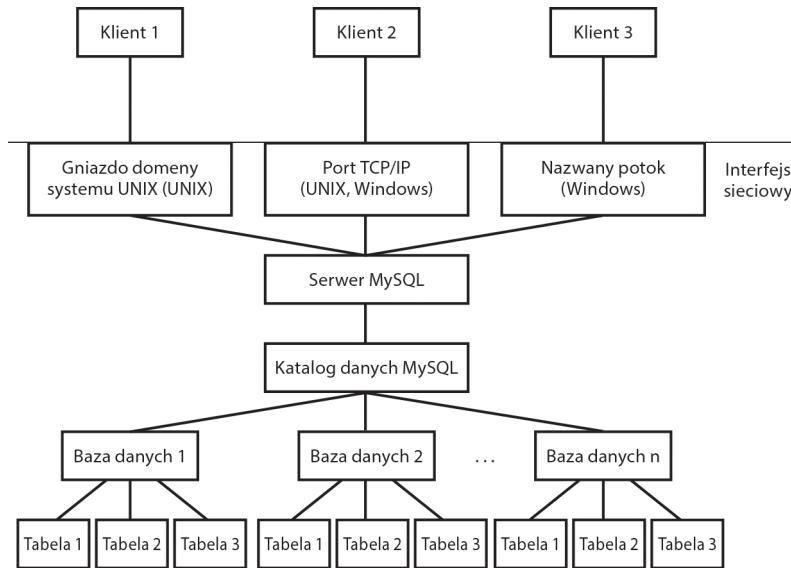
Katalog danych może zawierać także jeszcze inne pliki:

- Plik PID (ang. *Process ID*) serwera. Podczas uruchamiania serwer zapisuje w tym pliku identyfikator procesu, co pozwala innym programom na poznanie wspomnianej wartości, jeśli muszą wysłać jakikolwiek sygnał do serwera.
- Generowane przez serwer pliki stanu i dzienników zdarzeń. Wspomniane pliki zawierają ważne informacje o działalności serwera i są wręcz nieocenione dla administratorów, zwłaszcza w przypadku wystąpienia jakichkolwiek problemów, gdy zachodzi potrzeba znalezienia źródła problemu. Na przykład, jeśli określone zapytanie nie jest z powodzeniem wykonywane w serwerze, problem bardzo często można ustalić przez analizę plików dzienników zdarzeń. (Jeżeli skonfigurujesz serwer w taki sposób, aby rejestracja informacji odbywała się w tabelach bazy danych zamiast w plikach dzienników zdarzeń, to tabele zdarzeń znajdziesz w bazie danych mysql).
- Pliki powiązane z serwerem, na przykład plik klucza DES lub pliki kluczy i certyfikatów SSL serwera. Wymienione pliki administratorzy bardzo często umieszczają w katalogu danych MySQL.

11.2.1. W jaki sposób serwer MySQL zapewnia dostęp do danych?

Kiedy baza danych MySQL jest używana w typowej konfiguracji klient-serwer, wszystkie bazy danych znajdujące się w katalogu danych są zarządzane przez pojedynczą jednostkę, czyli serwer MySQL (mysqld). Programy klientów nie mają możliwości bezpośredniego przeprowadzania operacji na danych. Zamiast tego serwer zapewnia pojedynczy punkt kontaktu z bazą danych działający w charakterze pośrednika między programem klienta i danymi, których chce użyć. Tę architekturę pokazano na rysunku 11.1.

Podczas uruchamiania serwera następuje otworenie wszystkich obsługiwanych przez niego plików dzienników zdarzeń, a następnie udostępnienie katalogu danych interfejsowi sieciowemu przez nasłuchiwanie różnych rodzajów połączeń sieciowych. (Informacje szczegółowe dotyczące wyboru interfejsu sieciowego znajdziesz w punkcie 12.2.4, zatytułowanym „W jaki sposób serwer nasłuchuje połączeń?”). W celu uzyskania dostępu do danych program klienta nawiązuje połączenie z serwerem i wysyła zapytania SQL odpowiedzialnych za wykonanie określonych operacji, takich jak utworzenie tabeli, wybór rekordów lub uaktualnienie rekordów. Serwer wykonuje wskazane operacje i zwraca klientowi ich wynik. Ponieważ serwer jest wielowątkowy, jednocześnie może obsługiwać wiele połączeń z klientami. Jednak ponieważ jednocześnie można przeprowadzać tylko jedną operację uaktualniania, w efekcie serwer serializuje zapytania, aby dwa klienty nigdy nie miały szansy na jednoczesne uaktualnienie tego samego rekordu.



Rysunek 11.1. Sposób, w jaki serwer MySQL kontroluje dostęp do katalogu danych

W normalnych warunkach serwer działający w charakterze jedynej arbitra dostępu do bazy danych jest gwarancją uniknięcia wszelkiego rodzaju uszkodzeń, które mogłyby powstać na skutek jednoczesnego przetwarzania tabel bazy danych przez wiele klientów. Administratorzy powinni jednak zdawać sobie sprawę, że zdarzają się sytuacje, w których serwer nie ma wyłącznej kontroli nad katalogiem danych. Narzędzia bezpośredniego dostępu, takie jak `myisamchk`, są używane do przeprowadzania zadań administracyjnych względem tabel `MyISAM`, podczas rozwiązywania problemów, operacji naprawy lub kompresji. Tego rodzaju programy działają bezpośrednio na plikach odpowiadających tabelom. Ponieważ te narzędzia mogą zmienić zawartość tabel, używanie w tym samym czasie tabel przez serwer może doprowadzić do ich uszkodzenia.

Najbardziej oczywistym sposobem uniknięcia wymienionego problemu jest zatrzymanie serwera przed uruchomieniem programu narzędziowego. Jeśli nie ma takiej możliwości, konieczne jest poznanie sposobu zakazania serwerowi uzyskiwania dostępu do tabeli, gdy używasz narzędzia bezpośrednio działającego na plikach tej tabeli. W podrozdziale 14.2, zatytułowanym „Obsługa bazy danych w działającym serwerze”, znajdziesz informacje dotyczące współpracy z serwerem podczas używania programów narzędziowych. Alternatywą dla narzędzia `myisamchk` jest wykonanie zapytań takich jak `CHECK TABLE` i `REPAIR TABLE` (lub wykorzystanie narzędzia `mysqlcheck` wykonującego wymienione zapytania). Wymienione zapytania eliminują problem współpracy z serwerem, ponieważ samemu serwerowi nakazują przeprowadzenie operacji na tabelach.

11.2.2. Przedstawienie baz danych w systemie plików

Każda baza danych zarządzana przez serwer MySQL ma własny katalog. Istnieje on jako podkatalog w katalogu danych i ma taką samą nazwę jak baza danych, którą przedstawia. Na przykład, jeżeli *katalog_danych* określa położenie katalogu danych w serwerze, a nazwą bazy danych jest *moja_baza_danych*, wówczas katalogiem bazy danych jest *katalog_danych/moja_baza_danych* w systemach UNIX lub *katalog_danych\moja_baza_danych* w Windows.

Zapytanie `SHOW DATABASES` po prostu generuje listę podkatalogów znajdujących się w katalogu danych MySQL.

Zapytanie `CREATE DATABASE nazwa_bazy_danych` tworzy podkatalog o podanej nazwie w katalogu danych MySQL. Utworzony w ten sposób podkatalog będzie katalogiem bazy danych. Ponadto, w katalogu bazy danych wymienione zapytanie tworzy plik *db.opt* zawierający atrybuty bazy danych, na przykład domyślne kodowanie znaków i kolejność sortowania. W systemach UNIX właścicielem katalogu bazy danych jest użytkownik używany do uruchamiania serwera. Katalog bazy danych jest dostępny tylko dla tego użytkownika.

Zapytanie `DROP DATABASE` jest zaimplementowane niemal w taki sam prosty sposób. Zapytanie `DROP DATABASE nazwa_bazy_danych` powoduje usunięcie z katalogu danych MySQL podkatalogu o nazwie *nazwa_bazy_danych* wraz ze znajdującymi się w nim tabelami i innymi obiektami bazy danych, na przykład widokami i wyzwalaczami. Niemalże odpowiada to ręcznemu usunięciu katalogu bazy danych za pomocą polecenia poziomu systemu plików, takiego jak `rm` w systemie UNIX lub `del` w Windows. Jednak między zapytaniem `DROP DATABASE` i wymienionymi poleceniami systemu plików istnieją pewne różnice:

- W przypadku zapytania `DROP DATABASE` serwer używa jedynie plików rozpoznanych na podstawie rozszerzeń pliku jako tabele lub inne obiekty bazy danych. Jeżeli w katalogu bazy danych utworzyłeś inne pliki lub podkatalogi, serwer pozostawi je nietknięte. W takim przypadku katalog bazy danych nie może być usunięty, a zapytanie `DROP DATABASE` wygeneruje komunikat błędu. Jedną z konsekwencji jest, że zapytanie `SHOW DATABASES` nadal będzie wyświetlało nazwę tej bazy danych. Rozwiązanie takiego problemu polega na ręcznym usunięciu wszystkich plików w podkatalogu bazy danych, a następnie ponownym wykonaniu zapytania `DROP DATABASE`.
- Nie można bezpiecznie usunąć tabel InnoDB w bazie danych przez usunięcie jej katalogu. Dla każdej tabeli InnoDB silnik InnoDB ma odpowiedni wpis w systemowej przestrzeni tabel, która może również przechowywać zawartość tabeli. Jeżeli baza danych zawiera tabele InnoDB, konieczne jest wykonanie zapytania `DROP DATABASE`, aby silnik InnoDB mógł uaktualnić katalog danych i usunąć całą zawartość tabeli z przestrzeni tabel.

11.2.3. Przedstawienie tabel w systemie plików

MySQL obsługuje wiele silników bazy danych, między innymi InnoDB, MyISAM i MEMORY. Na dysku każda tabela MySQL jest przedstawiana przynajmniej za pomocą jednego pliku: to plik w formacie *.frm*, zawierający opis struktury tabeli. Serwer tworzy plik *.frm*, a poszczególne silniki bazy danych mogą tworzyć dodatkowe pliki zawierające dane rekordów i informacje o indeksach. Nazwy i struktura wspomnianych plików zależą od konkretnego silnika bazy danych.

Poniżej przedstawiono ogólne cechy charakterystyczne pewnych silników bazy danych w zakresie przechowywania plików na dysku. Więcej informacji szczegółowych dotyczących różnic między omawianymi silnikami bazy danych znajdziesz w punkcie 2.6.1, zatytułowanym „Cechy charakterystyczne silników bazy danych”.

Domyślnym silnikiem bazy danych jest InnoDB. W katalogu bazy danych dla każdej tabeli InnoDB znajduje się plik w formacie *.frm*, zawierający definicję struktury tabeli. Z kolei dla danych InnoDB stosuje dwa sposoby ich przedstawienia, oba oparte na przestrzeniach tabel:

- **Systemowa przestrzeń tabel.** Ta przestrzeń tabel składa się z jednego lub więcej ogromnych plików umieszczonych w katalogu danych. Wspomniane pliki przestrzeni tabel tworzą logiczną, ciągłą przestrzeń o wielkości równej sumie wielkości poszczególnych plików. Domyślnie, InnoDB przechowuje tabele w systemowej przestrzeni tabel. Dla tego rodzaju tabel jedynym plikiem charakterystycznym dla tabeli jest plik w formacie *.frm*.
- **Poszczególne przestrzenie tabel.** Istnieje możliwość konfiguracji silnika InnoDB w taki sposób, aby dla każdej tabeli tworzona była oddzielna przestrzeń tabel. W takim przypadku każda tabela InnoDB ma dwa charakterystyczne dla siebie pliki w katalogu bazy danych: plik *.frm* oraz plik *.ibd*, zawierający dane tabeli i jej indeksy.

Systemowa przestrzeń tabel jest używana także w innym celu. InnoDB zawiera wewnętrzny katalog danych przechowujący informacje o wszystkich tabelach. Wspomniany katalog znajduje się w systemowej przestrzeni tabel, która tym samym jest konieczna nawet w przypadku używania poszczególnych przestrzeni tabel do przechowywania zawartości pojedynczych tabel.

Każda tabela MyISAM jest przez MySQL przedstawiana w postaci trzech plików umieszczonych w katalogu bazy danych zawierającej daną tabelę. Nazwa bazowa każdego pliku odpowiada nazwie tabeli, natomiast jego rozszerzenie wskazuje przeznaczenie pliku. Na przykład, tabela MyISAM o nazwie *moja_tabela* ma trzy następujące pliki:

- *moja_tabela.frm* to plik formatu zawierający opis struktury tabeli;
- *moja_tabela.myd* to plik danych przechowujący zawartość rekordów tabeli;
- *moja_tabela.myi* to plik zawierający informacje o indeksach dla wszystkich indeksów utworzonych w danej tabeli.

Tabele typu MEMORY są obszarami w pamięci. Dla tabeli typu MEMORY w katalogu bazy danych znajduje się jedynie plik *.frm*, opisujący jej format. Tego rodzaju tabela nie jest w żaden inny sposób przedstawiana w systemie plików, ponieważ wszystkie dane i indeksy tabeli MEMORY serwer przechowuje w pamięci, a nie na dysku. Po zamknięciu serwera cała zawartość tabel typu MEMORY jest bezpowrotnie tracona. Po ponownym uruchomieniu serwera tabela typu MEMORY nadal istnieje (ponieważ w systemie plików jest jej plik *.frm*), ale pozostaje pusta.

11.2.4. Przedstawienie widoków i wyzwalaczy w systemie plików

Widoki i wyzwalacze są obiektami powiązаныmi z plikami w katalogu bazy danych zawierającym te obiekty.

Widok składa się z pliku *.frm* zawierającego definicję widoku oraz inne powiązane z nim atrybuty. Bazowa nazwa pliku widoku odpowiada nazwie widoku. Dlatego też widok *mój_widok* jest przedstawiany przez plik *mój_widok.frm*.

Wyzwalacz jest przechowywany w pliku *.trg*, zawierającym definicję wyzwalacza oraz inne powiązane z nim atrybuty. Plik wyzwalacza ma nazwę bazową odpowiadającą tabeli, do której należy. Na przykład, wyzwalacz o nazwie *mój_wyzwalacz* powiązany z tabelą *moja_tabela* jest przechowywany w pliku *moja_tabela.trg*, a nie *mój_wyzwalacz.trg*. Jeżeli tabela zawiera wiele wyzwalaczy, serwer przechowuje ich definicje w tym samym pliku *.trg*. Ponadto, każdy wyzwalacz ma plik *.trn* o nazwie wyzwalacza i zawierający nazwę tabeli, w której został zdefiniowany. Na przykład, wyzwalacz *mój_wyzwalacz* ma plik *mój_wyzwalacz.trn* zawierający nazwę tabeli *moja_tabela*.

11.2.5. Jak zapytania SQL są mapowane na operacje na pliku tabeli?

Każdy silnik bazy danych używa pliku *.frm* do przechowywania formatu (definicji) tabeli, aby dane wyjściowe zapytania `SHOW TABLES nazwa_bazy_danych` były takie same jak nazwy bazowe plików *.frm* w katalogu bazy danych o wskazanej nazwie.

W celu utworzenia tabeli dowolnego typu obsługiwanego przez MySQL należy wykonać zapytanie `CREATE TABLE` definiujące strukturę tabeli i zawierające klauzulę `ENGINE=nazwa_silnika` wskazującą używany silnik bazy danych. W przypadku pominięcia klauzuli `ENGINE` serwer MySQL użyje domyślnego silnika bazy danych (InnoDB, o ile tego nie zmieniłeś). Serwer tworzy plik *.frm* dla nowej tabeli i umieszcza w nim wewnętrznie zakodowaną definicję tabeli oraz nakazuje odpowiedniemu silnikowi bazy danych utworzenie wszelkich plików powiązanych z tabelą. Na przykład, InnoDB tworzy odpowiedni wpis w katalogu danych oraz inicjalizuje w odpowiedniej przestrzeni tabel informacje o danych i indeksach tabeli. Z kolei silnik MyISAM tworzy pliki danych (*.myd*) i indeksów (*.myi*), natomiast silnik CSV tworzy plik danych w formacie *.csv*. W systemach UNIX właścicielem wszystkich utworzonych w ten sposób plików przedstawiających tabelę jest użytkownik, którego konto jest używane do uruchamiania serwera.

Po wykonaniu zapytania ALTER TABLE serwer ponownie koduje plik *.frm* tabeli w celu odzwierciedlenia strukturalnych zmian wprowadzonych przez zapytanie i odpowiednio modyfikuje zawartość tabeli (dane i indeksy). To samo dzieje się również w przypadku zapytań CREATE INDEX i DROP INDEX, ponieważ są one obsługiwane przez serwer jako odpowiedniki zapytań ALTER TABLE. Jeżeli zapytanie ALTER TABLE powoduje zmianę silnika bazy danych, zawartość tabeli jest transferowana do nowego silnika, który ponownie zapisuje tę zawartość na dysku, używając odpowiedniego typu plików stosowanych do przedstawienia tabeli.

MySQL implementuje zapytanie DROP TABLE przez usunięcie plików przedstawiających tabelę. Jeżeli usuniesz tabelę InnoDB, silnik bazy danych InnoDB uaktualnia także jej katalog danych i oznacza jako wolną całą przestrzeń, która w systemowej przestrzeni tabel InnoDB jest powiązana z usuniętą tabelą.

W przypadku innych silników bazy danych, na przykład MyISAM, istnieje możliwość usunięcia tabeli przez ręczne usunięcie w katalogu bazy danych plików odpowiadających danej tabeli. W silnikach takich jak InnoDB lub MEMORY pewne fragmenty tabeli mogą nie być przedstawiane w plikach charakterystycznych dla danej tabeli, a więc zapytanie DROP TABLE nie ma odpowiednika w postaci polecenia systemu plików. Na przykład, tabela InnoDB przechowywana w systemowej przestrzeni tabel zawsze jest unikalnie powiązana z plikiem *.frm*, ale usunięcie wymienionego pliku nie powoduje całkowitego usunięcia danej tabeli. Katalog danych InnoDB musi być uaktualniony przez sam silnik InnoDB, a usunięcie pliku *.frm* pozostawia dane i indeksy tabeli „porzucone” w systemowej przestrzeni tabel.

Jeżeli tabela InnoDB znajduje się w oddzielnej przestrzeni tabel, w katalogu bazy danych jest przedstawiana w postaci plików *.frm* i *.ibd*. Jednak nadal nie można „usunąć” tabeli przez usunięcie wymienionych plików, ponieważ wtedy silnik InnoDB nie ma szansy na uaktualnienie katalogu danych. Konieczne jest użycie zapytania DROP TABLE, aby silnik InnoDB mógł usunąć pliki oraz uaktualnić katalog danych.

11.2.6. Ograniczenia systemu operacyjnego w zakresie nazw obiektów bazy danych

MySQL ma ogólne reguły dotyczące identyfikatorów dla nazw baz danych oraz innych obiektów, takich jak tabele. Wspomniane reguły zostały omówione w podrozdziale 2.2, zatytułowanym „Identyfikatory składni MySQL i reguły nadawania nazw”, ale poniżej pokrótce przedstawiono ich podsumowanie:

- Niecytowane identyfikatory mogą składać się z liter łacińskich od a do z o dowolnej wielkości, cyfr od 0 do 9, znaków dolara i podkreślenia oraz rozszerzonych znaków Unicode z zakresu od U+0080 do U+FFFF.
- Identyfikatory cytowane za pomocą odwróconych apostrofów mogą zawierać także inne znaki, na przykład `dziwna?nazwa!`. Cytowanie jest również konieczne, gdy jako identyfikator użyte zostało słowo zarezerwowane SQL. Po włączeniu trybu SQL o nazwie ANSI_QUOTES identyfikator może być cytowany za pomocą odwróconych apostrofów lub ujęty w cudzysłów.
- Identyfikator może mieć maksymalnie 64 znaki długości.

Poza wymienionymi regułami system operacyjny, w którym został uruchomiony serwer MySQL, może nakładać inne ograniczenia na identyfikatory. Wynikają one z konwencji nazw w systemie plików, ponieważ nazwy baz danych i tabel odpowiadają nazwom katalogów i plików. Każda baza danych jest przedstawiana w systemie plików za pomocą jej katalogu, natomiast każda tabela, niezależnie od używanego silnika bazy danych, jest reprezentowana w systemie plików przynajmniej przez plik *.frm*. Dlatego też zastosowanie mają następujące ograniczenia:

- MySQL pozwala, aby nazwa bazy danych lub tabeli składała się z maksymalnie 64 znaków, ale długość nazwy może być również ograniczona przez używany system plików.
- Rozróżnianie wielkości plików przez system plików wpływa na sposób nadawania nazw bazom danych i plikom oraz odwoływania się do nich. Jeżeli system plików rozróżnia wielkość liter (jak to zwykle ma miejsce w systemach UNIX), nazwy *abc* i *ABC* odwołują się do różnych plików. Jeśli system plików nie rozróżnia wielkości liter (jak ma to miejsce w systemach Windows i Mac OS X Extended), wtedy nazwy *abc* i *ABC* odwołują się do tego samego pliku. Powinieneś o tym pamiętać, tworząc bazę danych w serwerze rozpoznającym wielkość liter w nazwach plików, ponieważ istnieje prawdopodobieństwo, że będziesz ją przerosił lub replikował do serwera, w którym wielkość liter w nazwach plików nie ma znaczenia.

Serwer koduje w identyfikatorach znaki specjalne, które mogą sprawiać problemy w nazwach plików. Wspomniane kodowanie pozwala na użycie znaków takich jak */* i ** w nazwach występujących w zapytaniach SQL. Dowolny znak spoza zakresu cyfr i liter łacińskich jest mapowany w nazwie pliku na znak *@*, po którym znajduje się zakodowana wartość znaku. Na przykład, znaki *?* i *!* mają kody 003f i 0021, a więc struktura tabeli o nazwie *dziwna?nazwa!* będzie zapisana w pliku *.frm* o nazwie *dziwna@003fnazwa@0021.frm*. Pozostałe pliki powiązane z tabelą będą miały podobne nazwy.

Jak wcześniej wspomniano, rozróżnianie wielkości liter przez system plików wpływa na nazewnictwo baz danych i tabel. Jednym z rozwiązań jest używanie zawsze nazw o określonej wielkości liter. Inne rozwiązanie polega na uruchomieniu serwera wraz z przypisaną wartością 1 zmiennej systemowej *lower_case_table_names*, co ma dwa efekty:

- Serwer konwertuje nazwę tabeli na zapisaną małymi literami i dopiero wtedy tworzy odpowiadające jej pliki na dysku.
- Później, podczas odwoływania się do tabeli w zapytaniu, serwer konwertuje jej nazwę na zapisaną małymi literami i dopiero wtedy próbuje odszukać na dysku plik tabeli.

Wynikiem powyższych działań jest traktowanie nazw jako nierozróżniających wielkości liter, niezależnie od rozróżniania wielkości liter przez system plików. Dzięki temu można łatwiej przenosić bazy danych i tabele między systemami. Jednak jeśli planujesz zastosowanie przedstawionej strategii, konfigurację zmiennej *lower_case_table_names* musisz przeprowadzić *przed* rozpoczęciem tworzenia baz danych i tabel, a nie po.

Jeżeli wymienioną zmienną ustawisz już po utworzeniu baz danych lub tabel zawierających w nazwach wielkie litery, zmienna nie przyniesie oczekiwanego efektu, ponieważ nazwy plików na dysku nie są w całości zapisane za pomocą małych liter. Aby uniknąć wspomnianego problemu, nazwy wszystkich tabel, które zawierają wielkie litery, zmień na całkowicie małe i dopiero wtedy ustaw wartość zmiennej `lower_case_table_names`. (W celu zmiany nazwy tabeli można wykonać zapytanie `ALTER TABLE` lub `RENAME TABLE`). W przypadku dużej liczby tabel wymagających zmiany nazwy lub baz danych zawierających w nazwach wielkie litery łatwiejszym rozwiązaniem jest utworzenie kopii zapasowej zawartości tych baz danych, a następnie ich ponowne utworzenie już po ustawieniu zmiennej `lower_case_table_names`:

1. Za pomocą narzędzia `mysqldump` utwórz kopię zapasową zawartości wszystkich baz danych:

```
% mysqldump --databases nazwa_bazy_danych > nazwa_bazy_danych.sql
```

2. Usuń wszystkie bazy danych, wykonując zapytania `DROP DATABASE`.
3. Zatrzymaj serwer, przekonfiguruj go przez przypisanie wartości 1 zmiennej systemowej `lower_case_table_names`, a następnie ponownie uruchom serwer.
4. Za pomocą klienta `mysql` wczytaj zawartość wszystkich utworzonych wcześniej plików kopii zapasowych:

```
% mysql < nazwa_bazy_danych.sql
```

Po ustawieniu zmiennej `lower_case_table_names` wszystkie bazy danych i tabel zostaną ponownie utworzone i zapisane na dysku w plikach o nazwach składających się z małych liter.

Zmiennej `lower_case_table_names` można przypisać wiele wartości, jak to omówiono w dodatku D, zatytułowanym „Przewodnik po zmiennych systemowych, stanu i użytkownika”.

11.2.7. Czynniki wpływające na maksymalną wielkość tabeli

Wielkość tabeli w MySQL jest ograniczona, ale na to wpływa połączenie wielu czynników. Dlatego też nie zawsze można precyzyjnie wskazać te granice.

System operacyjny narzuca granicę maksymalnej wielkości pliku. Wspomniane ograniczenie ma zastosowanie względem plików przedstawiających tabele, na przykład plików `.myd` i `.myi` dla tabeli MyISAM, a także plików tworzących dowolną przestrzeń tabel InnoDB. Jednak ogólne wymagania w zakresie wielkości systemowej przestrzeni tabel InnoDB mogą bardzo łatwo przekroczyć maksymalną dopuszczalną wielkość pliku. Rozwiązaniem jest skonfigurowanie przestrzeni tabel jako składającej się z wielu plików, z których każdy może osiągnąć wielkość maksymalną. Innym sposobem pokonania ograniczenia w postaci maksymalnej wielkości plików jest wykorzystanie niezmodyfikowanych partycji jako przestrzeni tabel InnoDB. Komponenty przestrzeni tabel znajdujące się na niezmodyfikowanych partycjach mogą osiągnąć rozmiary samej partycji. Informacje

dotyczące konfiguracji tego rodzaju rozwiązania znajdziesz w podpunkcie 12.5.3.1, zatytułowanym „Konfiguracja przestrzeni tabel InnoDB”.

Poza ograniczeniami narzucanymi przez system operacyjny, MySQL ma własne, wewnętrzne ograniczenia wielkości tabel. Wspomniane ograniczenia zależą od używanego silnika bazy danych:

- W przypadku InnoDB maksymalna wielkość systemowej przestrzeni tabel InnoDB wynosi 4 miliardy stron o wielkości 16 KB. Maksymalna wielkość przestrzeni tabel jest związana także z wielkością poszczególnych przechowywanych w niej tabel InnoDB. Jeżeli silnik jest zdefiniowany w sposób powodujący użycie oddzielnych przestrzeni tabel, zawartość poszczególnych tabel InnoDB jest przechowywana w plikach *.ibd*. W takim przypadku na maksymalną wielkość tabeli wpływ mają nakładane przez system operacyjny ograniczenia związane z maksymalną wielkością pliku.
- W przypadku MyISAM wielkość plików *.myd* i *.myi* jest domyślnie ograniczona do 256 TB. Jednak jeśli w trakcie tworzenia tabeli zostaną użyte opcje `AVG_ROW_LENGTH` i `MAX_ROWS`, to wymienione pliki mogą mieć wielkość do 65 536 TB. (Zapoznaj się z opisem zapytania `CREATE TABLE` w dodatku E, zatytułowanym „Przewodnik po składni SQL”). Wymienione opcje wpływają na wielkość wewnętrznego wskaźnika, który określa maksymalną liczbę rekordów, które może przechowywać tabela. Gdy tabela MyISAM osiągnie swoją wielkość maksymalną i zacznie pojawiać się błędy o kodzie 135 i 136 dla operacji na tabeli, użyj zapytania `ALTER TABLE` do zwiększenia wartości wymienionych opcji. Aby bezpośrednio zmienić domyślną wielkość wskaźnika MyISAM, należy ustawić zmienną systemową `mysam_data_pointer_size`. Nowa wartość tej zmiennej ma zastosowanie również dla wcześniej utworzonych tabel.

W przypadku silników bazy danych przedstawiających dane i indeksy w oddzielnych plikach maksymalna wielkość tabeli zostaje osiągnięta, gdy dowolny z tworzących ją plików będzie miał maksymalną dopuszczalną wielkość. Dla tabeli MyISAM cechy charakterystyczne indeksowania wpływają na plik, który pierwszy osiągnie limit. Jeżeli tabela ma niewiele lub w ogóle nie ma indeksów, plik danych prawdopodobnie pierwszy osiągnie wielkość maksymalną. Z kolei w przypadku tabeli posiadającej wiele indeksów plik indeksu może być tym, który jako pierwszy osiągnie wielkość maksymalną.

Obecność kolumny `AUTO_INCREMENT` wyraźnie ogranicza liczbę rekordów, które mogą znaleźć się w tabeli. Na przykład, jeśli wymieniona kolumna została zdefiniowana jako typu `TINY UNSIGNED`, jej maksymalną wartością jest 255 i to jednocześnie jest maksymalna liczba rekordów, które mogą być przechowywane przez tabelę. Większe typy liczb całkowitych pozwalają na przechowywanie większej liczby rekordów. Ogólnie rzecz biorąc, zdefiniowanie w tabeli indeksu `PRIMARY KEY` lub `UNIQUE` ogranicza liczbę rekordów tej tabeli do maksymalnej liczby unikalnych wartości, jakie mogą być przechowywane przez utworzony indeks.

W celu określenia rzeczywistej wielkości tabeli, jaką można uzyskać, pod uwagę trzeba wziąć wiele czynników. Efektywna maksymalna wielkość tabeli będzie prawdopodobnie

wartością najmniejszego z wspomnianych czynników. Przyjmujemy założenie, że chcesz utworzyć tabelę typu MyISAM. MySQL określa maksymalną wielkość plików danych i indeksów na 256 TB każdy przy użyciu wskaźnika o domyślnej wielkości. Jednak jeśli system operacyjny nakłada ograniczenie wielkości pliku do 2 GB, to będzie efektywna wielkość maksymalna dla poszczególnych plików tabeli. Z drugiej strony, jeśli system plików obsługuje pliki o wielkości większej niż 256 TB, czynnikiem decydującym o maksymalnej wielkości tabeli będzie czynnik MySQL, a dokładnie wielkość jego wewnętrznego wskaźnika danych. To jest czynnik, nad którym masz kontrolę.

W związku z przechowywaniem tabel InnoDB w systemowej przestrzeni tabel pojedyncza tabela InnoDB może osiągnąć wielkość przestrzeni tabel, która z kolei może być utworzona z wielu plików, aby zapewnić odpowiednią wielkość do pomieszczenia wszystkich tabel. Jeśli (co jest bardzo prawdopodobne) masz wiele tabel InnoDB, wszystkie współdzielą tę samą przestrzeń i tym samym są ograniczone nie tylko wielkością przestrzeni tabel, ale również miejscem zajmowanym w niej przez inne table. Pojedyncza tabela InnoDB może zwiększać swoją wielkość, o ile przestrzeń tabel nie jest zapełniona. Po wykorzystaniu całej przestrzeni tabel żadna tabela nie może się zwiększyć aż do chwili dodania do przestrzeni tabel kolejnego komponentu, który w ten sposób ją powiększy. Alternatywne rozwiązanie polega na zastosowaniu automatycznie rozszerzającego się komponentu przestrzeni tabel. W takim przypadku komponent zwiększa swoją wielkość, dopóki nie osiągnie maksymalnej dozwolonej wielkości pliku w systemie operacyjnym lub nie wykorzysta całej dostępnej pamięci masowej. Informacje dotyczące konfiguracji przestrzeni tabel znajdziesz w podpunkcie 12.5.3.1, zatytułowanym „Konfiguracja przestrzeni tabel InnoDB”.

11.2.8. Wpływ struktury katalogu danych na wydajność systemu

Struktura katalogu danych MySQL jest łatwa do zrozumienia, ponieważ w naturalny sposób wykorzystuje hierarchiczną strukturę systemu plików. Jednocześnie wspomniana struktura wiąże się z pewnymi implikacjami w zakresie wydajności, szczególnie podczas wykonywania operacji otwierania plików przedstawiających table bazy danych.

W przypadku silników bazy danych przedstawiających poszczególne table w postaci własnych plików każde otworenie tabeli może wymagać deskryptora pliku. Jeżeli tabela jest przedstawiana za pomocą wielu plików, jej otworenie wymaga wielu deskryptorów plików, a nie tylko jednego. Serwer w sprytny sposób buforuje deskryptory plików, ale bardzo obciążony serwer może bardzo łatwo wykorzystywać wiele z nich podczas obsługi wielu jednoczesnych połączeń z klientami lub w trakcie wykonywania skomplikowanych zapytań odwołujących się do wielu tabel. To może stanowić poważny problem, ponieważ deskryptory plików szybko wyczerpują zasoby w wielu systemach, przede wszystkim w tych, które domyślnie mają ustawioną niską wartość deskryptorów plików. System operacyjny nakładający niskie ograniczenie na liczbę deskryptorów plików i niezwiększający ich liczby nie jest dobrym kandydatem do uruchamiania w nim bardzo obciążonego serwera MySQL.

Innym efektem przedstawiania każdej tabeli w postaci oddzielnych plików jest wydłużenie czasu otwierania tabeli wraz ze wzrostem liczby tabel. Operacje otwarcia tabeli są mapowane

na dostarczane przez system operacyjny operacje otwarcia plików. Efektywność wymienionych operacji zależy od efektywności procedur systemowych odpowiedzialnych za przeszukiwanie katalogu. Normalnie to nie jest żaden problem, ale sytuacja ulega zmianie, jeśli w bazie danych znajduje się ogromna liczba tabel. Na przykład, tabela MyISAM jest przedstawiana przez trzy pliki. Jeśli wymagane jest użycie 10 000 tabel MyISAM, wtedy katalog bazy danych będzie zawierał 30 000 plików. W przypadku tak dużej liczby plików na pewno zauważysz spowolnienie wynikające z czasu potrzebnego na przeprowadzenie operacji otwierania plików. Jeżeli to Cię martwi, rozważ zastosowanie systemu plików charakteryzującego się wysoką wydajnością podczas obsługi ogromnej ilości plików. Na przykład, XFS lub JFS zapewniają dobrą wydajność działania, nawet w przypadku ogromnej liczby małych plików. Jeśli zastosowanie innego systemu plików nie jest możliwe, wtedy konieczne może być ponowne przemyślenie struktury tabel względem wymagań aplikacji i odpowiednia zmiana tej struktury. Przede wszystkim odpowiedz sobie na pytanie, czy naprawdę konieczne jest używanie tak dużej liczby tabel; czasami aplikacja niepotrzebnie zwiększa ich liczbę. Aplikacja może na przykład tworzyć oddzielną tabelę dla każdego wyniku przeprowadzanej przez użytkownika operacji wyszukiwania wielu tabel, a wszystkie wspomniane tabele wynikowe będą miały identyczne struktury. Jeżeli spróbujesz je połączyć w pojedynczą tabelę, może się to okazać możliwe po dodaniu kolejnej kolumny identyfikującej użytkownika, którego dotyczy dany rekord. Jeśli takie rozwiązanie zmniejszy liczbę tabel, wydajność działania aplikacji niewątpliwie wzrośnie.

W przypadku każdego projektu bazy danych trzeba sprawdzić, czy określona strategia jest warta zastosowania w danej aplikacji. Poniżej wymieniono powody, dla których nie powinno się łączyć tabel w przedstawiony wcześniej sposób:

- Większa ilość wymaganego miejsca na dysku. Połączenie tabel zmniejsza całkowitą liczbę wymaganych tabel (skrócenie czasu otwierania tabel), ale wiąże się z dodaniem kolejnej kolumny (zwiększenie ilości wymaganego miejsca na dysku). To jest typowy kompromis — czas otwarcia pliku tabeli kontra ilość wymaganej pamięci masowej; samodzielnie musisz zdecydować, który z wymienionych czynników jest najważniejszy. Jeżeli ważna jest szybkość, to prawdopodobnie poświęcisz nieco dodatkowego miejsca na dysku twardym. Z kolei jeśli ilość pamięci masowej jest ograniczona, wtedy akceptowalne może być użycie większej liczby tabel i nieco większe opóźnienie w działaniu.
- Kwestie dotyczące bezpieczeństwa. Mogą one wpływać na możliwość łączenia tabel. Jednym z powodów stosowania oddzielnych tabel dla poszczególnych użytkowników jest zapewnienie dostępu do tabeli tylko dla pojedynczego konta MySQL i stosowania tym samym uprawnień na poziomie tabeli. Po połączeniu tabel dane wszystkich użytkowników będą znajdowały się w tej samej tabeli. MySQL nie ma systemu ograniczającego danemu użytkownikowi dostęp do określonych rekordów. Dlatego też połączenie tabel nie będzie mogło odbyć się bez złamania reguł kontroli dostępu. Jednym z możliwych rozwiązań jest użycie widoków pobierających rekordy dla bieżącego użytkownika i nadających uprawnienia dostępu poprzez wspomniane widoki. Ewentualnie, jeśli cały

dość do danych jest kontrolowany przez aplikację (użytkownik nigdy nie nawiązuje bezpośredniego połączenia z bazą danych), masz możliwość połączenia tabel i użycia logiki aplikacji w celu wymuszenia stosowania reguł dostępu na poziomie rekordów dla zwróconego wyniku.

Innym sposobem utworzenia wielu tabel bez konieczności tworzenia wielu poszczególnych plików jest wykorzystanie tabel InnoDB i przechowywanie ich w systemowej przestrzeni tabel. W takim przypadku silnik InnoDB z każdą tabelą powiązuje jedynie plik *.frm*, natomiast dane i indeksy wszystkich tabel InnoDB są już przechowywane razem. W ten sposób minimalizuje się liczbę plików na dysku wymaganych do przedstawienia tabel, a tym samym znacznie zmniejsza liczbę deskryptorów plików wymaganych do otwarcia tabel. InnoDB potrzebuje tylko jednego deskryptora dla pliku komponentu przestrzeni tabel (który na dodatek nie ulega zmianie w trakcie istnienia procesu serwera) i krótko mówiąc, deskryptor dla otwieranej przez niego tabeli jest odczytywany wraz z plikiem *.frm* tabeli.

11.2.9. Pliki dzienników zdarzeń i stanu MySQL

Poza podkatalogami baz danych, katalog danych MySQL zawiera także liczbę plików dzienników zdarzeń oraz stanu MySQL, które wymieniono w tabeli 11.1. Domyślnym położeniem wszystkich plików wymienionych w tabeli 11.1 jest katalog danych serwera, a nazwa domyślna wielu z nich pochodzi od nazwy komputera, w którym działa serwer, oznaczonej w tabeli jako *HOSTNAME*. Dzienniki zdarzeń binarne i przekazywania są tworzone w postaci ponumerowanej sekwencji plików, co zostało oznaczone jako *nnnnn*. W tabeli wymieniono jedynie pliki dzienników zdarzeń i stanu używane na poziomie serwera. Poszczególne silniki bazy danych również mogą tworzyć własne dzienniki zdarzeń oraz inne pliki. Na przykład, takie rozwiązanie jest stosowane przez InnoDB.

Tabela 11.1. Pliki dzienników zdarzeń i stanu MySQL

Typ pliku	Nazwa domyślna	Zawartość pliku
Plik PID	<i>HOSTNAME</i> .pid	Identyfikator procesu serwera.
Dziennik błędów	<i>HOSTNAME</i> .err	Zdarzenia podczas uruchamiania i zamykania serwera oraz ewentualne błędy.
Ogólny dziennik zapytań	<i>HOSTNAME</i> .log	Zdarzenia podczas nawiązywania i zamykania połączenia z serwerem oraz informacje o zapytaniach.
Dziennik binarny	<i>HOSTNAME</i> -bin. <i>nnnnn</i>	Binarny format zapytań modyfikujących dane.
Indeks dziennika binarnego	<i>HOSTNAME</i> -bin.index	Lista aktualnych nazw plików binarnych dzienników zdarzeń.

Tabela 11.1. Pliki dzienników zdarzeń i stanu MySQL (ciąg dalszy)

Typ pliku	Nazwa domyślna	Zawartość pliku
Dziennik przekazywania	<i>HOSTNAME-relay-bin.nnnnnn</i>	Daty modyfikacji otrzymanych przez serwer podległy z serwera głównego.
Indeks dziennika przekazywania	<i>HOSTNAME-relay-bin.index</i>	Lista aktualnych nazw plików dzienników przekazywania.
Plik informacyjny serwera głównego	<i>master.info</i>	Parametry nawiązania połączenia z serwerem głównym.
Plik informacyjny serwera podległego	<i>relay-log.info</i>	Stan przetwarzania dziennika przekazywania.
Dziennik wolnych zapytań	<i>HOSTNAME-slow.log</i>	Tekst zapytań, których przetworzenie wymaga dużej ilości czasu.

W przypadku dzienników zdarzeń ogólnego i wolno wykonywanych zapytań można zdecydować, czy serwer ma zapisywać zdarzenia w pliku dziennika, w tabeli bazy danych `mysql`, czy w obu wymienionych miejscach. Szczegółowe omówienie rejestracji zdarzeń w tabelach przedstawiono w punkcie 12.8.6, zatytułowanym „Rejestracja zdarzeń w tabelach”.

11.2.9.1. Plik PID

W chwili uruchamiania serwera zapisuje on w pliku PID identyfikator procesu (PID), natomiast podczas zamykania serwera MySQL wspomniany plik jest usuwany. Inne procesy mogą wykorzystać plik PID do ustalenia, czy serwer działa oraz jaki jest jego identyfikator procesu. Na przykład, jeśli w trakcie zamykania systemu operacyjnego uruchamia on skrypt `mysql.server` w celu zamknięcia serwera MySQL, skrypt analizuje plik PID i ustala tym samym proces, któremu trzeba wysłać sygnał zakończenia działania.

Jeżeli serwer nie będzie mógł utworzyć pliku PID, w dzienniku zdarzeń umieści odpowiedni komunikat błędu i będzie kontynuował działanie.

11.2.9.2. Dzienniki zdarzeń MySQL

Serwer MySQL może obsługiwać wiele typów plików dzienników zdarzeń. Większość operacji rejestrowania zdarzeń jest opcjonalna. Opcje podczas uruchamiania serwera można wykorzystać do włączenia jedynie potrzebnych dzienników zdarzeń i nadania im nazw, jeśli nie odpowiadają Ci ich nazwy domyślne. Pamiętaj, że wraz z upływem czasu pliki dzienników zdarzeń mogą osiągnąć ogromne rozmiary. Dlatego bardzo ważne jest uniemożliwienie im zapelnienia systemu plików. Aby zachować pod kontrolą ilość miejsca zajmowanego przez pliki dzienników zdarzeń, co pewien czas należy usuwać stare.

W tym punkcie pokrótce zostaną omówione niektóre pliki dzienników zdarzeń. Więcej informacji na temat dzienników zdarzeń oraz opcji kontrolujących zachowanie serwera w zakresie rejestracji zdarzeń i usuwania starych plików dzienników znajdziesz w podrozdziale 12.8, zatytułowanym „Dzienniki zdarzeń serwera”.

Dziennik zdarzeń błędów zawiera informacje diagnostyczne wygenerowane przez serwer po wystąpieniu zdarzenia wyjątkowego. Jeżeli próba uruchomienia serwera zakończy się niepowodzeniem lub serwer niespodziewanie zakończy działanie, ten dziennik zdarzeń okaże się bardzo użyteczny, ponieważ często zawiera informacje o przyczynach problemów.

Ogólny dziennik zdarzeń zawiera ogólne informacje o działaniu serwera: kto i skąd nawiązał połączenie oraz jakie wykonał zapytania. Binarny dziennik zdarzeń również zawiera informacje o zapytaniach, ale jedynie tych modyfikujących zawartość bazy danych. Ponadto, zawiera informacje takie jak znaczniki czasu wymagane do zapewnienia synchronizacji serwerów podległych z głównym po włączeniu replikacji. Zawartość binarnego dziennika zdarzeń jest zapisywana w formacie binarnym, jako „zdarzenia”, które mogą być wykonane i dostarczyć dane wejściowe dla klienta mysql. Towarzyszący dziennikowi binarnemu plik indeksu zawiera listę plików binarnych dzienników zdarzeń aktualnie używanych przez serwer.

Binarny dziennik zdarzeń jest użyteczny w przypadku wystąpienia awarii i konieczności przywrócenia danych z plików kopii zapasowej, ponieważ wtedy pozwala na odtworzenie zmian wprowadzonych po utworzeniu ostatniej kopii zapasowej. W ten sposób bazy danych zostają przywrócone do stanu, w którym znajdowały się w chwili awarii. Binarny dziennik zdarzeń jest używany także po włączeniu replikacji. W takiej konfiguracji działa w charakterze źródła uaktualnień, które muszą być przekazywane z serwera głównego do podległych. Więcej informacji na ten temat znajdziesz w rozdziale 14., zatytułowanym „Obsługa bazy danych, kopie zapasowe i replikacja”.

Dobrym rozwiązaniem jest zagwarantowanie, że pliki dzienników zdarzeń pozostaną bezpieczne i nie są dostępne dla zwykłych użytkowników, ponieważ mogą zawierać tekst zapytań obejmujących informacje wrażliwe, takie jak hasła. Na przykład, przedstawiony poniżej wpis w dzienniku zdarzeń zawiera hasło dla użytkownika root. Bez wątpienia to jest informacja, której lepiej nie udostępniać innym użytkownikom:

```
080412 16:47:24      44 Query      SET PASSWORD FOR
                    'root'@'localhost'=PASSWORD('secret')
```

Pliki dzienników zdarzeń są przez serwer domyślnie zapisywane w katalogu danych MySQL. Dlatego też dobrym sposobem zabezpieczenia dzienników zdarzeń jest zabezpieczenie samego katalogu danych, do którego dostęp powinien mieć jedynie użytkownik zajmujący się administracją instalacji MySQL. Szczegółowy opis procedury zabezpieczenia katalogu danych przedstawiono w punkcie 13.1.2, zatytułowanym „Zabezpieczenie instalacji MySQL”.

11.3. Przeniesienie zawartości katalogu danych

We wcześniejszej części rozdziału omówiono strukturę katalogu danych MySQL w jego konfiguracji domyślnej, czyli zawierającego wszystkie bazy danych, pliki dzienników zdarzeń oraz stanu. Jednak istnieje możliwość samodzielnego wskazania miejsca położenia zawartości katalogu danych. Serwer MySQL pozwala na zmianę położenia całego katalogu danych lub

tylko jego wybranych elementów. Istnieje kilka powodów, dla których możesz zdecydować się na tego typu rozwiązanie:

- System plików zawierający katalog danych został zapełniony i konieczne jest jego przeniesienie do systemu plików o większej pojemności.
- Jeżeli katalog danych znajduje się na intensywnie używanym dysku, jego przeniesienie na mniej używany pozwala na zrównoważenie aktywności dyskowej między poszczególnymi urządzeniami fizycznymi. Z tych samych powodów w różnych napędach można umieścić pliki baz danych i dzienników zdarzeń lub rozproszyć bazy danych. Podobnie, systemowa przestrzeń tabel InnoDB pod względem koncepcyjnym jest pojedynczym ogromnym blokiem pamięci masowej, ale jej poszczególne komponenty można umieścić w różnych napędach, poprawiając tym samym wydajność działania. Jeśli używane są tabele partycjonowane, takie samo rozwiązanie można zastosować względem poszczególnych partycji tabeli.
- Umieszczenie baz danych na jednym dysku, a dzienników zdarzeń na innym pomaga w minimalizacji uszkodzeń, które mogą powstać na skutek awarii jednego dysku.

W pozostałej części podrozdziału dowiesz się, które elementy katalogu danych mogą być przeniesione oraz jak przeprowadzać tego rodzaju operacje.

11.3.1. Metody przenoszenia katalogu danych lub jego elementów

Mamy dwa sposoby pozwalające na przeniesienie katalogu danych lub znajdujących się w nim elementów.

Pierwszy polega na tym, że na dowolnej platformie można zdefiniować opcje odczytywane przez serwer w chwili jego uruchamiania. Wspomniane opcje można podać w wierszu poleceń lub pliku opcji. Na przykład, aby wskazać położenie katalogu danych, należy uruchomić serwer wraz opcją `--datadir=nazwa_katalogu` w wierszu poleceń lub umieścić poniższe wiersze w pliku opcji:

```
[mysqld]
datadir=nazwa_katalogu
```

Zazwyczaj grupa opcji dla serwera nosi nazwę `[mysqld]`, jak pokazano w powyższym przykładzie. Jednak w zależności od okoliczności odpowiednie mogą być inne nazwy grup opcji. Na przykład, jeśli uruchamiasz wiele serwerów za pomocą `mysqld_multi`, nazwy grup mają postać `[mysqldn]`, gdzie *n* to liczba całkowita przypisana danemu egzemplarzowi serwera. W punkcie 12.2.3, zatytułowanym „Określanie opcji startowych serwera”, dokładnie przedstawiono, które grupy opcji mają zastosowanie do różnych metod uruchamiania serwera, a także zaprezentowano informacje dotyczące uruchamiania wielu serwerów MySQL.

Drugi sposób bazuje na fakcie, że w systemach UNIX można przenieść plik lub cały katalog, a następnie utworzyć dowiązanie symboliczne w położeniu początkowym prowadzące do nowego położenia.

Żadna z wymienionych metod nie jest uniwersalna i nie działa dla każdego elementu, którego położenie można zmienić. W tabeli 11.2 podsumowano możliwości w zakresie przenoszenia katalogu danych lub jego elementów oraz metody możliwe do zastosowania. Jeśli używasz pliku opcji, to opcje możesz podać w globalnym pliku opcji, takim jak `/etc/my.cnf` w systemach UNIX i `C:\my.ini` w Windows.

Tabela 11.2. Podsumowanie dostępnych metod przenoszenia katalogu danych lub jego elementów

Element do przeniesienia	Akceptowalne metody przenoszenia
Cały katalog danych	Opcje startowe lub dowiązanie symboliczne.
Poszczególne podkatalogi baz danych	Dowiązanie symboliczne.
Poszczególne tabele bazy danych	Dowiązanie symboliczne.
Pliki przestrzeni tabel InnoDB	Opcje startowe.
Plik PID	Opcje startowe.
Pliki dzienników zdarzeń	Opcje startowe.

11.3.2. Przygotowania do operacji przeniesienia

Przed podjęciem próby przeniesienia czegokolwiek z katalogu danych MySQL konieczne jest utworzenie jego kopii zapasowej, aby móc przywrócić wymieniony katalog, gdy operacja zakończy się niepowodzeniem. Ponadto, przed przeniesieniem należy zatrzymać serwer MySQL, a po przeprowadzeniu operacji ponownie go uruchomić. W przypadku pewnych rodzajów przeniesienia, na przykład katalogu bazy danych, czasami jest *możliwe* pozostawienie uruchomionego serwera, choć takie rozwiązanie nie jest zalecane. Jeśli się jednak zdecydujesz, to koniecznie upewnij się, że serwer nie uzyskuje dostępu do przenoszonych bazy danych. Przed przeniesieniem bazy danych powinieneś również wykonać zapytanie `FLUSH TABLES`, aby serwer zamknął wszystkie otwarte pliki tabel. Niewykonanie wymienionych zadań może doprowadzić do uszkodzenia tabel.

11.3.3. Uzyskanie dostępu do wyniku przeniesienia

Przed podjęciem próby przeniesienia czegokolwiek z katalogu danych MySQL upewnij się, że operacja przyniesie żądany efekt. Na przykład, w systemie UNIX można użyć poleceń `du`, `df` i `ls -l` w celu sprawdzenia ilości dostępnego miejsca na dysku. Konieczne jest przy tym prawidłowe zrozumienie układu systemu plików, aby wymienione polecenia były użyteczne.

Przedstawiony poniżej przykład pokazuje pułapkę, na którą trzeba uważać podczas przenoszenia katalogu danych MySQL. Przyjmujemy założenie, że katalogiem danych jest `/usr/local/mysql/data` i chcemy go przenieść do katalogu `/var/mysql`, ponieważ wynik działania polecenia `df` pokazuje, że system plików `/var` ma większą ilość wolnej przestrzeni:

```
% df -k /usr /var
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/sda5        28834716K 24078024K  3291968K  88% /usr
/dev/sda6        28834716K  9175456K 18194536K  34% /var
```

Aby przekonać się, ile przestrzeni w systemie plików `/usr` zostanie zwolnione po przeniesieniu katalogu danych, używamy polecenia `du -s`:

```
% du -s /usr/local/mysql/data
3264308K /usr/local/mysql/data
```

Dane wyjściowe pokazują, że przeniesienie katalogu `data` z `/usr` do `/var` spowoduje zwolnienie około 3 GB miejsca w systemie plików `/usr`. Czy na pewno? Aby się tego dowiedzieć, należy użyć polecenia `df` względem katalogu `data`. Załóżmy, że otrzymujemy następujące dane wyjściowe:

```
% df -k /usr/local/mysql/data
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/sda6        28834716K  9175456K 18194536K  34% /var
```

To dziwne. Dlaczego narzędzie `df` zgłasza użycie miejsca w systemie plików `/var`? Odpowiedź na to pytanie dostarczają dane wyjściowe polecenia `ls -l`:

```
% ls -l /usr/local/mysql/data
lrwxrwxr-x 1 mysql mysql 10 Dec 11 23:46 data -> /var/mysql
```

Powyższe dane wyjściowe pokazują, że katalog `/usr/local/mysql/data` jest dowiązaniem symbolicznym do `/var/mysql`. Innymi słowy, katalog danych MySQL już znajduje się w systemie plików `/var` i został zastąpiony przez dowiązanie symboliczne prowadzące do `/var`. Tak więc przeniesienie katalogu danych do systemu plików `/var` nie spowoduje zwolnienia miejsca w systemie `/usr`.

Jeżeli chcesz przenieść bazę danych do innego systemu plików w celu podjęcia próby redystrybucji zajmowanej przez nią przestrzeni na dysku, musisz pamiętać o jednym: w przypadku używania tabel InnoDB przechowywanych w systemowej przestrzeni tabel zawartość tych tabel *nie* jest przechowywana w katalogu bazy danych. Gdy baza danych składa się głównie z tabel InnoDB, zmiana położenia katalogu bazy danych powoduje przeniesienie jedynie plików `.frm` tabel, a nie ich treści. To ma niewielki wpływ na zajmowane miejsce na dysku.

Z przedstawionych przykładów płynie następujący wniosek. Poświęcenie chwili na sprawdzenie efektu przeniesienia katalogu danych MySQL może zapobiec zmarnowaniu znacznie większej ilości czasu na kopiowanie plików tylko po to, aby się przekonać, że żądanego celu nie udało się osiągnąć.

11.3.4. Przeniesienie całego katalogu danych

W celu przeniesienia całego katalogu danych należy zatrzymać serwer MySQL, a następnie przenieść katalog danych do nowego położenia. Po przeniesieniu trzeba uruchomić serwer wraz z opcją `--datadir`, wyraźnie wskazującą nowe położenie. W systemach UNIX alternatywą dla użycia wymienionej opcji jest utworzenie w starym położeniu katalogu danych dowiązania symbolicznego wskazującego nowe położenie.

11.3.5. Przeniesienie poszczególnych baz danych

Serwer zawsze szuka podkatalogów baz danych w katalogu danych MySQL, a więc jedynym sposobem na przeniesienie bazy danych jest utworzenie dowiązania symbolicznego. Procedura różni się w systemach UNIX i Windows.

W systemie UNIX przeniesienie bazy danych trzeba przeprowadzić w następujący sposób:

1. Zatrzymanie serwera, jeśli działa.
2. Przeniesienie katalogu bazy danych do nowego położenia, czyli jego skopiowanie do nowej lokalizacji i usunięcie w początkowej.
3. Utworzenie w katalogu danych MySQL dowiązania symbolicznego o nazwie takiej samej jak oryginalna baza danych i prowadzącego do nowej lokalizacji.
4. Ponowne uruchomienie serwera.

Poniższy przykład pokazuje, jak przenieść bazę danych `bigdb` z katalogu `/usr/local/mysql/data` do `/var/db`:

```
% mysqladmin -p -u root shutdown
Enter password: *****
% cd /usr/local/mysql/data
% tar cf - bigdb | (cd /var/db; tar xf -)
% rm -rf bigdb
% ln -s /var/db/bigdb bigdb
% mysql_safe &
```

Powyższe polecenie trzeba wykonać po zalogowaniu się jako administrator MySQL.

W systemie Windows procedura przeniesienia bazy danych jest nieco inna:

1. Zatrzymanie serwera, jeśli działa.
2. Przeniesienie katalogu bazy danych do nowego położenia, czyli jego skopiowanie do nowej lokalizacji i usunięcie w początkowej.
3. Utworzenie w katalogu danych MySQL pliku działającego w charakterze dowiązania symbolicznego, które pozwala serwerowi MySQL na znalezienie przeniesionego katalogu danych. Wspomniany plik powinien mieć rozszerzenie `.sym` i nazwę bazową odpowiadającą nazwie bazy danych. Na przykład, jeśli przenosisz bazę danych `sampdb` z katalogu `C:\mysql\data\sampdb` do `E:\mysql-book\sampdb`, utwórz plik o nazwie `E:\mysql\data\sampdb.sym` i umieść w nim następujący wiersz:

```
E:\mysql-book\sampdb\
```

4. Upewnij się o włączonej obsłudze dowiązania symbolicznego podczas ponownego uruchamiania serwera. Serwery Windows powinny domyślnie mieć włączoną wspomnianą obsługę dowiązań symbolicznych, ale możesz jeszcze użyć opcji `--symbolic-links` w wierszu poleceń lub umieścić poniższe wiersze w pliku opcji:

```
[mysqld]
symbolic-links
```

11.3.6. Przeniesienie poszczególnych tabel

Przeniesienie poszczególnych tabel jest możliwe jedynie w pewnych sytuacjach:

- Trzeba używać systemu UNIX, a przenoszona tabela musi być typu MyISAM.
- System operacyjny musi mieć prawidłowo działające wywołanie `realpath()`. W takim przypadku wynikiem wykonania poniższego zapytania będzie wartość YES:

```
mysql> SHOW VARIABLES LIKE 'have_symlink';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_symlink  | YES   |
+-----+-----+
```

Po spełnieniu obu wymienionych warunków można przenieść pliki *.myd* (dane) i *.myi* (indeksy) do nowego położenia, a następnie utworzyć do nich dowiązania symboliczne w katalogu danych MySQL, zachowując oryginalne nazwy plików danych i indeksów. (Plik *.frm* powinien pozostać w katalogu danych). Przed operacją należy zatrzymać serwer lub nałożyć blokadę na przenoszoną tabelę, aby uniemożliwić serwerowi jej używanie. Informacje na ten temat znajdziesz w podrozdziale 14.2, zatytułowanym „Obsługa bazy danych w działającym serwerze”.

11.3.7. Przeniesienie systemowej przestrzeni tabel InnoDB

Początkowa konfiguracja systemowej przestrzeni tabel InnoDB odbywa się przez umieszczenie w pliku opcji położenia plików komponentów przestrzeni tabel za pomocą zmiennych systemowych `innodb_data_home_dir` i `innodb_data_file_path`. (Więcej informacji na ten temat znajdziesz w podpunkcie 12.5.3.1, zatytułowanym „Konfiguracja przestrzeni tabel InnoDB”). Jeżeli utworzyłeś już przestrzeń tabel, to istnieje możliwość przeniesienia tworzących ją zwykłych plików, na przykład w celu rozmieszczenia plików w różnych systemach plików. Ponieważ położenie plików wskazują zmienne systemowe, przeniesienie kilku lub wszystkich z plików tworzących przestrzeń tabeli można przeprowadzić w następujący sposób:

1. Zatrzymanie serwera, jeśli działa.
2. Przeniesienie pliku lub plików przestrzeni tabel do lokalizacji docelowej.

3. Uaktualnienie pliku opcji zawierającego konfigurację InnoDB w celu odzwierciedlenia nowego położenia przeniesionych plików.
4. Ponowne uruchomienie serwera.

11.3.8. Przeniesienie plików dzienników zdarzeń i stanu

W celu przeniesienia pliku PID lub dziennika zdarzeń należy zatrzymać serwer, a następnie ponownie go uruchomić wraz z odpowiednią opcją wskazującą nowe położenie pliku. Na przykład, jeśli plik PID zostanie utworzony jako `/tmp/mysql.pid`, wtedy w wierszu poleceń użyj opcji `--pid_file=/tmp/mysql.pid` lub umieść poniższe wiersze w pliku opcji:

```
[mysql]
pid_file=/tmp/mysql.pid
```

Jeżeli nazwa pliku zostanie podana w postaci bezwzględnej ścieżki dostępu, serwer utworzy plik, używając podanej ścieżki dostępu. Jeśli użyto względnej ścieżki dostępu, serwer utworzy plik w katalogu danych MySQL. Na przykład, podanie opcji `--pid_file=mysql.pid` powoduje utworzenie pliku PID o nazwie `mysqld.pid` w katalogu danych MySQL.

Pewne systemy przechowują pliki PID w określonym katalogu, na przykład `/var/run`. W celu zachowania spójności w działającym serwerze plik PID MySQL możesz chcieć umieścić w tym samym katalogu. Podobnie, jeśli system używa katalogu `/var/log` do przechowywania plików dzienników zdarzeń, tam możesz umieścić też pliki dzienników zdarzeń MySQL. Jednak wiele systemów pozwala na uzyskanie dostępu do wymienionych katalogów tylko użytkownikowi root. Oznacza to konieczność uruchomienia serwera jako użytkownik root, co ze względów bezpieczeństwa nie jest dobrym pomysłem. Rozwiązaniem może być utworzenie podkatalogów `/var/run/mysql` i `/var/log/mysql` i ustawienie ich jako własności konta używanego do uruchamiania serwera. Na przykład, jeśli nazwa konta i użytkownika to `mysql`, wtedy jako użytkownik root należy wykonać poniższe polecenia:

```
# mkdir /var/run/mysql
# chown mysql /var/run/mysql
# chgrp mysql /var/run/mysql
# chmod u=rwx,go-rwx /var/run/mysql
# mkdir /var/log/mysql
# chown mysql /var/log/mysql
# chgrp mysql /var/log/mysql
# chmod u=rwx,go-rwx /var/log/mysql
```

W takim przypadku serwer nie będzie miał żadnych problemów z zapisywaniem plików w nowo utworzonych katalogach. Serwer możesz uruchomić wraz z opcjami wskazującymi pliki w tych nowych katalogach, na przykład:

```
[mysql]
pid_file = /var/run/mysql/mysql.pid
log_error = /var/log/mysql/log.err
```

```
general_log = 1  
general_log_file = /var/log/mysql/querylog  
log-bin = /var/log/mysql/binlog
```

Więcej informacji na temat opcji dotyczących plików dzienników zdarzeń oraz sposobu ich używania znajdziesz w podrozdziale 12.8, zatytułowanym „Dzienniki zdarzeń serwera”.

Skorowidz

A

- ACID, 189
- administracja bazą danych, 591, 621
- adres IP, 763, 917
- algorytm wyświetlania, 412
- aliasy powłoki, 115
- alokacja uchwytu zapytania, 432
- analiza skryptów DBI, 447
- API, 359
 - C, 359, 360
 - Perl DBI, 360
 - PHP, 359, 362
- aplikacje sieciowe, 510
- architektura klient-serwer, 45
- argument undef, 473
- atrybut
 - ASCII, 254
 - AUTO_INCREMENT, 72, 214, 238, 272–277
 - BINARY, 254
 - NULL, 60, 264
 - RaiseError, 488
 - SIGNED, 237
 - UNICODE, 254
 - UNSIGNED, 237
 - ZEROFILL, 237
- atrybuty
 - połączenia, 450, 451
 - typu ciągu tekstowego, 251
 - typu daty, 262
 - typu liczbowego, 237

- automatyczna
 - konwersja typu, 337
 - naprawa, 783, 784, 807
- automatyczne
 - właściwości typu daty, 263
 - zatwierdzenie, 576

B

- baza danych, DB, 42, 323
 - INFORMATION_SCHEMA, 164, 166
 - sampdb, 523
- bezpieczeństwo, 320, 386, 524, 715–774, 911
 - serwera, 594
 - systemu plików, 594
 - transakcji, 160
- bezpieczne
 - połączenia, 770
 - środowisko wykonywania, 190
- biblioteka
 - info_tables.so, 652
 - libmysqlclient, 538
 - mysqlnd, 538
- bieżące ustawienia, 133
- binarne kopie zapasowe, 786, 816
- blok try/catch, 544
- blokada globalna, 782
- blokowanie tabel, 353, 779–782
- błąd Access denies, 398
- błędy, 268, 377, 453, 478, 561
- błędy składni, 53, 401
- BMP, Basic Multilingual Plane, 130

bufor
kluczy, 677
zapytań, 680

C

cechy skryptu, 446
Certificate Authority, CA, 771
ciąg tekstowy, 217, 416, 878–891
binarny, 219, 221, 223
niebinarny, 219, 223
CRC, Cyclic Redundancy Check, 874
cyfry znaczące, 235
cytowanie, 730
ciągów tekstowych, 467
danych, 559

D

dane binarne, 405, 418
data i godzina, 92, 892–907
DBD, Database Driver, 361
DBI, Database Interface, 361
DDL, Data Definition Language, 192
definicja
FOREIGN KEY, 74, 198
kolumny, 230
kolumny member_id, 61
PRIMARY KEY, 74
definiowanie
haseł, 622
uprawnień, 734
widoków, 308
defragmentowanie tabel, 345
dezaktywacja indeksów, 351
deinstalacja wtyczki, 653
długość
identyfikatora, 127
prefiksu, 330
rekordów indeksu, 157
DNS, 730
dodawanie rekordów, 75
dokument HTML, 507
dokumentacja Perl DBI, 445

dołączanie
do zmiennych, 474
wyniku zapytania, 474
domyślne kodowanie znaków, 667
dopasowywanie wzorca, 95, 285, 532, 865
dostęp do
CGI.pm, 514
danych, 357, 358, 600
metadanych, 482
MySQL, 537, 538
odczytu i zapisu, 780
plików, 778
pliku nagłówkowego, 546
pliku opcji, 383
tabel, 353, 778
dostrajanie
serwera, 668, 670
silnika bazy danych, 673
dystrybucja sampdb, 548, 819
działanie optymalizatora zapytań, 333, 334
dziennik
błędów, 687, 688
przekazywania, 692
zapytań, 688, 689
zdarzeń, 612, 686, 693, 701
zdarzeń binarny, 690, 815
zdarzeń serwera, 684

E

edycja
rekordu, 500
wiersza danych, 116
efektywne
wczytywanie danych, 349
wykonywanie zapytań, 344
efektywność indeksu, 324
elementy struktury my_option, 390

F

format
rejestracji danych, 815
RTF, 39, 489, 492

- formaty DSN, 450
 - formularze, 574, 588
 - fsp, 831
 - funkcja
 - AGAINST(), 207
 - ASCII(), 296
 - AVG(), 107
 - CAST(), 251
 - CHARSET(), 130
 - check_response(), 581
 - COLLATION(), 130
 - CONVERT(), 222, 297
 - COUNT(), 98
 - COUNT(*), 207
 - CURRENT_USER(), 321
 - DATE_ADD(), 94, 296
 - DATE_SUB(), 94
 - DAYOFYEAR(), 109
 - display_events(), 529
 - display_scores(), 530
 - display_table_contents(), 526
 - display_table_names(), 525
 - escape(), 517
 - get_one_option(), 424
 - htmlspecialchars(), 552
 - insert_rows(), 432, 435
 - LAST_INSERT_ID(), 279
 - LENGTH(), 222
 - li(), 526
 - load_defaults(), 383, 386–389
 - load_image(), 419
 - main(), 375
 - MIN(), 102
 - MONTH(), 109
 - my_print_help(), 394
 - mysql_close(), 376, 379
 - mysql_error(), 379
 - mysql_fetch_row(), 402
 - mysql_field_count(), 406
 - mysql_field_seek(), 413
 - mysql_free_result(), 402
 - mysql_init(), 375
 - mysql_library_end(), 376
 - mysql_library_init(), 376
 - mysql_more_results(), 427
 - mysql_real_connect(), 375
 - mysql_real_escape_string(), 417
 - mysql_sqlstate(), 379
 - mysql_ssl_set(), 425
 - mysql_stmt_init(), 432
 - mysql_store_result(), 402, 406, 409–411
 - mysql_use_result(), 409–411
 - NOW(), 151
 - password_field(), 585
 - print_dashed(), 414
 - print_error(), 380, 456
 - process_call_result(), 443
 - process_result_set(), 404, 412, 420, 444
 - RAND(), 151
 - referential integrity, 72
 - sampdb_connect(), 545, 547, 551
 - search_members(), 532, 533
 - select_rows(), 437, 438
 - TIMESTAMPDIFF(), 93
 - TO_DAYS(), 94
 - update_entry(), 589
 - funkcje, 869
 - agregujące, 177, 907
 - ciągu tekstowego, 878–91
 - daty i godziny, 892–907
 - liczbowe, 873
 - nakładania blokad, 915
 - podsumowań, 907
 - porównań, 870
 - przestrzenne, 920
 - różne, 920
 - rzutowania, 872
 - składowane, 311, 314
 - XML, 919
 - zapewnienia bezpieczeństwa, 911
 - związane z adresem IP, 917
- ## G
- GA, General Availability, 27
 - generowanie
 - danych wyjściowych, 514
 - dokumentów HTML, 518
 - catalogu, 488, 492
 - listy, 491

generowanie
 łączy, 530
 podsumowania, 97
 sekwencji, 278
 strony HTML, 515
 globalizacja, 664
 globalne ustawienia, 125
 graficzny interfejs użytkownika, GUI, 45
 grupowanie, 221, 853

H

harmonogram, 352
 harmonogram zdarzeń, 318
 hasła, 622, 627, 745, 747
 hermetyzacja kodu, 545
 historia poleceń, 114
 HTML, 515

I

identyfikatory, 126, 127
 implementacje SQL, 361
 indeks
 typu BTREE, 157, 332
 typu FULLTEXT, 155, 205, 211, 534
 typu HASH, 155, 331
 typu PRIMARY KEY, 155
 typu SPATIAL, 155
 typu UNIQUE, 155
 indeksowanie, 324, 327
 krótkich wartości, 329
 prefiksów, 330
 tabel, 153
 indeksy
 klastrowanie, 330
 syntetyczne, 346
 unikalne, 154
 w silnikach, 154
 zwykłe, 154
 informacje
 o bazach danych, 164, 166
 o błędzie, 563
 o kolumnie, 414
 o strefie czasowej, 665
 z wielu tabel, 104, 183

inicjalizacja
 katalogu danych, 825
 tabel systemowych, 827
 tabel uprawnień, 825
 InnoDB, 27
 instalacja
 MySQL, 822, 823
 PDO, 829
 Perl DBI, 828
 PHP, 829
 integralność odwołań, 197, 198
 interaktywne wykonywanie zapytań, 420
 interfejs
 dostępu, 537
 sieciowy, 703
 wtyczek, 651
 izolacja transakcji, 194

J

jednoczesne
 operacje zapisu, 353
 wykonywanie zapytań, 426
 język
 C, 369–444
 Perl, 39
 Perl DBI, 445–536
 PHP, 537–89
 SQL, *Patrz* SQL
 języki interpretowane, 366

K

katalog
 danych, 593, 597, 614, 825
 Ligi Historycznej, 488
 phpapi, 537
 ssl, 772
 ushl, 553
 klauzula
 CHANGE, 159
 CHARSET, 131
 CONSTRAINT, 198
 DEFAULT, 254, 262, 832
 DEFINER, 321

- DO, 320
- ENGINE, 604
- FOREIGN KEY, 74, 198
- FROM, 81, 104, 168
- GROUP BY, 99
- HAVING, 101
- IDENTIFIED BY, 746, 747
- IDENTIFIED WITH, 751
- LEFT JOIN, 108
- LIKE, 132, 163, 285, 532
- LIMIT, 89, 101, 186, 527
- MODIFY, 159
- ON, 104, 106, 170
- ON DELETE, 199
- ON UPDATE, 199
- ON UPDATE CASCADE, 201
- ORDER BY, 87, 100, 174, 185
- PRIMARY KEY, 61
- REFERENCES, 199
- REQUIRE, 741, 742
- RETURNS, 312
- SET, 77
- USING, 187
- USING BTREE, 157
- USING(), 170
- WHERE, 82, 308
- WHERE FALSE, 485
- WITH ROLLUP, 102
- klient
 - MySQL, 52, 382, 389, 394, 592
 - mysqladmin, 1154
 - z obsługą SSL, 421
- klucz zewnętrzny, 75, 197, 200
- kod
 - HTML, 515
 - XHTML, 516
- kodowanie
 - Unicode, 130, 132, 134
 - znaków, 130–132, 219, 222, 517
 - znaków specjalnych, 416, 417
- kolejność
 - dołączania plików, 374
 - operatorów, 287, 852
 - sortowania, 131, 667
- kolumna
 - AUTO_INCREMENT, 275–278
 - GROUP BY, 106
- kolumny
 - uprawnień, 758
 - uwierzytelnienia, 759
 - zarządzania, 760
 - zasięgu, 762, 764, 766
- komentarze, 1112
- kompilacja, 370
 - pliku, 421, 426
 - serwerów, 705
- kompilator gcc, 372
- komponenty MySQL, 592
- kompresja, 911
- komunikacja z serwerem, 46
- komunikat
 - błędu, 456, 667
 - pomocy, 1116
 - Using where, 341
- konfiguracja
 - Apache, 512
 - bufora kluczy, 677
 - haseł, 627
 - mysqld, 630
 - obsługi stref czasowych, 664
 - połączeń, 770
 - puli bufora InnoDB, 675
 - serwerów, 705
 - serwerów replikacji, 809
 - silnika, 654, 656
 - silnika wyszukiwania, 211
 - struktury MYSQL_BIND, 435
 - systemowej przestrzeni tabel, 657–661
 - tablicy MYSQL_BIND, 436, 439
 - zapytań preinterpretowanych, 431
 - zmiennej PATH, 824
- koniec zapytania, 52
- konstrukcja
 - AS, 91
 - new PDO(), 543
 - qq{}, 469
 - switch, 585
 - try/catch, 544

konto, 623
 anonimowe, 625
 logowania, 823
 MySQL, 727
 początkowe, 624
 użytkownika, 725, 747
 w serwerze, 48

kontrola
 dostępu, 594, 715, 746, 754, 761
 nad serwerem, 641
 współbieżności, 45

konwersja
 podzapytań, 182
 tabeli, 160
 typu, 289, 294
 zapytania, 181

kopia
 zapasowa, 785, 790, 794, 808, 815
 zapasowa binarna, 793

kopiowanie baz danych, 796, 798

koszt indeksowania, 327

kradzież danych, 717

kwalifikator, 171

kwalifikatory identyfikatora, 127

kwalifikowane odwołania, 170

L

liczba
 sekwencji, 276
 zwracanych rekordów, 462

liczbowe typy danych, 228, 232, 234, 236

liczby, 215, 873

liczby całkowite, 234

liczebność kolumny, 329

linkowanie programów klienckich, 370

lista
 argumentów, 385
 przypisań, 77
 zdarzeń, 530

Ł

łącze do skryptu, 530

M

makra przenośności, 374

mapowanie zapytań, 604

mechanizm cron, 784

metadane
 bazy danych, 161
 zbioru wynikowego, 411, 482

metoda
 col_prompt(), 503
 connect(), 450
 do(), 456, 479
 errorCode(), 562
 exec(), 554
 execute(), 471
 fetch(), 552, 555
 fetchall_arrayref(), 466
 fetchrow_array(), 452, 458, 463–465
 fetchrow_arrayref(), 460, 465
 fetchrow_hashref(), 461
 finish(), 452
 interpret_argument(), 497
 prepare(), 451, 471
 query(), 544, 555
 quote(), 471, 560
 rollback(), 487
 rtf_format_entry(), 508
 search_members(), 535

metody pobierania rekordu, 461

miejsca zarezerwowane, 470

moduł CGI.pm, 445, 513, 519

moduły MySQL, 361

modyfikator IF EXISTS, 152

MySQL, 20–22

MySQL jako usługa, 48

MySQL Workbench, 45

N

nadawanie uprawnień, 732, 743

nakładanie blokad, 354, 779, 780, 915

naprawianie tabel, 798, 801

narzędzia
 bezpośredniego dostępu, 601
 MySQL, 592

narzędzie, *Patrz* program
 nasłuchiwanie połączeń, 639
 nawiasy
 klamrowe, 469
 kwadratowe, 831, 1003
 nawiązywanie połączenia, 113, 373, 521, 545, 641, 741
 nazwa komputera, 730
 nazwy, 128
 baz danych, 605
 kont użytkowników, 728
 kolumn danych wyjściowych, 90
 niepowtarzalne odczyty, 195
 niezatwierdzone odczyty, 195

O

obiekt PDO, 554
 obliczanie
 szerokości kolumny, 413
 wartości, 90
 wyrażeń, 280
 obsługa
 bazy danych, 595, 775, 777
 błędów, 377, 453, 561
 dopełnień, 256
 kodowania znaków, 130, 219
 opcji hasła, 394
 preinterpretowanego zapytania, 441
 przyrostków, 491
 skryptów CGI, 512
 SSL, 421
 stref czasowych, 664
 Unicode, 133
 wartości nieprawidłowych, 267
 wyjątków, 548, 563
 odbieranie uprawnień, 744
 odczyty widma, 195
 odsłona reklamy, hit, 42
 odwołania
 do kolumn, 170
 do tabel, 127
 ograniczenia
 danych wejściowych, 269
 relacji, 72
 systemu operacyjnego, 605
 wyników zapytania, 89
 zasobów, 743
 ograniczniki zapytań, 310
 określanie opcji, 1117
 opcja, 1117
 CLIENT_MULTI_STATEMENTS, 426
 host, 386
 register_globals, 567
 opcje
 analizy tabel, 1166
 charakterystyczne dla serwera, 1124
 charakterystyczne dla użytkownika, 1124
 dzienników zdarzeń, 704
 formatu danych, 1193
 globalne, 1124
 mysql, 49
 mysql.server, 1150
 mysql_config, 1151
 naprawy tabel, 801, 1166
 narzędzia mysqlcheck, 802
 optymalizacji tabel, 1167
 replikacji, 1177
 sprawdzania tabel, 1166
 standardowe mysamchk, 1132
 standardowe mysql, 1139
 standardowe MySQL, 1119
 standardowe mysql_install_db, 1152
 standardowe mysql_upgrade, 1153
 standardowe mysqladmin, 1154
 standardowe mysqlbinlog, 1159
 standardowe mysqlcheck, 1163
 standardowe mysqld, 1168
 standardowe mysqld_multi, 1180
 standardowe mysqld_safe, 1182
 standardowe mysqldump, 791, 1184
 standardowe mysqlimport, 1194
 standardowe mysqlshow, 1197
 standardowe perror, 1198
 standardowe SSL, 1122
 startowe, 707
 startowe serwera, 637
 tabeli, 144
 zapytania CHECK TABLE, 802

- operacje na danych, 92
- operandy, 289
- operator, 852
 - ALL, 179
 - ANY, 179
 - COLLATE, 130, 132
 - EXISTS, 179
 - IN, 178
 - LIKE, 95
 - NOT EXISTS, 179
 - NOT IN, 177
 - NOT LIKE, 95
 - REGEXP, 337
 - SOME, 179
- operatory
 - arytmetyczne, 84, 282, 854
 - bitowe, 284, 292, 861
 - dopasowania wzorca, 865
 - grupowania, 853
 - logiczne, 84, 283, 862
 - porównania, 84, 176, 284, 856
 - rzutowania, 864
- opis
 - modułu CGI.pm, 513
 - Perl DBI, 447
 - PHP, 539
 - typów danych, 228, 831
- opóźniony zapis kluczy, 351
- oprogramowanie, 819
- oprogramowanie MySQL, 47
- optymalizacja zapytań, 333, 334
 - indeksowanie, 324
 - wczytywanie danych, 349
 - wybór formatu tabeli, 347
 - wybór typu danych, 344
 - zapytanie EXPLAIN, 333, 338
- optymalizacje sprzętowe, 682
- organizacja repozytorium, 42

P

- pakiety, 821
- pakowanie danych, 346

- parametry
 - danych wejściowych, 567
 - połączenia, 382, 393, 475, 478, 545
 - procedury składowanej, 315
- partycjonowanie tabeli, 150
- PDO, PHP Data Objects, 538, 829
- Perl DBI, 39, 445–536
- pętla pobierania
 - DBI, 464
 - rekordu, 458
- PHP, 40
- plik, *Patrz także* skrypt
 - .my.cnf, 1128
 - .mysql_history, 118
 - connect2.c, 395
 - crontab, 630, 784
 - dump_members.pl, 464
 - exec_stmt_ssl.c, 422, 426
 - fill_help_tables.sql, 828
 - index.php, 539
 - kopii zapasowej, 797
 - Makefile, 372
 - my_global.h, 374
 - my_sys.h, 374
 - mysql.h, 374
 - opcji, 796
 - php.ini, 546
 - PID, 612
 - prepared_call.c, 444
 - sampdb.cnf, 522
 - sampdb_pdo.php, 548
 - show_opt.c, 388
 - sslopt-case.h, 424
 - sslopt-vars.h, 424
- pliki
 - .csm, 139
 - .csv, 139
 - .frm, 139, 603, 789, 796
 - .ibd, 139, 796
 - .idb, 139
 - .my.cnf, 772
 - .myd, 139
 - .myi, 139
 - .php, 538

- .pl, 446
- .sql, 54
- certyfikatu, 771, 773
- dziennika zdarzeń, 611, 695, 698, 796
- kopii zapasowej, 788
- mysql, 537
- mysqli, 538
- nagłówkowe, 371, 546
- opcji, 113, 383, 477, 1124–1128
- PDO, 538, 554
- startowe powłoki, 481
- systemowej przestrzeni tabel, 796
- wsadowe, 115
- pobieranie
 - danych, 167, 566
 - informacji, 80, 82, 183
 - metadanych, 161–166
 - parametrów połączenia, 382
 - pojedynczego rekordu, 458, 462, 473
 - rekordów, 556
 - serwera MySQL, 820
- podsumowanie, 97, 907
- podzapytania, 175, 182
 - kolumny, 175
 - rekordu, 175
 - skalarne, 175, 176
 - skorelowane, 180
 - tabeli, 175
 - w klauzuli FROM, 181
- podzapytanie
 - ALL, 178
 - ANY, 178
 - EXISTS, 179
 - IN, 177
 - NOT EXISTS, 179
 - NOT IN, 177
 - SOME, 178
- pojedynczy rekord, 462
- pola formularza, 588
- polecenia
 - edycji wiersza danych, 116, 117
 - programu mysql, 1145
- polecenie
 - chgrp, 826
 - chown, 826
 - include, 548
 - mysql, 48
 - print, 478
 - quit, 51
 - require, 548
 - use strict, 449
 - use warnings, 449
- połączenie
 - localhost, 729
 - z serwerem, 49, 55, 373, 395, 425, 521, 543, 561, 770
- położenie katalogu danych, 598
- ponowne wykonywanie zapytań, 805
- porównywanie, 221, 856, 870
 - kolumn, 335
 - wartości, 336
- poziom
 - DBD, 361
 - DBI, 361
- poziomy izolacji transakcji, 195, 196
- prefiks, 157, 330
- preinterpretowane zapytania, 440
- procedura składowana, 313
- procedury, 311, 314
- program
 - connect1, 373, 376
 - connect2, 399
 - make, 372
 - metadata, 416
 - my_print_defaults, 1128
 - myisamchk, 211, 592, 601, 1130, 1136
 - myisamchkmysql, 1137
 - opcje, 1139–1144
 - polecenia, 1145
 - zmienne, 1145
 - znaki zachęty, 1138, 1148
 - mysql_config, 372, 1150
 - mysql_upgrade, 1153
 - mysqladmin, 592
 - opcje, 1154, 1155
 - polecenia, 1156
 - zmienne, 1155
 - mysqlbinlog, 803, 805, 807
 - opcje, 1159–1162
 - zmienne, 1162

program

- mysqlcheck, 592, 801
- opcje, 1163–1165
- mysqldump, 167, 592, 786–790, 1184
 - opcje, 1185
 - opcje formatu danych, 1193
 - zmienne, 1193
- mysqlimport, 1194
 - opcje, 1194, 1195
 - opcje formatu danych, 1196
- mysqlshow, 166, 1196
- perror, 1198
- show_argv, 393
- show_opt, 387

programowanie MySQL, 355

programy składowane, 305, 309

projekt

- „Liga Historyczna USA”, 37
- „Oceny uczniów”, 40, 564

protokół

- SSL, 370, 421, 741, 759, 770
- TCP/IP, 762

prywatność, 1128

przeglądarka

- bazy danych, 523
- tabel, 528

przenoszenie

- katalogu danych, 614, 615, 617
- plików dzienników, 619
- stanu, 619
- systemowej przestrzeni tabel, 618
- tabel, 618

przenośność, 367

- skryptu DBI, 361
- tabel, 788

przepisywanie podzapytań, 181, 182

przeprowadzanie transakcji, 188, 486

przeźródzeń tabel InnoDB, 657–662

przetwarzanie

- argumentów, 387
- opcji, 395
- opcji polecenia, 377
- transakcyjne, 189, 486
- wyników zapytania, 554
- zapytań, 399, 407

przyrostki, 491

przywracanie

- bazy danych, 79, 803
- tabel, 804

punkty pośrednie transakcji, 194

puzzle uprawnień, 767

Q

quiz, 577

R

RDBMS, 19, 33, 42

reguły nadawania nazw, 128

rejestracja zdarzeń, 692, 815

rekord, 75

relacja

- klucza zewnętrznego, 199
- typu główny – podległy, 811

relacyjna baza danych, 42

replikacja, 595, 809–815

reprezentacja tabeli, 139

rodzaje

- indeksów, 154
- kont, 623

root, 624

rotacja

- plików dzienników zdarzeń, 695
- tabel dzienników zdarzeń, 701

RTF, Rich Text Format, 39, 489

rzutowanie, 864, 872

S

sekwencje, 270, 276

serwer

- dodatkowy, 627
- główny, 810
- MySQL, 123, 592
- mysqld, 1167
 - opcje, 1168–1176
 - opcje replikacji, 1177
 - zmienne, 1180

- podległy, 812, 815
- replikacji, 809
- WWW, 357
- silnik
 - InnoDB, 72, 140, 274, 348, 656, 796
 - MEMORY, 141, 275, 348
 - MyISAM, 140, 273
 - NDB, 142
- silniki
 - baz danych, 27, 137, 603, 655, 673
 - domyślne, 656
 - wyszukiwania, 211
- skanowanie
 - binarne, 325
 - liniowe, 325
- składnia
 - komentarzy, 1112
 - SQL, 1003–1113
- skrypt
 - count_members.pl, 462
 - db_browse.pl, 523–525
 - dump_members.php, 552
 - dump_members.pl, 451, 453, 550
 - edit_member.php, 582, 583, 589
 - edit_member.pl, 500, 501, 503, 504
 - flip_flop.pl, 520
 - gen_dir.pl, 490, 494, 510
 - index.php, 565
 - interests.pl, 506
 - mysql.server, 632, 633, 1150
 - mysql_install_db, 825, 1152
 - mysqld_multi, 632, 708
 - mysqld_safe, 632, 1181
 - need_renewal.pl, 495
 - pres_quiz.php, 577, 582
 - renewal_notify.pl, 497, 500
 - score_browse.pl, 528, 530
 - score_entry.php, 564, 568, 574
 - show_member.pl, 484, 499, 501
 - tabular.pl, 484
 - ushl_browse.pl, 532, 533
 - ushl_ft_browse.pl, 535
- skrypty
 - CGI, 512
 - DBI, 448
 - Perl, 446
 - PHP, 538, 541
 - powłoki, 115
 - sieciowe, 445
 - wiersza poleceń, 521
- słowo kluczowe
 - AS, 91
 - BEGIN, 311
 - DISTINCT, 97, 98
 - END, 311
 - GLOBAL, 125
 - LOCAL, 79
 - prompt, 1149
 - SCHEMA, 134
 - STRAIGHT_JOIN, 335
 - TEMPORARY, 153
 - UNION, 183
 - UNSIGNED, 235
- sortowanie, 131, 220, 250, 667
- sortowanie wyników zapytania, 87
- spacje, 53
- specyfikatory poziomu uprawnień, 738
- sprawdzanie, 572
 - konwersji typu, 294
 - opcji, 1128
 - operacji optymalizatora, 338
 - optymalizatora, 340
 - parametrów, 514
 - tabel, 798, 800, 801
 - wartości zwrotnej, 378
 - wyników podzapytania, 175
 - sprawdzenie wartości NULL, 558
- SQL, Structured Query Language, 21, 33, 44, 592, 1003–1113
- SSL, Secure Sockets Layer, 370, 741
- stan MySQL, 611
- status GA, 27
- strefy czasowe, 664
- strona pobierająca dane, 550
- struktura
 - katalogu, 599, 609
 - my_option, 390
 - MYSQL_BIND, 439
- superużytkownik, 748

system plików, 603, 614, 716
 szerokość kolumny, 413
 szyfrowanie, 774

Ś

śledzenie, 480, 481

T

tabela, 603

- absence, 75
- child, 203
- ENGINES, 138
- grade_event, 73
- member, 57
- parent, 203
- score, 73
- student, 71
- uprawnień, 825
- user, 761

tabele

- CSV, 789
- INFORMATION_SCHEMA, 165
- InnoDB, 796
- MEMORY, 789
- nadrzędne, 198, 202
- nietransakcyjne, 268
- partycjonowane, 150
- potomne, 198, 200
- transakcyjne, 197, 268
- tymczasowe, 145
- uprawnień, 624, 754–765

tablica

- \$score, 576
- atrybutów połączenia, 450
- hash, 450, 462
- MYSQL_BIND, 435, 439

tekst, 878–891

transakcje, 188, 190, 192, 486

tryb

- boolowski, 208
- obsługi błędów, 562
- PIPES_AS_CONCAT, 283
- SQL serwera

ANSI, 125

ANSI_QUOTES, 125, 126

PIPES_AS_CONCAT, 125

STRICT_ALL_TABLES, 124, 269

STRICT_TRANS_TABLES, 124, 269

TRADITIONAL, 124, 269

tryby pobierania rekordów, 556

tworzenie

- aplikacji, 364

- bazy danych, 54, 135

- binarnej kopii zapasowej, 793

- indeksów, 154

- klienta, 421

- konta logowania, 823

- kopii, 595

- kopii binarnej, 793, 794, 795

- kopii zapisowej, 785, 790, 795, 815

- programów, 372

- programów klienckich, 370

- programów MySQL

- przy użyciu C, 369–444

- przy użyciu Perl DBI, 445–536

- przy użyciu PHP, 537–589

- sekwencji, 277

- skryptu, 455

- stron, 518

- tabel, 55, 58, 64, 143

- warunkowe tabeli, 145

- wyrażeń, 281

- wyzwalaczy, 318

- zdarzenia, 320

- zmiennej, 97

typ danych, 213, 227, 831

- BINARY, 244

- BIT, 236, 837

- BLOB, 244

- CHAR, 242

- DATE, 59, 92, 259

- DATETIME, 259, 263

- ENUM, 71, 245, 845

- MYSQL_ROW, 403

- SET, 245, 247, 845

- TEXT, 244

- TIME, 259

- TIMESTAMP, 260, 263, 264

- VARBINARY, 244
 - VARCHAR, 59, 242, 347
 - YEAR, 261
 - typy
 - indeksów, 155
 - operatorów, 282
 - rzutowania, 150
 - typy danych
 - ciągi tekstowe, 219, 229, 303, 241, 838
 - ciągi tekstowe binarne, 840
 - ciągi tekstowe niebinarne, 842
 - DBI, 447
 - daty i godziny, 845
 - liczbowe, 228, 232, 236, 833, 836
 - wyliczeniowe, 71
- ## U
- uaktualnianie
 - MySQL, 712
 - rekordów, 110, 186, 202
 - uchwyt połączenia, 406
 - uchwyty, 447
 - udostępnienie katalogu, 507
 - układ tabeli, 65
 - ukryte pola formularza, 582
 - Unicode, 130, 132, 133
 - uprawnienia, 726, 732–744, 748–750, 764, 767, *Patrz także* tabele uprawnień, puzzle uprawnień
 - administracyjne, 743
 - funkcji składowanych, 314
 - procedur, 314
 - uprawnienie
 - EXECUTE, 320
 - FILE, 748, 749
 - TRIGGER, 318
 - USAGE, 741
 - uruchamianie
 - klientów, 712
 - plików skryptów, 118
 - serwera, 628, 707, 826
 - jako usługi Windows, 635
 - nasłuchiwanie połączeń, 639
 - opcje startowe, 637
 - w systemach UNIX, 631
 - w systemach Windows, 634
 - wielu serwerów, 701, 710
 - ustawianie zmiennych, 1123
 - ustawianie zmiennych systemowych, 647, 648
 - ustawienia
 - bieżące, 132
 - językowe, 668
 - usuwanie
 - bazy danych, 136
 - błędów, 478, 480
 - indeksu, 158
 - konta użytkownika, 746
 - rekordów, 110, 186
 - tabel, 75, 152
 - UTC, Universal Coordinated Time, 260
 - uwierzytelnianie użytkownika, 731, 751, 759
 - użytkownik
 - proxy, 751
 - root, 625, 629
 - używanie
 - bufora zapytań, 680
 - indeksów, 327
 - metadanych, 411
 - programów składowanych, 309
 - widoków, 306
- ## W
- wartości
 - boolowskie, 227
 - ciągu tekstowego, 217
 - daty i godziny, 226, 230, 256, 265, 293
 - domyślnych kolumn, 231
 - dopasowane, 181
 - liczbowe, 215
 - niedopasowane, 182
 - pól bitowych, 216
 - przestrzenne, 226
 - zmiennych systemowych, 646
 - wartość
 - arg_type, 392
 - NULL, 86, 227, 466, 558
 - var_type, 391
 - zmiennej stanu, 649

- weryfikacja uprawnień, 764
 - widoki, 305, 306, 604
 - wielkość
 - liter, 128, 129
 - tabeli, 607
 - wiersz poleceń, 166, 373, 387, 391
 - własne aplikacje MySQL, 355
 - właściwości
 - ACID, 189
 - AUTO_INCREMENT, 270, 272
 - wprowadzanie wyników, 568
 - współbieżność, 352
 - wstawianie rekordu, 76
 - wstawienie rekordów z pliku, 78
 - wtyczka, 651
 - LOCKS, 652
 - unix_auth, 752
 - wtyczki uwierzytelnienia, 751
 - wybór
 - API, 363
 - bazy danych, 134
 - danych z widoku, 308
 - domyślnego silnika, 656
 - formatu tabeli, 347
 - indeksów, 328
 - silnika, 655
 - systemu operacyjnego, 40
 - typu ciągu tekstowego, 254
 - typu danych, 297, 299, 344
 - typu liczbowego, 239
 - ustawień językowych, 668
 - wydajność, 365
 - działania aplikacji, 365, 366
 - serwera, 429
 - systemu, 609
 - wyjątek PDOException, 562
 - wykonywanie
 - wielu zapytań, 426
 - zapytań, 51
 - wymuszanie konwersji typu, 294
 - wyniki zapytania, 554
 - wyrażenia, 280, 281
 - wyrażenie MATCH, 207
 - wystąpienia znaków cytowania, 467
 - wysyłanie przypomnień, 495
 - wyszukiwanie, 531, 532, 534
 - członków, 506
 - pełnego tekstu, 204–208
 - rozszerzone, 210
 - w języku naturalnym, 208
 - wyświetlanie
 - bieżących ustawień, 133
 - danych binarnych, 405
 - danych wejściowych, 584
 - informacji, 582
 - komunikatów błędów, 667
 - komunikatu pomocy, 1116
 - metadanych, 416
 - ocen, 528
 - struktury tabeli, 61
 - tablicy argumentów, 385
 - uprawnień, 744
 - wyników, 568
 - wywoływanie programów MySQL, 1115
 - wyzwalacze, 316, 604, 737
 - wzorce, 95, 162, 285, 532, 763, 806, 865
- X**
- XHTML, 516
 - XML, 919
- Z**
- zabezpieczanie
 - dostępu, 716
 - instalacji, 622, 718
 - plików opcji, 724
 - pliku gniazda, 723
 - zakres wartości, 303
 - zakresy typów liczbowych, 233
 - zalety
 - architektury klient-serwer, 45
 - indeksowania, 324
 - prefiksów, 330
 - zamykanie serwera, 628
 - zapytania, 51, 1003–1113
 - generujące zbiór wynikowy, 555
 - modyfikujące rekordy, 401, 456, 554
 - niezłożone, 1004

- obsługi deklaracji, 1105
- obsługi kursora, 1107
- obsługi warunków, 1108
- ogólnego przeznaczenia, 405
- preinterpretowane, 428–432, 440, 470, 559
- typu DDL, 192
- złożone, 310, 1103
- zwracające dane, 402, 458
- zapytanie
 - ALTER DATABASE, 130, 134, 136
 - ALTER TABLE, 158, 273, 608
 - ANALYZE TABLE, 343
 - CALL, 440, 441, 442
 - CHANGE MASTER, 812, 813
 - CHARACTER SET, 254
 - CHECK TABLE, 777, 800
 - COMMIT, 192
 - CREATE DATABASE, 54, 130, 602
 - CREATE INDEX, 156
 - CREATE TABLE, 58, 137
 - CREATE TABLE ... LIKE, 147
 - CREATE TABLE ... SELECT, 147
 - CREATE TRIGGER, 316
 - DELETE, 110, 112, 186
 - DELIMITER, 310
 - DESCRIBE, 62, 81, 162
 - DROP, 136
 - DROP DATABASE, 134, 602, 806
 - DROP INDEX, 158, 351
 - DROP TABLE, 152
 - EXPLAIN, 333, 338, 340–343
 - FLUSH, 782
 - FLUSH PRIVILEGES, 627
 - GRANT USAGE ON, 774
 - INSERT, 76, 213, 350
 - INSTALL PLUGIN, 653
 - LOAD DATA, 76, 147, 349
 - LOAD DATA LOCAL, 79
 - LOCK TABLES, 353
 - PROCEDURE ANALYSE(), 345
 - PURGE BINARY LOGS, 814
 - RENAME USER, 727
 - REPAIR TABLE, 777, 801
 - REPLACE, 576
 - ROLLBACK, 190
 - SAVEPOINT, 194
 - SELECT, 80, 82, 167
 - SET, 97
 - SET PASSWORD, 627
 - SHOW, 131, 161
 - SHOW COLLATION, 220
 - SHOW COLUMNS, 253
 - SHOW DATABASES, 63, 602
 - SHOW FULL COLUMNS, 62, 64
 - SHOW SLAVE STATUS, 814
 - SHOW STATUS, 649
 - SHOW TABLES, 163, 164, 604
 - START SLAVE, 814
 - START TRANSACTION, 190
 - STOP SLAVE, 814
 - UNION, 168, 183, 184
 - UNION DISTINCT, 185
 - UNLOCK TABLES, 353
 - UPDATE, 110, 112, 186, 278, 402, 487
 - USE, 55
 - zarządzanie
 - dziennikami zdarzeń, 693
 - kontami, 725, 727
 - zasięg udzielanego dostępu, 757
 - zastosowania MySQL, 35
 - zatrzymanie serwera, 640
 - zdarzenia, 318
 - zdarzenie jednokrotne, 320
 - złączenie, 108
 - CROSS JOIN, 170
 - INNER JOIN, 70, 169, 172
 - JOIN, 170
 - LEFT JOIN, 171–173
 - NATURAL LEFT JOIN, 173
 - OUTER JOIN, 171
 - RIGHT JOIN, 172
 - zmiana
 - bazy danych, 136
 - hasła, 745
 - nazwy tabeli, 161
 - struktury tabeli, 158
 - typu danych, 159
 - właściciela, 826

- zmienna, 96, 1123
 - error_count, 645
 - event_scheduler, 319
 - ft_min_word_len, 211
 - innodb_buffer_pool_size, 645
 - log_output, 693
 - max_connections, 810
 - mysam_recover_options, 351
 - query_cache_type, 681
 - sql_mode, 124, 645
 - time_zone, 666
- zmienna środowiskowa
 - DBI_DSN, 475
 - DBI_TRACE, 481
 - PATH, 824
- zmienne
 - bufora kluczy, 678
 - narzędzia myisamchk, 1136
 - programu mysql, 1145
 - silnika, 663
 - stanu, 644, 650, 987–994
 - bufora zapytań, 998
 - InnoDB, 994
 - SSL, 999
 - systemowe, 223, 644–648, 927–972
 - InnoDB, 972–87
 - ogólnego przeznaczenia, 670
 - środowiskowe, 386, 1129
 - w PHP, 544
 - zdefiniowane przez użytkownika, 1000
- znak
 - !, 114
 - %, 95, 337, 767
 - _, 95
 - apostrofu, 468
 - cudzysłowu, 468
 - gwiazdki, 81
 - nowego wiersza, 78
 - odwrotnego apostrofu, 230
 - pionowej kreski, 1003
 - plus, 209
 - średnika, 52
 - wielokropka, 1003
 - zachęty, 53, 54, 1148
- znaki
 - BMP, 130
 - cytowania, 467
 - specjalne, 416, 467, 730
 - sterujące, 1127
 - w identyfikatorach, 126
 - wieloznaczne, 62, 209, 286, 336, 769
 - zachęty, 1138

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



- 1. ZAREJESTRUJ SIĘ**
- 2. PREZENTUJ KSIĄŻKI**
- 3. ZBIERAJ PROWIZJĘ**

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>



Wykorzystaj potencjał bazy danych MySQL!

MySQL to jedna z najpopularniejszych baz danych wykorzystywanych we współczesnych aplikacjach internetowych. Swoją sukces zawdzięcza prostej konfiguracji, niewielkim wymaganiom oraz wysokiej niezawodności. Jeżeli do tego dołożymy genialny współczynnik możliwości do ceny – baza jest darmowa – oraz doskonałą współpracę z językiem PHP, to otrzymamy rewelacyjny produkt. Warto pamiętać, że MySQL to prawdziwy system bazodanowy, wspierający zaawansowane techniki replikacji danych i mogący przechowywać bez problemu miliardy rekordów.

Ta książka to kompletny poradnik poświęcony wykorzystaniu bazy danych MySQL oraz administrowaniu tą bazą. W trakcie lektury dowiesz się, jak najlepiej składować dane w bazie, jak optymalizować zapytania oraz jak pobierać metadane. Ponadto nauczysz się uzyskiwać dostęp do bazy z użyciem języków C, Perl oraz PHP. Niezwykle istotna jest część trzecia niniejszej publikacji. W całości została poświęcona zagadnieniom związanym z administrowaniem bazą MySQL. Zrozumiesz dzięki temu, w jaki sposób baza przechowuje dane, jak zoptymalizować jej wydajność oraz co musisz zrobić, by zagwarantować najwyższy poziom bezpieczeństwa. Warto, żebyś zawniczasu zapoznał się z rozdziałem poświęconym wykonywaniu i odtwarzaniu kopii zapasowej. Książka jest doskonałym źródłem informacji zarówno dla programistów, jak i administratorów korzystających z tej bazy danych.

Dzięki tej książce:

- poznasz dogłębnie bazę danych MySQL
- zoptymalizujesz jej działanie
- podłączysz się do jej zasobów za pomocą języków C, Perl i PHP
- w pełni wykorzystasz potencjał bazy danych MySQL

helion.pl
księgarnia
internetowa

Nr katalogowy: 17975



Księgarnia internetowa
<http://helion.pl>



Zamówienia telefoniczne:
0 801 339900



0 601 339900

Informatyka w najlepszym wydaniu



Helion

Sprawdź najnowsze promocje:

- <http://helion.pl/promocje>
- Książki najchętniej czytane
- <http://helion.pl/bestsellery>
- Zamów informacje o nowościach:
- <http://helion.pl/novowosci>

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
<http://helion.pl>

sięgnij po **WIĘCEJ**



KOD KORZYŚCI

Addison
Wesley

cena: 149,00 zł

ISBN 978-83-246-8146-4



9 788324 681464