

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

MySQL. Podręcznik administratora

Autor: Paul Dubois

Tłumaczenie: Tomasz Pędziwiatr (rozdz. 5 – 10,

dod. A, B), Grzegorz Werner (rozdz. 1 – 4)

ISBN: 83-7361-687-X

Tytuł oryginału: [MySQL Administrators Guide](#)

Format: B5, stron: 712



Książka „MySQL. Podręcznik administratora” to zbiór informacji niezbędnych do pracy osobie odpowiadającej za poprawną pracę MySQL. Przedstawia wszystkie zagadnienia związane z instalacją, administrowaniem i optymalizowaniem bazy. Opisuje metody replikowania danych, zarządzania kontami użytkowników i zabezpieczania danych przed uszkodzeniem oraz zaawansowane mechanizmy składowania InnoDB i narzędzia wchodzące w skład pakietu MySQL.

- Instalowanie MySQL z plików binarnych i źródłowych
- Zarządzanie systemem przywilejów i kontami użytkowników
- Zapobieganie awariom i usuwanie ich skutków
- Replikacja danych
- Optymalizowanie wydajności bazy
- Klient MySQL i narzędzia użytkowe
- Mechanizmy składowania ISAM i InnoDB
- Usuwanie przyczyn błędów

Dzięki wiadomościom zawartym w tej książce optymalnie skonfigurujesz MySQL oraz zapewnisz jego maksymalną wydajność i najwyższy poziom bezpieczeństwa danych.



Spis treści

Rozdział 1. Informacje ogólne	11
1.1. Informacje o niniejszym przewodniku	11
1.1.1. Konwencje używane w podręczniku	12
1.2. Przegląd systemu zarządzania bazami danych MySQL	14
1.2.1. Historia MySQL	15
1.2.2. Najważniejsze cechy MySQL	16
1.2.3. Stabilność MySQL	19
1.2.4. Jak duże mogą być tabele MySQL?	20
1.2.5. Zgodność z rokiem 2000	21
1.3. Informacje o firmie MySQL AB	23
1.3.1. Model biznesowy i usługi MySQL AB	24
1.3.2. Informacje kontaktowe	26
1.4. Pomoc techniczna i licencjonowanie MySQL	28
1.4.1. Pomoc techniczna oferowana przez MySQL AB	28
1.4.2. Prawa autorskie i licencje MySQL	29
1.4.3. Licencje MySQL	29
1.4.4. Logo i znaki towarowe firmy MySQL AB	32
1.5. Plan rozwoju MySQL	34
1.5.1. MySQL 4.0 w pigułce	35
1.5.2. MySQL 4.1 w pigułce	37
1.5.3. MySQL 5.0: następna wersja rozwojowa	39
1.6. MySQL i przyszłość	39
1.6.1. Nowe cechy planowane w wersji 4.1	39
1.6.2. Nowe cechy planowane w wersji 5.0	39
1.6.3. Nowe cechy planowane w wersji 5.1	40
1.6.4. Funkcje planowane na najbliższą przyszłość	41
1.6.5. Funkcje planowane na dalszą przyszłość	44
1.6.6. Nowe cechy, których nie planujemy	46
1.7. Źródła informacji o MySQL	46
1.7.1. Listy dyskusyjne MySQL	46
1.7.2. Pomoc społeczności użytkowników MySQL na IRC (ang. Internet Relay Chat)	55
1.8. MySQL — zgodność ze standardami	55
1.8.1. Standardy obsługiwane przez MySQL	56
1.8.2. Wybór trybu SQL	56
1.8.3. Uruchamianie MySQL w trybie ANSI	57
1.8.4. Rozszerzenia standardu SQL w serwerze MySQL	57
1.8.5. Rozbieżności między MySQL a standardem SQL	60
1.8.6. Jak MySQL obsługuje ograniczenia?	67
1.8.7. Znane błędy i wady projektowe MySQL	70

Rozdział 2. Instalowanie MySQL	77
2.1. Ogólne kwestie instalacyjne.....	78
2.1.1. Systemy operacyjne obsługiwane przez MySQL.....	78
2.1.2. Wybór dystrybucji MySQL.....	81
2.1.3. Jak zdobyć MySQL?.....	92
2.1.4. Weryfikowanie integralności pakietu za pomocą sum kontrolnych MD5 albo narzędzia GnuPG.....	92
2.1.5. Układ katalogów instalacyjnych	96
2.2. Standardowa instalacja MySQL z dystrybucji binarnej	97
2.2.1. Instalowanie MySQL w Windows	98
2.2.2. Instalowanie MySQL w systemie Linux	110
2.2.3. Instalowanie MySQL w systemie Mac OS X.....	113
2.2.4. Instalowanie MySQL w systemie NetWare	116
2.2.5. Instalowanie MySQL w innych systemach typu Unix	118
2.3. Instalacja MySQL z dystrybucji źródłowej.....	121
2.3.1. Instalacja dystrybucji źródłowej.....	123
2.3.2. Typowe opcje skryptu configure.....	126
2.3.3. Instalowanie MySQL z drzewa kodu źródłowego.....	129
2.3.4. Rozwiązywanie problemów występujących podczas kompilowania MySQL.....	132
2.3.5. Uwagi dotyczące biblioteki MIT-pthreads.....	136
2.3.6. Instalowanie MySQL z kodu źródłowego w Windows	137
2.3.7. Kompilowanie klientów MySQL w Windows	141
2.4. Konfiguracja i testy poinstalacyjne.....	142
2.4.1. Procedury poinstalacyjne w Windows	142
2.4.2. Procedury poinstalacyjne w systemie Unix.....	143
2.4.3. Automatyczne uruchamianie i zatrzymywanie serwera MySQL	150
2.4.4. Uruchamianie serwera MySQL i rozwiązywanie problemów	152
2.4.5. Zabezpieczanie początkowych kont MySQL	156
2.5. Instalowanie nowszej lub starszej wersji MySQL.....	159
2.5.1. Aktualizowanie wersji 4.1 do 5.0.....	161
2.5.2. Aktualizowanie wersji 4.0 do 4.1	161
2.5.3. Aktualizowanie wersji 3.23 do 4.0.....	166
2.5.4. Aktualizowanie wersji 3.22 do 3.23	171
2.5.5. Aktualizowanie wersji 3.21 do 3.22	173
2.5.6. Aktualizowanie wersji 3.20 do 3.21	173
2.5.7. Aktualizowanie MySQL w Windows	174
2.5.8. Aktualizowanie tabel przywilejów	175
2.5.9. Kopiowanie baz danych MySQL do innego komputera.....	176
2.6. Uwagi dotyczące poszczególnych systemów operacyjnych.....	177
2.6.1. Uwagi dotyczące systemu Linux.....	177
2.6.2. Uwagi dotyczące systemu Mac OS X	186
2.6.3. Uwagi dotyczące systemu Solaris	187
2.6.4. Uwagi dotyczące systemu BSD	192
2.6.5. Uwagi dotyczące innych systemów typu Unix.....	196
2.6.6. Uwagi dotyczące systemu OS/2	206
2.6.7. Uwagi dotyczące systemu BeOS.....	207
2.7. Uwagi dotyczące instalacji języka Perl	207
2.7.1. Instalowanie języka Perl w systemie Unix.....	208
2.7.2. Instalowanie oprogramowania ActiveState Perl w Windows.....	209
2.7.3. Problemy z interfejsem DBI/DBD języka Perl.....	210

Rozdział 3. Korzystanie z programów MySQL	213
3.1. Przegląd programów MySQL	213
3.2. Uruchamianie programów MySQL.....	214
3.3. Określanie opcji programów	215
3.3.1. Używanie opcji w wierszu polecenia	216
3.3.2. Używanie plików opcji	217
3.3.3. Określanie opcji za pomocą zmiennych środowiskowych	221
3.3.4. Ustawianie zmiennych programu za pomocą opcji	222
Rozdział 4. Administrowanie bazą danych	225
4.1. Serwer MySQL i skrypty startowe.....	225
4.1.1. Skrypty i programy użytkowe działające po stronie serwera	225
4.1.2. Rozszerzony serwer MySQL mysqld-max.....	227
4.1.3. Skrypt startowy mysqld_safe	229
4.1.4. Skrypt startowy mysql.server.....	233
4.1.5. Program mysqld_multi do zarządzania wieloma serwerami MySQL	233
4.2. Konfigurowanie serwera MySQL	237
4.2.1. Opcje wiersza polecenia mysqld	238
4.2.2. Tryb SQL serwera	249
4.2.3. Zmienne systemowe serwera	251
4.2.4. Zmienne stanu serwera.....	281
4.3. Ogólne kwestie bezpieczeństwa.....	288
4.3.1. Ogólne wskazówki dotyczące bezpieczeństwa	288
4.3.2. Zabezpieczanie MySQL przed napastnikami	291
4.3.3. Opcje startowe mysqld związane z bezpieczeństwem.....	293
4.3.4. Kwestie bezpieczeństwa związane z instrukcją LOAD DATA LOCAL	295
4.4. System przywilejów dostępu MySQL.....	296
4.4.1. Funkcje systemu przywilejów	296
4.4.2. Jak działa system przywilejów?	296
4.4.3. Przywileje obsługiwane przez MySQL	300
4.4.4. Łączenie się z serwerem MySQL.....	304
4.4.5. Kontrola dostępu, etap 1.: weryfikacja połączenia	305
4.4.6. Kontrola dostępu, etap 2.: weryfikacja żądań.....	309
4.4.7. Kiedy uwzględniane są zmiany przywilejów?	312
4.4.8. Przyczyny błędów Access denied	313
4.4.9. Szyfrowanie haseł w MySQL 4.1.....	318
4.5. Zarządzanie kontami użytkowników MySQL.....	324
4.5.1. Nazwy użytkowników i hasła MySQL.....	324
4.5.2. Dodawanie nowych kont użytkownika MySQL.....	326
4.5.3. Usuwanie kont użytkownika MySQL	329
4.5.4. Ograniczanie zasobów używanych przez konto	330
4.5.5. Przypisywanie haseł kontom	331
4.5.6. Ochrona hasła.....	333
4.5.7. Używanie połączeń zabezpieczonych	334
4.6. Zapobieganie awariom i usuwanie ich skutków.....	341
4.6.1. Kopie zapasowe bazy danych	341
4.6.2. Utrzymanie tabel i usuwanie skutków awarii.....	343
4.6.3. Tworzenie harmonogramu utrzymania tabel.....	357
4.6.4. Uzyskiwanie informacji o tabeli.....	358
4.7. Lokalizacja i zastosowania międzynarodowe MySQL.....	365
4.7.1. Zestaw znaków używany do zapisu i sortowania danych.....	365
4.7.2. Ustawianie języka komunikatów o błędach	366
4.7.3. Dodawanie nowego zestawu znaków	367
4.7.4. Tablice definicji znaków	369

4.7.5. Obsługa sortowania łańcuchów.....	369
4.7.6. Obsługa znaków wielobajtowych.....	370
4.7.7. Problemy z zestawami znaków.....	370
4.8. Pliki dziennika MySQL.....	370
4.8.1. Dziennik błędów.....	371
4.8.2. Ogólny dziennik zapytań.....	372
4.8.3. Dziennik aktualizacji.....	372
4.8.4. Dziennik binarny.....	373
4.8.5. Dziennik powolnych zapytań.....	377
4.8.6. Utrzymanie plików dziennika.....	377
4.9. Uruchamianie wielu serwerów MySQL w jednym komputerze.....	378
4.9.1. Uruchamianie wielu serwerów w Windows.....	380
4.9.2. Uruchamianie wielu serwerów w systemie Unix.....	384
4.9.3. Używanie programów klienta w środowisku z wieloma serwerami.....	385
4.10. Pamięć podręczna zapytań MySQL.....	386
4.10.1. Sposób działania pamięci podręcznej zapytań.....	387
4.10.2. Opcje instrukcji SELECT związane z pamięcią podręczną zapytań.....	389
4.10.3. Konfiguracja pamięci podręcznej zapytań.....	389
4.10.4. Stan i utrzymanie pamięci podręcznej zapytań.....	390
Rozdział 5. Replikacja w MySQL.....	393
5.1. Wstęp do replikacji.....	393
5.2. Ogólny opis implementacji replikacji.....	394
5.3. Szczegóły implementacji replikacji.....	395
5.3.1. Stany wątku replikacji serwera nadrzędnego.....	397
5.3.2. Stany wątku wejścia-wyjścia serwera podrzędnego.....	398
5.3.3. Stany wątku SQL serwera podrzędnego.....	399
5.3.4. Pliki stanu i transmisyjne pliki replikacji.....	399
5.4. Sterowanie replikacją.....	402
5.5. Kompatybilność replikacji pomiędzy różnymi wersjami MySQL.....	407
5.6. Konfiguracja replikacji z aktualizacją wersji serwera.....	407
5.6.1. Replikacja z aktualizacją do wersji 4.0 lub 4.1.....	408
5.6.2. Replikacja z aktualizacją do wersji 5.0.....	408
5.7. Replikacja — możliwości i problemy.....	409
5.8. Opcje startowe replikacji.....	413
5.9. Replikacja — najczęściej zadawane pytania.....	424
5.10. Replikacja — rozwiązywanie problemów.....	431
5.11. Raportowanie błędów replikacji.....	432
Rozdział 6. Optymalizacja MySQL.....	435
6.1. Wstęp do optymalizacji.....	435
6.1.1. Ograniczenia projektowe MySQL i kompromisy.....	436
6.1.2. Projektowanie przenośnej aplikacji.....	436
6.1.3. Do czego wykorzystywaliśmy MySQL.....	438
6.1.4. Pakiet testowania wydajności MySQL.....	439
6.1.5. Wykorzystanie własnych testów wydajności.....	440
6.2. Optymalizowanie instrukcji SELECT oraz innych zapytań.....	441
6.2.1. Składnia polecenia EXPLAIN. Pobieranie informacji o instrukcji SELECT.....	441
6.2.2. Szacowanie wydajności zapytania.....	450
6.2.3. Szybkość wykonania zapytań SELECT.....	451
6.2.4. Jak MySQL optymalizuje klauzule WHERE.....	452
6.2.5. Jak MySQL optymalizuje klauzule OR.....	454
6.2.6. Jak MySQL optymalizuje IS NULL.....	454
6.2.7. Jak MySQL optymalizuje DISTINCT.....	455

6.2.8. Jak MySQL optymalizuje LEFT JOIN i RIGHT JOIN	456
6.2.9. Jak MySQL optymalizuje ORDER BY	457
6.2.10. Jak MySQL optymalizuje LIMIT	459
6.2.11. Jak uniknąć przeszukiwania tabeli	460
6.2.12. Szybkość wykonywania zapytań INSERT	461
6.2.13. Szybkość wykonywania zapytań UPDATE	463
6.2.14. Szybkość wykonywania zapytań DELETE	463
6.2.15. Inne wskazówki optymalizacyjne	463
6.3. Problemy związane z blokowaniem	467
6.3.1. Metody blokowania	467
6.3.2. Problemy związane z blokowaniem tabel	470
6.4. Optymalizowanie struktury bazy danych	472
6.4.1. Wybór projektu	472
6.4.2. Dane powinny być tak małe, jak to tylko możliwe	472
6.4.3. Indeksy na pojedynczej kolumnie	474
6.4.4. Indeksy na wielu kolumnach	474
6.4.5. Jak MySQL wykorzystuje indeksy	475
6.4.6. Pamięć podręczna klucza MyISAM	478
6.4.7. Jak MySQL zlicza otwarte tabele	484
6.4.8. Jak MySQL otwiera i zamyka tabele	485
6.4.9. Wady tworzenia wielu tabel w tej samej bazie danych	486
6.5. Optymalizowanie serwera MySQL	487
6.5.1. Czynniki związane z systemem serwera oraz optymalizacja jego parametrów startowych	487
6.5.2. Optymalizowanie parametrów serwera	488
6.5.3. Wpływ kompilacji i łączenia na wydajność MySQL	490
6.5.4. Jak MySQL wykorzystuje pamięć	492
6.5.5. Jak MySQL wykorzystuje DNS	494
6.6. Kwestie związane z dyskiem	495
6.6.1. Wykorzystanie dowiązań symbolicznych	496
Rozdział 7. Klient MySQL i programy użytkowe	501
7.1. Przegląd skryptów klienta oraz narzędzi	501
7.2. myisampack — generator skompresowanych tabel, przeznaczonych tylko do odczytu	503
7.3. mysql — narzędzie wiersza poleceń	510
7.3.1. Polecenia mysql	515
7.3.2. Wykonywanie instrukcji SQL z pliku tekstowego	519
7.3.3. mysql — wskazówki	519
7.4. mysqladmin — administrowanie serwerem MySQL	521
7.5. mysqlbinlog — narzędzie do rejestrowania binarnego	526
7.6. mysqlcc — centrum sterowania MySQL	530
7.7. mysqlcheck — program do utrzymania i naprawiania tabel	532
7.8. mysqldump — narzędzie do tworzenia kopii zapasowych bazy danych	535
7.9. mysqlhotcopy — narzędzie do tworzenia kopii zapasowych baz danych	542
7.10. mysqlimport — program do importowania danych	544
7.11. mysqlshow — pokazywanie baz danych, tabel i kolumn	547
7.12. perror — opisy kodów błędów	549
7.13. replace — narzędzie do zamiany łańcuchów znaków	549
Rozdział 8. Mechanizmy składowania i typy tabel MySQL	551
8.1. Mechanizm składowania MyISAM	553
8.1.1. Opcje startowe MyISAM	555
8.1.2. Obszar wymagany dla kluczy	557
8.1.3. Mechanizm składowania tabel MyISAM	557
8.1.4. Problemy związane z tabelami MyISAM	560

8.2. Mechanizm składowania MERGE	562
8.2.1. Problemy związane z tabelami MERGE	565
8.3. Mechanizm składowania MEMORY (HEAP)	566
8.4. Mechanizm składowania BDB (ang. BerkeleyDB).....	569
8.4.1. Systemy operacyjne obsługiwane przez BDB	569
8.4.2. Instalowanie BDB	570
8.4.3. Opcje startowe BDB	570
8.4.4. Cechy tabel BDB	572
8.4.5. Rzeczy wymagające naprawienia w BDB.....	574
8.4.6. Restrykcje do tabel BDB.....	574
8.4.7. Błędy, które mogą wystąpić, gdy używamy tabel BDB	574
8.5. Mechanizm składowania ISAM.....	575
Rozdział 9. Mechanizm składowania InnoDB	577
9.1. Ogólny przegląd InnoDB	577
9.2. Kontakt z twórcami InnoDB	578
9.3. InnoDB w MySQL 3.23.....	578
9.4. Konfiguracja InnoDB.....	579
9.5. Opcje startowe InnoDB.....	584
9.6. Tworzenie obszaru tabel InnoDB.....	588
9.6.1. Rozwiązywanie problemów inicjalizacji InnoDB	589
9.7. Tworzenie tabel InnoDB.....	590
9.7.1. Wykorzystanie transakcji InnoDB przy użyciu różnych API.....	591
9.7.2. Konwertowanie tabel MyISAM na InnoDB.....	591
9.7.3. Działanie kolumny AUTO_INCREMENT w InnoDB.....	592
9.7.4. Ograniczenia FOREIGN KEY	593
9.7.5. InnoDB i replikacja MySQL	597
9.7.6. Użycie dla każdej tabeli oddzielnego obszaru tabel	598
9.8. Dodawanie i usuwanie plików danych oraz dzienników InnoDB	600
9.9. Tworzenie kopii zapasowej i odzyskiwanie InnoDB	602
9.9.1. Wymuszanie odzyskania danych.....	603
9.9.2. Punkty kontrolne	605
9.10. Przenoszenie bazy danych InnoDB na inną maszynę.....	605
9.11. Model transakcyjny oraz blokowania InnoDB	606
9.11.1. InnoDB i AUTOCOMMIT	606
9.11.2. InnoDB i TRANSACTION ISOLATION LEVEL	607
9.11.3. Spójny odczyt bez zakładania blokady.....	609
9.11.4. Odczyty z blokadą SELECT ... FOR UPDATE i SELECT ... LOCK IN SHARE MODE	609
9.11.5. Blokowanie najbliższego klucza — zapobieganie problemom fantomów ..	610
9.11.6. Przykład pokazujący działanie spójnych odczytów w InnoDB.....	611
9.11.7. Blokady zakładane przez różne instrukcje SQL w InnoDB	612
9.11.8. Niejawne zatwierdzenie lub wycofanie transakcji przez MySQL	614
9.11.9. Detekcja zakleszczeń oraz wycofanie	614
9.11.10. Jak radzić sobie z zakleszczeniami	615
9.12. Optymalizacja wydajności InnoDB — wskazówki.....	616
9.12.1. SHOW INNODB STATUS i monitoruj InnoDB	618
9.13. Implementowanie wielokrotnej kontroli wersji.....	623
9.14. Struktury tabel i indeksów	624
9.14.1. Fizyczna struktura indeksu.....	625
9.14.2. Buforowanie wstawiania	625
9.14.3. Adaptujące się indeksy mieszające	625
9.14.4. Fizyczna struktura rekordu.....	626

9.15. Zarządzanie przestrzenią plików oraz dyskowymi operacjami wejścia-wyjścia	627
9.15.1. Dyskowe operacje wejścia-wyjścia	627
9.15.2. Bezpośrednie wykorzystanie urządzeń na obszar tablicowy	627
9.15.3. Zarządzanie przestrzenią pliku	628
9.15.4. Defragmentowanie tabeli	629
9.16. Obsługa błędów	630
9.16.1. Kody błędów InnoDB	630
9.16.2. Kody błędów systemu operacyjnego	631
9.17. Ograniczenia tabel InnoDB	636
9.18. Rozwiązywanie problemów związanych z InnoDB	638
9.18.1. Rozwiązywanie problemów związanych z operacjami na słownikach danych InnoDB	638
Rozdział 10. Wstęp do MaxDB	641
10.1. Historia MaxDB	641
10.2. Licencjonowanie i wsparcie techniczne	642
10.3. Linki związane z MaxDB	642
10.4. Podstawowe założenia MaxDB	642
10.5. Różnice pomiędzy MaxDB a MySQL	642
10.6. Współdziałanie pomiędzy MaxDB a MySQL	643
10.7. Zarezerwowane słowa MaxDB	644
Dodatek A Rozwiązywanie problemów związanych z programami MySQL	649
A.1. Jak ustalić przyczynę problemu	649
A.2. Popularne błędy powstające podczas używania programów MySQL	651
A.2.1. Access denied	651
A.2.2. Can't connect to [local] MySQL server	651
A.2.3. Client does not support authentication protocol	654
A.2.4. Password Fails When Entered Interactively	655
A.2.5. Host 'nazwa_komputera' is blocked	655
A.2.6. Too many connections	655
A.2.7. Out of memory	656
A.2.8. MySQL server has gone away	656
A.2.9. Packet too large	658
A.2.10. Błędy połączenia oraz połączenia przerwane	659
A.2.11. The table is full	660
A.2.12. Can't create/write to file	661
A.2.13. Commands out of sync	661
A.2.14. Ignoring user	661
A.2.15. Table 'nazwa_tabeli' doesn't exist	662
A.2.16. Can't initialize character set	662
A.2.17. File Not Found	663
A.3. Problemy związane z instalacją	664
A.3.1. Problemy przy konsolidacji z biblioteką klienta MySQL	664
A.3.2. Uruchamianie MySQL z konta zwykłego użytkownika	665
A.3.3. Problemy z uprawnieniami do plików	666
A.4. Problemy związane z administrowaniem	667
A.4.1. Jak ustawić hasło użytkownika root	667
A.4.2. Co zrobić w przypadku powtarzających się awarii MySQL	669
A.4.3. W jaki sposób MySQL obsługuje wypełniony dysk	672
A.4.4. Gdzie MySQL przechowuje pliki tymczasowe	673
A.4.5. Jak zabezpieczyć lub zmienić plik gniazda MySQL /tmp/mysql.sock	674
A.4.6. Problemy związane ze strefą czasową	675
Dodatek B Zmienne środowiska	677
Skorowidz	679

Rozdział 2.

Instalowanie MySQL

W tym rozdziale wyjaśniono, jak pobrać i zainstalować MySQL:

- 0. Ustalić, czy dana platforma jest obsługiwana.** Warto pamiętać, że nie we wszystkich obsługiwanych systemach oprogramowanie MySQL działa równie dobrze. W niektórych jest znacznie bardziej odporne i stabilne. Szczegółowe informacje na ten temat można znaleźć w punkcie 2.1.1. „Systemy operacyjne obsługiwane przez MySQL”.
- 0. Wybrać dystrybucję do zainstalowania.** Dostępnych jest kilka wersji MySQL, a większość z nich ma kilka formatów dystrybucyjnych. Można wybierać spośród spakowanych dystrybucji zawierających programy binarne (wstępnie skompilowane) albo kod źródłowy. Udostępniamy też bieżące drzewo kodu źródłowego dla tych, którzy chcą zapoznać się z najnowszymi postępami i pomóc nam w testowaniu kodu. Aby ustalić, która wersja i typ dystrybucji są najodpowiedniejsze w danej sytuacji, należy przeczytać punkt 2.1.2. „Wybór dystrybucji MySQL”.
- 0. Pobrać dystrybucję do zainstalowania.** Lista witryn, z których można pobrać MySQL znajduje się w punkcie 2.1.3. „Jak zdobyć MySQL?”. Integralność dystrybucji można sprawdzić według wskazówek podanych w punkcie 2.1.4. „Weryfikowanie integralności pakietu za pomocą sum kontrolnych MD5 albo narzędzia GnuPG”.
- 0. Zainstalować dystrybucję.** Aby zainstalować MySQL z dystrybucji binarnej, należy postępować według wskazówek podanych w podrozdziale 2.2. „Standardowa instalacja MySQL z dystrybucji binarnej”. Aby zainstalować MySQL z dystrybucji źródłowej albo z bieżącego drzewa kodu źródłowego, należy postępować według wskazówek podanych w podrozdziale 2.3. „Instalacja MySQL z dystrybucji źródłowej”.

Ci, którzy aktualizują serwer MySQL do nowszej wersji, a nie instalują go po raz pierwszy, powinni przeczytać podrozdział 2.5 „Instalowanie nowszej lub starszej wersji MySQL”. Znajdują się tam informacje o procedurze aktualizacji oraz o kwestiach, które należy rozważyć przed zainstalowaniem nowszej wersji.

Jeśli podczas instalacji wystąpią problemy, należy zajrzeć do podrozdziału 2.6. „Uwagi dotyczące poszczególnych systemów operacyjnych”, w którym omówiono rozwiązywanie problemów związanych z poszczególnymi platformami.

- 0. **Wykonać konfigurację poinstalacyjną.** Po zainstalowaniu MySQL należy przeczytać podrozdział 2.4. „Konfiguracja i testy poinstalacyjne”. Podrozdział ten zawiera ważne informacje o tym, jak upewnić się, że serwer MySQL działa prawidłowo. Opisuje też zabezpieczanie wstępnych kont MySQL, które *nie mają hasła*, dopóki użytkownik ich nie przypisze. Podrozdział ten dotyczy zarówno binarnej, jak i źródłowej dystrybucji MySQL.
- 0. Do uruchomienia skryptów testujących MySQL niezbędna jest obsługa języka Perl. Zobacz podrozdział 2.7. „Uwagi dotyczące instalacji języka Perl”.

2.1. Ogólne kwestie instalacyjne

Przed zainstalowaniem MySQL należy wykonać poniższe czynności:

- 0. Ustalić, czy MySQL działa na wybranej platformie.
- 0. Wybrać dystrybucję do zainstalowania.
- 0. Pobrać dystrybucję i sprawdzić jej integralność.

Niniejszy podrozdział zawiera informacje niezbędne do wykonania tych czynności. Po ich wykonaniu można skorzystać ze wskazówek podanych w następnych podrozdziałach, aby zainstalować wybraną dystrybucję.

2.1.1. Systemy operacyjne obsługiwane przez MySQL

W tym punkcie wymieniono systemy operacyjne, w których można uruchomić MySQL.

Używamy programu GNU Autoconf, aby MySQL można było przenieść do wszystkich nowoczesnych systemów operacyjnych, które mają kompilator C++ i działającą implementację wątków POSIX. (Obsługa wątków jest potrzebna do działania serwera. Do skompilowania kodu klienta wystarczy kompilator C++.) My używamy oprogramowania MySQL i rozwijamy je głównie w systemach Linux (SuSE i Red Hat), FreeBSD oraz Sun Solaris (wersje 8 i 9).

MySQL można skompilować w poniższych kombinacjach systemu operacyjnego i pakietu wątków. W wielu systemach operacyjnych natywna obsługa wątków działa tylko w najnowszych wersjach.

- ♦ AIX 4.x, 5.x z natywną obsługą wątków (zobacz podpunkt 2.6.5.3. „Uwagi dotyczące systemu IBM-AIX”),
- ♦ Amiga,
- ♦ BSDI 2.x z pakietem MIT-pthreads (zobacz podpunkt 2.6.4.5. „Uwagi dotyczące systemu BSD/OS 2.x”),

- ◆ BSDI 3.0, 3.1 i 4.x z natywną obsługą wątków (zobacz podpunkt 2.6.4.5. „Uwagi dotyczące systemu BSD/OS 2.x”),
- ◆ DEC UNIX 4.x z natywną obsługą wątków (zobacz podpunkt 2.6.5.5. „Uwagi dotyczące systemu Alpha-DEC-Unix (Tru64”),
- ◆ FreeBSD 2.x z pakietem MIT-pthreads (zobacz podpunkt 2.6.4.1. „Uwagi dotyczące systemu FreeBSD”),
- ◆ FreeBSD 3.x i 4.x z natywną obsługą wątków (zobacz podpunkt 2.6.4.1. „Uwagi dotyczące systemu FreeBSD”),
- ◆ FreeBSD 4.x z biblioteką LinuxThreads (zobacz podpunkt 2.6.4.1. „Uwagi dotyczące systemu FreeBSD”),
- ◆ HP-UX 10.20 z wątkami DCE albo pakietem MIT-pthreads (zobacz podpunkt 2.6.5.1. „Uwagi dotyczące systemu HP-UX 10.20”),
- ◆ HP-UX 11.x z natywną obsługą wątków (zobacz podpunkt 2.6.5.2. „Uwagi dotyczące systemu HP-UX 11.x”),
- ◆ Linux 2.0+ z biblioteką LinuxThreads 0.7.1+ albo glibc 2.0.7+ (zobacz punkt 2.6.1. „Uwagi dotyczące systemu Linux”),
- ◆ Mac OS X (zobacz punkt 2.6.2. „Uwagi dotyczące systemu Mac OS X”),
- ◆ NetBSD 1.3 i 1.4 Intel oraz NetBSD 1.3 Alpha (wymaga programu GNU make; zobacz podpunkt 2.6.4.2. „Uwagi dotyczące systemu NetBSD”),
- ◆ Novell NetWare 6.0 (zobacz punkt 2.2.4. „Instalowanie MySQL w systemie NetWare”),
- ◆ OpenBSD > 2.5 z natywną obsługą wątków, OpenBSD < 2.5 z pakietem MIT-pthreads (zobacz podpunkt 2.6.4.3. „Uwagi dotyczące systemu OpenBSD 2.5”),
- ◆ OS/2 Warp 3, FixPack 29 oraz OS/2 Warp 4, FixPack4 (zobacz punkt 2.6.6. „Uwagi dotyczące systemu OS/2”),
- ◆ SCO OpenServer z nową wersją pakietu FSU Pthreads (zobacz podpunkt 2.6.5.8. „Uwagi dotyczące systemu SCO”),
- ◆ SCO UnixWare 7.1.x (zobacz podpunkt 2.6.5.9. „Uwagi dotyczące systemu SCO UnixWare 7.1.x”),
- ◆ SGI Irix 6.x z natywną obsługą wątków (zobacz podpunkt 2.6.5.7. „Uwagi dotyczące systemu SGI Irix”),
- ◆ Solaris 2.5 i nowsze wersje z natywną obsługą wątków na procesorach SPARC i x86 (zobacz punkt 2.6.3. „Uwagi dotyczące systemu Solaris”),
- ◆ SunOS 4.x z pakietem MIT-pthreads (zobacz punkt 2.6.3. „Uwagi dotyczące systemu Solaris”),
- ◆ Tru64 Unix,
- ◆ Windows 9x, Me, NT, 2000 oraz XP (zobacz punkt 2.2.1. „Instalowanie MySQL w Windows”).

Nie na wszystkich platformach oprogramowanie MySQL działa z taką samą wydajnością. Zdolność danej platformy do obsługi mocno obciążonego, krytycznego serwera MySQL zależy od następujących czynników:

- ♦ Ogólnej stabilności biblioteki wątków. Platforma może cieszyć się doskonałą reputacją, ale jeśli biblioteka wątków zawiera niestabilny kod wywoływany przez MySQL, to oprogramowanie MySQL również będzie niestabilne, nawet jeśli wszystko inne działa idealnie.
- ♦ Zdolności jądra systemu oraz biblioteki wątków do wykorzystania symetrycznych systemów wieloprocesorowych (SMP). Innymi słowy, wątek utworzony przez proces powinien móc działać na innym procesorze niż pierwotny proces.
- ♦ Zdolności jądra systemu oraz biblioteki wątków do wykonywania wielu wątków, które często zakładają i zwalniają muteks na krótkim, krytycznym obszarze kodu, bez nadmiernego przełączania kontekstów. Jeśli implementacja funkcji `pthread_mutex_lock()` zbyt szybko oddaje czas procesora, wydajność MySQL znacznie się zmniejsza. Jeśli ta kwestia nie zostanie rozwiązana, dodanie kolejnych procesorów tylko spowolni działanie MySQL.
- ♦ Ogólnej stabilności i wydajności systemu plików.
- ♦ Jeśli tabele są duże — od tego, czy system plików w ogóle potrafi obsługiwać duże pliki, a jeśli tak, to czy obsługuje je wydajnie.
- ♦ Wiedzy programistów firmy MySQL AB o danej platformie. Jeśli dobrze znamy platformę, włączamy specyficzne optymalizacje i poprawki podczas kompilacji oprogramowania. Możemy również doradzić, jak optymalnie skonfigurować system na użytek MySQL.
- ♦ Liczby testów, które wykonaliśmy wewnątrz w podobnych konfiguracjach.
- ♦ Liczby użytkowników, którzy z powodzeniem używają MySQL na podobnie skonfigurowanych platformach. Jeśli ta liczba jest duża, prawdopodobieństwo wystąpienia problemów specyficznych dla platformy jest znacznie mniejsze.

Według powyższych kryteriów, obecnie najlepszą platformą do uruchamiania MySQL jest komputer x86 z systemem SuSE Linux i jądrem 2.4 oraz systemem plików ReiserFS (albo podobną dystrybucją systemu Linux) lub komputer SPARC z systemem Solaris (2.7-9). FreeBSD zajmuje trzecie miejsce, ale liczymy na to, że po ulepszeniu biblioteki wątków dołączy do elity. Mamy też nadzieję, że kiedyś do tej kategorii będzie można zaliczyć wszystkie inne platformy, na których można skompilować i uruchomić MySQL, choć z nieco gorszą stabilnością i wydajnością. Będzie to wymagało współpracy z twórcami systemów operacyjnych i komponentów bibliotecznych, z których korzysta MySQL. Osoby, które są zainteresowane ulepszeniem któregoś z tych komponentów, mają wpływ na ich rozwój i potrzebują dokładniejszych informacji o tym, co trzeba zrobić, aby oprogramowanie MySQL działało lepiej, powinny przesłać wiadomość na listę dyskusyjną `internals`. Zobacz podpunkt 1.7.1.1. „Listy dyskusyjne MySQL”.

Chcemy podkreślić, że powyższe porównanie nie ma sugerować, że jeden system operacyjny jest gorszy albo lepszy od drugiego. Mówimy tylko o wyborze systemu operacyjnego do specyficznego celu: uruchamiania serwera MySQL. Wynik porównania

byłby inny, gdybyśmy wzięli pod uwagę więcej czynników. W niektórych przypadkach wyższość jednego systemu operacyjnego nad drugim może wynikać z tego, że włożyliśmy więcej pracy w testowanie i optymalizowanie serwera pod kątem konkretnej platformy. Podajemy tylko nasze spostrzeżenia, aby ułatwić użytkownikowi wybór platformy dla serwera MySQL.

2.1.2. Wybór dystrybucji MySQL

Podczas przygotowywania się do instalacji MySQL należy zdecydować się na którąś z wersji. Oprogramowanie MySQL jest rozwijane w kilku seriach, a użytkownik może wybrać tę, która najlepiej odpowiada jego wymaganiom. Po wybraniu wersji można wybrać format dystrybucji. Poszczególne wersje są dostępne w formacie binarnym lub źródłowym.

2.1.2.1. Wybór wersji MySQL

Użytkownik musi najpierw zdecydować, czy chce korzystać z wersji produkcyjnej (stabilnej), czy rozwojowej. Istnieje wiele serii oprogramowania MySQL znajdujących się w różnych fazach dojrzałości:

- ◆ MySQL 5.0 to najnowsza seria wersji rozwojowych, do których dodawanych jest wiele nowych funkcji. Do niedawna była dostępna tylko w postaci demonstracyjnej w repozytorium kodu źródłowego BitKeeper. Obecnie wydano wersję alfa, aby mogło ją przetestować więcej użytkowników.
- ◆ MySQL 4.1 to seria wersji rozwojowych, do których dodano ważne nowe funkcje. Obecnie ma status beta. Dostępny jest kod źródłowy i pliki binarne, które można testować w systemach pilotażowych.
- ◆ MySQL 4.0 to bieżąca seria wersji produkcyjnych (stabilnych). Nowe wydania mają na celu poprawienie usterek. Nie dodaje się żadnych nowych funkcji, które mogłyby naruszyć stabilność kodu.
- ◆ MySQL 3.23 to poprzednia seria wersji produkcyjnych (stabilnych). Ta seria nie jest już rozwijana, więc nowe wydania mają na celu tylko poprawienie krytycznych usterek.

Nie jesteśmy zwolennikami całkowitego zamrożenia kodu, ponieważ eliminuje to również poprawki usterek oraz rzeczy, które „trzeba zrobić”. Sformułowanie „prawie zamrożony” oznacza, że możemy wprowadzić niewielkie zmiany, które „niemal na pewno nie wpłyną na to, co już działa prawidłowo”. Oczywiście, poprawki starszych wersji są przenoszone również do nowszych.

Tym, którzy zaczynają korzystać z oprogramowania MySQL albo próbują przenieść je do systemu, który nie ma dystrybucji binarnej, polecamy skorzystanie z bieżącej wersji produkcyjnej. Obecnie jest to MySQL 4.0. Wszystkie wersje MySQL, również te z serii rozwojowych, są sprawdzane za pomocą narzędzi MySQL i obszernych zestawów testowych, zanim zostaną opublikowane.

Ci, którzy korzystają ze starego systemu i chcą go zaktualizować, ale woleliby uniknąć związanych z tym problemów, powinni zainstalować najnowszą wersję z obecnie używanej serii (taką, w której tylko ostatnia część numeru wersji jest większa od obecnie używanej). W tej wersji staraliśmy się naprawić tylko krytyczne usterki i wprowadzić niewielkie, względnie bezpieczne zmiany.

Użytkownicy chcący korzystać z nowych funkcji, których nie ma w bieżącej serii produkcyjnej, mogą zainstalować wersję z serii rozwojowej. Należy jednak pamiętać, że wersje rozwojowe są mniej stabilne od produkcyjnych.

Ci, którzy chcą używać najnowszych wersji źródłowych ze wszystkimi bieżącymi poprawkami usterek, mogą skorzystać z jednego spośród naszych repozytoriów BitKeeper. Nie są to „wydania” jako takie, ale udostępniamy je jako wersję demonstracyjną kodu, na którym będą oparte przyszłe wydania.

W schemacie nazewniczym MySQL nazwy wydań składają się z trzech liczb i przyrostka, na przykład `mysql-4.1.2-alpha`. Liczby w nazwie wydania należy interpretować następująco:

- ♦ Pierwsza liczba (4) wskazuje główny numer wersji, a także format pliku. Wszystkie wydania wersji 4 mają ten sam format pliku.
- ♦ Druga liczba (1) to poziom wydania. Główny numer wersji oraz poziom wydania razem tworzą numer serii.
- ♦ Trzecia liczba (2) to numer wersji w obrębie danej serii. Jest on zwiększany w każdym nowym wydaniu. Zwykle należy zaopatrzyć się w najnowszą wersję z wybranej serii.

Podczas każdej pomniejszej aktualizacji zwiększana jest ostatnia liczba w numerze wersji. Kiedy wprowadzane są nowe funkcje albo pomniejsze niezgodności z poprzednią wersją, zwiększana jest druga liczba. Kiedy zmienia się format pliku, zwiększana jest pierwsza liczba.

W nazwach wydań znajduje się również przyrostek, który wskazuje stabilność wydania. W poszczególnych wydaniach należących do danej serii przyrostek zmienia się zgodnie z rosnącym poziomem stabilności. Oto możliwe przyrostki:

- ♦ Przyrostek `alpha` wskazuje, że wydanie zawiera duże fragmenty nowego kodu, który nie został przetestowany w 100 procentach. Znane usterki (zwykle takich nie ma) powinny być udokumentowane w rozdziale „News” podręcznika online pod adresem <http://dev.mysql.com/doc/mysql/en/News.html>. W większości wydań alfa znajdują się także nowe polecenia i rozszerzenia. Wydanie alfa jest aktywnie rozwijane, co może prowadzić do istotnych zmian w kodzie, ale wszystko jest testowane przed opublikowaniem wydania. Z tej przyczyny w żadnym wydaniu MySQL nie powinno być znanych usterek.
- ♦ Przyrostek `beta` oznacza, że nowy kod został przetestowany. Nie dodano żadnych nowych funkcji, które mogłyby spowodować uszkodzenie starego kodu. W wersji beta nie powinno być żadnych znanych usterek. Zmiana

wersji alfa na beta następuje wtedy, gdy nikt nie zgłosi krytycznych usterek w wersji alfa przez co najmniej miesiąc i nie zamierzamy dodawać nowych funkcji, które mogłyby zakłócić działanie starych poleceń.

- ◆ Wersja gamma to wersja beta, która od pewnego czasu znajduje się w obiegu i wydaje się działać prawidłowo. Do wersji gamma dodawane są tylko niewielkie poprawki. Taką wersję inne firmy nazywają wydaniem.
- ◆ Jeśli w nazwie brak przyrostka, oznacza to, że dana wersja była używana przez pewien czas w wielu różnych ośrodkach bez raportów o usterekach (nie licząc usterek specyficznych dla platformy). Do takiego wydania dodawane są tylko poprawki krytycznych usterek. Taką wersję nazywamy wydaniem produkcyjnym (stabilnym).

MySQL używa nieco odmiennej konwencji nazewnictwa niż inne produkty. Ogólnie rzecz biorąc, można względnie bezpiecznie używać każdej wersji, która jest dostępna od kilku tygodni i nie została zastąpiona nową wersją z określonej serii.

Wszystkie wydania MySQL są poddawane standardowym testom poprawności i wydajności, co ma gwarantować względne bezpieczeństwo ich użycia. Ponieważ standardowe testy są z czasem rozszerzane o funkcje wykrywania wszystkich uprzednio znalezionych usterek, zestaw testowy staje się coraz lepszy.

Wszystkie wydania są testowane przynajmniej za pomocą:

- ◆ Wewnętrznego zestawu testowego
Katalog *mysql-test* zawiera obszerny zbiór przypadków testowych. Wykonujemy te testy na praktycznie każdym wydaniu binarnym serwera.
- ◆ Zestawu testów wydajnościowych MySQL
Ten zestaw wykonuje różne często używane zapytania. Pozwala też sprawdzić, czy najnowsze optymalizacje rzeczywiście przyspieszyły kod. Zobacz punkt 6.1.4. „Pakiet testowania wydajności MySQL”.
- ◆ Testu *crash-me*
Ten test próbuje ustalić, jakie funkcje obsługuje baza danych i jakie są jej możliwości oraz ograniczenia. Zobacz punkt 6.1.4. „Pakiet testowania wydajności MySQL”.

Inny test polega na tym, że używamy najnowszej wersji MySQL w naszym wewnętrznym środowisku produkcyjnym, w przynajmniej jednym serwerze. Mamy ponad 100 GB danych, z którymi może pracować serwer.

2.1.2.2. Wybór formatu dystrybucji

Po wybraniu wersji MySQL trzeba zdecydować, czy zainstalowana zostanie dystrybucja binarna, czy źródłowa. W większości przypadków lepiej jest użyć dystrybucji binarnej, jeśli dostępna jest wersja przeznaczona dla danej platformy. Dystrybucje binarne są dostępne w natywnych formatach wielu platform, takich jak linuksowe pliki RPM albo pakiety DMG dla systemu Mac OS X. Dostępne są też w postaci archiwów *Zip* albo skompresowanych plików *tar*.

Oto powody, dla których warto wybrać dystrybucję binarną:

- ♦ Dystrybucje binarne zwykle łatwiej jest zainstalować niż dystrybucje źródłowe.
- ♦ Aby spełnić wymagania różnych użytkowników, udostępniamy dwie wersje binarne: jedną skompilowaną z nietransakcyjnymi mechanizmami składowania (mały, szybki plik binarny) i jedną uzupełnioną najważniejszymi rozszerzeniami, takimi jak tabele transakcyjne. Obie wersje są kompilowane z tej samej dystrybucji źródłowej. Wszystkie natywne klienty MySQL mogą łączyć się z obiema wersjami serwera MySQL.

Rozszerzona dystrybucja binarna MySQL jest oznaczona przyrostkiem `-max` i skonfigurowana z tymi samymi opcjami co `mysqld-max`. Zobacz punkt 4.1.2. „Rozszerzony serwer MySQL `mysqld-max`”.

Ci, którzy chcą użyć pakietu RPM MySQL-Max, muszą najpierw zainstalować pakiet RPM MySQL-Server.

W niektórych okolicznościach lepiej jest zainstalować MySQL z dystrybucji źródłowej:

- ♦ Serwer MySQL ma być zainstalowany w konkretnej lokalizacji. Standardowe dystrybucje binarne mogą działać w dowolnym katalogu, ale niektórzy chcą mieć jeszcze większą swobodę rozmieszczania poszczególnych komponentów MySQL.
- ♦ Serwer `mysqld` ma być wyposażony w dodatkowe funkcje, które nie są dołączone do standardowej dystrybucji źródłowej. Oto lista najczęściej używanych opcji dodatkowych:
 - ♦ `--with-innodb` (opcja domyślna w MySQL 4.0 i nowszych wersjach)
 - ♦ `--with-berkeley-db` (nieдоступna na niektórych platformach)
 - ♦ `--with-raid`
 - ♦ `--with-libwrap`
 - ♦ `--with-named-z-libs` (używana podczas kompilowania niektórych plików binarnych)
 - ♦ `--with-debug[=full]`
- ♦ Serwer `mysqld` ma być pozbawiony niektórych funkcji, które są dołączone do standardowej dystrybucji źródłowej. Dystrybucje są na przykład kompilowane z obsługą wszystkich zestawów znaków. Ci, którzy chcą zmniejszyć serwer, mogą skompilować go z obsługą tylko tych zestawów, które są im potrzebne.
- ♦ Serwer będzie kompilowany za pomocą specjalnego kompilatora (takiego jak `gcc`) albo z opcjami zoptymalizowanymi pod kątem konkretnego procesora. Dystrybucje binarne są kompilowane z opcjami, które powinny działać na różnych procesorach z danej rodziny.
- ♦ Użytkownik chce wykorzystać najnowsze źródła z repozytorium BitKeeper, aby uzyskać dostęp do najnowszych poprawek. Jeśli ktoś na przykład odkryje usterkę i poinformuje o niej zespół programistów MySQL, poprawka zostanie umieszczona w repozytorium źródłowym. Poprawka zostanie dołączona do wersji binarnej dopiero podczas publikacji kolejnego wydania MySQL.

- ◆ Użytkownik chce przeczytać (lub zmodyfikować) kod C i C++ składający się na MySQL. W takim przypadku należy pobrać dystrybucję źródłową, ponieważ kod źródłowy zawsze jest najbardziej definitywnym podręcznikiem.
- ◆ Dystrybucje źródłowe zawierają więcej testów i przykładów niż dystrybucje binarne.

2.1.2.3. Jak i kiedy publikowane są aktualizacje?

Serwer MySQL rozwija się bardzo szybko, a firma MySQL AB chętnie dzieli się najnowszymi postęпами z innymi użytkownikami MySQL. Próbujemy opublikować nową wersję, kiedy dodamy do niej przydatne funkcje, których potrzebują inni użytkownicy.

Próbujemy także pomóc użytkownikom proszącym o dodanie funkcji, które są łatwe w implementacji. Uwzględniamy życzenia naszych licencjonowanych użytkowników, a szczególną uwagę zwracamy na to, czego potrzebują abonenci pomocy technicznej, i próbujemy im pomóc.

Nie każdy musi pobierać nową wersję. W rozdziale „News” podręcznika online można sprawdzić, czy nowa wersja zawiera coś, co jest rzeczywiście potrzebne. Zobacz <http://dev.mysql.com/doc/mysql/en/News.html>.

Aktualizacje MySQL podlegają poniższym zasadom:

- ◆ Publikowane są wydania w ramach każdej serii. W każdym wydaniu ostatnia liczba numeru wersjii jest większa o jeden niż w poprzednim wydaniu z tej samej serii.
- ◆ Wydania produkcyjne (stabilne) pojawiają się raz lub dwa razy na rok. W razie znalezienia niewielkich usterek zostanie jednak opublikowane wydanie zawierające tylko ich poprawki.
- ◆ Wydania robocze oraz poprawki starszych wydań pojawiają się co 4 – 8 tygodni.
- ◆ Tworzymy dystrybucje binarne głównych wydań dla niektórych platform. Inni tworzą dystrybucje binarne dla pozostałych systemów, ale prawdopodobnie z mniejszą częstotliwością.
- ◆ Udostępniamy poprawki, kiedy tylko zidentyfikujemy i poprawimy niewielkie, ale irytujące usterek. Poprawki są natychmiast udostępniane w repozytoriach BitKeeper i zostają dołączone do następnego wydania.
- ◆ W razie znalezienia krytycznego błędu w wydaniu, jak najszybciej przygotowujemy nowe wydanie (chcielibyśmy, żeby inne firmy również tak postępowały!).

2.1.2.4. Filozofia publikowania wydań — żadnych znanych usterek w wydaniu

Wkładamy mnóstwo wysiłku w to, aby wydania były wolne od usterek. O ile nam wiadomo, nie opublikowaliśmy choćby jednej wersji MySQL ze *znaną* „krytyczną” usterką (usterka „krytyczna” to coś, co załamuje normalnie używany serwer MySQL, powoduje zwrócenie błędnej odpowiedzi na poprawne zapytanie albo narusza bezpieczeństwo).

Dokumentujemy wszystkie nierozwiązane problemy, usterki i kwestie mające związek z decyzjami projektowymi. Zobacz punkt 1.8.7. „Znane błędy i wady projektowe MySQL”.

Naszym celem jest naprawienie wszystkiego, co można naprawić, bez naruszania stabilności danej wersji MySQL. W niektórych przypadkach oznacza to, że możemy rozwiązać problem w wersjach rozwojowych, ale nie w stabilnych (produkcyjnych). Oczywiście, dokumentujemy takie problemy, aby użytkownicy o nich wiedzieli.

Oto procedura budowania nowych wersji MySQL:

- ♦ Monitorujemy usterki przesyłane na naszą listę pomocy technicznej, do bazy usterek pod adresem <http://bugs.mysql.com/> oraz na zewnętrzne listy dyskusyjne MySQL.
- ♦ Wszystkie zgłoszone usterki są wprowadzane do bazy usterek.
- ♦ Kiedy poprawiamy usterkę, staramy się utworzyć odpowiedni przypadek testowy i dołączyć go do naszego zestawu testowego, aby upewnić się, że w razie ponownego wystąpienia usterki zostanie ona wykryta (niemal 90 procent poprawionych usterek ma swój przypadek testowy).
- ♦ Tworzymy przypadki testowe dla wszystkich nowych funkcji, które dodajemy do MySQL.
- ♦ Zanim zaczniemy budować nowe wydanie MySQL, sprawdzamy, czy wszystkie zgłoszone, powtarzalne usterki poszczególnych wersji (3.23.x, 4.0.x itp.) zostały poprawione. Jeśli czegoś nie da się poprawić (ze względu na pewną wewnętrzną decyzję projektową), dokumentujemy to w podręczniku. Zobacz punkt 1.8.7. „Znane błędy i wady projektowe MySQL”.
- ♦ Budujemy dystrybucję binarną na wszystkich obsługiwanych przez nas platformach (ponad 15 platform) i na każdej z nich wykonujemy testy poprawności i wydajności.
- ♦ Nie publikujemy dystrybucji binarnej dla platformy, na której test poprawności albo wydajności zakończył się niepowodzeniem. Jeśli problem wynika z ogólnego błędu w kodzie źródłowym, poprawiamy błąd i ponownie budujemy oraz testujemy dystrybucję na wszystkich platformach.
- ♦ Proces budowy i testowania trwa od 2 do 3 dni. Jeśli podczas tego procesu otrzymamy raport o krytycznej usterce (na przykład takiej, która powoduje załamanie serwera), poprawiamy ją i rozpoczynamy proces budowy od nowa.
- ♦ Po opublikowaniu plików binarnych pod adresem <http://dev.mysql.com/> wysyłamy powiadomienie na listy dyskusyjne `mysql` oraz `announce`. Zobacz podpunkt 1.7.1.1. „Listy dyskusyjne MySQL”. Powiadomienie zawiera informacje o wszystkich zmianach wprowadzonych w wydaniu i o wszystkich znanych problemach. Opis znanych problemów był potrzebny tylko w nielicznych wydaniach.
- ♦ Aby użytkownicy mogli szybko uzyskać dostęp do najnowszych wersji MySQL, publikujemy nowe wydanie co 4 – 8 tygodni. Migawki kodu źródłowego są budowane codziennie i dostępne pod adresem <http://downloads.mysql.com/snapshots.php>.

- ◆ Jeśli pomimo naszych starań po opublikowaniu wydania otrzymamy informacje, że wersja dla którejś z platform zawiera krytyczne usterki, poprawiamy je natychmiast i budujemy nowe wydanie dla tej platformy, oznaczając je literą a. Dzięki dużej społeczności użytkowników problemy są odkrywane szybko.
- ◆ Publikowane przez nas wydania są zwykle dobrej jakości. Na 150 ostatnich wydań musieliśmy przebudować mniej niż 10. W trzech spośród tych przypadków usterka była spowodowana błędną biblioteką `glibc` w jednym z komputerów używanych do budowania wydań, a ustalenie tego zajęło nam sporo czasu.

2.1.2.5. Dystrybucje binarne MySQL kompilowane przez MySQL AB

Firma MySQL AB udostępnia zbiór binarnych dystrybucji MySQL kompilowanych w naszych systemach albo tam, gdzie zwolennicy MySQL pozwalają nam korzystać ze swoich maszyn.

Oprócz plików binarnych dostarczanych w pakietach specyficznych dla platformy oferujemy dystrybucje binarne dla niektórych platform w postaci skompresowanych plików *tar* (*.tar.gz*). Zobacz podrozdział 2.2. „Standardowa instalacja MySQL z dystrybucji binarnej”.

Informacje o dystrybucjach dla Windows znajdują się w punkcie 2.2.1. „Instalowanie MySQL w Windows”.

Dystrybucje te są generowane przez skrypt *Build-tools/Do-compile*, który kompiluje kod źródłowy i tworzy binarne archiwum *tar.gz* za pomocą skryptu *scripts/make_binary_distribution*.

Pliki binarne są konfigurowane i budowane z wykorzystaniem wymienionych niżej kompilatorów i opcji. Informacje te możemy również uzyskać, patrząc na zmienne `COMP_ENV_INFO` oraz `CONFIGURE_LINE` w skrypcie *bin/mysqlbug* dołączonym do każdej binarnej dystrybucji *tar*.

Poniższe pliki binarne są budowane w systemach firmy MySQL AB:

- ◆ Linux 2.4.xx x86 z kompilatorem gcc 2.95.3:

```
CFLAGS="-O2 -mcpu=pentiumpro" CXX=gcc CXXFLAGS="-O2 -mcpu=pentiumpro -felide-constructors" ./configure --prefix=/usr/local/mysql --with-extra-charsets=complex --enable-thread-safe-client --enable-local-infile --enable-asm --disable-shared --with-client-ldflags=-all-static --with-mysqld-ldflags=-all-static
```

- ◆ Linux 2.4.xx Intel Itanium 2 z kompilatorem ecc (Intel C++ Itanium Compiler 7.0):

```
CC=ecc CFLAGS="-O2 -tpp2 -ip -nolib_inline" CXX=ecc CXXFLAGS="-O2 -tpp2 -ip -nolib_inline" ./configure --prefix=/usr/local/mysql --with-extra-charsets=complex --enable-thread-safe-client --enable-local-infile
```

- ◆ Linux 2.4.xx Intel Itanium z kompilatorem ecc (Intel C++ Itanium Compiler 7.0):

```
CC=ecc CFLAGS=-tpp1 CXX=ecc CXXFLAGS=-tpp1 ./configure --prefix=/usr/local/mysql --with-extra-charsets=complex --enable-thread-safe-client --enable-local-infile
```

- ♦ **Linux 2.4.xx alpha z kompilatorem ccc (Compaq C V6.2-505 / Compaq C++ V6.3-006):**

```
CC=ccc CFLAGS="-fast -arch generic" CXX=cxx CXXFLAGS="-fast -arch generic -noexceptions -nortti" ./configure --prefix=/usr/local/mysql --with-extra-charset=complex --enable-thread-safe-client --enable-local-infile --with-mysqld-ldflags=-non_shared --with-client-ldflags=-non_shared --disable-shared
```

- ♦ **Linux 2.x.xx ppc z kompilatorem gcc 2.95.4:**

```
CC=gcc CFLAGS="-O3 -fno-omit-frame-pointer" CXX=gcc CXXFLAGS="-O3 -fno-omit-frame-pointer -felide-constructors -fno-exceptions -fno-rtti" ./configure --prefix=/usr/local/mysql --localstatedir=/usr/local/mysql/data --libexecdir=/usr/local/mysql/bin --with-extra-charset=complex --enable-thread-safe-client --enable-local-infile --disable-shared --with-embedded-server --with-innodb
```

- ♦ **Linux 2.4.xx s390 z kompilatorem gcc 2.95.3:**

```
CFLAGS="-O2" CXX=gcc CXXFLAGS="-O2 -felide-constructors" ./configure --prefix=/usr/local/mysql --with-extra-charset=complex --enable-thread-safe-client --enable-local-infile --disable-shared --with-client-ldflags=-all-static --with-mysqld-ldflags=-all-static
```

- ♦ **Linux 2.4.xx x86_64 (AMD64) z kompilatorem gcc 3.2.1:**

```
CXX=gcc ./configure --prefix=/usr/local/mysql --with-extra-charset=complex --enable-thread-safe-client --enable-local-infile --disable-shared
```

- ♦ **Sun Solaris 8 x86 z kompilatorem gcc 3.2.3:**

```
CC=gcc CFLAGS="-O3 -fno-omit-frame-pointer" CXX=gcc CXXFLAGS="-O3 -fno-omit-frame-pointer -felide-constructors -fno-exceptions -fno-rtti" ./configure --prefix=/usr/local/mysql --localstatedir=/usr/local/mysql/data --libexecdir=/usr/local/mysql/bin --with-extra-charset=complex --enable-thread-safe-client --enable-local-infile --disable-shared --with-innodb
```

- ♦ **Sun Solaris 8 SPARC z kompilatorem gcc 3.2:**

```
CC=gcc CFLAGS="-O3 -fno-omit-frame-pointer" CXX=gcc CXXFLAGS="-O3 -fno-omit-frame-pointer -felide-constructors -fno-exceptions -fno-rtti" ./configure --prefix=/usr/local/mysql --with-extra-charset=complex --enable-thread-safe-client --enable-local-infile --enable-assembly --with-named-z-libs=no --with-named-curses-libs=-lcurses --disable-shared
```

- ♦ **Sun Solaris 8 SPARC (64-bitowy) z kompilatorem gcc 3.2:**

```
CC=gcc CFLAGS="-O3 -m64 -fno-omit-frame-pointer" CXX=gcc CXXFLAGS="-O3 -m64 -fno-omit-frame-pointer -felide-constructors -fno-exceptions -fno-rtti" ./configure --prefix=/usr/local/mysql --with-extra-charset=complex --enable-thread-safe-client --enable-local-infile --with-named-z-libs=no --with-named-curses-libs=-lcurses --disable-shared
```

- ♦ **Sun Solaris 9 SPARC z kompilatorem gcc 2.95.3:**

```
CC=gcc CFLAGS="-O3 -fno-omit-frame-pointer" CXX=gcc CXXFLAGS="-O3 -fno-omit-frame-pointer -felide-constructors -fno-exceptions -fno-rtti" ./configure --prefix=/usr/local/mysql --with-extra-charset=complex --enable-thread-safe-client --enable-local-infile --enable-assembly --with-named-curses-libs=-lcurses --disable-shared
```

◆ Sun Solaris 9 SPARC z kompilatorem cc-5.0 (Sun Forte 5.0):

```
CC=cc-5.0 CXX=CC ASFLAGS="-xarch=v9" CFLAGS="-Xa -xstrconst -mt -D_FORTEC_
-xarch=v9" CXXFLAGS="-noex -mt -D_FORTEC_ -xarch=v9" ./configure --prefix=
/usr/local/mysql --with-extra-charsets=complex --enable-thread-safe-client
--enable-local-infile --enable-asmbler --with-named-z-libs=no --enable-
thread-safe-client --disable-shared
```

◆ IBM AIX 4.3.2 ppc z kompilatorem gcc 3.2.3:

```
CFLAGS="-O2 -mcpu=powerpc -Wa,-many " CXX=gcc CXXFLAGS="-O2 -mcpu=powerpc
-Wa,-many -felide-constructors -fno-exceptions -fno-rtti" ./configure --
prefix=/usr/local/mysql --with-extra-charsets=complex --enable-thread-safe-
client --enable-local-infile --with-named-z-libs=no --disable-shared
```

◆ IBM AIX 4.3.3 ppc z kompilatorem x1C_r (IBM Visual Age C/C++ 6.0):

```
CC=x1C_r CFLAGS="-ma -O2 -qstrict -qoptimize=2 -qmaxmem=8192" CXX=x1C_r
CXXFLAGS="-ma -O2 -qstrict -qoptimize=2 -qmaxmem=8192" ./configure --
prefix=/usr/local/mysql --localstatedir=/usr/local/mysql/data --libexecdir=
usr/local/mysql/bin --with-extra-charsets=complex --enable-thread-safe-client
--enable-local-infile --with-named-z-libs=no --disable-shared --with-innodb
```

◆ IBM AIX 5.1.0 ppc z kompilatorem gcc 3.3:

```
CFLAGS="-O2 -mcpu=powerpc -Wa,-many" CXX=gcc CXXFLAGS="-O2 -mcpu=powerpc
-Wa,-many -felide-constructors -fno-exceptions -fno-rtti" ./configure --
prefix=/usr/local/mysql --with-extra-charsets=complex --enable-thread-safe-
client --enable-local-infile --with-named-z-libs=no --disable-shared
```

◆ IBM AIX 5.2.0 ppc z kompilatorem x1C_r (IBM Visual Age C/C++ 6.0):

```
CC=x1C_r CFLAGS="-ma -O2 -qstrict -qoptimize=2 -qmaxmem=8192" CXX=x1C_r
CXXFLAGS="-ma -O2 -qstrict -qoptimize=2 -qmaxmem=8192" ./configure --
prefix=/usr/local/mysql --localstatedir=/usr/local/mysql/data --libexecdir=
usr/local/mysql/bin --with-extra-charsets=complex --enable-thread-safe-client
--enable-local-infile --with-named-z-libs=no --disable-shared --with-embedded-
server --with-innodb
```

◆ HP-UX 10.20 pa-risc1.1 z kompilatorem gcc 3.1:

```
CFLAGS="-DHPUX -I/opt/dce/include -O3 -fPIC" CXX=gcc CXXFLAGS="-DHPUX -
I/opt/dce/include -felide-constructors -fno-exceptions -fno-rtti -O3 -fPIC"
./configure --prefix=/usr/local/mysql --with-extra-charsets=complex --enable-
thread-safe-client --enable-local-infile --with-pthread --with-named-thread-
libs=-ldce --with-lib-ccflags=-fPIC --disable-shared
```

◆ HP-UX 11.00 pa-risc z kompilatorem aCC (HP ANSI C++ B3910B A.03.50):

```
CC=cc CXX=aCC CFLAGS=+DAportable CXXFLAGS=+DAportable ./configure --prefix=
usr/local/mysql --localstatedir=/usr/local/mysql/data --libexecdir=/usr/
local/mysql/bin --with-extra-charsets=complex --enable-thread-safe-client --
enable-local-infile --disable-shared --with-embedded-server --with-innodb
```

◆ HP-UX 11.11 pa-risc2.0 (64-bitowy) z kompilatorem aCC (HP ANSI C++ B3910B A.03.33):

```
CC=cc CXX=aCC CFLAGS=+DD64 CXXFLAGS=+DD64 ./configure --prefix=/usr/local/
mysql --with-extra-charsets=complex --enable-thread-safe-client --enable-
local-infile --disable-shared
```

- ♦ HP-UX 11.11 pa-risc2.0 (32-bitowy) z kompilatorem aCC (HP ANSI C++ B3910B A.03.33):

```
CC=cc CXX=aCC CFLAGS="+DAportable" CXXFLAGS="+DAportable" ./configure --
prefix=/usr/local/mysql --localstatedir=/usr/local/mysql/data --libexecdir=/
usr/local/mysql/bin --with-extra-charsets=complex --enable-thread-safe-client
--enable-local-infile --disable-shared --with-innodb
```

- ♦ HP-UX 11.22 ia64 (64-bitowy) z kompilatorem aCC (HP aC++/ANSI C B3910B A.05.50):

```
CC=cc CXX=aCC CFLAGS="+DD64 +DSitanium2" CXXFLAGS="+DD64 +DSitanium2"
./configure --prefix=/usr/local/mysql --localstatedir=/usr/local/mysql/data --
libexecdir=/usr/local/mysql/bin --with-extra-charsets=complex --enable-thread-
safe-client --enable-local-infile --disable-shared --with-embedded-server --
with-innodb
```

- ♦ Apple Mac OS X 10.2 powerpc z kompilatorem gcc 3.1:

```
CC=gcc CFLAGS="-O3 -fno-omit-frame-pointer" CXX=gcc CXXFLAGS="-O3 -fno-omit-
frame-pointer -felide-constructors -fno-exceptions -fno-rtti" ./configure --
prefix=/usr/local/mysql --with-extra-charsets=complex --enable-thread-safe-
client --enable-local-infile --disable-shared
```

- ♦ FreeBSD 4.7 i386 z kompilatorem gcc 2.95.4:

```
CFLAGS=-DHAVE_BROKEN_REALPATH ./configure --prefix=/usr/local/mysql --with-
extra-charsets=complex --enable-thread-safe-client --enable-local-infile --
enable-asmbler --with-named-z-libs=not-used --disable-shared
```

- ♦ FreeBSD 4.7 i386 z biblioteką LinuxThreads, z kompilatorem gcc 2.95.4:

```
CFLAGS="-DHAVE_BROKEN_REALPATH -D_USE_UNIX98 -D_REENTRANT -D_THREAD_SAFE -
I/usr/local/include/pthread/linuxthreads" CXXFLAGS="-DHAVE_BROKEN_REALPATH -
D_USE_UNIX98 -D_REENTRANT -D_THREAD_SAFE -I/usr/local/include/pthread/
linuxthreads" ./configure --prefix=/usr/local/mysql --localstatedir=/usr/
local/mysql/data --libexecdir=/usr/local/mysql/bin --enable-thread-safe-client
--enable-local-infile --enable-asmbler --with-named-thread-libs="-DHAVE_
GLIBC2_STYLE_GETHOSTBYNAME_R -D_THREAD_SAFE -I /usr/local/include/pthread/
linuxthreads -L/usr/local/lib -llthread -llgcc_r" --disable-shared --with-
embedded-server --with-innodb
```

- ♦ QNX Neutrino 6.2.1 i386 z kompilatorem gcc 2.95.3qnx-nto 20010315:

```
CC=gcc CFLAGS="-O3 -fno-omit-frame-pointer" CXX=gcc CXXFLAGS="-O3 -fno-omit-
frame-pointer -felide-constructors -fno-exceptions -fno-rtti" ./configure --
prefix=/usr/local/mysql --with-extra-charsets=complex --enable-thread-safe-
client --enable-local-infile --disable-shared
```

Poniższe pliki binarne są budowane w systemach udostępnionych firmie MySQL AB przez innych użytkowników. Firma MySQL AB nie ma pełnej kontroli nad tymi systemami, więc możemy zaoferować tylko ograniczoną pomoc techniczną związaną z budowanymi w nich plikami binarnymi.

- ♦ SCO Unix 3.2v5.0.6 i386 z kompilatorem gcc 2.95.3:

```
CFLAGS="-O3 -mpentium" LDFLAGS=-static CXX=gcc CXXFLAGS="-O3 -mpentium -
felide-constructors" ./configure --prefix=/usr/local/mysql --with-extra-
charsets=complex --enable-thread-safe-client --enable-local-infile --with-
named-z-libs=no --enable-thread-safe-client --disable-shared
```

◆ SCO OpenUnix 8.0.0 i386 z kompilatorem CC 3.2:

```
CC=cc CFLAGS="-O" CXX=CC ./configure --prefix=/usr/local/mysql --with-extra-
charsets=complex --enable-thread-safe-client --enable-local-infile --with-
named-z-libs=no --enable-thread-safe-client --disable-shared
```

◆ Compaq Tru64 OSF/1 V5.1 732 alpha z kompilatorem cc i cxx (Compaq C V6.3-029i / DIGITAL C++ V6.1-027):

```
CC="cc -pthread" CFLAGS="-O4 -ansi_alias -ansi_args -fast -inline speed -
speculate all" CXX="cxx -pthread" CXXFLAGS="-O4 -ansi_alias -fast -inline
speed -speculate all -noexceptions -nortti" ./configure --prefix=/usr/local/
mysql --with-extra-charsets=complex --enable-thread-safe-client --enable-
local-infile --with-prefix=/usr/local/mysql --with-named-thread-libs="-
lpthread -lmach -lexc -lc" --disable-shared --with-mysqld-ldflags=-all-static
```

◆ SGI Irix 6.5 IP32 z kompilatorem gcc 3.0.1:

```
CC=gcc CFLAGS="-O3 -fno-omit-frame-pointer" CXXFLAGS="-O3 -fno-omit-frame-
pointer -felide-constructors -fno-exceptions -fno-rtti" ./configure --prefix=/
usr/local/mysql --with-extra-charsets=complex --enable-thread-safe-client --
enable-local-infile --disable-shared
```

◆ FreeBSD/sparc64 5.0 z kompilatorem gcc 3.2.1:

```
CFLAGS=-DHAVE_BROKEN_REALPATH ./configure --prefix=/usr/local/mysql --
localstatedir=/usr/local/mysql/data --libexecdir=/usr/local/mysql/bin --with-
extra-charsets=complex --enable-thread-safe-client --enable-local-infile --
disable-shared --with-innodb
```

Poniższe opcje były używane do kompilowania pakietów binarnych, które firma MySQL AB udostępniała w przeszłości. Pliki te nie są już aktualizowane, ale poniżej zamieszczono listę opcji w celach poglądowych:

◆ Linux 2.2.xx SPARC z kompilatorem egcs 1.1.2:

```
CC=gcc CFLAGS="-O3 -fno-omit-frame-pointer" CXX=gcc CXXFLAGS="-O3 -fno-omit-
frame-pointer -felide-constructors -fno-exceptions -fno-rtti" ./configure --
prefix=/usr/local/mysql --with-extra-charsets=complex --enable-thread-safe-
client --enable-local-infile --enable-asm --disable-shared
```

◆ Linux 2.2.x x686 z kompilatorem gcc 2.95.2:

```
CFLAGS="-O3 -mpentiumpro" CXX=gcc CXXFLAGS="-O3 -mpentiumpro -felide-
constructors -fno-exceptions -fno-rtti" ./configure --prefix=/usr/local/
mysql --enable-asm --with-mysqld-ldflags=-all-static --disable-shared
--with-extra-charsets=complex
```

◆ SunOS 4.1.4 2 sun4c z kompilatorem gcc 2.7.2.1:

```
CC=gcc CXX=gcc CXXFLAGS="-O3 -felide-constructors" ./configure --prefix=/
usr/local/mysql --disable-shared --with-extra-charsets=complex --enable-
asm
```

◆ SunOS 5.5.1 (i nowsze wersje) sun4u z kompilatorem egcs 1.0.3a lub 2.90.27 albo gcc 2.95.2 i nowszymi wersjami:

```
CC=gcc CFLAGS="-O3" CXX=gcc CXXFLAGS="-O3 -felide-constructors -fno-exceptions
-fno-rtti" ./configure --prefix=/usr/local/mysql --with-low-memory --with-
extra-charsets=complex --enable-asm
```

♦ SunOS 5.6 i86pc z kompilatorem gcc 2.8.1:

```
CC=gcc CXX=gcc CXXFLAGS=-O3 ./configure --prefix=/usr/local/mysql
--with-low-memory --with-extra-charsets=complex
```

♦ BSDI BSD/OS 3.1 i386 z kompilatorem gcc 2.7.2.1:

```
CC=gcc CXX=gcc CXXFLAGS=-O ./configure --prefix=/usr/local/mysql
--with-extra-charsets=complex
```

♦ BSDI BSD/OS 2.1 i386 z kompilatorem gcc 2.7.2:

```
CC=gcc CXX=gcc CXXFLAGS=-O3 ./configure --prefix=/usr/local/mysql
--with-extra-charsets=complex
```

♦ AIX 4.2 z kompilatorem gcc 2.7.2.2:

```
CC=gcc CXX=gcc CXXFLAGS=-O3 ./configure --prefix=/usr/local/mysql
--with-extra-charsets=complex
```

Każdy, kto zna bardziej optymalne opcje dla wymienionych wyżej konfiguracji, może przesłać je we wiadomości e-mail na listę dyskusyjną `internals`. Zobacz podpunkt 1.7.1.1. „Listy dyskusyjne MySQL”.

Dystrybucje RPM przed wersją 3.22 MySQL były dostarczane przez użytkowników. Od wersji 3.22 dystrybucje RPM są tworzone przez firmę MySQL AB.

Osoby, które chcą skompilować wersję MySQL przeznaczoną do debugowania, powinny dodać opcje `--with-debug` lub `--with-debug=full` do wierszy konfiguracyjnych i usunąć opcje `-fomit-frame-pointer`.

2.1.3. Jak zdobyć MySQL?

Informacje o bieżącej wersji i instrukcja jej pobierania znajdują się na stronie głównej MySQL (<http://www.mysql.com/>).

Nasza główna witryna lustrzana znajduje się pod adresem <http://mirrors.sunsite.dk/mysql/>.

Pełną, aktualną listę witryn lustrzanych MySQL można znaleźć pod adresem <http://dev.mysql.com/downloads/mirrors.html>. Znajdują się tam również informacje, jak założyć witrynę lustrzaną i jak informować o niedostępnych lub nieaktualnych witrynach lustrzanych.

2.1.4. Weryfikowanie integralności pakietu za pomocą sum kontrolnych MD5 albo narzędzia GnuPG

Po pobraniu, a przed zainstalowaniem wybranego pakietu MySQL należy upewnić się, że jest on nienaruszony i nikt go nie zmodyfikował. MySQL AB oferuje trzy sposoby sprawdzania integralności:

- ◆ sumy kontrolne MD5,
- ◆ podpisy kryptograficzne utworzone za pomocą narzędzia GnuPG (ang. *GNU Privacy Guard*),
- ◆ w przypadku pakietów RPM — wbudowany mechanizm weryfikowania integralności pakietu.

W poniższych podpunktach opisano, jak korzystać z tych metod.

Jeśli suma kontrolna lub podpis GPG są nieprawidłowe, najpierw należy jeszcze raz pobrać odpowiedni pakiet, być może z innej witryny lustrzanej. Jeśli kilkakrotnie nie uda się zweryfikować integralności pakietu, prosimy powiadomić nas o tym — wysłać wiadomość na adres webmaster@mysql.com lub build@mysql.com, z podaniem pełnej nazwy pakietu oraz nazwy witryny lustrzanej. Nie należy zgłaszać problemów z pobieraniem MySQL za pomocą systemu raportowania usterek.

2.1.4.1. Weryfikowanie sumy kontrolnej MD5

Po pobraniu pakietu MySQL należy upewnić się, że jego suma kontrolna MD5 odpowiada tej podanej na stronach pobierania MySQL. Każdy pakiet ma własną sumę kontrolną, którą można zweryfikować za pomocą poniższego polecenia (*nazwa_pakietu* to nazwa pobranego pakietu):

```
shell> md5sum nazwa_pakietu
```

Przykład:

```
shell> md5sum mysql-standard-4.0.17-pc-linux-i686.tar.gz  
60f5fe969d61c8f82e4f7f62657e1f06 mysql-standard-4.0.17-pc-linux-i686.tar.gz
```

Należy sprawdzić, czy obliczona suma kontrolna (łańcuch cyfr szesnastkowych) odpowiada tej, którą podano na stronie pobierania tuż pod nazwą odpowiedniego pakietu.

Nie wszystkie systemy operacyjne zawierają polecenie `md5sum`. W niektórych ma ono nazwę `md5`, a w innych jest niedostępne. W systemie Linux stanowi ono część pakietu GNU Text Utilities dostępnego w wersjach dla wielu odmian tego systemu. Kod źródłowy pakietu można pobrać pod adresem <http://www.gnu.org/software/textutils/>. Jeśli zainstalowany jest pakiet OpenSSL, można również użyć polecenia `openssl md5 nazwa_pakietu`. Implementacja polecenia `md5` dla systemów DOS i Windows jest dostępna pod adresem <http://fourmilab.ch/md5/>.

2.1.4.2. Sprawdzanie podpisu cyfrowego za pomocą GnuPG

Inną metodą weryfikowania integralności i autentyczności pakietu jest użycie podpisów kryptograficznych. Jest to metoda bardziej niezawodna od sum kontrolnych MD5, ale wymaga więcej pracy.

Od wersji 4.0.10 MySQL (luty 2003) firma MySQL AB zaczęła podpisywać przeznaczone do pobrania pakiety za pomocą narzędzia GnuPG (ang. *GNU Privacy Guard*). GnuPG to zamiennik typu Open Source dobrze znanego programu Pretty Good Privacy

(PGP) Phila Zimmermana. Więcej informacji o programie GnuPG oraz o jego pobieraniu i instalowaniu w systemie można znaleźć pod adresem <http://www.gnupg.org/>. Większość dystrybucji systemu Linux domyślnie instaluje program GnuPG. Więcej informacji o OpenPGP znajduje się pod adresem <http://www.openpgp.org/>.

Aby zweryfikować podpis konkretnego pakietu, najpierw należy pobrać kopię publicznego klucza GPG firmy MySQL AB. Klucz można pobrać z witryny <http://www.keyserver.net/>; jego nazwa to `build@mysql.com`. Można również wyciąć i wkleić klucz bezpośrednio z poniższego tekstu:

```
Key ID:
pub 1024D/5072E1F5 2003-02-03
    MySQL Package signing key (www.mysql.com) <build@mysql.com>
Fingerprint: A4A9 4068 76FC BD3C 4567 70C8 8C71 8D3B 5072 E1F5
```

Public Key (ASCII-armored):

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.0.6 (GNU/Linux)
Comment: For info see http://www.gnupg.org
```

```
mQGIBd4+owwRBAC14GI fuFcyEDSIePvEW3SAFUdJBtoQHh/nJKZyQT7h9bP1UWC3
RODjQReyCITRrdwyrKUGku2FmeVGwn2u2WmDMNABLnpPrwPkBdCk96+OmSLN9brZ
fw2v0UgCmYv2hw0hyDHuvY1QA/BThQoADgJ8AW6/0Lo7V1W9/8VuHP0gQwCgvzV3
Bq0xRznNCRcRxAuAuVztHRcEAJooQK1+iSiunZMYD1WufeXfshc57S/+yeJkegNW
hxwR9pRWArNYJdDRT+rF2RUE3vpquKNQU/hnEIUHJRQqYHo8gTvxvXNQC7fJYLV
K2HtkrPbP72vwsEKMYhhr0eKcBtLGf1s9krjJ6sBgACyP/Vb7hiPwxh6rDZ7ITe
kYpXBACmWpP8NJTkamEnPCia2ZoOHODANwpUkP43I7jsDmgtobZX9qnrAXw+uNDI
QJEXM6fSbi0LLtZciN1YsafwAPEOMDKpMqAK6IyisNtPvald81H0bPANwqcyefep
rv0sxxqUEMcM3o7wwgfN83P0kDasDbs3pjwPhxvhz6//62zQJ7Q77X1TUUwUGFj
a2FnZSBzaWduaW5nIGt1eSAd3d3Lm15c3FsLmNvbSkpPGJ1awxkQG15c3FsLmNv
bT6IXQQTQEQIAHQUCj6jDAUJCwYBgAULBwoDBAMVAwIDFgIBAheAAAoJEIxxjTtQ
cuH1cY4AnilUwTXn8MatQ0iG0a/bPxrVk/gCAJ4oiNSNZRYTnb1ChwFaazt7PF3q
zIhMBBMRagAMBQI+PqPRBYMJZgC7AAoJEE1Q4SqycpHyJOEAnImxHijft00bKXvu
cSo/pECUmppiaJ41M9MRVj5VcdH/KN/KjRtW6tHFPYhMBBMRagAMBQI+QoIDBYMJ
YiKJAAoJELb1zU3GuqiQ/1pEAoIhpp6BozKI8p6eaabzF5M1JH58pAKCu/ROofK8J
Eg2aLos+5zEYrB/LsrkCDQQ+PqMdEAgA7+GJfxbMdY4ws1PnjH9rF4N2qfwsEN/1
xaZoJyc3a6M02wCnH16ahT2/tBK2w1QI4YFteR47gCvtgb601JHf0o2HfLmRDRi
Rjd1DTCheqYX7CHhcgchj/dNR1W2Z015QFecmV9U0Vhp3aFfWC4Ujfs3LU+hkAwE
7zaD5cH9J7yv/6xuZVw411x0h4UqsTcwMu0iM1BzELqX1DY7LwoPEb/09Rkbf4fm
Le11EzIaCa4PqARXQZc4dhSinMt6K3X4BrRsKTfozBu74F47D8I1bf5VYHbuE5p
/loIDznkg/p8kW+3FxlwrycciqFTcNz215yyX39LXFnlLzKUb/F5GwADBQf+Lwqq
a8CGrRfs0AJxim63CHftY5mUc5rUSnTs1GYEIOCR1BeQauyPZbPDSDD9MZ1ZaSaF
anFwvFG6L1x9xkU7tzq+vKLowkm4u5xf3vn55VjnSd1aQ9eQnUcXiL4cNBGoTbOW
I39Ecyzgs1zBdC++MPjCQtCA7p6JUVsP6oAB3FQWg54tuUo0Ec8bsM8b3Ev42Lmu
QT5NdKHGwHsXTpt10k1k4bQk40ajHsijYBMahpT27jWjJ1MiJc+IwJ0mgkhKht92
6s/ymfdf5HkdQ1cyvsz5tryVI3F78XeSYfQvuuwqp2H139pXGEGk0n6Kdu0etdZ
whe70YGNPw1yJwJT1IhMBBGRagAMBQI+PqMdBQkJZGAAAoJEIxxjTtQcuH17p4A
n3r1QpVC9yhnlw2cSAjq+kr72GX0eAJ4295k16NxyEuFApmr1+0uUq/S1sQ==
=YJKx
```

```
-----END PGP PUBLIC KEY BLOCK-----
```

Aby zaimportować klucz do osobistego zbioru kluczy publicznych GPG, należy użyć polecenia `gpg --import`.

Jeśli na przykład klucz został zapisany w pliku o nazwie *mysql_pubkey.asc*, polecenie importu powinno wyglądać tak:

```
shell> gpg --import mysql_pubkey.asc
```

Więcej informacji o pracy z kluczami publicznymi można znaleźć w dokumentacji GPG.

Po pobraniu i zaimportowaniu klucza publicznego należy pobrać żądany pakiet MySQL oraz odpowiedni podpis, który również jest dostępny na stronie pobierania. Plik podpisu ma tę samą nazwę co plik dystrybucji, z rozszerzeniem *.asc*. Na przykład:

Plik dystrybucji	<i>mysql-standard-4.0.17-pc-linux-i686.tar.gz</i>
Plik podpisu	<i>mysql-standard-4.0.17-pc-linux-i686.tar.gz.asc</i>

Należy sprawdzić, czy pliki są zapisane w tym samym katalogu, a następnie wydać poniższe polecenie, aby zweryfikować podpis pliku dystrybucji:

```
shell> gpg --verify nazwa_pakietu.asc
```

Przykład:

```
shell> gpg --verify mysql-standard-4.0.17-pc-linux-i686.tar.gz.asc
gpg: Warning: using insecure memory!
gpg: Signature made Mon 03 Feb 2003 08:50:39 PM MET
using DSA key ID 5072E1F5
gpg: Good signature from
"MySQL Package signing key (www.mysql.com) <build@mysql.com>"
```

Komunikat Good signature oznacza, że wszystko jest w porządku. Ostrzeżenie insecure memory można zignorować.

2.1.4.3. Sprawdzanie podpisów za pomocą RPM

W przypadku pakietów RPM nie ma dodatkowego podpisu. Zawierają one wbudowany podpis GPG i sumę kontrolną MD5. Pakiet można zweryfikować za pomocą poniższego polecenia:

```
shell> rpm --checksig nazwa_pakietu.rpm
```

Przykład:

```
shell> rpm --checksig MySQL-server-4.0.10-0.i386.rpm
MySQL-server-4.0.10-0.i386.rpm: md5 gpg OK
```

Uwaga: ci, którzy używają RPM 4.1 i otrzymają komunikat (GPG) NOT OK (MISSING KEYS: GPG#5072e1f5), mimo że zaimportowali klucz publiczny MySQL do osobistego zbioru kluczy GPG, muszą najpierw zaimportować klucz do zbioru kluczy RPM. Od wersji 4.1 RPM nie używa osobistego zbioru kluczy GPG (ani samego programu GPG). Korzysta z oddzielnego zbioru kluczy, ponieważ jest aplikacją ogólnosystemową, a osobisty zbiór kluczy GPG jest plikiem specyficznym dla użytkownika. Aby zaimportować klucz publiczny MySQL do zbioru kluczy RPM, należy najpierw pobrać go w sposób opisany w poprzednim podpunkcie, a następnie zaimportować za pomocą polecenia `rpm --import`. Jeśli na przykład klucz publiczny jest zapisany w pliku o nazwie *mysql_pubkey.asc*, należy zaimportować go za pomocą polecenia:

```
shell> rpm --import mysql_pubkey.asc
```

2.1.5. Układ katalogów instalacyjnych

W tym rozdziale opisano układ katalogów tworzonych podczas instalowania binarnej lub źródłowej dystrybucji dostarczonej przez MySQL AB. Dystrybucje innych firm mogą mieć inny układ.

W Windows domyślny katalog instalacyjny to *C:\mysql*. Zawiera on następujące podkatalogi:

Katalog	Zawartość katalogu
<i>bin</i>	Programy klienta oraz serwer <i>mysqld</i>
<i>data</i>	Pliki dziennika, bazy danych
<i>Docs</i>	Dokumentacja
<i>examples</i>	Przykładowe programy i skrypty
<i>include</i>	Pliki nagłówkowe
<i>lib</i>	Biblioteki
<i>scripts</i>	Skrypty narzędziowe
<i>share</i>	Pliki z komunikatami o błędach

Instalacja z linuksowych dystrybucji RPM powoduje umieszczenie plików w następujących katalogach systemowych:

Katalog	Zawartość katalogu
<i>/usr/bin</i>	Programy klienta i skrypty
<i>/usr/sbin</i>	Serwer <i>mysqld</i>
<i>/var/lib/mysql</i>	Pliki dziennika, bazy danych
<i>/share/doc/packages</i>	Dokumentacja
<i>/usr/include/mysql</i>	Pliki nagłówkowe
<i>/usr/lib/mysql</i>	Biblioteki
<i>/usr/share/mysql</i>	Pliki z komunikatami o błędach
<i>/usr/share/sql-bench</i>	Testy wydajnościowe

W systemie Unix dystrybucja binarna w postaci pliku *tar* jest rozpakowywana w katalogu wybranym przez użytkownika (zwykle */usr/local/mysql*) i tworzy w nim następujące podkatalogi:

Katalog	Zawartość katalogu
<i>bin</i>	Programy klienta oraz serwer <i>mysqld</i>
<i>data</i>	Pliki dziennika, bazy danych
<i>docs</i>	Dokumentacja, dziennik zmian
<i>include</i>	Pliki nagłówkowe
<i>lib</i>	Biblioteki
<i>scripts</i>	<i>mysql_install_db</i>
<i>share/mysql</i>	Pliki z komunikatami o błędach
<i>sql-bench</i>	Testy wydajnościowe

Dystrybucja źródłowa jest instalowana po jej skonfigurowaniu i skompilowaniu. Pliki są domyślnie instalowane w katalogu */usr/local*, w poniższych podkatalogach:

Katalog	Zawartość katalogu
<i>bin</i>	Programy klienta i skrypty
<i>include/mysql</i>	Pliki nagłówkowe
<i>info</i>	Dokumentacja w formacie info
<i>lib/mysql</i>	Biblioteki
<i>libexec</i>	Serwer <i>mysqld</i>
<i>share/mysql</i>	Pliki z komunikatami o błędach
<i>sql-bench</i>	Testy wydajnościowe i test <i>crash-me</i>
<i>var</i>	Bazy danych i pliki dziennika

W katalogu instalacyjnym układ dystrybucji źródłowej różni się od dystrybucji binarnej w następujących kwestiach:

- ◆ serwer *mysqld* jest instalowany w katalogu *libexec*, a nie *bin*,
- ◆ katalog danych to *var*, a nie *data*,
- ◆ skrypt *mysql_install_db* jest instalowany w katalogu *bin*, a nie *scripts*,
- ◆ katalog plików nagłówkowych oraz bibliotek to *include/mysql* oraz *lib/mysql*, a nie *include* i *lib*.

Aby utworzyć własną instalację binarną ze skompilowanej dystrybucji źródłowej, można wykonać skrypt *scripts/make_binary_distribution*, który znajduje się w najwyższym katalogu dystrybucji źródłowej.