

Index

SYMBOLS AND NUMERICS

+ (addition operator), 272
 <> (angle brackets), 15
 * (asterisk), 233–234, 354–355, 669
 @ (at symbol), 241, 555
 \. (backslash, period), 445
 > (bitwise shift right operator), 297
 ^ (caret), 286
 %> (closing scriptlet tag), 678, 734
 := (colon, equals sign), 241
 , (comma)
 basic joins, 371, 374
 tables, creating, 20
 {} (curly brackets), 15, 726
 / (division operator), 272
 \$ (dollar sign), 620
 && (double ampersands), 292
 -- (double dashes), 502
 || (double vertical pipes), 292
 = (equals sign)
 C#, 726
 as comparison operator, 278, 280
 server variables, reading, 502
 ! (exclamation point), 292
 != (exclamation point, equals sign), 264
 > (greater than operator), 238
 > (greater than sign), 278
 - (minus sign), 238
 % (modulo operator), 272
 * (multiplication operator), 272
 <% (opening scriptlet tag), 678, 734
 () (parentheses), 17, 310
 % (percent character), 180, 284, 527
 .* (period, asterisk), 286
 ; (semicolon), 19, 620, 621
 << (shift left operator, bitwise), 297, 299

[] (square brackets), 15, 17
 - (subtraction operator), 272
 ... (three periods), 15
 ::= (two colons, equal sign), 15
 - (unary operator), 272
 _ (underscore), 284
 | (vertical bar), 15, 18, 601–602
 64-bit operation, bitwise operators and, 299

A

access control

connections, authenticating, 527–529
 described, 25–26, 517–518
 grant tables
 columns_priv, 523–524, 532
 db, 520–521
 host, 521–522
 privileges listed, 524–526
 tables_priv, 522–523, 531–532
 user, 518–520
 verification, 529–533

access to multiple tables

DELETE statement joins, creating, 394–395
 DELETE subqueries, 411–413
 indexes, benefits of using, 563
 joins, when to use, 369–370
 SELECT statement joins, creating
 basic joins, 374–377
 described, 370–371
 inner joins and cross joins, 377–380
 left outer joins, 386–388
 natural joins, 391–392
 outer joins, 385–391
 right outer joins, 389–390
 sample, 381–385
 sample tables, 372–373
 straight joins, 380–381
 syntax, 371

access to multiple tables (continued)

access to multiple tables (continued)

- SELECT subqueries
 - ALL operator, 400
 - ANY and SOME operators, 399–400
 - comparison operators, 396–399
 - EXISTS and NOT EXISTS operators, 402–403
 - grouped data, 403–407
 - IN and NOT IN operators, 401–402
 - sample, 407–410
- unions joining SELECT statements, 414–417
- UPDATE statement joins, creating, 393–394
- UPDATE subqueries, 410–411

accounts

- adding and assigning privileges (GRANT statement)
 - granting or revoking to others (WITH clause), 543–544
 - privileges (GRANT clause), 534–536
 - sample, 537–541, 546–551
 - secure connection, stipulating (REQUIRE clause), 541–543
 - syntax, 533–534
 - tables or databases, specifying (ON clause), 536–537
 - to users (TO clause), 537
- current status, displaying, 94
- dropping and revoking privileges
 - described, 553
 - DROP USER statement, 555
 - REVOKE statement, 553–555
 - sample, 555–558
- identifying, PHP application, 37
- passwords, setting
 - administrative (mysql utility), 100
 - changing, 552–553
 - configuration file, 103–106
 - grant tables, reloading, 100–101, 551–552
 - root account, 101–102
 - SET PASSWORD statement, 551
- server connection, specifying, 487
- tar files on Linux, 51–52
- viewing privileges (SHOW GRANTS statement), 544–545

ACID test (atomic, consistent, isolated, and durable), 456

actions, grant table, 80–81

adding

- date/time functions (ADDDATE() and DATE_ADD()), 339–340
- indexes to existing tables, 175–178

adding data to table

- ASPNET/C# application, 748–773
- described, 189–190
- FOUND_ROWS(), 362–363
- INSERT statement
 - copying data into existing table, 435–436
 - foreign key tables, 199–203
 - lookup tables, DVDRentals database, 195–198
 - one row at a time (<set option> method), 203–205
 - people tables, DVDRentals database, 198–199
 - REPLACE statement versus, 205
 - rows, one or more (<values option>), 191–195
 - syntax, 190–191
- Java/J2EE, 692–716
- LAST_INSERT_ID(), 363

- mysql utility
 - LOAD DATA statement, 438–444
 - mysql command, 446–448
 - source command, 444–446
- mysqlimport utility, 449–452
- performance, optimizing, 577–578
- PHP, 639–640, 656–660

addition operator (+), 272

address, converting to numeric value (INET_ATON() and INET_NTOA()), 362

administration

- log files
 - binary logging, setting up, 507–515
 - error log, 506
 - Linux query and binary logging, 508–512
 - query logging, setting up, 507, 508–515
 - Windows query and binary logging, 512–515
- mysql utility
 - account passwords, assigning, 100
 - batch mode from command prompt, 97–100
 - current database, changing (use \u command), 94
 - data folder, 78–79
 - discontinuing statements (clear \c command), 91–92
 - help command (\? or ?), 92
 - in interactive mode, 90–91
 - interactive use example, 94–96
 - launching, 96–97
 - prompt display (\R command), 92–93
 - query, executing in specified file (source \. command), 93–94
 - quitting (exit or quit \q), 92
 - session information (status \s command), 94
 - text file, saving input and output to (tee \T command), 94
- mysqladmin utility, 89, 488–494
- options, 486–487
- privilege columns, 519
- server configuration
 - command line, specifying at, 502–503
 - option file, specifying in, 503
 - runtime, specifying at, 503–506
- system variables
 - configuring, 494–495
 - settings, retrieving, 495–502
- utilities, 485–486

aggregate functions

- averaging values (AVG()), 350–351
- bit, 354–356
- COUNT(), 353–354
- described, 349–350
- example, 356–357
- minimum or maximum column value (MIN() and MAX()), 352–353
- sum of expression (SUM()), 351–352

aliases

- C# string class, 727
- tables, joining, 375–376

ALL operator, 400

ampersands, double (&&), 292

- AND operator**
 - aggregate function, 354–355
 - bitwise, 297, 301
 - described, 292, 294
 - angle brackets (<>), 15**
 - anomalies**
 - dirty reads, 472–473, 476
 - nonrepeatable reads, 473–474, 476
 - phantom reads, 474–475, 476
 - anonymous users**
 - db table, 82
 - Linux user table, 81
 - removing, 556
 - ANSI SQL standard, 9, 11–12**
 - ANY and SOME operators, 399–400**
 - application**
 - ASP.NET/C#
 - connecting to MySQL database, 727–728
 - converting values, 731–732
 - creating basic, 735–748
 - declaring and initializing variables, 725–727
 - deleting data, 773–777
 - described, 723–725
 - exceptions, managing, 734–735
 - files, including, 733–734
 - HTML forms, 732–733
 - inserting and updating data, 748–773
 - redirecting browsers, 733
 - retrieving data from MySQL database, 728–731
 - web page, setting up, 725
 - data, converting for use in, 320–322
 - Java/J2EE
 - described, 679–682
 - index.jsp file, 682–692
 - PHP
 - basic, creating, 628–639
 - call-level interface, 30
 - connecting to MySQL database, 33–34, 619–621
 - data, 656–660
 - data, retrieving from MySQL database, 621–625
 - deleting data, 660–664
 - described, 618–619
 - HTML forms, 626–627
 - include files, 627–628
 - inserting data, 639–640, 656–660
 - updating data, 640–656
 - Web-based applications, popularity of, 33
 - application server, 32–33**
 - argument, 37**
 - arithmetic functions**
 - averaging values (AVG ()), 350–351
 - date/time (ADDDATE () and DATE_ADD ()), 339–340
 - operators, 272–277, 282
 - ASCII data type**
 - column definition, 149
 - string function, 323
 - ASP.NET application**
 - connecting to MySQL database, 727–728
 - converting values, 731–732
 - creating basic, 735–748
 - declaring and initializing variables, 725–727
 - deleting data, 773–777
 - described, 723–725
 - exceptions, managing, 734–735
 - files, including, 733–734
 - HTML forms, 732–733
 - inserting and updating data
 - described, 748–750
 - functions, adding to sample application, 750–773
 - redirecting browsers, 733
 - retrieving data from MySQL database
 - described, 728–729
 - result set, processing, 729–730
 - string data, manipulating, 730–731
 - web page, setting up, 725
 - asterisk (*), 233–234, 354–355, 669**
 - at symbol (@), 241, 555**
 - atomic, consistent, isolated, and durable (ACID) test, 456**
 - atomic transaction property, 456**
 - attribute, 122**
 - authenticating connections, 527–529**
 - autocommit mode, 468–472**
 - auto-increment columns, 18, 155–156**
 - automatically committed statements, listed, 460**
 - averaging values (AVG ()), 350–351**
- ## B
- backslash, period (\.), 445**
 - backups**
 - all databases on system, 595–596
 - entire single database, 588–594
 - to file, 90, 588, 596–599
 - individual tables, 594
 - multiple databases, 594–599
 - specific databases, 595
 - basic joins, 374–377**
 - batch mode, administrative functions, 97–100**
 - BDB table type, 159**
 - binary data type, 149, 150**
 - binary files, 45**
 - binary logging**
 - displaying, 89
 - enabling and disabling, 603
 - setting up, 507–515
 - BINARY operator, 303–304, 305–306**
 - bit functions, 354–356**
 - bitwise operators, 296–302**
 - bitwise shift right operator (>), 297**
 - blank values**
 - column, ability to handle, 152–153
 - columns, accepting, 152–153
 - as default, 153–154
 - natural joins, 392
 - outer joins, 385, 388
 - restricting use of, 580
 - blocking former users**
 - described, 553
 - DROP USER statement, 555
 - REVOKE statement, 553–555
 - sample, 555–558
 - users to other users (WITH clause), 543–544

brackets, curly ({}), 15, 726

browser

- mechanics, 618
- redirecting, 627, 678, 733

bug reports, generating on Unix systems, 89

built-in functions, 310

C

C API, 801

C++ API, 801

C# application

- connecting to MySQL database, 727–728
- converting values, 731–732
- creating basic, 735–748
- declaring and initializing variables, 725–727
- deleting data, 773–777
- described, 723–725
- exceptions, managing, 734–735
- files, including, 733–734
- HTML forms, 732–733
- inserting and updating data
 - described, 748–750
 - functions, adding to sample application, 750–773
- redirecting browsers, 733
- retrieving data from MySQL database
 - described, 728–729
 - result set, processing, 729–730
 - string data, manipulating, 730–731
- web page, setting up, 725

C programming languages

- APIs, 801
- call-level interface, 30
- embedded SQL, 29
- SQL versus, 10

C+ programming languages

- APIs, 801
- call-level interface, 30
- embedded SQL, 29
- SQL versus, 10

caching query results, 246, 580–583

call-level interface execution, 30

capital letters

- GRANT statement scope columns, 534
- name conversion issues, 141
- string data, changing (LCASE(), LOWER(), UCASE(), and UPPER()), 326–327

caret (^), 286

Cartesian product, 374–375

case

- GRANT statement scope columns, 534
- name conversion issues, 141
- string data, changing (LCASE(), LOWER(), UCASE(), and UPPER()), 326–327

CASE() flow control function, 316–317

categories of information

- identifying
 - described, 122–124
 - DVDRentals data model, 128–130
- normalizing, 124
- reasons to use, 116

CD, copying Linux tar file from, 52–53

CEIL() numeric data functions, 333

CEILING() numeric data functions, 333

certificate authorities, 541, 542

CHANGE MASTER statement, 613–614

changing current database, changing (use \u command), 94
changing data

- ASPNET/C# application, 748–750, 750–773
 - INSERT statement
 - copying data into existing table, 435–436
 - foreign key tables, 199–203
 - lookup tables, DVDRentals database, 195–198
 - lookup tables, sample, 195–198
 - one row at a time (<set option> method), 203–205
 - people tables, DVDRentals database, 198–199
 - people tables, sample, 198–199
 - REPLACE statement versus, 205
 - rows, one or more (<values option>), 191–195
 - rows, one or more (<values option> method), 191–195
 - syntax, 190–191
 - internal table statistics, 571–573
 - Java/J2EE, 692–693, 693–716
 - joins, creating multiple table data access, 393–394
 - maximum permitted per hour, 544
 - nonrepeatable reads, 473–474, 476
 - performance, optimizing, 578–579
 - phantom reads, 474–475, 476
 - PHP
 - described, 623–625
 - edit.php file, creating, 640–654
 - insert.php file, creating, 655
 - update.php file, creating, 655–656
 - REPLACE statement
 - copying data into existing table, 436–438
 - INSERT statement versus, 205
 - <set option> alternative, 207–208, 207–210, 208–210
 - syntax, 206
 - <values option> alternative, 206–208
 - restoration from binary log files
 - direct, 601–602
 - from text file, 602
 - subqueries, 410–411
 - syntax, 210
 - UPDATE statement
 - joined tables, 216–218
 - single table, 211–216
- character set**
- assigning to value (CAST functions), 320–322
 - data type, 149
 - databases
 - changing, 142
 - creating, 140
 - directory, specifying, 486
 - string data, identifying (CHARSET()), 324
- characters**
- numeric data, maximum displayed, 145
 - string data
 - from ends, returning (LEFT() and RIGHT()), 327
 - extracting, PHP (substr()), 624–625

child-parent relationships

- foreign keys, 156
- hierarchical model, 3–4
- network model, 5–6

classes, Java/J2EE, 669**clause, 24****client options, listed, 89–90****client utility, mysql**

- account passwords, assigning, 100
- batch mode, 97–100, 600
- current database, changing (use `\u` command), 94
- described, 22–23
- direct invocation, 27
- discontinuing statements (clear `\c` command), 91–92
- help command (`\?` or `?`), 92
- importing data into table
 - `LOAD DATA` statement, 438–444
 - `mysql` command, 446–448
 - `source` command, 444–446
- interactive mode
 - described, 90–91
 - example, 94–96
 - reloading database, 600–601
- launching, 95, 96–97
- programs, 97–100
- prompt display (`\R` command), 92–93
- query, executing in specified file (source `\.` command), 93–94
- quitting (exit or quit `\q`), 92
- `SELECT` statement, 25
- session information (status `\s` command), 94
- text file, saving input and output to (tee `\T` command), 94

closing scriptlet tag (%>), 678, 734**code, predefined**

- built-in versus user-defined, 310
- comparing data
 - first integer listed as argument `INTERVAL()` and `STRCMP()`, 312–314
 - first nonnull values, returning or status as all null (`COALESCE()` and `ISNULL()`), 311–312
 - `GREATEST()` and `LEAST()`, 310–311
- connection (`CONNECTION_ID()`), 361–362
- control flow
 - `CASE()`, 316–317
 - `IF()`, 314–315
 - `IFNULL()` and `NULLIF()`, 315
 - using in `SELECT` statement, 317–319
- converting values to specific type (`CAST()`), 320–322
- database (`DATABASE()`), 361–362
- date/time
 - add (`ADDDATE()` and `DATE_ADD()`), 339–340
 - current date and time, retrieving (`CURDATE()`, `CURRENT_DATE()`, `CURTIME()`, `CURRENT_TIME()`, `CURRENT_TIMESTAMP()`, and `NOW()`), 341–342
 - differences between (`DATEDIFF()` and `TIMEDIFF()`), 344
 - example, 346–349
 - extracting specific information (`DATE()`, `MONTH()`, `MONTHNAME()`, and `YEAR()`), 342–343

- pull day-related values (`DAY()`, `DAYOFMONTH()`, `DAYNAME()`, `DAYOFWEEK()`, and `DAYOFYEAR()`), 344–345
- specific part, extracting (`EXTRACT()`), 341
- subtracting (`SUBDATE()` or `DATE_SUB()`), 340–341
- time parts, extracting (`SECOND()`, `MINUTE()`, `hour()`, and `TIME()`), 345–346
- described, 37, 270, 309
- MySQL version (`VERSION()`), 361–362
- network address, converting to numeric value (`INET_ATON()` and `INET_NTOA()`), 362
- numeric data
 - `CEIL()`, `CEILING()`, and `FLOOR()`, 333
 - cotangent of number (`COT()`), 333
 - `PI()`, 334
 - `POW()` and `POWER()`, 334
 - remainder from dividing two numbers (`MOD()`), 334
 - `ROUND()` and `TRUNCATE()`, 334–335
 - sample, 336–339
 - square root `SQRT()`, 335
- query and insert
 - `FOUND_ROWS()`, 362–363
 - `LAST_INSERT_ID()`, 363
- string data
 - case, changing (`LCASE()`, `LOWER()`, `UCASE()`, and `UPPER()`), 326–327
 - character set, identifying (`CHARSET()`), 324
 - characters from ends, returning (`LEFT()` and `RIGHT()`), 327
 - collation, identifying (`COLLATION()`), 303, 324–325
 - concatenating (`CONCAT()` and `CONCAT_WS()`), 325
 - length (`LENGTH()`), 324
 - multibyte value of first character, identifying (`ORD()`), 323
 - number of characters (`CHAR_LENGTH()` and `CHARACTER_LENGTH()`), 323
 - numeric value of first character, identifying (`ASCII()`), 323
 - repeating specified times (`REPEAT()`), 327–328
 - reversing order (`REVERSE()`), 328
 - sample, 329–332
- substrings
 - finding after specified point (`LOCATE()`), 326
 - locating (`INSTR()`), 325–326
 - returning (`SUBSTRING()`), 328–329
- summarizing data
 - averaging values (`AVG()`), 350–351
 - bit, 354–356
 - `COUNT()`, 353–354
 - described, 349–350
 - example, 356–357
 - minimum or maximum column value (`MIN()` and `MAX()`), 352–353
 - sum of expression (`SUM()`), 351–352
- system encryption
 - `ENCODE()` and `DECODE()`, 358–359
 - `PASSWORD()`, `MD5()`, `SHA()`, and `SHA1()`, 359–360
- system-related operations, sample, 363–366
- user-name and hostname, current session (`CURRENT_USER()`, `SESSION_USER()`, `SYSTEM_USER()`, and `USER()`), 361
- value, assigning character set to (`CAST()`), 320–322

collation

collation

- database
 - changing, 142
 - creating, 140
- string data
 - identifying (COLLATION()), 324–325
 - sorting, 303

colon, equals sign (:=), 241

columns

- all, returning, 233–234
- auto-increment, 155–156
- compared, optimizing, 579
- data model, 120
- default values, 18, 153–154
- defining
 - data types, 18
 - date/time data types, 151–152
 - name, 21, 270
 - numeric data types, 145–148
 - string data types, 148–151
 - syntax, 144–145
- including, 25
- linked, inner or cross joins, 379
- normalizing data
 - first normal form, 111
 - second normal form, 114
- nullability, 152–153
- privileges, limiting to, 536
- relational database design, 110
- search conditions, 564

columns_priv grant table, 523–524, 532

combined functions

- averaging values (AVG()), 350–351
- bit, 354–356
- COUNT(), 353–354
- described, 349–350
- example, 356–357
- minimum or maximum column value (MIN() and MAX()), 352–353
- sum of expression (SUM()), 351–352

comma (,)

- basic joins, 371, 374
- tables, creating, 20

command line

- data, importing, 446–448
- options, specifying, 83–85
- prompt display (\R command), 92–93
- returning from mysql, 92
- server configuration, specifying, 502–503

commands, listing available, 95–96

comments

- syntax, 20
- Windows configuration file, 88

comparing data

- functions
 - first integer listed as argument (INTERVAL() and STRCMP()), 312–314
 - first nonnull values, returning or status as all null (COALESCE() and ISNULL()), 311–312
 - GREATEST() and LEAST(), 310–311

strings

- in C# (equals()), 731
- in Java (equals()), 675
- in PHP (strcmp()), 624

comparison operators

- listed, 277–291
- SELECT subqueries, 396–399

composite primary key, 113, 124

compressing data, 486

computer system

- cache performance, optimizing, 580–583
- character sets and collations, viewing available, 140
- databases, backing up all, 595–596
- encryption functions (ENCODE() and DECODE()), 358–359
- information
 - DESCRIBE statements, 183–187
 - listing databases and tables, 90
 - SHOW SLAVE STATUS statement, 614
 - SHOW statements, 180–183
 - user accounts (SHOW GRANTS statement), 545
- operations, sample functions, 363–366
- settings, retrieving
 - dynamic system (SELECT statement), 497–498
 - sample, 500–502
 - server (SHOW VARIABLES statement), 495–496
 - server status (SHOW STATUS statement), 498–500

computer system administration

log files

- binary logging, setting up, 507–515
- error log, 506
- Linux query and binary logging, 508–512
- query logging, setting up, 507, 508–515
- Windows query and binary logging, 512–515

mysql utility

- account passwords, assigning, 100
- batch mode from command prompt, 97–100
- current database, changing (use \u command), 94
- data folder, 78–79
- discontinuing statements (clear \c command), 91–92
- help command (\? or ?), 92
- in interactive mode, 90–91
- interactive use example, 94–96
- launching, 96–97
- prompt display (\R command), 92–93
- query, executing in specified file (source \. command), 93–94
- quitting (exit or quit \q), 92
- session information (status \s command), 94
- text file, saving input and output to (tee \T command), 94

mysqladmin utility, 89, 488–494

options, 486–487

privilege columns, 519

server configuration

- command line, specifying at, 502–503
- option file, specifying in, 503
- runtime, specifying at, 503–506

system variables

- configuring, 494–495
- settings, retrieving, 495–502

utilities, 485–486

concatenating string data, 325, 731**configuration file**

- options, specifying, 85–88
- server, 503

connections

- ASPNET/C# application, 727–728
- authenticating, 527–529
- database version, 361–362
- establishing, 37
- failures, 38
- ID, displaying, 94, 361–362
- Java/J2EE, 671–674
- maximum permitted per hour, 544
- PHP application, 619–621
- retrying, 487
- secure, requiring, 519
- slave server
 - altering (CHANGE MASTER statement), 613–614
 - listing (SHOW SLAVE HOSTS statement), 612
 - version, 361–362

connectivity, 8**consistent transaction property, 456****constructor, 671****control access**

- connections, authenticating, 527–529
- described, 25–26, 517–518
- grant tables
 - columns_priv, 523–524, 532
 - db, 520–521
 - host, 521–522
 - privileges listed, 524–526
 - tables_priv, 522–523, 531–532
 - user, 518–520
- verification, 529–533

control flow functions

- CASE(), 316–317
- IF(), 314–315
- IFNULL() and NULLIF(), 315
- using in SELECT statement, 317–319

converting values

- ASPNET/C# application, 731–732
- date/time, PHP, 625
- Java/J2EE, 675–677
- to specific type or assigning character set to value (CAST()), 320–322

copying data into table

- described, 431
- existing table
 - INSERT statement, 435–436
 - REPLACE statement, 436–438
- new table, 432–435

copying files

- tar on Linux, 52–53
- text, data to/from, 90

Core SQL, 12**cotangent of number (COT()) function, 333****COUNT() function, 353–354****crash recovery**

- backups
 - all databases on system, 595–596
 - entire single database, 588–594

- to file, 596–599
- individual tables, 594
- specific databases, 595

described, 587

replication

- described, 606–608
- implementing, 608–610
- master server, 610–612
- slave server, 612–615
- restoration
 - binary logging, enabling and disabling, 603
 - reloading database, 600–601
 - sample, 603–606
 - steps, 599
 - updating from binary log files, 601–602

cross joins, 377–380**curly brackets ({}), 15, 726****current database, changing (use \u command), 94****current date and time, retrieving (CURDATE(),**

CURRENT_DATE(), CURTIME(), CURRENT_TIME(),
CURRENT_TIMESTAMP(), and NOW()), 341–342

D**dashes, double (--), 502****data anomalies**

- dirty reads, 472–473, 476
- nonrepeatable reads, 473–474, 476
- phantom reads, 474–475, 476

data directory, backing up, 588**data manipulation**

- INSERT statement
 - foreign key tables, 199–203
 - lookup tables, sample, 195–198
 - one row at a time (<set option> method), 203–205
 - people tables, sample, 198–199
 - rows, one or more (<values option> method), 191–195
 - syntax, 190–191
- performance, optimizing, 578–579
- PHP from MySQL database, 623–625
- REPLACE statement
 - INSERT statement versus, 205
 - <set option> alternative, 207–208, 208–210
 - syntax, 206
 - <values option> alternative, 206–208

data model

- elements, 128
- entities, identifying, 128–130
- normalizing data, 130–132
- refining, 133–136
- relationships between tables, 132–133

data retrieval

- ASPNET/C# application
 - described, 728–729
 - result set, processing, 729–730
 - string data, manipulating, 730–731
- Java/J2EE application
 - described, 674–673
 - result set, processing, 673–674
 - string data, manipulating, 674–675

data retrieval (continued)

- performance, optimizing, 566–571
- PHP application
 - date values, converting, 625
 - described, 621–622
 - manipulating data, 623–625
 - result set, processing, 622–623
- SELECT statement
 - analyzing specific (EXPLAIN statement), 566–571
 - creating, 236–237
 - described, 245–249
 - displaying values, 243–245
 - expressions, 238–241
 - GROUP BY clause, 254–259
 - HAVING clause, 259–261
 - LIMIT clause, 265–267
 - ORDER BY clause, 261–264
 - syntax, 230–235
 - usefulness, 229
 - variables, 241–243
 - WHERE clause, 250–254

data summary functions

- averaging values (AVG ()), 350–351
- bit, 354–356
- COUNT (), 353–354
- described, 349–350
- example, 356–357
- minimum or maximum column value (MIN () and MAX ()), 352–353
- sum of expression (SUM ()), 351–352

data types

- column definitions, 18, 145–152
- length, specifying correct, 579–580
- values, converting, 320–322

data updates

- ASPNET/C# application, 748–750, 750–773
- INSERT statement
 - copying data into existing table, 435–436
 - foreign key tables, 199–203
 - lookup tables, DVDRentals database, 195–198
 - one row at a time (<set option> method), 203–205
 - people tables, DVDRentals database, 198–199
 - REPLACE statement versus, 205
 - rows, one or more (<values option>), 191–195
 - syntax, 190–191
- internal table statistics, 571–573
- Java/J2EE, 692–693, 693–716
- joins, creating multiple table data access, 393–394
- maximum permitted per hour, 544
- nonrepeatable reads, 473–474, 476
- phantom reads, 474–475, 476
- PHP
 - edit.php file, creating, 640–654
 - insert.php file, creating, 655
 - update.php file, creating, 655–656
- REPLACE statement
 - copying data into existing table, 436–438
 - INSERT statement versus, 205
 - <set option> alternative, 207–210
 - syntax, 206
 - <values option> alternative, 206–208

- restoration from binary log files
 - direct, 601–602
 - from text file, 602
- subqueries, 410–411
- syntax, 210
- UPDATE statement
 - joined tables, 216–218
 - single table, 211–216

data value, 18

database

- current status, displaying, 94
- DDL statements and, 23
- defined, 2–3
- entire single, backing up, 588–594
- hierarchical model, 3–4
- identifying, 37
- information, retrieving (SHOW statements), 180–181
- listing, 90
- name, displaying, 92
- network model, 5–6
- PHP connection, 620
- privileges, limiting to, 536
- relational model, 6–7
- script creating and populating, 88
- user account privileges, specifying (ON clause), 536–537

database management system (DBMS)

- functions, 7–8
- MySQL RDBMS, 8–9

database object

- creating, 140–141
- deleting, 142
- modifying, 142

data-driver applications

- application server, 32–33
- creating, 35–40
- MySQL database, connecting, 33–34

date/time data

- adding (ADDDATE () and DATE_ADD ()), 339–340
- column definitions, 151–152
- converting
 - format, 320
 - Java, 676, 711
 - PHP from MySQL database, 625, 638
- current, retrieving (CURDATE (), CURRENT_DATE (), CURTIME (), CURRENT_TIME (), CURRENT_TIMESTAMP (), and NOW ()), 341–342
- day-related values (DAY (), DAYOFMONTH (), DAYNAME (), DAYOFWEEK (), and DAYOFYEAR ()), 344–345
- differences between (DATEDIFF () and TIMEDIFF ()), 344
- month, extracting, 342–343
- specific part, extracting (EXTRACT ()), 341
- subtracting (SUBDATE () or DATE_SUB ()), 340–341
- time parts, extracting (SECOND (), MINUTE (), HOUR (), and TIME ()), 345–346
- year, extracting, 342–343

day, listing current, 344–345

db grant tables, 520–521

DBMS (database management system)

- functions, 7–8
- MySQL RDBMS, 8–9

DCL (access control) statements, 25–26

DDL (database structure, defining) statements, 23–24

debug edition, MySQL, 46

decimal

column places, defining, 145

data type, 147, 148

declaring and initializing variables

ASPNET/C# application, 725–727

Java/J2EE, 670–671

DECODE () system encryption, 358–359

default settings

column values, 18, 153–154

performance, improving, 578

privileges, 80

system variables, 494

definitions, tables, 167–169

defragmenting tables and indexes, 571–573

delays

dirty reads, 472–473, 476

inserting data, 578

nonrepeatable reads, 473–474, 476

deleting

database object, 142

indexes, 179

tables from database, 169–171

deleting data

ASPNET/C# application, 773–777

DELETE statement

joining multiple tables, 394–395

subqueries, 411–413

Java/J2EE, 716–720

from joined tables

<from join delete> alternative, 222–224

<using join delete> alternative, 224–226

PHP, 660–664

from single table, 220–221

syntax, 219

TRUNCATE statement, 226–227

delimited text file

creating, 422

importing data from, 438–444, 449–452

DESCRIBE statements, 183–187

differences, date/time, 340–341, 344

direct invocation, 26–28

directive

ASPNET, 725

JSP, 669, 678

directory structure

data, backing up, 588

data folder

administrative database, 78–79

described, 77–78

described, 73–74

file storage

Linux RPM installation, 74–75

Linux tar installation, 75–76

Windows installation, 76

grant tables

actions and privileges, 80–81

anonymous users, 82

Linux implementation, 81–82

listed, 79

user table, 81

Windows implementation, 82

ownership, changing, 55

dirty reads, 472–473, 476

disaster recovery

backups

all databases on system, 595–596

entire single database, 588–594

to file, 596–599

individual tables, 594

specific databases, 595

described, 587

replication

described, 606–608

implementing, 608–610

master server, 610–612

slave server, 612–615

restoration

binary logging, enabling and disabling, 603

reloading database, 600–601

sample, 603–606

steps, 599

updating from binary log files, 601–602

discontinuing statements (clear \c command), 91–92

displaying

binary logging, 89

connection ID, 94

database name, 92

database status, 94

mysql client utility prompt (\R command), 92–93

values from SELECT statement, 243–245

version number and other information, 487

distribution files, downloading, 46–48

distribution type, 45–46

division operator (/), 272

DML statements, 24–25

dollar sign (\$), 620

double ampersands (&&), 292

double dashes (--), 502

DOUBLE data type, 147

double vertical pipes (||), 292

driver applications

application server, 32–33

creating, 35–40

MySQL database, connecting, 33–34

driver class, Java, 671

dropping and revoking privileges

described, 553

DROP USER statement, 555

REVOKE statement, 553–555

sample, 555–558

users to other users (WITH clause), 543–544

dump file

described, 588

DVDRentals sample, 431

exporting data, 430–431

specifying, 422

duplicate databases

all on system, 595–596

entire single, 588–594

duplicate databases (continued)

to file, 90, 588, 596–599
individual tables, 594
specific, 595

duplicate rows, returning all, 246, 247

duplicating data from other tables

described, 431
existing table
 INSERT statement, 435–436
 REPLACE statement, 436–438
new table, 432–435

duplicating files

tar on Linux, 52–53
text, data to/from, 90

durable transaction property, 456

DVDRentals database example

copying data into new table, 433–435
data manipulation (REPLACE statement)
 <set option> alternative, 207–208
 <values option> alternative, 207–208
data model
 elements, 128
 entities, identifying, 128–130
 normalizing data, 130–132
 refining, 133–136
 relationships between tables, 132–133
database object, creating, 141
deleting data (DELETE statement)
 from joined tables, <from join delete> alternative,
 223–224
 from joined tables, <using join delete> alternative,
 225–226
 from single table, 221
exporting data to sample out file, 428–430
importing data into table
 LOAD DATA statement, 443–444
 mysql command, 447–448
 mysqlimport utility, 451–452
 source command, 445–446
lookup tables, 195–198
people tables, 198–199
purpose, 127
relationships between tables, 132–133
sample data to dump file, 431
tables, managing, 160–167
user accounts privileges, adding and assigning (GRANT
 statement), 546–551

dynamic system settings, retrieving (SELECT statement), 497–498

E

edition

MySQL, 46
Windows, 64–65

edition, MySQL version 5

changes, highlights of, 805–806
stored procedures, 806–807
triggers, 807–808
views, 808–809

Eiffel API, 802

embedded SQL in programming language

asterisk, restricted use of, 234
execution, 29

ENCODE() **system encryption, 358–359**

entities

identifying
 described, 122–124
 DVDRentals data model, 128–130
normalizing, 124
reasons to use, 116

equals sign (=)

C#, 726
as comparison operator, 278, 280
server variables, reading, 502

errors

bug reports, generating on Unix systems, 89
dropping database, 486
exceptions, managing
 ASPNET/C# application, 734–735
 Java/J2EE, 679
Java query result, 708
logging, 506
PHP query result, 622

exclamation point (!), 292

exclamation point, equals sign (!=), 264

exclusive or (XOR) operator

bit aggregate function, 355–356
bitwise, 297, 301
described, 292, 293

execution

call-level interface, 30
direct invocation, 26–28
embedded SQL in programming language, 29
module binding, 29

EXISTS and NOT EXISTS operators, 402–403

exporting data

to an out file, 423–430
described, 421–422
to a dump file, 430–431
DVDRentals sample out file, 428–430
syntax, 422

expressions

operators
 described, 269–270
 grouping operators, 271
 logical, 291–296
 precedence, 270–271
retrieving data (SELECT statement), 238–241
returned values
 averaging, 350–351
 sum, 351–352

F

failures

backups
 all databases on system, 595–596
 entire single database, 588–594
 to file, 596–599

- individual tables, 594
- specific databases, 595
- described, 587
- replication
 - described, 606–608
 - implementing, 608–610
 - master server, 610–612
 - slave server, 612–615
- restoration
 - binary logging, enabling and disabling, 603
 - reloading database, 600–601
 - sample, 603–606
 - steps, 599
 - updating from binary log files, 601–602
- false, 291–296**
- file**
 - backups, 596–599
 - copying
 - tar on Linux, 52–53
 - text, data to/from, 90
 - including, ASPNET/C# application, 733–734
 - ownership, changing, 55
 - query, executing in specified (source \ . command), 93–94
 - storage setup
 - Linux RPM, 74–75
 - Linux tar, 75–76
 - Windows, 76
- first nonnull values, returning or status as all null**
(`COALESCE()` and `ISNULL()`), **311–312**
- first string character**
 - multibyte or numeric value (`ORD()` function), 323
 - numeric value, identifying (`ASCII()`), 323
- first string integer, 312–314**
- FLOAT data type, 147–148**
- FLOOR() numeric data functions, 333**
- flow control functions**
 - `CASE()`, 316–317
 - `IF()`, 314–315
 - `IFNULL()` and `NULLIF()`, 315
 - using in `SELECT` statement, 317–319
- foreign key**
 - DVDRentals database example, creating, 164–167
 - index, 565
 - `INSERT` statement, 199–203
 - managing, 156–158
 - numbers, generating automatically, 155–156
 - relational database design, 120, 121
- formatting values**
 - ASPNET/C# application, 731–732
 - date/time, PHP, 625
 - Java/J2EE, 675–677
 - to specific type or assigning character set to value
(`CAST()`), 320–322
- forms**
 - ASPNET/C# application, 732–733
 - Java/J2EE, 677–678, 707–708, 744
 - PHP application, creating, 628–629, 644–647
 - PHP rendering, 33–34, 618, 626–627
 - tags, 36
- `FOUND_ROWS()` **query and insert function, 362–363**
- fractional data type, 145, 147–148**
- FreeBSD, 45**
- full-text index**
 - `ALTER TABLE` statement, 176–177
 - `CREATE INDEX` statement, 177–178
 - described, 174–175, 565
- functions**
 - built-in versus user-defined, 310
 - comparing data
 - first integer listed as argument `INTERVAL()` and `STRCMP()`, 312–314
 - first nonnull values, returning or status as all null
(`COALESCE()` and `ISNULL()`), 311–312
 - `GREATEST()` and `LEAST()`, 310–311
 - connection (`CONNECTION_ID()`), 361–362
 - control flow
 - `CASE()`, 316–317
 - `IF()`, 314–315
 - `IFNULL()` and `NULLIF()`, 315
 - using in `SELECT` statement, 317–319
 - converting values to specific type (`CAST()`), 320–322
 - database (`DATABASE()`), 361–362
 - date/time
 - add (`ADDDATE()` and `DATE_ADD()`), 339–340
 - current date and time, retrieving (`CURDATE()`, `CURRENT_DATE()`, `CURTIME()`, `CURRENT_TIME()`, `CURRENT_TIMESTAMP()`, and `NOW()`), 341–342
 - differences between (`DATEDIFF()` and `TIMEDIFF()`), 344
 - example, 346–349
 - extracting specific information (`DATE()`, `MONTH()`, `MONTHNAME()`, and `YEAR()`), 342–343
 - pull day-related values (`DAY()`, `DAYOFMONTH()`, `DAYNAME()`, `DAYOFWEEK()`, and `DAYOFYEAR()`), 344–345
 - specific part, extracting (`EXTRACT()`), 341
 - subtracting (`SUBDATE()` or `DATE_SUB()`), 340–341
 - time parts, extracting (`SECOND()`, `MINUTE()`, `HOURL()`, and `TIME()`), 345–346
 - described, 37, 270, 309
 - MySQL version (`VERSION()`), 361–362
 - network address, converting to numeric value
(`INET_ATON()` and `INET_NTOA()`), 362
 - numeric data
 - `CEIL()`, `CEILING()`, and `FLOOR()`, 333
 - cotangent of number (`COT()`), 333
 - `PI()`, 334
 - `POW()` and `POWER()`, 334
 - remainder from dividing two numbers (`MOD()`), 334
 - `ROUND()` and `TRUNCATE()`, 334–335
 - sample, 336–339
 - square root `SQRT()`, 335
 - query and insert
 - `FOUND_ROWS()`, 362–363
 - `LAST_INSERT_ID()`, 363
 - string data
 - case, changing (`LCASE()`, `LOWER()`, `UCASE()`, and `UPPER()`), 326–327
 - character set, identifying (`CHARSET()`), 324

functions (continued)

functions (continued)

- characters from ends, returning (`LEFT()` and `RIGHT()`), 327
- collation, identifying (`COLLATION()`), 303, 324–325
- concatenating (`CONCAT()` and `CONCAT_WS()`), 325
- length (`LENGTH()`), 324
- multibyte value of first character, identifying (`ORD()`), 323
- number of characters (`CHAR_LENGTH()` and `CHARACTER_LENGTH()`), 323
- numeric value of first character, identifying (`ASCII()`), 323
- repeating specified times (`REPEAT()`), 327–328
- reversing order (`REVERSE()`), 328
- sample, 329–332
- substrings
 - finding after specified point (`LOCATE()`), 326
 - locating (`INSTR()`), 325–326
 - returning (`SUBSTRING()`), 328–329
- summarizing data
 - averaging values (`AVG()`), 350–351
 - bit, 354–356
 - `COUNT()`, 353–354
 - described, 349–350
 - example, 356–357
 - minimum or maximum column value (`MIN()` and `MAX()`), 352–353
 - sum of expression (`SUM()`), 351–352
- system encryption
 - `ENCODE()` and `DECODE()`, 358–359
 - `PASSWORD()`, `MD5()`, `SHA()`, and `SHA1()`, 359–360
- system-related operations, sample, 363–366
- user-name and hostname, current session
 - (`CURRENT_USER()`, `SESSION_USER()`, `SYSTEM_USER()`, and `USER()`), 361
- value, assigning character set to (`CAST()`), 320–322

G

global privileges, 536, 538

GPL (GNU General Public License), 9

GRANT statement

- granting or revoking to others (`WITH` clause), 543–544
- privileges (`GRANT` clause), 534–536
- sample, 537–541, 546–551
- secure connection, stipulating (`REQUIRE` clause), 541–543
- syntax, 533–534
- tables or databases, specifying (`ON` clause), 536–537
- to users (`TO` clause), 537

grant tables

- actions and privileges, 80–81
- `columns_priv`, 523–524, 532
- db, 520–521
- host, 521–522
- Linux implementation, 81–82
- listed, 79
- privileges listed by table, 524–526
- reloading
 - `FLUSH PRIVILEGES` statement, 100–101, 551–552
 - with `mysqladmin`, 491
- `tables_priv`, 522–523, 531–532

- user, 81, 518–520
- user accounts, reloading, 100–101
- Windows implementation, 82

greater than operator (>), 238

greater than sign (>), 278

GREATEST() and LEAST() data comparison, 310–311

group accounts, 51–52

group file or directory ownership, 55

grouped data

- `GROUP BY` data clause, 254–259
- `SELECT` subqueries, 403–407

grouping operators, 271

guest users

- db table, 82
- Linux user table, 81
- removing, 556

H

HAVING clause, data retrieval, 259–261, 405

help command (\? or ?), 92, 95–96, 486

hierarchical database model, 3–4

hierarchical relationships

- foreign keys, 156
- model, 3–4
- network model, 5–6

host

- anonymous user access control, 82

configuration

- command line, specifying at, 502–503
- option file, specifying in, 503
- runtime, specifying at, 503–506
- Windows, 61–64

connection

- authenticating, 527–529
- Windows root user, 82

host grant tables, 521–522

identifying, PHP application, 37

master server

- connected slave hosts, listing (`SHOW SLAVE HOSTS` statement), 612
- resetting (`RESET MASTER` statement), 611
- status (`SHOW MASTER STATUS` statement), 611–612
- user account, setting up, 608–609
- name, specifying, 486
- options, listed, 88–89
- privileges, specifying, 537
- settings, retrieving (`SHOW VARIABLES` statement), 495–496

slave

- `CHANGE MASTER` statement, 613–614
- connected, listing (`SHOW SLAVE HOSTS` statement), 612
- files, 607–608, 610
- principles, 606–607
- `RESET SLAVE` statement, 614
- `SHOW SLAVE STATUS` statement, 614
- `START SLAVE` statement, 614–615
- `STOP SLAVE` statement, 615
- status (`SHOW STATUS` statement), 498–500
- version, displaying, 94

- Web
 - JSP-based application, 668–669, 671–674
 - PHP-based application, 618
 - hostname, current session, 361**
 - hour, extracting, 345–346**
 - HTML (Hypertext Markup Language) forms**
 - ASP.NET/C# application, 732–733
 - Java/J2EE, 677–678, 707–708, 744
 - PHP application, creating, 628–629, 644–647
 - PHP rendering, 33–34, 618, 626–627
 - tags, 36
 - HTTP (Hypertext Transfer Protocol)**
 - header, sending data through, 677, 732
 - PHP application, 34
 - Web browsers, redirecting, 627
- I**
- IBM, 3, 11**
 - IF () flow control, 314–315**
 - if...else result set, processing, 39**
 - IFNULL () flow control, 315**
 - importing data into table**
 - mysql utility
 - LOAD DATA statement, 438–444
 - mysql command, 446–448
 - source command, 444–446
 - mysqlimport utility, 449–452
 - IMS (Information Management System), 3**
 - IN and NOT IN operators, 283, 401–402**
 - include files**
 - Java/J2EE, 678–679
 - PHP, 627–628
 - indexes**
 - creating
 - adding to existing tables, 175–178
 - defining when creating tables, 172–175
 - syntax, basic, 19
 - DDL statements and, 23
 - defragmenting, 571–573
 - disabling in full backup, 593
 - performance, optimizing, 561–565, 573–577
 - removing, 179
 - showing, 182
 - types, 171–172
 - updating and deleting data, 579
 - individual accounts**
 - adding and assigning privileges (GRANT statement)
 - granting or revoking to others (WITH clause), 543–544
 - privileges (GRANT clause), 534–536
 - sample, 537–541, 546–551
 - secure connection, stipulating (REQUIRE clause), 541–543
 - syntax, 533–534
 - tables or databases, specifying (ON clause), 536–537
 - to users (TO clause), 537
 - current status, displaying, 94
 - dropping and revoking privileges
 - described, 553
 - DROP USER statement, 555
 - REVOKE statement, 553–555
 - sample, 555–558
 - identifying, PHP application, 37
 - passwords, setting
 - administrative (mysql utility), 100
 - changing, 552–553
 - configuration file, 103–106
 - grant tables, reloading, 100–101, 551–552
 - root account, 101–102
 - SET PASSWORD statement, 551
 - server connection, specifying, 487
 - tar files on Linux, 51–52
 - viewing privileges (SHOW GRANTS statement), 544–545
 - information categories**
 - identifying
 - described, 122–124
 - DVDRentals data model, 128–130
 - normalizing, 124
 - reasons to use, 116
 - Information Management System (IMS), 3**
 - information, system**
 - DESCRIBE statements, 183–187
 - listing databases and tables, 90
 - SHOW SLAVE STATUS statement, 614
 - SHOW statements
 - database-related, 180–181
 - table-related, 181–183
 - user accounts (SHOW GRANTS statement), 545
 - initializing and declaring variables**
 - ASP.NET/C# application, 725–727
 - Java/J2EE, 670–671
 - inner joins, 377–380**
 - InnoDB table, 159**
 - input, saving to text file (tee \T command), 94**
 - inserting data**
 - ASP.NET/C# application, 748–773
 - described, 189–190
 - FOUND_ROWS (), 362–363
 - INSERT statement
 - copying data into existing table, 435–436
 - foreign key tables, 199–203
 - lookup tables, DVDRentals database, 195–198
 - one row at a time (<set option> method), 203–205
 - people tables, DVDRentals database, 198–199
 - REPLACE statement versus, 205
 - rows, one or more (<values option>), 191–195
 - syntax, 190–191
 - Java/J2EE, 692–716
 - LAST_INSERT_ID (), 363
 - performance, optimizing, 577–578
 - PHP, 639–640, 656–660
 - installing MySQL**
 - ease of, 8, 43
 - initializing, 54–55
 - preliminary decisions
 - distribution type, 45–46
 - edition, 46
 - platform, 44–45
 - platforms, 47–48
 - version, 44
 - RPM files, on Linux, 49–51
 - starting server
 - automatically, 56–57
 - manually, 55–56

installing MySQL (continued)

- tar files on Linux
 - copying, 52–53
 - unpacking distribution file, 53–54
 - user and group accounts, creating, 51–52

- testing
 - on Linux, 65–68
 - in Windows, 68–70
- on Windows
 - editions, 64–65
 - installation program, 57–61
 - server, configuring, 61–64

integer data type, 145–147

interactive mode

- administrative, 90–91
- sample, 94–96

ISAM table, 159

ISO (International Organization for Standardization), 12

isolated transaction property, 456

isolation level

- data anomalies
 - dirty reads, 472–473, 476
 - nonrepeatable reads, 473–474, 476
 - phantom reads, 474–475, 476
- listed, 475–476
- setting (SET TRANSACTION statement), 476–479

J

Java Database Connectivity (JDBC), 30, 668

Java/J2EE (Java2 Enterprise Edition)

- classes, importing, 669
- connecting to MySQL database, 671–674
- converting values, 675–677
- declaring and initializing variables, 670–671
- deleting data from MySQL database, 716–720
- described, 667–669
- exceptions, managing, 679
- HTML forms, 677–678
- including JSP files, 678–679
- insert and update functions, adding to application, 693–716
- inserting and updating data in MySQL database, 692–693
- redirecting browsers, 678
- retrieving data from MySQL database
 - described, 674–673
 - result set, processing, 673–674
 - string data, manipulating, 674–675
- sample application, creating
 - described, 679–682
 - index.jsp file, 682–692

JavaScript, 629–631

JDBC (Java Database Connectivity), 30, 668

joined tables

- deleting data (DELETE statement)
 - <from join delete> alternative, 222–224
 - <using join delete> alternative, 224–226
- indexes, benefits of using, 563, 564
- performance, improving, 573
- SELECT statement, 246

- subqueries, rewriting as, 397
- updating data (UPDATE statement), 216–218
- when to use, 369–370

junction table, 126–127, 136

K

keys

foreign

- DVDRentals database example, creating, 164–167
- index, 565
- INSERT statement, 199–203
- managing, 156–158
- numbers, generating automatically, 155–156
- relational database design, 120, 121

primary

- columns, defining, 154–155
- defining, 18, 21
- foreign key and, 120
- index, 176–177, 565
- normalizing data, 112–116, 124
- relational database model, 110
- relationships, 117–118, 126–127
- tables, managing, 154–155

keyword

- arithmetic, 272–277
- bitwise, 296–302
- comparison, 277–291
- DDL statements, 24
- defined, 16
- expressions, creating
 - described, 269–270
 - grouping operators, 271
 - precedence, 270–271
- logical, 291–296
- sort, 302–306

L

languages, programming

C

- APIs, 801
- call-level interface, 30
- embedded SQL, 29
- SQL versus, 10

C#, 724

- direct communication with SQL database, 30

embedded SQL

- asterisk, restricted use of, 234
- execution, 29
- module binding, 29
- multiple, use of, 628
- PHP and, 618
- procedural nature of, 14
- SQL versus, 10
- user-defined functions, 310

- LAST_INSERT_ID() functions, 363

launching mysql utility, 96–97

left outer joins, 386–388

legacy databases, 4**length**

- columns, numeric data, 145
- data types, specifying correct, 579–580
- string
 - C# functions, 730–731
 - data functions, 324
 - Java functions, 675
 - PHP functions, 624

letters, string

- from ends, returning (`LEFT()` and `RIGHT()`), 327
- extracting, PHP (`substr()`), 624–625

library, 88**LIMIT clause, 265–267****linked tables**

- deleting data (`DELETE` statement)
 - <from join delete> alternative, 222–224
 - <using join delete> alternative, 224–226
- indexes, benefits of using, 563, 564
- performance, improving, 573
- `SELECT` statement, 246
- subqueries, rewriting as, 397
- updating data (`UPDATE` statement), 216–218
- when to use, 369–370

Linux

- anonymous users, removing, 556
- benefits of using, 48
- configuration file, 85, 86
- debug edition, 46
- grant tables, 81–82
- MySQL options, 44, 45, 94
- names, case-sensitivity of, 141
- open-source movement and, 9
- password configuration file, 103–105
- PHP application support, 33
- query and binary logging, 508–512
- RPM installation
 - file storage, 74–75
 - files, 49–51
 - naming convention, 47
- tar files
 - copying, 52–53
 - file storage, 75–76
 - unpacking distribution file, 53–54
 - user and group accounts, creating, 51–52

list data types, 151**literal values, 270****LOAD DATA statement, importing data into table**

- described, 438–444
- performance, improving, 578

loading

- server
 - automatically, 56–57
 - manually, 55–56
 - slave, 614–615
- transactions (`START TRANSACTION` statement), 458

locking tables

- full database back up, 593
- transactions, 456, 480

log files

- data directory, 77
- error log, 506

logging

- binary
 - displaying, 89
 - enabling and disabling, 603
 - setting up, 507–515
 - query, 507, 508–515

logging on

- ASPNET/C# application, 727–728
- authenticating, 527–529
- database version, 361–362
- establishing, 37
- failures, 38
- ID, displaying, 94, 361–362
- Java/J2EE, 671–674
- maximum permitted per hour, 544
- PHP application, 619–621
- retrying, 487
- secure, requiring, 519
- slave server
 - altering (`CHANGE MASTER` statement), 613–614
 - listing (`SHOW SLAVE HOSTS` statement), 612
 - version, 361–362

logical operators, 291–296**logon accounts**

- adding and assigning privileges (`GRANT` statement)
 - granting or revoking to others (`WITH` clause), 543–544
 - privileges (`GRANT` clause), 534–536
 - sample, 537–541, 546–551
 - secure connection, stipulating (`REQUIRE` clause), 541–543
 - syntax, 533–534
 - tables or databases, specifying (`ON` clause), 536–537
 - to users (`TO` clause), 537
- current status, displaying, 94
- dropping and revoking privileges
 - described, 553
 - `DROP USER` statement, 555
 - `REVOKE` statement, 553–555
 - sample, 555–558
- identifying, PHP application, 37
- passwords, setting
 - administrative (mysql utility), 100
 - changing, 552–553
 - configuration file, 103–106
 - grant tables, reloading, 100–101, 551–552
 - root account, 101–102
 - `SET PASSWORD` statement, 551
- server connection, specifying, 487
- tar files on Linux, 51–52
- viewing privileges (`SHOW GRANTS` statement), 544–545

lookup tables, DVDRentals database

- example, creating, 160–162
- `INSERT` statement, 195–198

lowercase

- `GRANT` statement scope columns, 534
- name conversion issues, 141
- string data, changing (`LCASE()`, `LOWER()`, `UCASE()`, and `UPPER()`), 326–327

M

Mac OS X, 45

manipulating data

- INSERT statement
 - foreign key tables, 199–203
 - lookup tables, sample, 195–198
 - one row at a time (<set option> method), 203–205
 - people tables, sample, 198–199
 - rows, one or more (<values option> method), 191–195
 - syntax, 190–191
- performance, optimizing, 578–579
- PHP from MySQL database, 623–625
- REPLACE statement
 - INSERT statement versus, 205
 - <set option> alternative, 207–208, 208–210
 - syntax, 206
 - <values option> alternative, 206–208

master server replication

- connected slave hosts, listing (SHOW SLAVE HOSTS statement), 612
- resetting (RESET MASTER statement), 611
- status (SHOW MASTER STATUS statement), 611–612
- user account, setting up, 608–609

mathematical functions

- averaging values (AVG()), 350–351
- date/time (ADDDATE() and DATE_ADD()), 339–340
- operators, 272–277, 282

MD5() system encryption function, 359–360

memory, caching query results, 246, 580–583

MERGE table, 160

metadata, 2

Microsoft Windows

- access control, 82
- anonymous users, removing, 556
- benefits of using, 48
- configuration file, 86–88, 103, 105–106
- debug edition, 46
- editions, 64–65
- file storage, 76
- installation program, 47, 57–61
- MySQL options, 44, 94
- names, case-insensitivity of, 141
- query and binary logging, 512–515
- root account password, setting, 101
- server, configuring, 61–64
- testing installation, 68–70

minimum or maximum column value (MIN() and MAX()) functions, 352–353

minus sign (-), 238

minute, extracting, 345–346

model, data

- elements, 128
- entities, identifying, 128–130
- normalizing data, 130–132
- refining, 133–136
- relationships between tables, 132–133

module binding, 29

modulo operator (%), 272

month, day of, 344–345

.msi file, installing MySQL from, 57–61

multibyte value of first character, identifying (ORD()), 323

multiple database backups, 594–599

multiple table data access

- DELETE statement joins, creating, 394–395
- DELETE subqueries, 411–413
- indexes, benefits of using, 563
- joins, when to use, 369–370
- SELECT statement joins, creating
 - basic joins, 374–377
 - described, 370–371
 - inner joins and cross joins, 377–380
 - left outer joins, 386–388
 - natural joins, 391–392
 - outer joins, 385–391
 - right outer joins, 389–390
 - sample, 381–385
 - sample tables, 372–373
 - straight joins, 380–381
 - syntax, 371
- SELECT subqueries
 - ALL operator, 400
 - ANY and SOME operators, 399–400
 - comparison operators, 396–399
 - EXISTS and NOT EXISTS operators, 402–403
 - grouped data, 403–407
 - IN and NOT IN operators, 401–402
 - sample, 407–410
 - unions joining SELECT statements, 414–417
- UPDATE statement joins, creating, 393–394
- UPDATE subqueries, 410–411

multiplication operator (*), 272

MyISAM tables

- binary logging, displaying, 89
- full-text index, adding, 174–175
- listing, 90
- syntax, 160

MySQL

- data-driver applications, 33–34
- described, 8–9
- open-source movement, 9
- version 5
 - changes, highlights of, 805–806
 - stored procedures, 806–807
 - triggers, 807–808
 - views, 808–809

MySQL C API, 801

MySQL C++ API, 801

mysql client utility

- account passwords, assigning, 100
- batch mode, 97–100, 600
- current database, changing (use \u command), 94
- described, 22–23
- direct invocation, 27
- discontinuing statements (clear \c command), 91–92
- help command (\? or ?), 92
- importing data into table
 - LOAD DATA statement, 438–444
 - mysql command, 446–448
 - source command, 444–446

interactive mode
 described, 90–91
 example, 94–96
 reloading database, 600–601
 launching, 95, 96–97
 programs, 97–100
 prompt display (`\R` command), 92–93
 query, executing in specified file (source `\.` command), 93–94
 quitting (exit or quit `\q`), 92
`SELECT` statement, 25
 session information (status `\s` command), 94
 text file, saving input and output to (tee `\T` command), 94

MySQL Connector/J API, 802

MySQL Connector/ODBC, 802

mysql database

data folder, 78–79
 user grant table, 81–82

MySQL Eiffel API, 802

MySQL Perl API, 802

MySQL PHP API, 802

MySQL Python API, 802

MySQL Tcl API, 803

mysqldadmin utility

commands, 488–494
 described, 89, 485–486
 grant tables, reloading, 101

MySQLdb, 802

mysqldump command

all database tables, 588–594
 all databases, 595–596
 individual tables, 594
 specific multiple databases, 595

mysqlimport utility, 449–452

N

name

columns, 21, 270
 conventions, normalizing data, 124–125
 converting, case-sensitivity and, 141
 database, displaying, 92
 of day, 344–345

natural joins, 391–392

negative values, 145

NetWare system, 90

network address, converting to numeric value

(`INET_ATON()` and `INET_NTOA()`), 362

network database model, 5–6

new table, copying data into, 432–435

nonnull values, returning first or status as all null

(`COALESCE()` and `ISNULL()`), 311–312

nonregistered users

db table, 82
 Linux user table, 81
 removing, 556

nontransactional tables, locking

described, 479
 example, 481–483
`LOCK TABLES` statement, 480
`UNLOCK TABLES` statement, 480–481

nonunique index

`ALTER TABLE` statement, 176–177
`CREATE INDEX` statement, 177–178
 described, 565
 uses of, 173–174

normalizing data

DVDRentals data model, 130–132
 first normal form, 111–113
 second normal form, 113–115
 third normal form, 115–116

NOT IN operator, 401–402

NOT operator

bitwise, 297
 described, 281–282, 292

notee (\t) command, 94

NULL, 291–296

NULL operator, 278, 281

null values

column, ability to handle, 152–153
 columns, accepting, 152–153
 as default, 153–154
 natural joins, 392
 outer joins, 385, 388
 restricting use of, 580

NULLIF () flow control, 315

number of characters (CHAR_LENGTH () and CHARACTER_LENGTH ()), string data functions, 323

number, of month, retrieving, 343

numbers

column definitions, 145–148
 converting, 320
 network address, converting to (`INET_ATON()` and `INET_NTOA()`), 362
 numeric data, maximum displayed, 145
 string data
 from ends, returning (`LEFT()` and `RIGHT()`), 327
 extracting, PHP (`substr()`), 624–625
 first character, identifying (`ASCII()`), 323

numeric data functions

`CEIL()`, `CEILING()`, and `FLOOR()`, 333
 cotangent of number (`COT()`), 333
`PI()`, 334
`POW()` and `POWER()`, 334
 remainder from dividing two numbers (`MOD()`), 334
`ROUND()` and `TRUNCATE()`, 334–335
 sample, 336–339
 square root `SQRT()`, 335

O

object, database

creating, 140–141
 deleting, 142
 modifying, 142

object variable, 670

object-relational model, 13

ODBC, 802

ODBC (Open Database Connectivity), 30, 724

OOP (object-oriented programming)

C#, 724
 Java, 668

opening scriptlet tag (<%)

opening scriptlet tag (<%), 678, 734

open-source code, 8, 9, 618

operating system

Linux

- anonymous users, removing, 556
- benefits of using, 48
- configuration file, 85, 86
- debug edition, 46
- grant tables, 81–82
- MySQL options, 44, 45, 94
- names, case-sensitivity of, 141
- open-source movement and, 9
- password configuration file, 103–105
- PHP application support, 33
- query and binary logging, 508–512
- RPM installation, 47, 49–51, 74–75
- tar files, 51–54, 75–76

MySQL installation, 44–45, 47–48

Unix-like

- backup, quick, 90
- bug reports, generating, 89
- choosing, 45
- names, case-sensitivity of, 141
- script, starting and stopping server, 88

Windows

- access control, 82
- anonymous users, removing, 556
- benefits of using, 48
- configuration file, 86–88, 103, 105–106
- debug edition, 46
- editions, 64–65
- file storage, 76
- installation program, 47, 57–61
- MySQL options, 44, 94
- names, case-insensitivity of, 141
- query and binary logging, 512–515
- root account password, setting, 101
- server, configuring, 61–64
- testing installation, 68–70

operators

- arithmetic, 272–277
- bitwise, 296–302
- comparison, 277–291
- expressions, creating
 - described, 269–270
 - grouping operators, 271
 - precedence, 270–271
- logical, 291–296
- sort, 302–306

optimizing performance

- deleting data, 578–579
- described, 561
- indexing, 561–565, 573–577
- inserting data, 577–578
- modifying data, 578–579
- normalizing data, 111
- queries, 565
- retrieving data
 - described, 566
 - SELECT statement, analyzing specific (EXPLAIN statement), 566–571

- system cache, 580–583
- tables, 571–573, 579–580

option file

- server, 503
- specifying, 85–88

optional language, 15

OR operator

- bit aggregate function, 355
- bitwise, 297
- described, 292, 294

ORDER BY clause, retrieving data, 261–264

out file

- exporting data, 423–430
- specifying, 422

outer joins, 385–391

output, saving to text file (tee \T command), 94

P

parent-child relationships

- foreign keys, 156
- hierarchical model, 3–4
- network model, 5–6

parentheses (()), 17, 310

PASSWORD () system encryption function, 359–360

passwords

- account, assigning, 100
- authenticating connections, 527–529
- encryption method
 - data, 358–359
 - Java connections and, 687
 - PHP connections and, 635
- grant privileges, specifying, 537
- identifying, 37
- root, 63
- server, specifying, 487
- setting
 - changing, 552–553
 - grant tables, reloading (FLUSH PRIVILEGES statement), 551–552
 - SET PASSWORD statement, 551
- user accounts, assigning
 - administrative (mysql utility), 100
 - configuration file, 103–106
 - grant tables, reloading, 100–101
 - root account, 101–102

path, shortening use of, 85

people tables, DVDRentals database

- creating, 163–164
- INSERT statement, 198–199

percent character (%), 180, 284, 527

performance, optimizing

- deleting data, 578–579
- described, 561
- indexing, 561–565, 573–577
- inserting data, 577–578
- modifying data, 578–579
- normalizing data, 111
- queries, 565

- retrieving data
 - described, 566
 - SELECT statement, analyzing specific (EXPLAIN statement), 566–571
 - system cache, 580–583
 - tables, 571–573, 579–580
- period, asterisk (*), 286**
- Perl API, 802**
- permanent tables, locking**
 - described, 479
 - example, 481–483
 - LOCK TABLES statement, 480
 - UNLOCK TABLES statement, 480–481
- permissions. See privileges**
- persistent stored module (PSM), 13**
- PHP (PHP: Hypertext Preprocessor)**
 - application, creating basic, 628–639
 - call-level interface, 30, 802
 - connecting to MySQL database, 33–34, 619–621
 - data, retrieving
 - date values, converting, 625
 - described, 621–622
 - manipulating data (strlen() and strcmp()), 623–625
 - result set, processing (while command), 622–623
 - deleting data, 660–664
 - described, 618–619
 - HTML forms, 626–627
 - include files, 627–628
 - inserting data into table, 639–640
 - updating data
 - edit.php file, creating, 640–654
 - index.php file, modifying, 656–660
 - insert.php file, creating, 655
 - update.php file, creating, 655–656
 - Web-based applications, popularity of, 33
- PI() numeric data function, 334**
- platform, MySQL installation, 44–45, 47–48**
- port number, server connection, 487**
- portability, 8**
- posting methods, HTML forms, 626**
- POW() and POWER() numeric data functions, 334**
- precedence operators, 270–271**
- predefined code**
 - built-in versus user-defined, 310
 - comparing data
 - first integer listed as argument INTERVAL() and STRCMP(), 312–314
 - first nonnull values, returning or status as all null (COALESCE() and ISNULL()), 311–312
 - GREATEST() and LEAST(), 310–311
 - connection (CONNECTION_ID()), 361–362
 - control flow
 - CASE(), 316–317
 - IF(), 314–315
 - IFNULL() and NULLIF(), 315
 - using in SELECT statement, 317–319
 - converting values to specific type (CAST()), 320–322
 - database (DATABASE()), 361–362
 - date/time
 - add (ADDDATE() and DATE_ADD()), 339–340
 - current date and time, retrieving (CURDATE(), CURRENT_DATE(), CURTIME(), CURRENT_TIME(), CURRENT_TIMESTAMP(), and NOW()), 341–342
 - differences between (DATEDIFF() and TIMEDIFF()), 344
 - example, 346–349
 - extracting specific information (DATE(), MONTH(), MONTHNAME(), and YEAR()), 342–343
 - pull day-related values (DAY(), DAYOFMONTH(), DAYNAME(), DAYOFWEEK(), and DAYOFYEAR()), 344–345
 - specific part, extracting (EXTRACT()), 341
 - subtracting (SUBDATE() or DATE_SUB()), 340–341
 - time parts, extracting (SECOND(), MINUTE(), HOUR(), and TIME()), 345–346
 - described, 37, 270, 309
 - MySQL version (VERSION()), 361–362
 - network address, converting to numeric value (INET_ATON() and INET_NTOA()), 362
 - numeric data
 - CEIL(), CEILING(), and FLOOR(), 333
 - cotangent of number (COT()), 333
 - PI(), 334
 - POW() and POWER(), 334
 - remainder from dividing two numbers (MOD()), 334
 - ROUND() and TRUNCATE(), 334–335
 - sample, 336–339
 - square root SQRT(), 335
 - query and insert
 - FOUND_ROWS(), 362–363
 - LAST_INSERT_ID(), 363
 - string data
 - case, changing (LCASE(), LOWER(), UCASE(), and UPPER()), 326–327
 - character set, identifying (CHARSET()), 324
 - characters from ends, returning (LEFT() and RIGHT()), 327
 - collation, identifying (COLLATION()), 303, 324–325
 - concatenating (CONCAT() and CONCAT_WS()), 325
 - length (LENGTH()), 324
 - multibyte value of first character, identifying (ORD()), 323
 - number of characters (CHAR_LENGTH() and CHARACTER_LENGTH()), 323
 - numeric value of first character, identifying (ASCII()), 323
 - repeating specified times (REPEAT()), 327–328
 - reversing order (REVERSE()), 328
 - sample, 329–332
 - substrings
 - finding after specified point (LOCATE()), 326
 - locating (INSTR()), 325–326
 - returning (SUBSTRING()), 328–329
 - summarizing data
 - averaging values (AVG()), 350–351
 - bit, 354–356
 - COUNT(), 353–354
 - described, 349–350
 - example, 356–357
 - minimum or maximum column value (MIN() and MAX()), 352–353
 - sum of expression (SUM()), 351–352

predefined code (continued)

predefined code (continued)

- system encryption
 - ENCODE () and DECODE (), 358–359
 - PASSWORD (), MD5 (), SHA (), and SHA1 (), 359–360
- system-related operations, sample, 363–366
- user-name and hostname, current session
 - (CURRENT_USER (), SESSION_USER (), SYSTEM_USER (), and USER ()), 361
- value, assigning character set to (CAST ()), 320–322

primary key

- columns, defining, 154–155
- defining, 18, 21
- foreign key and, 120
- index
 - adding to table (ALTER TABLE statement), 176–177
 - described, 565
- normalizing data
 - data model, 124
 - first normal form, 112–113
 - second normal form, 113–115
 - third normal form, 115–116
- relational database model, 110
- relationships
 - many-to-many, 118, 126–127
 - one-to-many, 118
 - one-to-one, 117
- tables, managing, 154–155

primitive variable, 670

prioritizing statements, 246

privilege columns

- columns_priv table, 523
- db table, 520
- defined, 518
- host table, 521
- tables_priv table, 522, 531–532
- user table, 519

privileges

- adding and assigning (GRANT statement), 546–551
- client program supporting, 89
- dropping and revoking
 - described, 553
 - DROP USER statement, 555
 - REVOKE statement, 553–555
 - sample, 555–558
- execute, changing, 56
- grant tables (access control), 80–81
- granting or revoking to others (WITH clause), 543–544
- privileges (GRANT clause), 534–536
- sample, 537–541
- secure connection, stipulating (REQUIRE clause), 541–543
- syntax, 533–534
- tables or databases, specifying (ON clause), 536–537
- to users (TO clause), 537
- verifying, 529–533

problems

- anomalies
 - dirty reads, 472–473, 476
 - nonrepeatable reads, 473–474, 476
 - phantom reads, 474–475, 476
- bug reports, generating on Unix systems, 89

- dropping database, 486
- exceptions, managing
 - ASP.NET/C# application, 734–735
 - Java/J2EE, 679
- Java query result, 708
- logging, 506
- PHP query result, 622

problems, recovery from

- backups
 - all databases on system, 595–596
 - entire single database, 588–594
 - to file, 596–599
 - individual tables, 594
 - specific databases, 595
- described, 587
- replication
 - described, 606–608
 - implementing, 608–610
 - master server, 610–612
 - slave server, 612–615
- restoration
 - binary logging, enabling and disabling, 603
 - reloading database, 600–601
 - sample, 603–606
 - steps, 599
 - updating from binary log files, 601–602

programming languages

C

- APIs, 801
- call-level interface, 30
- embedded SQL, 29
- SQL versus, 10

C#, 724

- direct communication with SQL database, 30

embedded SQL

- asterisk, restricted use of, 234
- execution, 29
- module binding, 29
- multiple, use of, 628
- PHP and, 618
- procedural nature of, 14
- SQL versus, 10
- user-defined functions, 310

programs, MySQL

- administrative (mysql utility)
 - account passwords, assigning, 100
 - in batch mode from command prompt, 97–100
 - current database, changing (\u command), 94
 - discontinuing statements (clear \c command), 91–92
 - help command (\? or ?), 92
 - in interactive mode, 90–91
 - interactive use example, 94–96
 - launching, 96–97
 - prompt display (\R command), 92–93
 - query, executing in specified file (source \. command), 93–94
 - quitting (exit or quit \q), 92
 - session information (status \s command), 94
 - text file, saving input and output to (tee \T command), 94
- client, listed, 89–90

options, specifying
 at command prompt, 83–85
 in configuration file, 85–88
 server-related, listed, 88–89

prompt, command line
 data, importing, 446–448
 display (`\R` command), 92–93
 options, specifying, 83–85
 returning from mysql, 92
 server configuration, specifying, 502–503

protocol, connection, 487

PSM (persistent stored module), 13

Python API, 802

Q

qualified column name, 217

queries. See also SELECT statement
 assigning to variable, 38
 executing in specified file (source `\.` command), 93–94
`FOUND_ROWS()`, 362–363
 Java application connected to MySQL database, 674–673
`LAST_INSERT_ID()`, 363
 logging, 507, 508–515
 number permitted in hour, 543
 performance, optimizing
 alternative forms, 573
 indexing, 561–565
 reasons for, 565

query cache, 246, 580–583

query logging, 507, 508–515

quitting (exit or quit `\q`) administrative (mysql utility), 92

R

READ COMMITTED isolation level, 475–476

READ UNCOMMITTED isolation level, 475–476

reads, dirty, 472–473, 476

recovery
 backups
 all databases on system, 595–596
 entire single database, 588–594
 to file, 596–599
 individual tables, 594
 specific databases, 595

described, 587

replication
 described, 606–608
 implementing, 608–610
 master server, 610–612
 slave server, 612–615

restoration
 binary logging, enabling and disabling, 603
 reloading database, 600–601
 sample, 603–606
 steps, 599
 updating from binary log files, 601–602

redirecting browsers
 ASPNET/C# application, 733
 Java/J2EE, 678

REGEXP operator, 285–287

registered users
 adding and assigning privileges (`GRANT` statement)
 granting or revoking to others (`WITH` clause), 543–544
 privileges (`GRANT` clause), 534–536
 sample, 537–541, 546–551
 secure connection, stipulating (`REQUIRE` clause),
 541–543
 syntax, 533–534
 tables or databases, specifying (`ON` clause), 536–537
 to users (`TO` clause), 537
 current status, displaying, 94
 dropping and revoking privileges
 described, 553
`DROP USER` statement, 555
`REVOKE` statement, 553–555
 sample, 555–558
 identifying, PHP application, 37
 passwords, setting
 administrative (mysql utility), 100
 changing, 552–553
 configuration file, 103–106
 grant tables, reloading, 100–101, 551–552
 root account, 101–102
`SET PASSWORD` statement, 551
 server connection, specifying, 487
 tar files on Linux, 51–52
 viewing privileges (`SHOW GRANTS` statement), 544–545

regular index
`ALTER TABLE` statement, 176–177
`CREATE INDEX` statement, 177–178
 described, 565
 uses of, 173–174

relational database
 current status, displaying, 94
 data model
 column types, identifying, 120
 described, 6–7
 elements, 119–120, 128
 entities, identifying, 122–124, 128–130
 foreign key, 121
 normalizing data, 124–125, 130–132
 refining, 126–127, 133–136
 relationships between tables, 125–126, 132–133
 steps, 122

DDL statements and, 23
 defined, 2–3
 entire single, backing up, 588–594
 hierarchical model, 3–4
 identifying, 37
 information, retrieving (`SHOW` statements), 180–181
 listing, 90
 model, 109–110
 MySQL, generally, 8–9
 name, displaying, 92
 network model, 5–6
 normalized data
 defined, 111
 first normal form, 111–113
 model, 124–125
 second normal form, 113–115
 third normal form, 115–116
 PHP connection, 620

relational database (continued)

- privileges, limiting to, 536
- relational model, 6–7
- script creating and populating, 88
- table relationships
 - defined, 116
 - identifying, 125–126, 132–133
 - many-to-many, 119, 126–127
 - one-to-many, 117–118
 - one-to-one, 117
- user account privileges, specifying (ON clause), 536–537

relational model

- introduction of, 109–110
- primary key, 110
- rows, 110
- table, 110

reloading database

- batch mode, 600
- interactive mode, 600–601

remainder, dividing two numbers (MOD()), 334

removing

- database object, 142
- indexes, 179
- tables from database, 169–171

removing data

- ASPNET/C# application, 773–777
- DELETE statement
 - joining multiple tables, 394–395
 - subqueries, 411–413
- Java/J2EE, 716–720
- from joined tables
 - <from join delete> alternative, 222–224
 - <using join delete> alternative, 224–226
- PHP 660–664
- from single table, 220–221
- syntax, 219
- TRUNCATE statement, 226–227

REPEATABLE READ isolation level, 475–476

repeating

- default values, usefulness of, 153
- specified times (REPEAT()), 327–328

REPLACE statement

- copying data into existing table, 436–438
- INSERT statement versus, 205
- <set option> alternative, 207–210
- syntax, 206
- <values option> alternative, 206–208

replication

- described, 606–608
- implementing, 608–610
- master server
 - connected slave hosts, listing (SHOW SLAVE HOSTS statement), 612
 - described, 610
 - resetting (RESET MASTER statement), 611
 - status (SHOW MASTER STATUS statement), 611–612
- slave server
 - CHANGE MASTER statement, 613–614
 - described, 612
 - RESET SLAVE statement, 614
 - SHOW SLAVE STATUS statement, 614

- START SLAVE statement, 614–615
- STOP SLAVE statement, 615

reserved words

- arithmetic, 272–277
- bitwise, 296–302
- comparison, 277–291
- DDL statements, 24
- defined, 16
- expressions, creating
 - described, 269–270
 - grouping operators, 271
 - precedence, 270–271
- logical, 291–296
- sort, 302–306

resetting

- master server (RESET MASTER statement), 611
- slave server (RESET SLAVE statement), 614

restoration

- binary logging, enabling and disabling, 603
- reloading database
 - batch mode, 600
 - interactive mode, 600–601
- sample, 603–606
- updating from binary log files
 - direct, 601–602
 - from text file, 602

result set

- ASPNET/C# application, 729–730
- assigning, 38–39
- caching, 580–583
- converting values
 - ASPNET/C# application, 731–732
 - date/time, PHP 625
 - Java/J2EE, 675–677
 - to specific type or assigning character set to value (CAST()), 320–322
- Java/J2EE, 673–674
- PHP data retrieval, 622–623, 637

retrieving data

- ASPNET/C# application
 - described, 728–729
 - result set, processing, 729–730
 - string data, manipulating, 730–731
- Java/J2EE application
 - described, 674–673
 - result set, processing, 673–674
 - string data, manipulating, 674–675
- performance, optimizing, 566–571
- PHP application
 - date values, converting, 625
 - described, 621–622
 - manipulating data, 623–625
 - result set, processing, 622–623
- SELECT statement
 - analyzing specific (EXPLAIN statement), 566–571
 - creating, 236–237
 - described, 245–249
 - displaying values, 243–245
 - expressions, 238–241
 - GROUP BY clause, 254–259
 - HAVING clause, 259–261

- LIMIT clause, 265–267
 - ORDER BY clause, 261–264
 - syntax, 230–235
 - usefulness, 229
 - variables, 241–243
 - WHERE clause, 250–254
 - reversing string order (REVERSE ()), 328**
 - revising data**
 - ASPNET/C# application, 748–750, 750–773
 - INSERT statement
 - copying data into existing table, 435–436
 - foreign key tables, 199–203
 - lookup tables, DVDRentals database, 195–198
 - lookup tables, sample, 195–198
 - one row at a time (<set option> method), 203–205
 - people tables, DVDRentals database, 198–199
 - people tables, sample, 198–199
 - REPLACE statement versus, 205
 - rows, one or more (<values option>), 191–195
 - rows, one or more (<values option> method), 191–195
 - syntax, 190–191
 - internal table statistics, 571–573
 - Java/J2EE, 692–693, 693–716
 - joins, creating multiple table data access, 393–394
 - maximum permitted per hour, 544
 - nonrepeatable reads, 473–474, 476
 - performance, optimizing, 578–579
 - phantom reads, 474–475, 476
 - PHP
 - described, 623–625
 - edit.php file, creating, 640–654
 - insert.php file, creating, 655
 - update.php file, creating, 655–656
 - REPLACE statement
 - copying data into existing table, 436–438
 - INSERT statement versus, 205
 - <set option> alternative, 207–208, 207–210, 208–210
 - syntax, 206
 - <values option> alternative, 206–208
 - restoration from binary log files
 - direct, 601–602
 - from text file, 602
 - subqueries, 410–411
 - syntax, 210
 - UPDATE statement
 - joined tables, 216–218
 - single table, 211–216
 - revoking privileges**
 - described, 553
 - DROP USER statement, 555
 - REVOKE statement, 553–555
 - sample, 555–558
 - users to other users (WITH clause), 543–544
 - rewriting values**
 - ASPNET/C# application, 731–732
 - date/time, PHP, 625
 - Java/J2EE, 675–677
 - to specific type or assigning character set to value (CAST ()), 320–322
 - right outer joins, 389–390**
 - ROLLBACK TO SAVEPOINT statement, 465–466
 - root user**
 - Linux implementation, 81
 - password, assigning, 101–102
 - Windows implementation, 82
 - ROUND () and TRUNCATE () **numeric data functions, 334–335**
 - rows**
 - copying with INSERT statement, 435–436
 - counting all groups, 354
 - DDL statements and, 23
 - grouping by column or column value, 254–255
 - inserting
 - one at a time (<set option> method), 203–205
 - one or more (<values option>), 191–195
 - locking during transaction, 456
 - normalizing data
 - first normal form, 111–112
 - second normal form, 113–114
 - third normal form, 115–116
 - relational model, 110
 - replacing with data from another table, 436–437
 - result set
 - C# processing, 729–730
 - Cartesian product, joins using, 375
 - including based on HAVING clause, 259–261
 - Java processing, 673–674
 - limiting, 265–267
 - PHP processing (while statement), 622–623, 637
 - returning all, 246, 247
 - tables, relationships between
 - many-to-many, 119, 126
 - one-to-many, 117–118
 - one-to-one, 117
 - RPM files, Linux**
 - file storage, 74–75
 - installing MySQL, 49–51
 - naming convention, 47
 - runtime, specifying server at, 503–506**
- ## S
- sample database, DVDRentals**
 - copying data into new table, 433–435
 - data manipulation (REPLACE statement)
 - <set option> alternative, 207–208
 - <values option> alternative, 207–208
 - data model
 - elements, 128
 - entities, identifying, 128–130
 - normalizing data, 130–132
 - refining, 133–136
 - relationships between tables, 132–133
 - database object, creating, 141
 - deleting data (DELETE statement)
 - from joined tables, <from join delete> alternative, 223–224
 - from joined tables, <using join delete> alternative, 225–226
 - from single table, 221

sample database, DVDRentals (continued)

- exporting data to sample out file, 428–430
 - importing data into table
 - LOAD DATA statement, 443–444
 - mysql command, 447–448
 - mysqlimport utility, 451–452
 - source command, 445–446
 - lookup tables, 195–198
 - people tables, 198–199
 - purpose, 127
 - relationships between tables, 132–133
 - sample data to dump file, 431
 - tables, managing, 160–167
 - user accounts privileges, adding and assigning (GRANT statement), 546–551
- ## savepoints
- creating (SAVEPOINT statement), 464–465
 - described, 457, 463–464
 - ROLLBACK TO SAVEPOINT statement, 465–466
 - sample using, 466–468
- ## saving transaction (COMMIT statement), 458–459
- ## scalability, 8
- ## scope columns
- columns_priv table, 523
 - db table, 520
 - defined, 518
 - host table, 521
 - tables_priv table, 522, 531–532
 - user table, 519
- ## script
- JavaScript, 629–631
 - processes, managing, 89
 - starting and stopping MySQL server, 88, 89
- ## second, extracting, 345–346
- ## secure connection
- encryption-related privilege columns, 519
 - user account, stipulating (REQUIRE clause), 541–543
- ## Secure Sockets Layer (SSL), 541, 542
- ## security
- connections, authenticating, 527–529
 - described, 8, 517–518
 - encryption functions, 358–359
 - JSP account name and password, 684
 - passwords, setting, 551–553
 - privileges
 - adding and assigning (GRANT statement), 533–544
 - dropping and revoking, 553–558
 - grant tables, 518–526
 - information, showing (SHOW GRANTS statement), 545
 - listed by table, 524–526
 - verification, 529–533
- ## SELECT statement
- analyzing specific (EXPLAIN statement), 566–571
 - comparison operators, 287–291
 - control flow functions, using in, 317–319
 - described, 24–25, 91
 - Java application connected to MySQL database, 674–673
 - joins, creating
 - basic joins, 374–377
 - described, 370–371
 - inner joins and cross joins, 377–380
 - left outer joins, 386–388
 - natural joins, 391–392
 - outer joins, 385–391
 - right outer joins, 389–390
 - sample, 381–385
 - sample tables, 372–373
 - straight joins, 380–381
 - syntax, 371
 - performance, optimizing, 574–577
 - PHP application connected to MySQL database, 35–40, 621–622
 - subqueries
 - ALL operator, 400
 - ANY and SOME operators, 399–400
 - comparison operators, 396–399
 - EXISTS and NOT EXISTS operators, 402–403
 - grouped data, 403–407
 - IN and NOT IN operators, 401–402
 - sample, 407–410
- ## semicolon (;), 19, 620, 621
- ## SEQUEL (Structured English Query Language), 11
- ## sequence, tables, 142
- ## SERIALIZABLE isolation level, 475–476
- ## server
- anonymous user access control, 82
 - configuration
 - command line, specifying at, 502–503
 - option file, specifying in, 503
 - runtime, specifying at, 503–506
 - Windows, 61–64
 - connection
 - authenticating, 527–529
 - Windows root user, 82
 - host grant tables, 521–522
 - identifying, PHP application, 37
 - master server
 - connected slave hosts, listing (SHOW SLAVE HOSTS statement), 612
 - resetting (RESET MASTER statement), 611
 - status (SHOW MASTER STATUS statement), 611–612
 - user account, setting up, 608–609
 - name, specifying, 486
 - options, listed, 88–89
 - privileges, specifying, 537
 - settings, retrieving (SHOW VARIABLES statement), 495–496
 - slave
 - CHANGE MASTER statement, 613–614
 - connected, listing (SHOW SLAVE HOSTS statement), 612
 - files, 607–608, 610
 - principles, 606–607
 - RESET SLAVE statement, 614
 - SHOW SLAVE STATUS statement, 614
 - START SLAVE statement, 614–615
 - STOP SLAVE statement, 615
 - status (SHOW STATUS statement), 498–500
 - version, displaying, 94
 - Web
 - JSP-based application, 668–669, 671–674
 - PHP-based application, 618

session status \s command, 94SET PASSWORD **statement, 551**SHA () **system encryption function, 359–360****shift left operator, bitwise (<<), 297, 299**SHOW SLAVE STATUS **statement, 614****showing**

- binary logging, 89
- connection ID, 94
- database name, 92
- database status, 94
- mysql client utility prompt (\R command), 92–93
- SHOW statements
 - database, 180–181
 - server, 495–496, 498–500, 611–612
 - table, 181–183
- values from SELECT statement, 243–245
- version number and other information, 487

signing on

- ASPNET/C# application, 727–728
- authenticating, 527–529
- database version, 361–362
- establishing, 37
- failures, 38
- ID, displaying, 94, 361–362
- Java/J2EE, 671–674
- maximum permitted per hour, 544
- PHP application, 619–621
- retrying, 487
- secure, requiring, 519
- slave server
 - altering (CHANGE MASTER statement), 613–614
 - listing (SHOW SLAVE HOSTS statement), 612
 - version, 361–362

single database, backing up entire, 588–594**single row, inserting (<set option> method), 203–205****single table**

- backing up, 594
- deleting data (DELETE statement), 220–221
- updating data (UPDATE statement), 211–216

64-bit operation, bitwise operators and, 299**slave server**

- CHANGE MASTER statement, 613–614
- connected, listing (SHOW SLAVE HOSTS statement), 612
- files, 607–608, 610
- principles, 606–607
- RESET SLAVE statement, 614
- SHOW SLAVE STATUS statement, 614
- START SLAVE statement, 614–615
- STOP SLAVE statement, 615

software

- administrative (mysql utility)
 - account passwords, assigning, 100
 - in batch mode from command prompt, 97–100
 - current database, changing (\u command), 94
 - discontinuing statements (clear \c command), 91–92
 - help command (\? or ?), 92
 - in interactive mode, 90–91
 - interactive use example, 94–96
 - launching, 96–97
 - prompt display (\R command), 92–93

- query, executing in specified file (source \. command), 93–94

- quitting (exit or quit \q), 92

- session information (status \s command), 94

- text file, saving input and output to (tee \T command), 94

ASPNET/C#

- connecting to MySQL database, 727–728
- converting values, 731–732
- creating basic, 735–748
- declaring and initializing variables, 725–727
- deleting data, 773–777
- described, 723–725

- exceptions, managing, 734–735

- files, including, 733–734

- HTML forms, 732–733

- inserting and updating data, 748–773

- redirecting browsers, 733

- retrieving data from MySQL database, 728–731

- web page, setting up, 725

- client, listed, 89–90

- data, converting for use in, 320–322

- Java/J2EE

- described, 679–682

- index.jsp file, 682–692

- options, specifying

- at command prompt, 83–85

- in configuration file, 85–88

- PHP

- basic, creating, 628–639

- call-level interface, 30

- connecting to MySQL database, 33–34, 619–621

- data, 656–660

- data, retrieving from MySQL database, 621–625

- deleting data, 660–664

- described, 618–619

- HTML forms, 626–627

- include files, 627–628

- inserting data, 639–640, 656–660

- updating data, 640–656

- Web-based applications, popularity of, 33

- server-related, listed, 88–89

Solaris, 44**sort operators, 302–306****sorting order**

- database

- changing, 142

- creating, 140

- string data

- identifying (COLLATION ()), 324–325

- performing, 303

source code MySQL distribution, 45–46**source command, data importing, 444–446****specific database backups, 595****speed, 8****speeding performance**

- deleting data, 578–579

- described, 561

- indexing, 561–565, 573–577

- inserting data, 577–578

- modifying data, 578–579

- normalizing data, 111

speeding performance (continued)

- queries, 565
- retrieving data
 - described, 566
 - SELECT statement, analyzing specific (EXPLAIN statement), 566–571
- system cache, 580–583
- tables, 571–573, 579–580

SQL

- acronym versus standalone term, 9
- described, 10–11
- execution
 - call-level interface, 30
 - direct invocation, 26–28
 - embedded SQL in programming language, 29
 - module binding, 29
- history
 - ANSI standard, 11–12
 - IBM and, 11
 - object-relational model, 13
- implementation-specific, 31–32
- nonprocedural nature, 14
- pronouncing, 10
- statements
 - access control (DCL), 25–26
 - creating, 19–23
 - database information, manipulating (DML), 24–25
 - database structure, defining (DDL), 23–24
 - syntax, 14–19

SQL Server 2000 table, 32

square brackets ([]), 15, 17

square root SQRT () numeric data function, 335

SSL (Secure Sockets Layer), 541, 542

standardizing data

- DVDRentals data model, 130–132
- first normal form, 111–113
- second normal form, 113–115
- third normal form, 115–116

starting

- server
 - automatically, 56–57
 - manually, 55–56
 - slave, 614–615
- transactions (START TRANSACTION statement), 458

statements

- access control (DCL), 25–26
- creating, 19–23
- database information, manipulating (DML), 24–25
- database structure, defining (DDL), 23–24
- executing
 - call-level interface, 30
 - direct invocation, 26–28
 - embedded SQL in programming language, 29
 - module binding, 29
- syntax, 14–19

static method, 675

status

- master server (SHOW MASTER STATUS statement), 611–612
- MySQL, installing, 64
- server version, 490

stopping

- logging to file, 94
- saving transaction (COMMIT statement), 458–459
- slave server (STOP SLAVE statement), 615
- transaction without saving (ROLLBACK statement), 459–460

stored procedures

- described, 13
- MySQL version 5, 806–807

straight joins, 380–381

string character

- first multibyte or numeric value (ORD () function), 323
- first numeric value, identifying (ASCII ()), 323

string data

- ASP.NET/C# application, 730–731
- C#
 - alias, 727
 - assessing, 730–731
 - converting, 732
- column definitions, 148–151
- functions
 - case, changing (LCASE () , LOWER () , UCASE () , and UPPER ()), 326–327
 - character set, identifying (CHARSET ()), 324
 - characters from ends, returning (LEFT () and RIGHT ()), 327
 - collation, identifying (COLLATION ()), 303, 324–325
 - concatenating (CONCAT () and CONCAT_WS ()), 325
 - encrypting and decrypting, 358–359
 - length (LENGTH ()), 324
 - multibyte value of first character, identifying (ORD ()), 323
 - number of characters (CHAR_LENGTH () and CHARACTER_LENGTH ()), 323
 - numeric value of first character, identifying (ASCII ()), 323
 - repeating specified times (REPEAT ()), 327–328
 - reversing order (REVERSE ()), 328
- indexing, 564
- Java/J2EE, 674–677
- performance, improving, 580
- PHP, 624–625, 638–639
- sort operator, 303, 304
- substring
 - finding after specified point (LOCATE ()), 326
 - locating (INSTR ()), 325–326
 - returning (SUBSTRING ()), 328–329

Structured English Query Language (SEQUEL), 11

style definition, 629, 681

sublanguage, 14

subqueries

- DELETE statement, 411–413
- SELECT statement, 396–410
- turning into joins, 573
- UPDATE statement, 410–411

substring

- finding after specified point (LOCATE ()), 326
- locating (INSTR ()), 325–326
- returning (SUBSTRING ()), 328–329

subtracting

- data, 272
- date/time (SUBDATE () or DATE_SUB ()), 340–341, 344

subtraction operator (-), 272**summarizing data**

- averaging values (`AVG()`), 350–351
- bit, 354–356
- `COUNT()`, 353–354
- described, 349–350
- example, 356–357
- minimum or maximum column value (`MIN()` and `MAX()`), 352–353
- sum of expression (`SUM()`), 351–352

system

- cache performance, optimizing, 580–583
- character sets and collations, viewing available, 140
- databases, backing up all, 595–596
- encryption functions (`ENCODE()` and `DECODE()`), 358–359
- information
 - `DESCRIBE` statements, 183–187
 - listing databases and tables, 90
 - `SHOW SLAVE STATUS` statement, 614
 - `SHOW` statements, 180–183
 - user accounts (`SHOW GRANTS` statement), 545
- operations, sample functions, 363–366
- settings, retrieving
 - dynamic system (`SELECT` statement), 497–498
 - sample, 500–502
 - server (`SHOW VARIABLES` statement), 495–496
 - server status (`SHOW STATUS` statement), 498–500

system administration

- log files
 - binary logging, setting up, 507–515
 - error log, 506
 - Linux query and binary logging, 508–512
 - query logging, setting up, 507, 508–515
 - Windows query and binary logging, 512–515
- mysql utility
 - account passwords, assigning, 100
 - batch mode from command prompt, 97–100
 - current database, changing (use `\u` command), 94
 - data folder, 78–79
 - discontinuing statements (clear `\c` command), 91–92
 - help command (`\?` or `?`), 92
 - in interactive mode, 90–91
 - interactive use example, 94–96
 - launching, 96–97
 - prompt display (`\R` command), 92–93
 - query, executing in specified file (source `\.` command), 93–94
 - quitting (exit or quit `\q`), 92
 - session information (status `\s` command), 94
 - text file, saving input and output to (tee `\T` command), 94
- mysqladmin utility, 89, 488–494
- options, 486–487
- privilege columns, 519
- server configuration
 - command line, specifying at, 502–503
 - option file, specifying in, 503
 - runtime, specifying at, 503–506
- system variables
 - configuring, 494–495
 - settings, retrieving, 495–502
- utilities, 485–486

System R, 11**T****tables**

- backing up all, 588–594
 - column default values, 153–154
 - column definitions
 - date/time data types, 151–152
 - null values, ability to accept, 152–153
 - numeric data types, 145–148
 - string data types, 148–151
 - syntax, 144–145
 - copying data into, 431
 - creating
 - DDL statements, 20–23
 - implementation-specific SQL, 31
 - syntax, 143–144, 160–167
 - definitions, modifying, 167–169
 - defragmenting, 571–573
 - deleting from database, 169–171
 - details, viewing, 567
 - foreign keys, 156–158
 - indexes
 - adding to existing, 175–178
 - defining when creating, 172–175
 - individual, backing up, 594
 - information, retrieving, 181–183
 - inserting data
 - auto-increment columns, 155–156
 - PHP, 639–640
 - internal statistics, updating, 571–573
 - junction, for many-to-many relationships, 126–127
 - options, 158–160
 - performance, optimizing, 579–580
 - primary keys, 154–155
 - privileges, limiting to, 536
 - relationships between
 - defined, 116
 - identifying, 125–126, 132–133
 - many-to-many, 119, 126
 - model, 110
 - one-to-many, 117–118
 - one-to-one, 117
 - sequence, 142
 - user account privileges, specifying (`ON` clause), 536–537
- tables_priv grant table, 522–523, 531–532**
- taking away privileges**
- described, 553
 - `DROP USER` statement, 555
 - `REVOKE` statement, 553–555
 - sample, 555–558
 - users to other users (`WITH` clause), 543–544
- tar files, Linux**
- file storage, 75–76
 - installing MySQL
 - copying, 52–53
 - unpacking distribution file, 53–54
 - user and group accounts, creating, 51–52
- Tcl API, 803**
- temporary tables**
- creating, 144
 - deleting, 169
 - query results, placing, 246

tenary relationship, 136

testing

- Linux installation, 65–68
- Windows installation, 68–70

text data type, 150

text file

- data, copying into, 90
- delimited
 - creating, 422
 - importing data from, 438–444, 449–452
- saving input and output to (tee `\T` command), 94
- updating from binary log files, 602

text index

- ALTER TABLE statement, 176–177
- CREATE INDEX statement, 177–178
- described, 174–175, 565

three periods (...), 15

time out, setting server, 486

time parts, functions extracting (SECOND(), MINUTE(), HOUR(), and TIME()), 345–346

time/date data

- adding (ADDDATE() and DATE_ADD()), 339–340
- column definitions, 151–152
- converting
 - format, 320
 - Java, 676, 711
 - PHP from MySQL database, 625, 638
- current, retrieving (CURDATE(), CURRENT_DATE(), CURTIME(), CURRENT_TIME(), CURRENT_TIMESTAMP(), and NOW()), 341–342
- day-related values (DAY(), DAYOFMONTH(), DAYNAME(), DAYOFWEEK(), and DAYOFYEAR()), 344–345
- differences between (DATEDIFF() and TIMEDIFF()), 344
- month, extracting, 342–343
- specific part, extracting (EXTRACT()), 341
- subtracting (SUBDATE() or DATE_SUB()), 340–341
- time parts, extracting (SECOND(), MINUTE(), HOUR(), and TIME()), 345–346
- year, extracting, 342–343

timestamp, retrieving (CURRENT_TIMESTAMP()), 341–342

transactions

- autocommit mode, 468–472
- described, 455–457
- isolation levels
 - data anomalies, 472–475
 - described, 472
 - listed, 475–476
 - setting (SET TRANSACTION statement), 476–479
- nontransactional tables, locking
 - described, 479
 - example, 481–483
 - LOCK TABLES statement, 480
 - UNLOCK TABLES statement, 480–481
- performance level, improving, 578
- performing
 - automatically committed statements, listed, 460
 - creating, 460–462
 - described, 457–458
 - START TRANSACTION statement, 458

- terminating transaction with saves (COMMIT statement), 458–459

- terminating transaction without saves (ROLLBACK statement), 459–460

savepoints

- creating (SAVEPOINT statement), 464–465
- described, 463–464
- ROLLBACK TO SAVEPOINT statement, 465–466
- sample using, 466–468

triggers, 807–808

true/false, 291–296

try/catch block, 679, 734, 739, 744

two colons, equal sign (::=), 15

types, data

- column definitions, 18, 145–152
- length, specifying correct, 579–580
- values, converting, 320–322

U

unary operator (-), 272

unchanging tables, locking

- described, 479
- example, 481–483
- LOCK TABLES statement, 480
- UNLOCK TABLES statement, 480–481

underscore (_), 284

Unicode data type, 149

Uniform Resource Locator (URL), 627

unions joining SELECT statements, 414–417

unique index

- ALTER TABLE statement, 176–177
- CREATE INDEX statement, 177–178
- described, 173, 565

Unix-like operating systems

- backup, quick, 90
- bug reports, generating, 89
- choosing, 45
- names, case-sensitivity of, 141
- script, starting and stopping server, 88
- UNLOCK TABLES statement, 480–481

unpacking tar file, 53–54

updating data

- ASPNET/C# application, 748–750, 750–773
- INSERT statement
 - copying data into existing table, 435–436
 - foreign key tables, 199–203
 - lookup tables, DVDRentals database, 195–198
 - one row at a time (<set option> method), 203–205
 - people tables, DVDRentals database, 198–199
 - REPLACE statement versus, 205
 - rows, one or more (<values option>), 191–195
 - syntax, 190–191
- internal table statistics, 571–573
- Java/J2EE, 692–693, 693–716
- joins, creating multiple table data access, 393–394
- maximum permitted per hour, 544
- nonrepeatable reads, 473–474, 476
- phantom reads, 474–475, 476

PHP

- edit.php file, creating, 640–654
- insert.php file, creating, 655
- update.php file, creating, 655–656

REPLACE statement

- copying data into existing table, 436–438
- INSERT statement versus, 205
- <set option> alternative, 207–210
- syntax, 206
- <values option> alternative, 206–208

restoration from binary log files

- direct, 601–602
- from text file, 602
- subqueries, 410–411
- syntax, 210

UPDATE statement

- joined tables, 216–218
- single table, 211–216

uppercase

- GRANT statement scope columns, 534
- name conversion issues, 141
- string data, changing (LCASE(), LOWER(), UCASE(), and UPPER()), 326–327

URL (Uniform Resource Locator), 627**user accounts**

- adding and assigning privileges (GRANT statement)
 - granting or revoking to others (WITH clause), 543–544
 - privileges (GRANT clause), 534–536
 - sample, 537–541, 546–551
 - secure connection, stipulating (REQUIRE clause), 541–543
 - syntax, 533–534
 - tables or databases, specifying (ON clause), 536–537
 - to users (TO clause), 537
- current status, displaying, 94
- dropping and revoking privileges
 - described, 553
 - DROP USER statement, 555
 - REVOKE statement, 553–555
 - sample, 555–558
- identifying, PHP application, 37
- passwords, setting
 - administrative (mysql utility), 100
 - changing, 552–553
 - configuration file, 103–106
 - grant tables, reloading, 100–101, 551–552
 - root account, 101–102
 - SET PASSWORD statement, 551
- server connection, specifying, 487
- tar files on Linux, 51–52
- viewing privileges (SHOW GRANTS statement), 544–545

user table

- directory structure, 81
- dropping account from, 553–555
- encryption functions, 360
- privileges, verifying, 518–520, 529
- showing, 545

user-defined functions, 310**user-name and hostname, current session**

- (CURRENT_USER(), SESSION_USER(), SYSTEM_USER(), and USER()) functions, 361

USING clause, cross joins, 378–379**utility, mysql client**

- account passwords, assigning, 100
- batch mode, 97–100, 600
- current database, changing (use \u command), 94
- described, 22–23
- direct invocation, 27
- discontinuing statements (clear \c command), 91–92
- help command (\? or ?), 92
- importing data into table
 - LOAD DATA statement, 438–444
 - mysql command, 446–448
 - source command, 444–446
- interactive mode
 - described, 90–91
 - example, 94–96
 - reloading database, 600–601
- launching, 95, 96–97
- programs, 97–100
- prompt display (\R command), 92–93
- query, executing in specified file (source \. command), 93–94
- quitting (exit or quit \q), 92
- SELECT statement, 25
- session information (status \s command), 94
- text file, saving input and output to (tee \T command), 94

V**values**

- arguments, 37
- data, 18
- Java variables, initializing, 670
- multiple clauses, using, 578
- range of, specifying with operator, 283–284

variables

- defined, 37
- retrieving data (SELECT statement), 241–243

verifying privileges, 529–533**version**

- installing MySQL, 44
- number and other information, displaying, 487

vertical bar (|), 15, 18, 601–602**vertical pipes, double (||), 292****views, MySQL version 5, 808–809****W****Web applications, 32–33****Web browser**

- mechanics, 618
- redirecting, 627, 678, 733

Web page

- forms
 - ASP.NET/C# application, 732–733
 - Java/J2EE, 677–678, 707–708, 744
 - PHP application, creating, 628–629, 644–647
 - PHP rendering, 33–34, 618, 626–627
 - tags, 36
 - setting up, 725

Web server

Web server

- JSP-based application, 668–669, 671–672
- PHP-based application, 618

Web site, MySQL, 45

week, day of, 344–345

WHERE clause

- EXISTS and NOT EXISTS operators, 402–403
- joined tables, limiting, 375–376
- retrieving data (SELECT statement), 250–254, 287–291
- subqueries, adding, 396–399

wildcard character, 180, 527, 573

Windows (Microsoft)

- access control, 82
- anonymous users, removing, 556
- benefits of using, 48
- configuration file, 86–88, 103, 105–106
- debug edition, 46
- editions, 64–65
- file storage, 76
- installation program, 47, 57–61
- MySQL options, 44, 94
- names, case-insensitivity of, 141
- query and binary logging, 512–515

- root account password, setting, 101
- server, configuring, 61–64
- testing installation, 68–70

WinZip, 57

X

x509 certificate, 542

XOR operator

- bit aggregate function, 355–356
- bitwise, 297, 301
- described, 292, 293

Y

year, day of, 344–345

Z

zero data type, 145

.zip file, installing MySQL from, 57–61