

INDEX

[] (square brackets) in regular expressions, 50

A

abstract keyword, 3
abstract methods, 353
access permissions, views, 223–224
aCFB encryption, 504
ACID, 238
 atomicity
 non-transactional tables, 238–239
 transactional tables, 239–240
 consistency, 240–241
 durability, 247–250
 foreign keys, 38
 isolation, 242
 levels, 247
 READ_COMMITTED, 244–245
 READ_UNCOMMITTED, 245–246
 REPEATABLE_READ, 242–244
 replication, 247
 SERIALIZABLE, 246–247
 relational integrity, 35
ACLs (Access Control Lists), 473–474
Aksyonoff, Andrew, 369
algorithms, UDFs, 277
aliases, MySQL, 22
ALTER ROUTINE privilege, 230
alternation in regular expressions, 51
AND operator (MySQL), 33
anonymous functions, 78
ANSII control codes, 523
APC (Alternative PHP Cache), 132
 configuration, 132–133
 installation, 132–133
 serialization, 135
 when to use, 138
appending, memcached, 166–167
application sample view, 67–69

Archive storage engine, 109–110
args key, 544
arguments
 checking for string, 285
 functions, in extensions, 331–334
 type, checking, 274
 UDFs, 192
ArrayAccess interface, 77–78
arrays
 accessing, 347–349
 associative, 70, 348
 building, 347–349
 tracking unlinks, 77
associative arrays, 70, 348
atomicity (ACID)
 non-transactional tables, 238–239
 transactional tables, 239–240
auditing, 500–501
authorization, memcached, 140
autoconf, 292
 macros, 292
automation attacks, 514–515

B

b-tree indexes, 365
back-references in regular expressions, 53
basic authentication, 461–463
Beanstalk, 442
bi-directional encoding
 mycrypt extension, 503–505
 XOR Cipher, 501–503
binary, definition, 327
binary logging, 500
 file analysis, 256–257
 replication and, 251
bindings (SOAP), 537–538
bison, 308
Blackhole storage engine, 108–109

blocking sockets, 346
 BlogEntry class, polymorphism, 2
 Boolean query syntax, 385
 boundary characters in regular expressions, 49
 breaking replication, 261–262
 breakpoints in debugging, 302–303
 built-in functions, 335–336
 built-in rewrite maps, 451
 bytecode caching, 132

C

C

C++ and, 308
 UDFs, 273
 Zend engine, 308
 C++
 C and, 308
 UDFs, 273
 C-libraries, UDFs and, 272
 cache replication, 145
 caching, 180–188
 bytecode, 132
 cache locality, 171–173
 cache replication, 145
 cache status, checking, 135–138
 catalogue table, 142
 deterministic, 144
 file system, 145
 introduction, 131
 LRU (Least Recently Used), 140
 memcached, 138–140
 authorization, 140
 consistent hashing, 140
 replication, 140
 non-deterministic, 144
 opcode, 132
 page, 145
 partial page, 145
 proactive, 145
 read-through, 141, 216–219
 session, 145
 state, 145
 strategies, 144–145
 user, enabling, 133
 write-back, 142
 write-through caching, 141
`_call()` magic method, 11
 callback functions, lambda functions, 78

callback parameter, 81
 callbacks, read-through caching callback, 165
 calling methods, 356
`_callStatic()` magic method, 11
 CAPTCHA (Complete Automated Public Turing test to tell Computers and Humans Apart), 514
 cardinality, 550–551
 CAS (check and set), 174–176, 201–202
 case sensitivity, joins (MySQL), 26–27
 catalogue table, 142
 CBC (Cipher Block Chaining) encryption, 504
 CentOS, memcached installation, 145–146
 CFB (Cipher-Feedback Mode) encryption, 504
 character classes (MySQL), 60
 character sets, 486–487
 collations, 487–488
 ciphers, 502
 C10K problem, 132
 class entry, 350
`class` key, 544
`_CLASS_` magic constant, 8
 classes
 CodeGen_Pecl, example, 320–321
 command-line scripts, 521–522
 constants, 354
 definition, 350–351
 EmailValidator, 56–57
 FileLog, 9–10
 ForumTopic, 4
 inheritance, 350–351
 interfaces
 implementing, 7
 inheritance, 7
 LoggingProxy, 13–14
 methods, magic, 9–11
 MultitonExample, 12–13
 namespaces, 350–351
 Node, defining, 3–4
 Pagination, 75–76
 SingletonExample, 11–12
 variables, 3
 CLI (Command Line Interface), 312
 command-line scripts, 517
 clients
 memcached, 153
 PECL/memcached, 162–163
`_clone()` magic method, 11
`clone` keyword, 12
 closures, 81–83
 code, critical section, 360

- CodeGen_PECL, 314
 - classes, example, 320–321
 - constants, defining, 318
 - dependencies, 338
 - extensions, as XML documents, 314
 - functions, defining, 316–317
 - globals, 318–319
 - INI directives, 318–319
 - interfaces, 321
 - methods, defining, 320
 - objects, defining, 319
 - properties, defining, 320
 - XML files, creating, 314–316
- Cole, Jeremy, 483
- collations of character sets, 487–488
- colored text, 523–524
- column privileges, 490–491
- command-line scripts
 - classes, 521–522
 - CLI, 517
 - color, 523–524
 - creating, 517–524
 - cron jobs, 522–523
 - input
 - prompting for, 520–521
 - reading, 518–520
 - shebang, 518
- commands
 - CREATE USER, 496
 - DROP USER, 497–498
 - GRANT, 268, 497
 - mysql_secure_installation, 494
 - replication, 261
 - REVOKE, 268, 497
 - SHOW, 491–492
 - SHOW TABLE STATUS, 92
 - SHOW. ENGINES, 89–90
- comments
 - functions, 330
 - regular expressions, 49, 54
- COMMIT command, implied, 250
- complex joins (MySQL), 27
- complex subqueries (MySQL), 47–49
- compressed tables, MyISAM storage engine, 96
- concrete decorators, 16
- CONDITION clause, 229–230
- conditional rewrite rules, 447–450
 - exclamation point, 448
 - testing for patterns, 449
 - testing to check file or path, 449
- conditions, 229–230
- config.m4 file, 310
- config.w32 file, 310
- consistency (ACID), 240–241
- consistent hashing, 140
- constants, 354
 - CodeGen_PECL, defining, 318
 - magic, 8–9
 - MINIT function, 357
- constraints, 35
 - NOT NULL, 35–36
 - UNIQUE KEY, 37
 - UNSIGNED, 36
- constructors, copy constructors, 329
- contexts, 343–344
- cookies, 471–473
- copy constructor, 329
- correlated subqueries (MySQL), 46
- count () method, 73
- countable interface, 73–74
- CREATE ROUTINE privilege, 230
- CREATE USER command, 496
- CREATE VIEW privilege, 223
- CREDITS file, 310
- critical section, 360
- cron jobs, 522–523
- crontab, 522
- CSRF attack, 511–514
- CSV (Comma Separated Value), 112
- CSV storage engine, 112
- CTR (Counter Mode) encryption, 504
- cURL, 528–529
- current () method, 72
- cursors, 229
- .cvsignore file, 310

D

- \d shorthand character, 50
- Danga Interactive, 139
- data capture
 - MySQL, 61
 - regular expressions, 53–54
- data persistence, Memory storage engine, 103
- data sources, Sphinx, 376–379
- data storage, disk, 275
- data types
 - ENUM, 36–37
 - function definitions file, 312–313
 - SET, 36–37

- data warehousing
 - InfiniDB, 124
 - Infobright, 125
 - views, 225
- databases, user authentication, 458
- Day, Eric, 412
- db_bytes() function, 283, 287
- db_bytes_deinit() function, 284
- db_bytes_init() function, 284
- DealNews, 402
- debug back-trace, 544
- debuggers
 - gdb, 273
 - attaching to running process, 301–302
 - Microsoft debugger, 273
 - MySQL and, 564–565
 - UDFs, 273
- debugging, 543
 - breakpoints, 302–303
 - debug back-trace, 544
 - literal values and, 304
 - MySQL
 - commands, 564
 - error log, 560–561
 - PROCESSLIST, 562–564
 - slow query log, 561–562
 - stack traces, creating, 544–545
 - stepping through code, 302–303
 - UDFs, 300–301
- DECLARE clause, 229
- decorating objects, 14
- decorator pattern, 14–16
- decrementing, memcached, 168–169
- default users, new installation, 494
- defining, interfaces, 7
- delete() method, 168
- DELETE command, triggers, 235–236
- DELETE statement, 556–558
- delta indexes, 389–394
- dependencies, CodeGen_Pecl, 338
- dependent queries, 45
- derived tables (MySQL), 46–47
- deriving interfaces, 7
- design patterns
 - decorator, 14–16
 - façade, 13–14
 - multiton, 11–13
 - observer, 18–19
 - proxy, 13–14
 - publisher/subscriber, 18–19
 - singleton, 11–13
- destructors, 5
- deterministic cache, 144
- Digest Authentication, 507
- digest authentication, 463–466
- Digg, 402
- _DIR_ magic constant, 9
- directives, rewrite rules, 444
 - RewriteCond, 447
- directories
 - ext, 308
 - main, 308
 - project directory organization, 281–283
 - Zend, 308
- disk space, InnoDB storage engine, 99–100
- disk storage, 275
- distributed indexes, 373
- DML statements, Blackhole storage engine, 108
- drivers, MySQL, 88
- Drizzle, 124, 126
- DROP USER command, 497–498
- dumb server, 140
- durability (ACID), 247–250

E

- eAccelerator, 132
 - configuration, 133–134
 - installation, 133–134
 - opcode, optimizer, 134
 - serialization, 135
 - when to use, 138
- eBay Memory storage engine, 127
- ECB (Electronic Code Book) encryption, 503
- elements, unsetting, 77
- EmailValidator class, 56–57
- encoding
 - bi-directional
 - mycrypt extension, 503–505
 - XOR Cipher, 501–503
 - single-directional, 505–506
- encryption
 - aCFB, 504
 - CBC, 504
 - CFB, 504
 - CTR, 504

- ECB, 503
- mycrypt extension, 503–505
- nOFB, 504
- OFB, 504
- STREAM, 503
- ENFORCE_SAFE_MODE flag, 342
- engines, Zend, 308
- ENUM data type, 36–37
- envelopes, SOAP requests, 532
- error handling, memcached, 178–179
- error log, debugging MySQL, 560–561
- errors, stack overflow, 544
- errors/warnings, 346–347
 - values, 346–347
- escaping quotes, 80
- events
 - altering, 267–268
 - creating, 265–266
 - cron and, 264
 - disabling, 267–268
 - dropping, 267–268
 - event scheduler, enabling, 267
 - meta data, 269–270
 - privileges, 268
 - re-enabling, 268
- eventual consistency, 402
- excerpts, 395
- expert regular expressions, 52–54
- EXPERIMENTAL file, 310
- EXPLAIN statement, 547–549
- EXPLAIN syntax, 102
- expressions. *See* regular expressions
- ext directory, 308
- extensibility, Federated storage engine, 113
- extensions, 307
 - CodeGen_PECL, 314
 - as XML documents, 314
 - XML files, 314–316
- ext_skel, 309
- functions in
 - arguments, 331–334
 - built-in functions, 335–336
 - comments, 330
 - definitions, 329–331
 - helper functions, 339–340
 - returning values, 334–335
 - standard output, 340
- printing, 361
- source code, 308
- UNIX/Linux, 308

- variables in, 321–329
 - zval, 321–328
- ext_skel, 309
 - function definitions file, 309

F

- façade design pattern, 13–14
- factory methods, 17
- Falcon storage engine
 - overview, 118
 - parameters, 119
 - tables, 119–120
- Federated storage engine
 - characteristics, 113–114
 - Federated Server, 116–117
 - files, 117
 - read operations, 112
 - standards, 113
 - table creation, 114–115
 - write statements, 112
- feet_to_meters() function, 274
- feet_to_meters_deinit() function, 274
- feet_to_meters_init() function, 274
- fetch method, 134
- fetch worker (Gearman), 430–433
- fetchall_hash_result() method, 299
- file key, 544
- _FILE_ magic constant, 8
- file system caching, 145
- FileLog class, 9–10
- files
 - config.m4, 310
 - config.w32, 310
 - CREDITS, 310
 - .cvsignore, 310
 - EXPERIMENTAL, 310
 - help, generating, 313
 - LICENSE, 316
- final keyword, 3
- Fitzpatrick, Brad, 139, 401
- flags in streams, 341–342
- flex, 308
- flow control (MySQL), 34–35, 230
 - CASE function, 34
 - IF() function, 34
 - IFNULL() function, 35
 - NULLIF() function, 35
- foreign keys, 37–40

ForumTopic class, 4

- polymorphism, 2

fstat() function, 275

FULLTEXT indexes, 365

- Boolean mode, 367–368
- creating, 366
- issues, 368–369
- Natural Language mode, 367
- searches, 366
 - modifier values, 366

function definitions file

- ext_skel, 309
- using, 312–313

function key, 544

Function magic constant, 9

functions

- anonymous functions, 78
- closures, 81–83
- CodeGen_PECL, defining, 316–317
- creating dynamically, 79
- db_bytes(), 283, 287
- db_bytes_deinit(), 284
- db_bytes_init(), 284
- extensions
 - arguments, 331–334
 - built-in functions, 335–336
 - comments, 330
 - definitions, 329–331
 - helper functions, 339–340
 - returning values, 334–335
- feet_to_meters(), 274
- feet_to_meters_deinit(), 274
- feet_to_meters_init(), 274
- fstat(), 275
- hashing functions, 506
- helper functions, 339–340
- init(), 285
- lambda functions, 65, 78
- lambda-style, 78–81
- lstat(), 290
- MATCH(), 366
- memcached, 191–192
- MINIT, 350, 357
- PHP API functions, consuming, 336–338
- php_printf, 340, 361
- serialize(), 9
- setToCache(), 184
- sprintf(), 290
- standard output, 340
- stat(), 275

- stored, 226
- stream functions, 341–343
- thread-safe, 273
- user defined (*See* UDFs (user defined functions))
- zend_parse_parameters, 334

G

gcc, 308

gdb debugger, 273

- running process, attaching to, 301–302

GDB (GNU Debugger), 564

Gearman, 142, 144

- DealNews, 402
- Digg, 402
- eventual consistency, 402
- Gearman MySQL UDFs, 407–409
 - gman_do(), 408
 - gman_servers_set(), 407–408
 - gman_sum(), 408–409
- gearmand
 - installation, 404
 - running, 406–407
- installation
 - Gearman MySQL UDFs, 405–406
 - gearmand, 404
 - job server, 404
 - PECL/Gearman, 404–405
- job server, running, 406–407
- language independence, 402
- Livejournal and, 401, 402
- map/reduce operations, 403
- Narada and, 412
- overview, 402–404
- PHP and, 409–412
- SixApart, 402
- workers
 - fetch, 430–433
 - index, 434–435
 - insert, 429–430
 - running, 440
 - search, 433–434
- Xing.com, 402
- Yahoo!, 402

Gearman MySQL UDFs, 272

- gman_do(), 408
- gman_servers_set(), 407–408
- gman_sum(), 408–409
- installation, 405–406

GearmanClient object, 410

GearmanWorker object, 410
`_get()` magic method, 10
`get` method, 134
`get_class()` method, 6
`getDelayedByKey()` method, 173–174
`get_fields()` method, 299
`getFromCache()` method, 183
`getMulti()` method, 169–170
`getServers()` method, 178
 globals, 357

- accessing, 359
- `CodeGen_Pecl`, defining, 318–319
- critical section, 360
- defining, 358–359
- thread safety, 359–361
- True Global Resources, 360

`gman_do()` UDF, 408
`gman_servers_set()` UDF, 407–408
`gman_sum()` UDF, 408–409
 GNU General Public License, 133
 GPL (General Public License), 316
 GRANT command, 268, 497
 GROUP BY (MySQL), 30–31
 GUID (Globally Unique Identifier), 363

H

Hadoop, 442
 HANDLER clause, 229–230
 handlers, 229–230
 hash tables

- accessing, 348, 349
- looping through, 349
- modifying, 349

 hashing, 505–506
 hashing functions, 506
`HashTable` data type, 348
 HAVING clause (MySQL), 32
 help files, generating, 313
 helper functions, 339–340
 host permissions, 499
 HTTP-based user authentication, 460

- basic, 460, 461–463
- digest, 460, 463–466

 human readability, 222

I

IGNORE_PATH flag, 341
 IGNORE_URL flag, 342

implementation, interfaces, 7
 implied COMMIT, 250
 incrementing, memcached, 168–169
 index workers (Gearman), 434–435
 indexes

- common mistakes, 552–554
- identifying, 549–550
- optimizing, 552
- types, 551

 indexing

- attributes, 376
- b-tree indexes, 365
- columns, 376
- delta indexes, 389–394
- distributed indexes, 373
- fields, 376
- FULLTEXT indexes, 365
 - Boolean mode, 367–368
 - creating, 366
 - issues, 368–369
 - Natural Language mode, 367
- inherited, Sphinx, 381
- merging indexes, 394–395
- searches, 365–366
- Sphinx, 372–373
 - Narada, 416–417
 - TokuDB, 124–125

 InfiniDB, 124
 Infobright, 125
 INFORMATION_SCHEMA. *See* I_S
 INFORMATION_SCHEMA schema, 289–292
 inheritance, 350–351

- classes inheriting interfaces, 7

 inherited indexes, 381
 INI directives, 357

- `CodeGen_Pecl`, defining, 318–319
- defining, 358–359

`init()` function, 285
 initialization, lazy initialization, 12
 inner joins (MySQL), 21–23

- ON, 22
- syntax, 22–23
- USING, 22

 InnoDB storage engine, 97–98

- disk space, 99–100
- MyISAM comparison, 99–102
- parameters, 98
- primary key order, 100
- SQL optimization, 102–103
- tables, 98–99

INSERT command, triggers, 233–234
 INSERT statement, 558
 triggers, 233
 insert worker (Gearman), 429–430
 INSERT...ON DUPLICATE KEY UPDATE statement,
 558–560
 installation
 APC, 132–133
 eAccelerator, 133–134
 Gearman, 404–407
 libmemcached, 155
 memcached
 CentOS, 145–146
 OpenSolaris, 147
 from source, 147–148
 testing, 151–152
 Ubuntu, 146–147
 Memcached Functions for MySQL, 193–196
 Sphinx, 370–371
 UDFs, 294–295
 checking, 289–292
 instanceof keyword, 6
 instances, memcached, 160
 instantiation, 2–6
 interfaces, 7
 integer values, incrementing/decrementing, 204–206
 integration points for triggers, 232
 interface keyword, 7
 interfaces, 7–8
 ArrayAccess, 77–78
 CodeGen_Pecl, 321
 countable, 73–74
 creating, 354–355
 defining, 7
 deriving, 7
 implementing, 7
 inheritance, 7
 instantiation, 7
 Iterator interface, 69–73
 methods, 70
 ReadableNode, 7
 SeekableIterator, 75–76
 testing variables against, 7
 internal maps, 451
 _invoke() magic method, 11
 I_S
 character sets, 486–487
 collations, 487–488
 column privileges, 490–491

 extensions, 491
 metadata, 486
 MySQL
 status variables, 484–485
 system variables, 485–486
 privilege tables, 488
 profiling table columns, 483–484
 schema privileges, 489
 SHOW command, 478
 SHOW_PROCESSLIST, 482
 status tables, 482
 table objects, 479, 481–482
 table privileges, 489–490
 tables, 478, 479–480
 user privileges, 488
 isolation (ACID), 242
 levels, 247
 READ_COMMITTED, 244–245
 READ_UNCOMMITTED, 245–246
 REPEATABLE_READ, 242–244
 replication, 247
 SERIALIZABLE, 246–247
 _isset() magic method, 10
 isValidCipher() method, 519
 Iterator interface, 69–73
 IteratorIterator, 75
 iterators, 67
 actions, 70
 IteratorIterator, 75
 methods, 71
 rewinding, 71

J

job servers, Gearman, 142
 joins (MySQL), 19–21
 aliases, 22
 case sensitivity, 26–27
 complex, 27
 inner joins, 21–23
 ON, 22
 syntax, 22–23
 USING, 22
 left joins, 25
 multiple tables, 20
 outer joins, 23–24
 right joins, 24–25
 syntax, 25–27

K

Kasindorf, Alan, 140
 key() method, 72, 77
 keywords
 abstract, 3
 clone, 12
 final, 3
 instanceof, 6
 interface, 7
 parent, 2
 private, 3
 protected, 3
 public, 3
 WITH ROLLUP, 31–32
 static, 3, 12
 Kickfire storage engine, 128

L

lambda functions, 65, 78
 lambda-style functions, 78–81
 lazy initialization, 12
 lazy processing, 143
 left joins (MySQL), 25
 libgearman, UDFs, 272
 libmemcached, 153–154
 features, 154
 installation, 155
 Memcached Functions for MySQL, 193
 UDFs, 272
 utility programs, 154
 memcat, 155–156
 memcp, 156
 memerror, 158
 memflush, 156
 memrm, 157
 memslap, 157–158
 memstat, 157
 libmysql, 88
 LIB_MYSQLUDF_PREG library, 61
 libraries
 interfacing PHP with MySQL, 88
 LIB_MYSQLUDF_PREG, 61
 MySQL Stored Routines Library, 231
 library dependencies, modules, compiling, 311
 LICENSE file, 316
 line key, 544
 LINE magic constant, 8

literal values, 304
 Livejournal, 402
 Gearman and, 401
 load() method, 182
 login, rewrite rules, 450
 LoggingProxy class, 13–14
 logic operators (MySQL), 33–34
 logos, 362–363
 lookaheads, 52–53
 lookbehinds, 52–53
 lookups
 random, 451–452
 text, 451–452
 loops, 65–67
 hash tables, 349
 iterators, 67
 LRU (Least Recently Used) cache, 140
 lstat() function, 290

M

Maesaka, Turo, 140
 magic constants, 8–9
 CLASS, 8
 DIR, 9
 FILE, 8
 Function, 9
 LINE, 8
 METHOD, 9
 NAMESPACE, 9
 magic methods, 8
 _call(), 11
 _callStatic(), 11
 classes, 9–11
 _clone(), 11
 _get(), 10
 _invoke(), 11
 _isset(), 10
 _set(), 10
 _set_state(), 11
 _sleep(), 9
 _unset(), 10
 _wakeup(), 9
 main directory, 308
 map/reduce operations, 403
 maps
 internal, 451
 rewrite maps, 450–455
 built-in, 451

- Maria storage engine, 120–121
- MariaDB, 126
- `MATCH()` function, 366
- materialized view, 225
- mcrypt extension, 503–505
- mcrypt library, white space, 505
- md5 hash, decrypting, 459
- member variables
 - private, serialization, 10
 - visibilities, 3
- memcached, 138–140. *See also* PECL/memcached
 - add method, 163
 - appending, 166–167
 - authorization, 140
 - cache locality, 171–173
 - CAS (check and swap), 174–176
 - class methods, 163–166
 - clients, 153
 - consistent hashing, 140
 - decrementing, 168–169
 - `delete()` method, 168
 - dumb server, 140
 - error handling, 178–179
 - functions, 191–192
 - get method, 164–165
 - `getDelayedByKey()` method, 173–174
 - `getMulti()` method, 169–170
 - `getServers()` method, 178
 - `-h` switch, 148
 - incrementing, 168–169
 - installation
 - CentOS, 145–146
 - OpenSolaris, 147
 - from source, 147–148
 - testing, 151–152
 - Ubuntu, 146–147
 - instances, 140, 160
 - mailing list, 140
 - memcapable key-value stores, 186
 - moxi proxy, 185–186
 - multi-get, 169–170
 - multi-set, 170–171
 - Narada and, 412
 - prepending, 166–167
 - read-through caching, 141
 - replace method, 164
 - replication, 140
 - servers
 - connecting to, 197–198
 - connection, 160–161
 - set method, 164
 - starting, 148–149
 - startup scripts, 149
 - Debian-based, 150
 - OpenSolaris, 151
 - Redhat-based, 150–151
 - statistics, 176–178
 - Tokyo Tyrant, 187–188
 - user contract, 140
 - using, 141–144
 - values, storing, 198–199
 - write-back caching, 142
 - write-through caching, 141
- Memcached Functions for MySQL, 191–192
 - behavioral functions
 - `memc_behavior_get`, 207
 - `memc_behavior_set`, 207–208
 - `memc_list_behaviors`, 206–207
 - `memc_list_distribution_types`, 209
 - `memc_list_hash_types`, 209
 - data fetching functions, 204
 - data setting functions
 - `memc_add`, 200
 - `memc_append`, 202–203
 - `memc_cas`, 201
 - `memc_delete`, 203
 - `memc_prepend`, 202
 - `memc_replace`, 200–201
 - `memc_set`, 198–199
 - `memc_set_by_key`, 199–200
 - incrementing/decrementing
 - `memc_decrement`, 205–206
 - `memc_increment`, 204–205
 - installation
 - building source, 194
 - configuring source, 194
 - func table, 196
 - installing UDF, 195
 - libmemcached, 196–197
 - Perl install utility, 195
 - source distribution directory, 212–214
 - statistical functions, 209–210
 - `memc_stat_get_keys`, 211
 - `memc_stat_get_value`, 211
 - `memc_stats`, 210–211

- triggers, 212
 - execution, 214–216
- values, 272
- version functions, 211
 - memc_libmemcached_version, 212
 - memc_udf_version, 212
- memc_add function, 200
- memc_append function, 202–203
- memcat, 155–156
- memc_behavior_get function, 207
- memc_behavior_set function, 207–208
- memc_cas function, 201
- memc_decrement function, 205–206
- memc_delete function, 203
- memc_get function, 204
- memc_increment function, 204–205
- memc_libmemcached_version function, 212
- memc_list_behaviors function, 206–207
- memc_list_distribution_types function, 209
- memc_list_hash_types function, 209
- memcp, 156
- memc_prepend function, 202
- memc_replace function, 200–201
- memc_server_count, 198
- memc_servers_set, 197–198
- memc_set function, 198–199
- memc_set_by_key function, 199–200
- memc_stat_get_keys function, 211
- memc_stat_get_value function, 211
- memc_stats function, 210–211
- memc_udf_version function, 212
- memerror, 158
- memflush, 156
- memory management, 328–329
- Memory storage engine
 - data persistence, 103
 - parameters, 103
 - table-level locking, 104
 - table size maximum, 104
 - table usage, 103–107
- memrm, 157
- memslap, 157–158
- memstat, 157
- Merge storage engine, 110–111
- merging indexes, 394–395
- metadata, 477
 - events, 269–270
 - I_S, 486
 - stored routines, 230–231
- _METHOD_ magic constant, 9
- methods, 351–353
 - abstract, defining, 353
 - calling, 356
 - CodeGen_PECL, defining, 320
 - count(), 73
 - current(), 72
 - factory methods, 17
 - fetch, 134
 - fetchall_hash_result(), 299
 - get, 134
 - get_class(), 6
 - get_fields(), 299
 - getFromCache(), 183
 - interface, 70
 - isValidCipher(), 519
 - iterator methods, 71
 - key(), 72, 77
 - load(), 182
 - magic, 8
 - classes, 9–11
 - next(), 73
 - print_r(), 543
 - rewind(), 71, 76
 - save(), 182
 - seek(), 75–76
 - set, 134
 - setMulti(), 170–171
 - touchPage(), 77
 - type hinting, 6
 - unlink(), 78
 - unset(), 78
 - valid(), 72
 - var_dump(), 543
 - visibilities, 3
- Microsoft debugger, 273
- MINIT function, 350, 357
- moxi (memcached proxy), 185–186
- MTA (Mail Transfer Agent), 55
- multiton design patterns, 11–13
- MultitonExample class, 12–13
- MVCC (multiversion concurrency control), 242
- MX (Mail Exchange), 55
- MyISAM storage engine, 90–91, 93–94
 - compressed tables, 96
 - high read environment, 97
 - high write environment, 97
 - InnoDB comparison, 99–102
 - Merge storage engine and, 110–111

- MySQL instance crashes, 97
- parameters, 94
- row formats, 94–95
- version, 96
- when to use, 97
- MySQL
 - binaries, custom, 89
 - character classes, 60
 - character sets, 486–487
 - collations, 487–488
 - connecting/disconnecting, 296–297
 - constraints, 35
 - debuggers, 564–565
 - debugging
 - commands, 564
 - error log, 560–561
 - PROCESLIST, 562–564
 - slow query log, 561–562
 - drivers, 88
 - flow control, 32, 34–35
 - CASE function, 34
 - IF () function, 34
 - IFNULL () function, 35
 - NULLIF () function, 35
 - FULLTEXT indexes, 365
 - HAVING clause, 32
 - joins, 19–21
 - aliases, 22
 - case sensitivity, 26–27
 - complex, 27
 - inner joins, 21–23
 - left joins, 25
 - multiple tables, 20
 - outer joins, 23–24
 - right joins, 24–25
 - syntax, 25–27
 - libraries, interfacing with PHP, 88
 - logic operators, 33–34
 - logical operations, 32
 - logical operators
 - AND, 33
 - NOT, 33
 - OR, 33
 - XOR, 33
 - Memcached Functions for MySQL (*See* Memcached Functions for MySQL)
 - privilege tables, 488
 - queries
 - dependent queries, 45
 - GROUP BY, 30–31
 - subqueries, 45–46
 - regular expressions, 59–62
 - data capture, 61
 - query filters, 62
 - replacing strings, 62
 - relational integrity, 35
 - replication
 - benefits, 264
 - binary log file analysis, 256–257
 - breaking, 261–262
 - commands, 261
 - configuration options, 260
 - issues with, 263–264
 - master analysis, 255–256
 - selective, 262–263
 - slave analysis, 257–258
 - testing, 254–255, 259
 - security model, 496–497
 - SELECT statements, 28–29
 - server, hardening, 493–501
 - silent column changes, 45
 - status tables, I_S, 482
 - status variables, 484–485
 - storage engines, 87, 89
 - Archive, 109–110
 - available engines, 89–90
 - Blackhole, 108–109
 - CSV, 112
 - default, 93
 - defining, 90–91
 - Falcon, 118–120
 - features, 89
 - Federated, 112–118
 - information, 89–92
 - InnoDB, 97–103
 - Maria, 120–121
 - Memory, 103–107
 - Merge, 110–111
 - MyISAM, 93–97
 - non-transactional, 89–90
 - table, confirming, 91–92
 - transactional, 89–90
 - system variables, 485–486
 - tables, derived tables, 46–47

- unauthenticated users, 494
- UNION, 28–29
- user privileges, 488
- MySQL Native Driver (for PHP), 88
- MySQL Stored Routines Library, 231
- MySQLi Result objects, 307
- mysqlnd, 88
 - as native PHP extension, 88
- mysql_secure_installation command, 494

N

- names, views, 225
- __NAMESPACE__ magic constant, 9
- namespaces, 350–351
- Narada
 - components, 413–415
 - configuration file, 420–422
 - database tables, 415–416
 - downloading, 413
 - flow chart depiction, 439
 - Gearman and, 412
 - workers, running, 440
 - index page, 435–438
 - memcached and, 412
 - Narada class, 422–429
 - get () method, 425
 - getContent () method, 426
 - NaradaWorkerCallback () method, 428
 - runWorker () method, 428
 - searchIndex () method, 427
 - storeContent () method, 425
 - verify () method, 428
 - NaradaIndexTimer.php, 438–439
 - open source, 441–442
 - overview, 412
 - running, 439–441
 - Sphinx and, 412
 - setup, 416–429
 - Sun and, 412
- Natural Language mode, 367
- NDB storage engine, 127
- networking
 - clients, building, 344–345
 - servers, building, 345–346
- next () method, 73
- NitroEDB storage engine, 127


- Node class
 - defining, 3–4
 - polymorphism, 2
- nOFB encryption, 504
- non-blocking sockets, 346
- non-deterministic cache, 144
- non-transactional tables, 238–239
- Norbye, Trond, 140
- NOT NULL constraint, 35–36
- NOT operator (MySQL), 33

O

- object-oriented PHP, 2–19
 - interfaces, 7–8
 - magic constants, 8–9
- objects
 - CodeGen_PECL, defining, 319
 - decorating, 14
 - identifying, 355–356
 - methods, 351–353
 - permissions, 496
 - properties, 353–354
 - table objects, creating, 223–224
- observer pattern, 18–19
- OFB (Output-Feedback Mode) encryption, 504
- opcode caching, 132
- open source projects
 - creating, 281–283
 - Narada, 441–442
 - subdirectories, 282
 - top-level directories, 282–283
- open source UDFs, 272
- OpenSolaris, memcached installation, 147
- operating system, security, 495
- optimization, SQL, InnoDB storage engine and, 102–103
- OR operator (MySQL), 33
- Our Delta patches, 126
- outer joins (MySQL), 23–24

P

- page caching, 145
- Pagination class, 75–76
- parameters
 - Callback, 81

- Memory storage engine, 103
 - MyISAM storage engine, 94
 - stored functions, 226–227
 - stored procedures, 226–227
 - parent keyword, 2
 - parsing, quoted strings, 80
 - partial page caching, 145
 - passwords
 - salts, 460
 - SHA-1 hash, 459
 - PBXT (Primebase Transaction) storage engine, 122–123
 - PEAR packages, 314
 - PECL extensions, 132
 - PECL/Gearman, installation, 404–405
 - PECL/memcached, 158–159
 - client behavior, 162–163
 - operations, 159–160
 - putting and retrieving data, 163–166
 - PECL (PHP Extension Community Library), 309
 - Percona builds for storage engine patches, 125
 - Perl, install utility, 195
 - permissions, 495–497
 - host permissions, 499
 - objects, 496
 - restarting, 498
 - triggers, 237
 - valid, 496
 - views, 223–224
 - PHP
 - Gearman and, 409–412
 - libraries, interfacing with MySQL, 88
 - pure authentication, 466–473
 - cookies, 471–473
 - sessions, user authentication and, 467–471
 - Sphinx and, 395–399
 - PHP API functions, consuming, 336–338
 - PHP license, 316
 - PHP_FE, 351
 - phpize program, 312
 - PHP_ME, 351
 - PHP_METHOD, 351
 - php_network.h, 344–345
 - php_printf function, 340, 361
 - php_socket_t, 345
 - php_stream, 341
 - polymorphism, 2–6
 - BlogEntry class, 2
 - ForumTopic class, 2
 - Node class, 2
 - ports, 537
 - prepending, memcached, 166–167
 - primary keys, InnoDB storage engine, 100
 - printing, extensions, 361
 - print_r() method, 543
 - private keyword, 3
 - private member variables, serialization, 10
 - privilege tables (I_S), 488
 - privileges
 - appropriate use, 499
 - column privileges, 490–491
 - event privileges, 268
 - removing, 497
 - schema privileges, 489
 - stored routines, 230–231
 - table privileges, 489–490
 - user privileges, 488
 - proactive caching, 145
 - PROCESSLIST, 562–564
 - project directory organization, 281–283
 - properties, 353–354
 - CodeGen_PECL, defining, 320
 - protected keyword, 3
 - protocol wrappers, 522
 - disabled, 529
 - prototyping, query builder and, 83–85
 - Proven Scaling, 483
 - proxy design pattern, 13–14
 - PSEA (pluggable storage engine architecture), 89
 - InnoDB Plugin, 121–122
 - PBXT, 122–123
 - XtraDB, 123
 - public keyword, 3
 - publisher/subscriber pattern, 18–19
- 
- QEP (Query Execution Plan), 102
 - quantifiers in regular expressions, 50
 - queries (MySQL)
 - dependent queries, 45
 - derived tables, 46–47
 - filtering, 62
 - GROUP BY, 30–31
 - subqueries
 - complex, 47–49
 - correlated subqueries, 46
 - query builder, prototyping, 83–85
 - quote syntax in regular expressions, 51

quoted strings, parsing, 80
 quotes, escaping, 80

R

rainbow tables, 459
 random lookups, 451–452
 read operations, Federated storage engine, 112
 read-through caching, 141, 216–219
 read-through caching callback, 165
 ReadableNode interface, 7
 READ_COMMITTED, 244–245
 READ_UNCOMMITTED, 245–246
 records
 returning, 72–73
 seeking specific, 75–76
 validating, 72–73
 regular expressions
 [] (square brackets), 50
 alternation, 51
 back-references, 53
 boundary character, 49
 comments, 49, 54
 data capture, 53–54
 exclamation point, 448
 expert, 52–54
 general patterns, 49–51
 MySQL, 59–62
 Perl-style, 54
 POSIX-style, 54
 quantifiers, 50
 quote syntax, 51
 replacing strings, 58–59
 relational integrity (MySQL), 35
 REPEATABLE_READ, 242–244
 REPLACE command, triggers, 232, 236–237
 REPLACE statement, 558
 replacing strings
 MySQL, 62
 regular expressions, 58–59
 replay attacks, 507–509
 replication
 benefits, 264
 binary log file analysis, 256–257
 binary logging, 251
 Blackhole storage engine, 108
 breaking, 261–262
 commands, 261
 configuration, options, 260
 isolation (ACID), 247
 issues with, 263–264
 master analysis, 255–256
 memcached, 140
 purposes, 250–251
 selective, 262–263
 setup
 master configuration, 251–252
 slave configuration, 252
 slave operation, 253
 slave analysis, 257–258
 testing, 254–255, 259
 REPORT_ERRORS flag, 342
 REST (Representational State Transfer), 524–531
 controller, 525–526
 cURL, 528–529
 error codes, 533
 requests, 529–531
 resources, 525
 status codes, 527–528
 verbs, 524
 result set array, returning, 297–300
 returning records, 72–73
 REVOKE command, 268, 497
 rewind() method, 71, 76
 rewinding an iterator, 71
 rewrite maps, 450–455
 built-in, 451
 rewrite rules
 conditional, 447–450
 exclamation point, 448
 testing for patterns, 449
 testing to check file or path, 449
 directives, 444
 RewriteCond, 447
 flags
 chain, 446
 env, 447
 forbidden, 445
 gone, 446
 last, 446
 multiple, 447
 next, 446
 nocase, 446
 noescape, 446
 nosubreq, 446
 passthrough, 446
 proxy, 446
 qsappend, 446

- redirect, 445
 - skip, 446
 - type, 446
- flow, 445–447
- logging, 450
- optimization, 450
- purposes, 443, 444–445
- random lookups, 451–452
- structure, 444–445
- RewriteCond directive, 447
- right joins (MySQL), 24–25
- root user, 494
 - changing, 498–499
- routines, 226
- RPCs (Remote Procedure Calls), 532

S

- \s shorthand character, 50
- Sallings, Dustin, 140
- save() method, 182
- scaling out, 138
- schema privileges, 489
- search engines, Narada, 412–442
- search worker (Gearman), 433–434
- searchd (Sphinx), 374–376
- searches
 - FULLTEXT indexes, 366
 - indexes, 365–366
 - modifier values, 366
 - Sphinx, 369, 383
 - Boolean query syntax, 385
 - extended query syntax, 385–386
 - search modes, 384
 - search utility, 386–388
 - sort modes, 384
- security
 - additional, 500
 - auditing, 500–501
 - automation attacks, 514–515
 - backdoor, 498
 - bypassing, 498
 - CSRF, 511–514
 - ideal, 499–500
 - introduction, 493
 - operating system, 495
 - permissions, 495–497
 - replay attacks, protecting against, 507–509
 - SQL injection, protecting against, 506–507
 - XSS (cross-site scripting) attacks, 509–511
- seek() method, 75–76
- SeekableIterator interface, 75–76
- seeking specific records, 75–76
- SELECT statement, 217, 546–555
- SELECT statements (MySQL), 28–29
- selective replication, 262–263
- SERIALIZABLE, 246–247
- serialization, 9
 - APC, 135
 - eAccelerator, 135
 - private member variables, 10
- serialize() function, 9
- Server SQL, modes, 40–44
- servers, memcached, connecting to, 197–198
- session caching, 145
- _set() magic method, 10
- SET data type, 36–37
- set method, 134
- setMulti() method, 170–171
- _set_state() magic method, 11
- setToCache() function, 184
- SHA-1 hash, 459
- shebang, 518
- shortened URL applications, 452–455
- shorthand characters, 50
- SHOW command, 478, 491–492
- SHOW MASTER STATUS command, 256
- SHOW TABLE STATUS command, 92
- SHOW .ENGINES command, 89–90
- SHOW_PROCESSLIST, 482
- shutdown, restricting, 498
- silent column changes in MySQL, 45
- single-directional encoding, 505–506
- singleton design patterns, 11–13
- SingletonExample class, 11–12
- SixApart, 402
- skeleton code
 - compiling, 309–312
 - creating, 309–312
- skip-grants option, 498
- _sleep() magic method, 9
- slow query log, 561–562
- SOAP (Simple Object Access Protocol)
 - bindings, 537–538
 - client, setup, 540–541
 - library, WSDL, 533–534

- requests, envelopes, 532
- server, setup, 538–540
- web services, 532–533
- socket pointers, valid, 55–56
- sockets, blocking, 346
- source distribution directory (Memcached Functions for MySQL), 212–214
- Sphinx (SQL Phrase Index)
 - Aksyonoff, Andrew, 369
 - applications using, 395–399
 - data sources, 372, 376–379
 - delta indexes, 391–394
 - distributed indexes, 373
 - indexer, 374
 - indexer options, 382
 - indexes, 372–373
 - attributes, 376
 - columns, 376
 - defining, 379–381
 - fields, 376
 - inherited, 381
 - installation, 370–371
 - Narada and, 412, 416–417
 - PHP and, 395–399
 - programs, 369–370
 - sakila_main data source, 379
 - searchd, 374–376
 - searchd options, 382
 - searches, 383
 - Boolean query syntax, 385
 - extended query syntax, 385–386
 - modes, 384
 - search utility, 386–388
 - sort modes, 384
 - Sphinx.conf, 371–382
 - SphinxQL, 388–389
 - starting
 - running indexer, 382–383
 - searchd, 383
 - utilities, 369–370
- sphinx_counter, 390
- sprintf() function, 290
- SQL injection, protecting against, 506–507
- sql-mode-traditional-mysam-sql, 42–43
- sql-mode-traditional.sql, 42
- SQL (Structured Query Language)
 - DELETE statement, 556–558
 - EXPLAIN statement, 547–549
- indexes
 - common mistakes, 552–554
 - identifying, 549–550
 - optimizing, 552
 - types, 551
- INSERT statement, 558
- INSERT...ON DUPLICATE KEY UPDATE statement, 558–560
- optimization, InnoDB storage engine and, 102–103
- queries, optimizing, 545–558
- REPLACE statement, 558
- SELECT statement, 546–555
- statements
 - identifying, 545–546
 - reducing, 545
 - UDFs, 300
 - UPDATE statement, 556
- sql_mode=NO_ENGINE_SUBSTITUTION, 43–44
- sql_mode=TRADITIONAL, 43
- square brackets ([]) in regular expressions, 50
- stack overflow errors, 544
- stack traces
 - creating, 544–545
 - keys, 544
- standalone storage engines, 124–125
- standard output from functions, 340
- stat() function, 275
- state caching, 145
- statement blocks, 228
- statements
 - DELETE, 556–558
 - EXPLAIN, 547–549
 - INSERT, 558
 - INSERT...ON DUPLICATE KEY UPDATE, 558–560
 - REPLACE, 558
 - SELECT, 546–555
 - UPDATE, 556
- static keyword, 3, 12
- statistics, memcached, 176–178
- stepping through code, 302–303
- storage engine
 - eBay Memory, 127
 - integrity, 44–45
 - Kickfire, 128
 - MySQL, 87, 89
 - Archive, 109–110
 - available engines, 89–90
 - Blackhole, 108–109

- CSV, 112
- default, 93
- defining, 90–91
- Falcon, 118–120
- features, 89
- Federated, 112–118
- information, 89–92
- InnoDB, 97–103
- Maria, 120–121
- Memory, 103–107
- Merge, 110–111
- MyISAM, 93–97
- non-transactional, 89–90
- table, confirming, 91–92
- transactional, 89–90
- NDB, 127
- NitroEDB, 127
- patch products
 - Our Delta, 126
 - Percona builds, 125
- PSEA
 - InnoDB Plugin, 121–122
 - PBXT, 122–123
 - XtraDB, 123
- standalone products, 124
- Virident, 128
- stored functions, 226
 - characteristics, 228
 - parameters, 226–227
- stored procedures, 226
 - characteristics, 228
 - parameters, 226–227
- stored routines
 - conditions, 229–230
 - disadvantages, 231
 - extending, 231
 - flow control, 230
 - handlers, 229–230
 - logic
 - cursors, 229
 - variables, 228–229
 - meta data, 230–231
 - MySQL Stored Routines Library, 231
 - privileges, 230–231
- STREAM encryption, 503
- stream functions, 341–343
 - php_register_url_stream_wrapper, 343
 - php_register_url_stream_wrapper_
 - volatile, 343
 - php_stream_close, 343
 - php_stream_context_alloc, 343
 - php_stream_context_del_link, 343
 - php_stream_context_free, 343
 - php_stream_context_get_link, 343
 - php_stream_context_get_option, 343
 - php_stream_context_set, 343
 - php_stream_context_set_link, 343
 - php_stream_context_set_option, 343
 - php_stream_copy_to_mem, 343
 - php_stream_copy_to_stream, 343
 - php_stream_copy_tostream_ex, 343
 - php_stream_eof, 343
 - php_stream_flush, 343
 - php_stream_free, 343
 - php_stream_from_persistent_id, 343
 - php_stream_getc, 343
 - php_stream_get_line, 343
 - php_stream_get_record, 343
 - php_stream_locate_url_wrapper, 343
 - php_stream_mkdir, 343
 - php_stream_opendir, 343
 - php_stream_open_wrapper, 343
 - php_stream_open_wrapper_ex, 343
 - php_stream_putc, 343
 - php_stream_puts, 343
 - php_stream_read, 343
 - php_stream_readdir, 343
 - php_stream_rewind, 343
 - php_stream_rmdir, 343
 - php_stream_scandir, 343
 - php_stream_seek, 343
 - php_stream_set_options, 343
 - php_stream_stat, 343
 - php_stream_tell, 343
 - php_stream_wrapper_log_error, 343
 - php_stream_write, 343
 - php_unregister_url_stream_wrapper, 343
 - php_unregister_url_stream_wrapper_
 - volatile, 343
 - STREAM_ASSUME_REALPATH flag, 342
 - STREAM_DISABLE_OPEN_BASEDIR flag, 342
 - STREAM_DISABLE_URL_PROTECTION flag, 342
 - STREAM_MUST_SEEK flag, 342
 - STREAM_ONLY_GET_HEADERS flag, 342
 - STREAM_OPEN_FOR_INCLUDE flag, 342
 - STREAM_OPEN_PERSISTENT flag, 342
- streams
 - flags, 341–342
 - validation, 341
- STREAM_USE_URL flag, 342

STREAM_WILL_CAST flag, 342

strings

converting to, 327–328

quoted, parsing, 80

replacing

MySQL, 62

regular expressions, 58–59

as `zvals`, 326–328

subqueries (MySQL), 45–46

complex, 47–49

correlated subqueries, 46

derived tables, 46–47

Sun, Narada and, 412

syntax

joins, 25–27

triggers, 233

T

table-level locking, 104

table objects, creating, 223–224

table privileges, 489–490

table storage engines (MySQL), confirming, 91–92

tables

building, 362

non-transactional, 238–239

transactional, 239–240

tables InnoDB storage engine, 98–99

tables (MySQL)

aliases, 22

derived tables, 46–47

terminal types, 6

testing

memcached installation, 151–152

replication, 254–255, 259

unit testing, factory methods, 17

variables against interfaces, 7

`zval`, 323–324

`tests` file, 310

text lookups, 451–452

thread-safe functions, 273

thread safety, globals, 359–361

TokuDB, 124–125

Tokyo Tyrant, 187–188

`touchPage()` method, 77

tracking, unlinks, 77

transactional tables, 239–240

transactions, 237–238

isolation level, 242

transparency, Federated storage engine, 113

triggers

defining, 232

DELETE command, 235–236

INSERT command, 233–234

integration points, 232

Memcached Functions for MySQL, 212

execution, 214–216

permissions, 237

REPLACE command, 236–237

syntax, 233

UPDATE statement, 234–235

True Global Resources, 360

TSRM (Thread Safe Resource Management), 359

`tsrm_mutex`, 360

Turck MMCache, 133

Turing Test, automation attacks, 514

type hinting, 6

type key, 544

U

Ubuntu, memcached installation, 146–147

UDFs (user defined functions), 191, 231, 271

algorithms, 277

arguments, 192

building, 292–294

C-libraries, 272

coding, 281

debuggers, 273

debugging, 300–301

breakpoints, 302–303

`gdb`, 301–302

literal values and, 304

stepping through code, 302–303

design, high-level, 275–277

developing, requirements, 273–274

example, 274–275

installation, 294–295

checking, 289–292

open source, 272

project creation, 281–283

overview, 272–273

PHP and, 296–300

required functions, 274

- result set array, returning, 297–300
- running, 295–296
- source code implementation, 283–289
- SQL statements, 300
- thread-safe functions, 273
- unauthenticated users, 494
- UNION (MySQL), 28–29
- UNIQUE KEY constraint, 37
- unit testing, factory methods, 17
- unlink() method, 78
- unlinks, tracking, 77
- _unset() magic method, 10
- unset() method, 78
- unseting elements, 77
- UNSIGNED constraint, 36
- UPDATE statement, 219–220, 556
 - triggers, 234–235
- updating, views, 225
- URLs
 - rewrite rules, 444
 - shortened, applications, 452–455
- USE_PATH flag, 342
- user authentication
 - database, 458
 - HTTP-based, 460
 - basic, 460, 461–463
 - digest, 460, 463–466
 - need for, 457
 - password, as salted md5 hash, 459
 - pure PHP authentication, 466–473
 - cookies, 471–473
 - sessions, 467–471
 - unauthenticated users, 494
 - username, 458–459
- user caching
 - enabling, 133
 - storage, 134
- user privileges (I_S), 488
- users
 - default, new installation, 494
 - root, 494
 - changing, 498–499
- utilities, libmemcached, 154
 - memcat, 155–156
 - memcp, 156
 - memerror, 158
 - memflush, 156
 - memrm, 157

- memslap, 157–158
- memstat, 157

V

- valid() method, 72
- valid permissions, 496
- validation
 - records, 72–73
 - streams, 341
- values, returning, 334–335
- var_dump() method, 543
- variables, 228–229
 - classes, 3
 - configuration, default values, 266
 - extensions, 321–329
 - zval, 321–328
 - member, serialization, 10
 - testing against interfaces, 7
- views, 221–222
 - access permissions, 223–224
 - data warehousing, 225
 - materialized, 225
 - names, 225
 - updating, 225
- Virident storage engine, 128
- visibilities, 3

W

- \w shorthand character, 50
- Wafflegrid, 129
- _wakeup() magic method, 9
- warnings, 346–347
- web services, 517
 - REST (Representational State Transfer), 524–531
 - SOAP, 532–533
- WITH ROLLUP keyword, 31–32
- write-back caching, 142
- write statements, Federated storage engine, 112
- write-through caching, 141
- WSDL (Web Service Descriptor Language), 533–534
 - defining service, 538
 - messages, defining, 536
 - ports, 537
 - types, 535–536
 - complex, 535

X

- Xing.com, 402
- XML documents
 - creating, 314–316
 - extensions as, 314
- XOR Cipher, 501–503
- XOR operator (MySQL), 33
- XSS (cross-site scripting) attacks, 509–511
- XtraDB storage engine, 123

Y

- Yahoo!, 402

Z

- Zend directory, 308
- Zend engine, 308
 - memory management, 328–329
- zend_parse_parameters function, 334
- ZTS (Zend Thread Safety), 359
- zval, 321–322
 - comparing, 324–326
 - reading, 324–326
 - setting, 323–324
 - testing, 323–324
- zvals, strings as, 326–328

