

# Index

## A

- a2enmod**, 428, 467, 480, 486
- A9 (Amazon)**, 708
- Aas, Gisle**, 8
- ABC** 80, 17
- ABC microcomputers**, 17
- abstraction**, 212
- access control**
  - Apache, 449, 450
  - cookies and, 643
  - MySQL, 75, 84
- AccessFileName**, 454
- accessor methods**, 217–221, 338–339
  - modifying to use cache, 348–350
- AccessTestHandler**, 613, 614, 619, 621, 626, 630
- account management statements (MySQL)**, 80–84
- account\_levels**, 332, 333. *See also* user application
- account.tt2**, 729
- ACID (atomicity, consistency, isolation, durability)**, 18, 113, 118, 119, 121, 122, 817
- add()**, 325, 350
- AddHandler**, 461
- AddIcon**, 456
- AddIconByEncoding**, 457
- AddIconByType**, 457
- AddInputFilter**, 461–462
- addNumbers()**, 202
- AddOutputFilter**, 462
- AddType (directive)**, 443–444
- aggregate functions**, 60–62
  - Ajax (Asynchronous JavaScript and XML), 1, 707–738
  - AJAX v., 707
  - asynchronous and, 710
  - core functionality, 708, 738
  - example 1, 708–710
    - instantiation of XMLHttpRequest object, 708, 709
    - mod\_perl handler for, 710
  - example 2 (display returned HTML table), 711–713
    - mod\_perl handler for, 711–712
    - template code, 713
  - example 3 (building table on client), 714–716
    - mod\_perl handlers for, 716
  - example 4 (MySQL Ajax client), 716–722
    - in action, 720–722
    - mod\_perl handler for, 719–720
  - example 5 (reading raw POST data), 722–724
    - mod\_perl handler for, 724–725
  - example 6 (using Prototype JavaScript Framework), 725–727
  - example 7 (account creation with Ajax), 728–738
    - account handler, 732–735
    - account page mod\_perl handler, 728–729
    - account page template, 729–732
    - in action, 736–738
    - sendEmail(), 735–736
  - JavaScript and, 707, 738
  - mod\_perl applications and, 707–738
  - overview of, 707–708
  - Professional Ajax*, 707
  - XMLHttpRequest object and, 708, 709, 711, 714
- AjaxAccountHandler**, 732–735
- /ajax\_handler URL**, 710, 711, 715, 724, 727, 732
- Aker, Brian**, 359, 383, 529. *See also* DBIx::Password; libmemcached; Memcached Functions for MySQL
- Aksyonoff, Andrew**, 10, 152, 158, 160, 161, 766
- aliasing**, 39–40
- AllowOverride**, 454
- ALTER TABLE**, 33–34, 55
- Amazon's A9**, 708
- ANSI**
  - read-only views, 293
  - SQL/MED, 125, 128
  - X3.135, 266
- Apache**, 417–501
  - access control, 449, 450
  - API, 417, 419–420
  - authentication, 450–451
  - authentication code refactoring, 421
  - authorization, 451
  - clickstream analysis, 466–467
  - configuration, 435–483. *See also* directives
    - online information, 435
    - uploads directory in, 676–677
  - configuration file, 420, 435–436
  - configuration schemes, 483–500
    - CentOS/Redhat variants, 486–487
    - source install, 484
    - SUSE, 487–489
    - Ubuntu/Debian install, 484–486
    - Windows, 489–492
  - directives
    - authorization, 451–453
    - basic, 440–444
    - CGI, 457–458
    - client handling, 462
    - Configuration Section Container Directives, 436–440
    - error, 448–449
    - filter, 461–462
    - handler, 460–461
    - .htaccess file, 453–454
    - indexing, 454–457
    - logging, 446–448
    - mod\_proxy, 481–483
    - rewrite, 468–478
    - server tuning, 444–446
    - SSL, 463–466
    - User Directory, 458–459
    - virtual hosting, 459–460
  - extensibility, 417, 419–420
  - features, 419
  - filtering and, 420, 422, 423, 566
  - functions of, 417–419
  - history of, 418–419
  - hooks, 422, 423, 575, 601
  - HTTP and, 4, 8, 9, 106, 417–418

## Apache (continued)

### Apache (continued)

installation, 424–432  
   on Apple OS X, 429  
   components, 424–425  
   on Redhat-based systems, 428–429  
   on Ubuntu/Debian-based Linux, 427–428  
   on UNIX, 429–432  
   on Windows, 425–427  
 LAMP and, 3–4  
 large file support, 421  
 Linux and, 419  
 market share of, 3  
 mod\_rewrite and, 434, 454, 468–478, 501  
 modules, 419–420  
 overview, 417–419  
 popularity of, 417, 419, 501  
 regex and, 421  
 reverse proxying, 478–480  
 server control, 483  
 setup, CGI, 504. *See also* contact list application  
 source code, 422  
 tasks, 492–500, 501  
   configuring virtual host, 493–495  
   Digest authentication, 496  
   HTTP basic authentication, 495–496  
   reverse proxy with virtual hosts, 499–500  
   secure server configuration, 497–498  
   secure server configuration iwith certificate, 498–499  
 TLS and, 417, 423, 463  
 UNIX and, 417, 419  
 version 2.2 v. 1.3, 419  
   features, 420–421  
   module changes, 423–424  
   request phases, 421–423  
 WAMP and, 3  
 wildcards and, 436, 437, 485

**Apache Hadoop**, 13  
**Apache life cycle**  
   mod\_perl handlers and, 8, 9, 576, 601, 612  
   overview, 577  
   phase name, 577

**Apache Portable Runtime (APR)**, 419, 491, 566  
**Apache Portable Runtime modules**, 585  
**Apache request library**. *See* libapreq2  
**Apache request phase cycle**, 423, 461, 576, 584  
**Apache Software Foundation**, 3, 4, 417, 489  
**Apache Tomcat**, 4, 479  
**Apache2 Perl modules**, 585–598  
   connection record, 586, 590  
   constants, 586–587  
   filter record, 586, 590–591  
   logging, 586, 592  
   performance, 586, 594, 595  
   request/request record, 586, 587–590  
   resource, 595  
   server configuration, 586, 592–594  
   server record, 586, 591  
   status, 586, 596–598

**Apache2::Access**, 589  
**Apache2::Connection**, 590  
**Apache2::Const**, 586–587, 636  
**Apache2::Cookie**, 565, 644, 645, 647, 649, 667, 668, 705  
**Apache2::DBI**, 595  
**Apache2::Directive**, 592–594  
**Apache2::Filter**, 590–591  
**Apache2::Log**, 592  
**Apache2::MPM**, 594  
**Apache2::PerlSections**, 594  
**Apache2::Reload**, 594–595

**Apache2::Request**, 587, 702  
**Apache2::RequestIO**, 589, 636  
**Apache2::RequestRec**, 588, 636  
**Apache2::RequestUtil**, 588  
**Apache2::Resource**, 595  
**Apache2::Response**, 589  
**Apache2::ServerRec**, 591  
**Apache2::ServerUtil**, 591  
**Apache2::SizeLimit**, 595–596  
**Apache2::Status**, 596–598  
**Apache2::Upload**, 565, 677, 681  
**Apache2::URI**, 589  
**Apache2::Util**, 590  
**apachectl graceful**, 483  
**apachectl start**, 483  
**apachectl stop**, 483

**API**  
   Apache, 417, 419–420  
   Cache::Memcached, 325–328  
   DBI, 247  
   DOM object, 708  
   UDF, 105, 106

**Apple OS X**  
   Apache installation on, 429  
   include path and, 195

**Application Packaging Standard (APS)**, 8  
**app.pl**, 516  
**APR (Apache Portable Runtime)**, 419, 491, 566  
**APR modules**, 585  
**APS (Application Packaging Standard)**, 8  
**apt-cache search**, 247, 319, 427, 428, 742, 805  
**apt-get**, 247, 319, 320, 427, 794, 804, 805  
**apxs**, 433, 434, 435, 487  
**Arabs**, 30  
**Archive (storage engine)**, 112, 123–125  
   characteristics, 123–124  
   under the hood, 124  
   table  
     creation, 125  
     maintenance, 125

**Arno, Kaj**, 17  
**array references**, 170  
**@\_array**, 188  
**arrays**, 167  
   adding, 175  
   @ISA, 197, 221, 234, 235, 237, 239, 244  
   functions and, 176–177  
   hashes as, 179–180  
   last subscript value of, 177–178  
   lists v., 174  
   printing, 178  
   slices, 178  
   splicing, 175–176  
   usage, 174–178

**arrow method call**, 216  
**ASP**, 4  
**asterisk (\*)**, 41  
**asynchronous, Ajax functionality and**, 710  
**Asynchronous JavaScript and XML**. *See* Ajax  
**asynchronous replication**, 133, 134  
**@\_array**, 188  
**Athen**, 576  
**@ISA array**, 197, 221, 234, 235, 237, 239, 244. *See also* inheritance

**atomicity**, 18, 121, 122. *See also* ACID  
**attackPrey**, 231, 233, 239  
**\$attr\_hashref**, 270  
   \_attr\_default(), 224  
   \_attr\_exists(), 224

## Cache::Memcached

**\$attrib\_hashref**, 271  
**\_attrib\_keys()**, 224  
**\$ATTRIBS**, 224, 228, 229, 237  
**attribute accessor methods**, 217–221  
**attributes**  
   defined, 212  
   retrieving, 217–218  
   setting, 218–221  
   statement handle, 265–269  
**\$attributes**, 273  
**\$attributes**, 253–254  
**%attributes**, 274  
**AuthBasicProvider**, 452–453  
**AuthDBMType**, 453  
**AuthDBMUserFile**, 453  
**authentication**, 450–451  
   Basic Digest Authentication, 452, 496  
   code refactoring, 421  
   Digest, setting up, 496  
   HTTP, setting up, 495–496  
**AuthGroupFile**, 453  
**AuthName**, 452  
**authorization, directives**, 451–453  
**AuthType**, 452  
**AuthUserFile**, 453  
**Authz**, 576  
**AuthZTestHandler**, 619, 620, 621, 626, 631  
**AutoCommit**, 254  
**autoconf**, 807  
**AUTO\_INCREMENT**, 35, 36, 54, 55, 115  
**AUTOLOAD()**, 221–231, 244  
   issues with, 225–226, 229  
**\$AUTOLOAD**, 221, 223, 224, 226, 227, 228, 231  
**automake**, 807  
**automatic error handling**, 280–281  
**auto\_prepend\_file**, 483. *See also* ProxyPreserveHost  
**AutoReconnect**, 254  
**available\_drivers()**, 248  
**AVG()**, 60

**B**

**back-references**, 469–470  
**“backronyms,”** 164  
**backups, MySQL**, 817–823. *See also* mysqldump  
**bar ()**, 96, 97  
**base**. *See* 'use base'  
**Basic Digest Authentication**, 452, 496  
**basic directives (Apache)**, 440–444  
**BEGIN**, 88  
**begin\_work**, 276, 277  
**behavioral functions, libmemcached**, 366–369  
**BigTable project**, 13  
**binary log**, 134, 141. *See also* replication  
**bind\_col()**, 263–264  
**bind\_columns()**, 264  
**binding**, 261  
   defined, 256  
   input parameters, 262–263  
   methods, 261–264  
   output parameters, 263–264  
**bind\_param()**, 261–263  
**bind\_param\_inout()**, 261  
**\$bind\_vals**, 336  
**@bind\_values**, 270, 271, 274  
**-binlog-do-db**, 137  
**-binlog-ignore-db**, 137  
**Bit operator**, 58

**Blackhole (storage engine)**, 112, 132–133  
**bless()**, 211, 214, 215, 216, 219, 223, 227, 236, 238, 244, 334, 348, 404, 532, 661  
**bling query expansion**, 150  
**BLOBS**, 91, 113, 115, 124, 675, 748, 750  
**bomb alert, ModPerl::Registry scripts and**, 643  
**BOOLEAN**, 89  
**BOOLEAN MODE**, 149, 150, 151  
**Boolean operator**, 58  
**Boolean query syntax**, 159  
**brackets**  
   curly, 171, 190  
   square, 87, 687  
**Brahmagupta**, 30  
**Brahmasputha Siddhanta**, 30  
**brigades, bucket**, 566, 579, 580, 599, 627, 629  
**BrowserMatch**, 454, 462  
**b-tree indexes**, 149  
**bucket brigades**, 566, 579, 580, 599, 627, 629  
**Buddhist philosophy**, 30  
**BuildExcerpts()**, 774, 775  
**buildInsert()**, 659–660  
**buildRange()**, 776  
**buildUpdate()**, 658–659  
**bulk inserts**, 36–37  
**Bunce, Tim**, 247, 365, 821

**C**

**C++**, 4, 18, 19, 105, 163, 216  
**C programming language**, 4, 5, 9, 11. *See also* libmemcached  
   libmysql, 246, 252, 256  
   MySQL and, 18  
   UDFs and, 105  
**cache stamped**, 352–355, 357  
**cache/caching**, 1–2. *See also* memcached  
   apt-cache search, 247, 319, 427, 428, 742, 805  
   contact list application and, 506  
   deterministic, 318, 346, 352  
   file system, 318  
   issues, 352–357  
   JSON and, 316  
   lazy, 315, 316  
   methods, contact list application, 549–560  
   nondeterministic, 318, 346  
   page, 318  
   partial page, 318  
   proactive, 318  
   read-through, 315, 316, 357  
     with SELECT, 412–415  
   replication, 318  
   session, 318  
   state, 318  
   strategies, 318  
   templates, 693  
   user application (memcached example)  
     implementation plan, 345  
     key scheme, 346  
     strategies, 318  
     where to add, 345–346  
   user data  
     get method modifications, 351–352  
     set method modifications, 350–351  
   write-through, 315–317, 357  
**cacheCities()**, 347, 349  
**Cache::Memcached**, 205, 314, 323, 357. *See also*  
   libmemcached  
   API, 325–328

## Cache::Memcached (continued)

### Cache::Memcached (continued)

- Cache::Memcached::libmemcached v., 377–380
  - instantiation, 324–325
  - libmemcached v., 359
  - methods, 325–328
  - performance comparison (test script/tool) and, 377–380
  - using, 323–328
- Cache::Memcached::libmemcached**, 375–380
  - Cache::Memcached v., 377–380
  - contact list application and, 530, 534, 549, 553
  - installing, 365
  - Memcached::libmemcached v., 365–366, 375
  - performance comparisons, 376, 381
  - test script/tool and, 377–380
  - purpose of, 376
- cacheStates()**, 347, 348, 349
- cacheUsers()**, 550–552
- Cacti**, 648, 823
- CALL**, 86, 89, 90, 93
- caller()**, 188
- CamelCase**, 221
- carp()**, 224, 225, 226, 228, 529
- cas**, 369, 392
- cascade deletion**, 55, 332
- CASE()**, 68–69
- case operator**, 68
- CentOS**, 428. *See also* Redhat-based systems
  - Apache configuration schemes on, 486–487
  - DBI/DBD: :mysql on, 247
  - memcached on, 318–319
- CGI (Common Gateway Interface)**, 4, 15–16
  - Apache setup, 504
  - contact list application and, 504–505. *See also* contact list application
  - development of, 504
  - directives, 457–458
  - FastCGI and, 480, 482
  - Perl and, 8
  - PHP and, 15–16
  - Web and, 15
- CGI program. See contact list application**
- CGI::Carp**, 516, 529
- CGI.pm**, 503, 516, 587
- \$cgi->Vars()**, 520, 568, 638, 640
- CHANGE MASTER**, 147, 148, 818, 820
- CHAR**, 31, 91, 115
- CHECK TABLE**, 117, 118
- children**, 46
- ChopBlanks=1** 0, 267
- cities**, 332, 333. *See also* user application
- class methods**, 217
- classes**, 213. *See also* Felidea class
  - defined, 212
  - derived, 212
  - writing (in Perl), 214
    - constructor creation, 214–217
    - package creation, 214
- Cleanup (mod\_perl request phase handler)**, 576
- Clickability**, 6, 7
- clickstream analysis**, 466–467
- client, in RDBMS**, 245–246
- client handling directives**, 462
- client library for memcached. See Cache::Memcached; libmemcached**
- client programs (MySQL)**, 20–25
- clone**, 243
- clone()**, 276
- Clone option**, 573
- closedir()**, 185
- cloud computing**, 1, 13
- clusters**
  - clustered indexes, 118
  - memcached, 323
    - diagram, 314
    - replicating data to, 355–357
  - MySQL and, 18
- cmp\_ok**, 202, 203
- code reuse. See reusable code**
- colons**, 250
  - double, 194, 235
  - semi, 30, 32, 88, 96, 97, 250
- \$cols**, 336
- column data types**, 31
- COLUMN\_PRIVILEGES**, 83
- Columns (attribute)**, 271, 272
- columns\_priv**, 76, 80
- comma-delimited data**, 22, 388
- command executor**, 19
- command options**
  - MySQL
    - basic, 813–815
    - InnoDB, 815–816
    - port setting, 814–815
    - socket setting, 815
    - tablespace, 815, 817
    - user, 814
  - replication, 137–139
    - adding, 141–143
- command-line contact list application. See contact list application**
- comma-separated values. See CSV**
- COMMENT**, 88
- commercial make**, 807
- commit()**, 276, 277
- Common Gateway Interface. See CGI**
- common.h**, 107
- Community Server**, 793, 795
- comparison operators**, 71–72
- compiler, GCC**, 2, 807
- complex data structures, Perl**, 180–182
- complex data types, memcached and**, 329–330
- components, Mason and**, 698, 700–702
- Comprehensive Perl Archive Network. See CPAN**
- computers. See also Internet; Web**
  - ABC microcomputers, 17
  - Internet-centric, 1
  - Larsson's computer store, 17
  - Linux and, 3
  - "network is the computer," 1
  - zero and, 30
- computing, cloud**, 1, 13
- CONCAT()**, 63
- concurrent connections**, 353, 800
- concurrent inserts**, 35, 115
- conditional pattern**, 471
  - RewriteRule and, 472–473
- confess()**, 529. *See also* carp()
- configuration directives (mod\_perl)**, 569–575. *See also* directives
  - directives
- configuration file**
  - Apache, 420, 435–436. *See also* Apache; directives
  - sphinx.conf. *See also* Sphinx
    - data sources, 153–154, 156
    - database connection options, 156
    - indexer section, 155
    - indexes, 154–155
    - searchd section, 156

## curly brackets

- configuration schemes, Apache**, 483–500
  - CentOS/Redhat variants, 486–487
  - source install, 484
  - SUSE, 487–489
  - Ubuntu/Debian install, 484–486
  - Windows, 489–492
- Configuration Section Container Directives**, 436–440
- configuring MySQL**, 811–829. *See also* MySQL
- connect()**, 249–254
  - arguments, 249
  - \$attributes, 253–254
  - \$dsn, 249–253
  - \$password, 253
  - \$username, 253
- connect\_cached**, 254–255
- connectDB()**, 334
- \_connectDB()**, 532–533
- connection cycle phase handlers (mod\_perl)**, 578–579
- connection functions (memcached)**, 366
- connection mod\_perl handlers**, 607–612
- connection record Apache2 Perl modules**, 586, 590
- \_connectMemcached()**, 533–534
- consistency**, 18, 122. *See also* ACID
- consistent hashing**, 360, 367
- constants Apache2 Perl modules**, 586–587
- constructors**, 213
  - creating, 214–217
  - purpose of, 214
- contact edit page**. *See* **userform display function**
- contact list application (CGI program)**, 503–564
  - Apache setup, 504
  - Cache::Memcached::libmemcached and, 530, 534, 549, 553
  - caching and, 505
  - caching methods, 549–560
  - CGI and, 504–505
  - changes, to mod\_perl application, 567–569. *See also* mod\_perl
  - connection
    - memcached, 533–534
    - MySQL, 532–534
  - database methods, 542–548
  - database storage requirements, 505, 513–515
  - DBIx::Password and, 503, 529–530, 532, 533
  - design, 505–513
    - conceptualizing, 505–506
    - contact view page, 513
    - HTML, 506–507
    - main user form, user listing, 507–510
    - program requirements, 506
    - user edit form, 510–513
    - user interface, 506–513
  - error handling and, 506
  - functionality of, 505
  - implementation, 516–517
  - MySQL/memcached and, 503–504
  - program flow, 515–516
  - skeletal frame, 515–516
  - testing, 563–564
  - WebApp class methods, 529–542
- contact list application (command-line interface)**, 285–312
  - contacts
    - deleting, 285, 303–304, 306–309
    - editing, 285, 297–301, 304–305
    - inserting, 285, 301–302, 305–306
    - listing, 285, 294–297
    - lookup of, 285, 309–312
    - updating, 285, 302–303
  - database connection, 287, 288
  - declarations, 287–290
    - entry point for, 290–292
    - information\_schema, 293–294
    - initializations, 287–290
    - operations for, 285
    - program listing (online), 312
    - schema design, 286
    - table creation subroutine, 292–304
    - table for, 286
    - wire-frame for, 283–287, 288, 290
- contact view page**. *See* **viewuser display function**
- contacts**
  - deleting, 285, 303–304, 306–309
  - editing, 285, 297–301, 304–305
  - inserting, 285, 301–302, 305–306
  - listing, 285, 294–297
  - lookup of, 285, 309–312
  - updating, 285, 302–303
- contacts table**, 292
- contacts.CSM file**, 131
- contacts.CSV file**, 131, 132
- contacts\_table\_exists()**, 292
- control flow functions**, 68–70
- CONV()**, 93
- conversion process**
  - mod\_perl handler to ModPerl::Registry script, 641–643
  - ModPerl::Registry script to mod\_perl handler, 633–634, 635–641
- CookieName**, 454, 467
- cookies**, 643–651. *See also* session management
  - access control and, 643
  - Apache2::Cookie, 565, 644, 645, 647, 649, 667, 668, 705
  - defined, 643
  - Gear6 and, 6, 7
  - as hack, 419
  - libapreq2 and, 424, 565
  - purpose of, 643
  - session ids and, 643
  - session management and, 662–663, 662–670
  - tools for testing, 649–651
  - user tracking and, 643
- Cookies tool**, 649–650
- CookieTestHandler**, 643–649
- corrupted tables**, 25, 26, 117, 118, 125, 151
- Cougar**, 213, 235, 236
- COUNT()**, 60
- COUNT**, 43
- COUNT DISTINCT**, 61
- CPAN (Comprehensive Perl Archive Network)**, 9, 205–206
  - DBI/DBD: :mysql and, 247
  - modules and, 205–206
  - reusable code and, 164
- Craig's List**, 4, 7
- CREATE DATABASE**, 281, 812
- CREATE USER**, 80
- createAccount()**, 730, 732
- create\_contacts\_db()**, 292
- create\_contacts\_table()**, 292, 294
- createdb**, 281
- createUser()**, 339
- croak()**, 222, 224, 228, 281
- CSS**, 508, 518, 650
- CSV (comma-separated values)**, 130
- CSV (storage engine)**, 112, 130–132
  - under the hood, 131–132
  - table creation, 130–131
- curl**. *See* **libcurl**
- curl\_easy\_perform()**, 109
- curl\_udf.c**, 107
- curly brackets**, 171, 190

## CURRENT\_USER()

**CURRENT\_USER()**, 60  
**CursorName**, 266  
**Curtis, Antony**, 9  
**CustomLog**, 446–447

## D

**Danga Interactive**, 6, 11, 317, 320, 740. *See also* memcached

**data**. *See also* **memcached**; **meta-data**  
 deleting, 52–56  
 inserting, 35–38  
 querying, 38–50  
 replacing, 56–58  
 replicating, to multiple memcached servers, 355–357  
 updating, 50–52  
 web applications and, 313

**data definition language (DDL)**, 26

**data design (user application example)**, 332–333

**data modification methods, user application and**, 339–342

**data retrieval**  
 functions, memcached and, 370–371  
 methods, user application and, 335–337

**data sources (sphinx.conf)**, 153–154, 156

**data storage worker**, 748, 749, 750, 751, 759, 762–764

**data structures (Perl), complex**, 180–182

**data types (Perl)**, 165–168  
 arrays. *See* arrays  
 file handles, 168, 182  
 hashes, 49, 167, 172  
 scalars, 165–167  
 subroutines, 168, 186–188  
 type globs, 168  
 usage, 168–192

**DATABASE()**, 60

**Database, Multifunctional**, 799

**Database (attribute)**, 266

**database application**. *See* **contact list application**

**database connection options (sphinx.conf)**, 156

**database handles**, 281, 569, 599  
 methods, 274–277  
 single (example, Memcached Functions for MySQL), 403–409

**database methods**  
 contact list application, 542–548  
 generic, 651–662

**database storage requirements (contact list application)**, 505, 513–515

**databases**. *See also* **master**; **MySQL**; **slaves**  
 Netrastructure, 18  
 RDBMS, 4, 16, 18, 85, 148, 245, 246, 293, 294  
 storing files in, 675

**datadir**, 813–814

**Data::Dumper**, 268, 328, 516, 530, 592, 666

**data-modification language (DML)**, 18, 26

**data\_sources()**, 248, 283

**DATE\_ADD()**, 67

**DATE\_FORMAT()**, 68

**DATE\_SUB()**, 67

**date/time functions**, 66–70

**Day, Eric**, 11, 740, 741. *See also* Gearman

**db**, 79

**dbDelete()**, 655–656

**DBD::mysql**, 9, 27, 50, 205, 245, 246–247  
 DBI and, 246–247  
 installation, 247

**dbGetRef()**, 652–653, 672

**DBI (Database Independent Interface)**, 245–283. *See also* contact list application  
 API, 247  
 connect(), 249–254  
   arguments, 249  
   \$attributes, 253–254  
   \$dsn, 249–253  
   \$password, 253  
   \$username, 253  
 connect\_cached, 254–255  
 database handle methods, 274–277  
 DBD::mysql and, 246–247  
 driver methods, 248–249  
 fetch methods, 258–261  
 installation, 247  
 loading, 248  
 multi-step utility methods, 269–274  
 statement handles, 255–261  
 stored procedures, 277–279  
 transactional methods, 276–277

**DBI::connect()**. *See* **connect()**

**dbinsert()**, 653–654

**DBIx::Password**, 572, 719  
 contact list application and, 503, 529–530, 532, 533

**dbUpdate()**, 654–655, 671

**\$DB\_VIRTUAL\_USER**, 530

**db\_virtual\_user parameter**, 719

**DDL (data definition language)**, 26

**Debian**, 20, 247. *See also* apt-get  
 Apache configuration on, 484–486  
 Apache installation and, 427–428  
 memcached and, 322  
 mod\_perl installation and, 427–428  
 startup scripts, 322

**debug()**, 232

**\_DEBUG**, 231–232

**debug option**, 324

**debugging**  
 dump\_results and, 265  
 Federated and, 130  
 search (utility program) and, 159

**declarations, contact list and**, 287–290

**decr()**, 326

**decrement function (memcached)**, 371

**DefaultIcon**, 457

**DEGREES()**, 62

**delayed inserts**, 37–38

**delete()**, 326

**DELETE**, 52–56, 97, 125

**DELETE FROM**, 53, 54

**delete function (memcached)**, 372

**delete\_contact()**, 303–304  
 testing, 306–309

**deleteMemcUIDList()**, 558–559

**deleteSession()**, 673

**deleteUser()**, 342

**deleteUserFromCache()**, 554–555

**deleteUsers()**, 545–547

**deleting contacts**, 285, 303–304, 306–309

**deleting data**, 52–56

**deletion, cascade**, 55, 332

**delimiters**, 88, 96, 97, 194  
 bar (|), 96, 97  
 colons, 250  
 double, 194, 235  
 semi, 30, 32, 88, 96, 97, 250  
 comma-delimited data, 22, 388  
 ideographic languages and, 152  
 tab-delimited data, 22

- DenierHandler**, 608, 611, 612, 613, 614, 630
- derived classes**, 212
- DESCRIBE**, 32
- design (contact list application)**, 505–513
  - conceptualizing, 505–506
  - contact view page, 513
  - main user form, user listing, 507–510
  - program requirements, 506
  - user edit form, 510–513
  - user interface, 506–513
- desktop applications, web applications as**, 707, 708.
  - See also *Ajax*
- DESTROY()**, 223, 224, 228
- DETERMINISTIC**, 88
- deterministic caching**, 318, 346, 352
- DIAB**, 17
- die()**, 224
- Digest Authentication**, 496
- digestCookieValue()**, 668, 670
- Digest::MD5**, 334, 616, 617, 732, 757, 779, 787
- Digg.com**, 4
- directives. See also specific directives**
  - Apache
    - authorization, 451–453
    - basic, 440–444
    - CGI, 457–458
    - client handling, 462
    - Configuration Section Container Directives, 436–440
    - error, 448–449
    - filter, 461–462
    - handler, 460–461
    - .htaccess file, 453–454
    - indexing, 454–457
    - logging, 446–448
    - mod\_proxy, 481–483
    - rewrite, 468–478
    - server tuning, 444–446
    - SSL, 463–466
    - User Directory, 458–459
    - virtual hosting, 459–460
    - mod\_perl
      - configuration, 569–575
      - handler, 575–585
  - directories**, 186. See also *files*
  - <Directory>**, 436
  - DirectoryIndex**, 454
  - <DirectoryMatch>**, 436
  - disconnect\_all**, 328
  - disk partitions, raw**, 119, 816
  - dispatch()**, 290, 291, 292
  - dispatch table**, 172
  - dispatcher() subroutine**, 519–522
    - modification, 638–641
  - display functions**, 507–513. See also *specific display functions*
  - display returned HTML table (Ajax example)**, 711–713
  - display\_menu()**, 290, 291, 292
  - distributed indexes**, 10, 152, 154, 155, 157, 158, 160, 752, 753, 755, 772, 775
  - dist\_urls**, 755
  - div DOM element**, 730
  - <div> tag**, 708, 709, 711
  - DML (data-modification language)**, 18, 26
  - do()**, 270
  - documentation**
    - Felidea, 233–234
    - modules, 197–201
    - POD, 197–201, 233, 366, 518, 574–575, 746
  - DocumentRoot (directive)**, 441
  - dog-piling**, 352–355, 357
  - dollar sign (\$), in JavaScript**, 726
  - DOM (Document Object Model)**, 738
    - Apache2::Directive and, 592
    - API object, 708
    - div, 730
    - firebug and, 650
    - JavaScript Prototype library and, 725, 726
    - objects, 316
    - onQueue() and, 782
    - tr, 786
    - userlist within, 711
  - Domestic Cat**, 212, 213, 235, 236
  - doMsg()**, 708, 710
  - do\_task()**, 747
  - double colon**, 194, 235
  - driver, Perl**, 246. See also *Memcached::libmemcached*
  - driver handle**, 249
  - driver methods (DBI)**, 248–249
  - Drizzle**, 13, 421
  - DROP DATABASE**, 281
  - DROP USER**, 80
  - drop\_contacts\_table()**, 292
  - dropdb**, 281
  - Drupal**, 8, 479, 480
  - \$dsn**, 249–253
  - DSOs (Dynamically Shared Object files)**, 419, 433
  - dual master configuration**, 135
  - duct tape**, 164, 419
  - dump\_results**, 265
  - durability**, 19, 121. See also *ACID*
  - dynamic scoping**, 190
  - Dynamically Shared Object files (DSOs)**, 419, 433

## E

- edit\_contact()**, 297–301
  - testing, 300, 304–305
- editing contacts**, 285, 297–301, 304–305
- emptiness**, 30
- empty strings**, 30, 31. See also *NULL*
- emulated prepared statements**, 250, 256, 257
- encapsulation**, 212, 217, 218, 222, 224, 237
- encodeCookieValue()**, 670
- encode\_json()**, 716
- encodeUserData()**, 562–563
- END**, 88
- Enterprise Monitor, MySQL**, 824
- equi-join**, 44
- ErrCount**, 254, 265
- error directives**, 448–449
- error handling**
  - automatic, 280–281
  - contact list application and, 506
  - HandleError, 254, 265
  - manual, 279–280
  - PrintError, 254, 265, 280
  - RaiseError, 254, 265, 280
- ErrorDocument**, 448–449
- ErrorLog**, 447
- errstr()**, 256, 279, 373
- excerpt**, 772
- execute()**, 255, 256, 257, 258
- expiration**, 313, 325, 330, 353
- expire-logs-days**, 137
- explicit inner join**, 44
- @EXPORT**, 196, 198, 214, 235, 662, 736

## Exporter

**Exporter**, 196, 197  
**@EXPORT\_OK**, 214, 235  
**extensibility, Apache**, 417, 419–420  
**eXternal Subroutine. See XS**

## F

**Facebook**, 7  
**Falcon**, 18, 85, 112  
**FastCGI**, 480, 482  
**Fawcett, Joe**, 707  
**feature pragma**, 208  
**Federated (storage engine)**, 112, 125–130  
   characteristics of, 125–126  
   under the hood, 129–130  
   table creation, 126–128  
**Federated Servers**, 128–129  
**FederatedX**, 113  
**Fedora**, 322, 428, 804. *See also* Redhat-based systems  
**feeds**  
   RSS, 316  
   XML, 47, 67, 103, 316  
**Felidea class**  
   attributes  
     retrieving, 217–218  
     setting, 218–221  
   AUTOLOAD() and, 221–231, 244  
   changes (summary) to, 224, 228, 230–231  
   constructor creation, 214–217  
   \_DEBUG, 231–232  
   directory  
     creating, 236  
     structure, 235  
   documentation, 233–234  
   furColor(), 218  
   get\_ <attribute>, 226–227  
   hasFur(), 217  
   inheritance and, 234–240  
   methods  
     attackPrey, 231, 233, 239  
     attribute accessor, 217–221  
     makeSound, 231, 233, 239  
   Moose and, 240–243  
   object hierarchy, 213, 235  
   package creation, 214  
   set\_ <attribute>, 226–227  
   subclasses, 234–240  
**Felidea.pm**, 214, 217, 218, 230, 231, 233, 235  
**Felinae**, 213  
**fetch()**, 258, 273  
**fetch methods**, 258–261  
**fetchall\_arrayref**, 259–260, 310, 717  
**fetchall\_hashref**, 260–261  
**fetch\_remote\_doc()**, 757, 759  
**fetchrow\_arrayref**, 258  
**fetchrow\_hashref**, 258  
**\$fields**, 289, 290, 295, 299, 309  
**\$field\_separator**, 265  
**file handles**, 168, 182  
   directory, 185  
   functions and, 182–183  
   naming, 182  
   process, 185–186  
**file system caching**, 318  
**file upload mod\_perl handler**, 675–685  
   database table, 676  
   handler() subroutine, 677–680  
   implementation, 676–682

  saveUpload() subroutine, 680–682  
   storing files in database, 675  
   uploads directory in Apache configuration, 676–677  
   using, 685  
   WebApp methods and, 682–684  
**fileExists()**, 683  
**\$file\_handle**, 265  
**files. See also specific files**  
   directories v., 186  
   opening, 182–183  
   reading, 183–184  
   writing to, 184  
**<Files>**, 437  
**<FilesMatch>**, 437  
**film\_main**, 157, 158, 160, 161  
**film\_text**, 150, 151, 156, 157, 161  
**filter chain**, 420, 423, 579, 586, 628  
**filter directives**, 461–462  
**filter handlers (mod\_perl)**, 579–580  
   example, 627–630  
**filter record Apache2 Perl modules**, 586, 590–591  
**filtering**  
   Apache and, 420, 422, 423, 566  
   mod\_perl handlers and, 627–630  
   PHP and, 627  
   replication and, 136  
   Template Toolkit and, 687  
**find\_contact()**, 309–310  
   testing, 310–312  
**finish()**, 261, 272, 277  
**Firebug**, 650, 750  
**Firefox**, 648, 649, 709  
**Fitzpatrick, Brad**, 11, 315, 317. *See also* Danga Interactive;  
   Gearman; memcached  
**Fixup**, 576  
**flags**, 20  
   RewriteCond and, 471–472  
   substitution and, 473–477  
**Flash**, 1  
**FLUSH TABLES**, 25, 131, 132  
**FLUSH TABLES WITH READ LOCK**, 820, 822  
**flush\_all**, 328  
**foo table**, 415–416  
**footer display function**, 507  
**foreign keys**, 119  
**for/foreach loop**, 174  
**forward slash**, 129, 603, 604  
**Franks, John**, 15  
   .frm files, 115, 116, 121, 129, 131, 813, 821  
**from\_json()**, 725  
**FULLTEXT indexes**, 149–152  
   ideographic languages and, 152  
   issues, 151–152  
   Sphinx v., 152  
   using, 150–151  
**full-text indexing**  
   MySQL, 754  
   search engine application and, 739, 747, 748, 749, 752,  
     754, 792  
   Sphinx, 11, 152, 739, 747, 748, 749, 752, 754, 792  
**full-text searching**, 148–161  
**func()**, 281  
**functions**, 59–70. *See also* Memcached Functions for MySQL;  
   subroutines; UDFs; *specific functions*  
   aggregate, 60–62  
   arrays and, 176–177  
   behavioral, libmemcached, 366–369  
   complete listing, 59, 71  
   connection, memcached, 366

## handler() subroutine

- control flow, 68–70
  - date, 66–68
  - date/time, 68–70
  - display, 507–513
  - file handles and, 182–183
  - geometrical, 62
  - get, memcached, 370–371
  - informational
    - libmemcached, 372
    - MySQL, 59–60
  - logarithmic, 62
  - memcached
    - data retrieval (get), 370–371
    - increment/decrement/delete, 371–372
    - informational, 372
    - setting values, 369–370
    - utility, 372
  - Memcached Functions for MySQL
    - behavioral, 397–400
    - decrementing, 396–397
    - establish connection to memcached Server, 388–389
    - fetching, 295
    - incrementing, 395–396
    - setting values, 389–394
    - statistical, 400–401
    - version, 401–402
  - numeric, 62
  - raising number to power and, 62
  - reusable code and, 86
  - stored procedures and, 86–94
  - string, 62–66
  - string comparison, 63–64
  - furColor()**, 218
- G**
- GaiaOnline**, 7
  - Galbraith, Patrick**, 383. *See also* Memcached Functions for MySQL
  - Gallery**, 482
  - GCC compiler**, 2, 807
  - gcc/g++**, 807
  - Gear6**, 6, 7
  - Gearman**, 11, 13, 317, 740–747. *See also* memcached; search engine application
    - installation, 741–742
    - Perl and, 746–747
    - visual diagram, 740
  - Gearman job server**. *See* gearmand
  - Gearman MySQL UDFs**, 11, 12, 741
    - installation, 741–742
    - using, 744–746
  - Gearman workers (Perl)**, 746–747
    - data storage worker, 748, 749, 750, 751, 759, 762–764
    - indexer worker, 748, 749, 759, 760, 764, 765
    - running, 764–766
    - search engine application and, 756–766
    - web client worker, 748, 749, 750, 757–762
  - Gearman::Client**, 721, 740, 741, 746, 747
  - gearmand (Gearman job server)**, 11, 12, 317, 739, 740
  - installation**, 741
  - running**, 743–744
  - Gearman::Worker**, 721, 740, 746, 757, 761, 762, 764
  - generic database methods (mod\_perl)**, 651–662
  - geographical information**. *See* user application
  - geometrical functions**, 62
  - get()**, 325
  - get functions, memcached**, 370–371
  - get\_<attribute>**, 226–227
  - getCities()**, 345, 347, 348, 349
  - getCity()**, 348, 349, 350
  - get\_contacts()**, 295, 296
  - getFiles()**, 682–683
  - get\_if\_modified()**, 760
  - getMemc()**, 354, 355
  - getMemcUIDList()**, 559–560
  - get\_multi()**, 325
  - GetOptions**, 289
  - get\_results\_from\_db()**, 775–776
  - getSession()**, 668
  - getSessionData()**, 671–672
  - getState()**, 337, 348, 349–350, 561–562
  - getStates()**, 560–561, 728
  - getStatesFromDB()**, 345, 348, 349, 350
  - getUser()**, 335, 345, 350, 537–538
    - reusable code and, 537–538
  - getUserFromDB()**, 354, 355
  - getUsers()**, 534–537
  - getUsersFromCache()**, 552–553
  - getXmlHttp()**, 711, 714, 717, 723
  - global**, 190
  - global system user, MySQL**, 75–76
  - globbing**, 184
  - glue code, XS as**, 9, 246
  - glue hash trickery**, 49
  - Gmail**, 708
  - gman\_do()**, 745
  - gman\_do\_background()**, 745, 749
  - gman\_do\_task\_background()**, 749
  - gman\_severs\_set()**, 744
  - gman\_sum()**, 745
  - gnu make**, 807
  - GNU Project**, 2
  - GNU Public License (GPL)**, 2, 152, 793
  - Google Maps**, 708
  - Google Suggest**, 708
  - GPL (GNU Public License)**, 2, 152, 793
  - GRANT**, 76, 81–83
  - GRANT OPTION**, 81
  - grant tables**, 76–80, 81, 84
  - greater than or equal (>=) operator**, 72
  - Grimmler, Lenz**, 822
  - GROUP BY**, 43
  - GROUP BY, ROLLUP**, 61
  - GROUP BY 'creation date' ORDER BY 'creation date'**, 103
  - GROUP\_CONCAT()**, 61
  - grouping**, 43
  - grouping optimization**, 208
  - gzip**, 112, 123, 124, 152, 205, 385
- H**
- hack, cookies as**, 419
  - handle methods, database**, 274–277
  - handle\_cookies()**, 645
  - HandleError**, 254, 265
  - handler directives**. *See also* mod\_perl handlers
    - Apache, 460–461
    - mod\_perl, 575–585
  - handler phase**, 422, 576
  - handlers**. *See* mod\_perl handlers
  - handler() subroutine**, 677–680, 732–734, 778–780, 787–790

## handles

**handles.** *See* database handles; file handles

`has`, 242

`hasFur()`, 217

**hash references**, 171

**hashes**, 49, 167

- as arrays, 179–180
- dispatch table and, 172
- glue hash trickery, 49
- iteration, 179
- keys and, 179, 180
- using, 179–181

**hashing, consistent**, 360, 367

**Header**, 454, 462

**header display function**, 507, 508

**Header Parser**, 421

**Header parser hook**, 422

**HeaderName**, 455, 456

**HeaderParser**, 576

**headers, tools for testing**, 649–651

**header()** subroutine, 518

**help facility (MySQL)**, 59, 70–72

`HELP SHOW`, 33

**herd, thundering**, 352–355, 357

**Herman, Eric**, 9

**hooks**, 422, 423, 575, 601

**host**, 76, 80

**Hotbackup, ibbackup (InnoDB)**, 822–823

**Hotmail**, 708

**.htaccess file**, 453–454

**HTML design (contact list application)**, 506–507. *See also* contact list application

**HTML::Entities**, 530

**HTML-esque tags**, 694–695

**HTML::Mason.** *See* Mason

**HTML::Template**, 686, 694–698

- `mod_perl` handler (example) and, 695–697
- tags, 694–696
- template for, 697–698

**HTTP (Hypertext Transfer Protocol)**, 4, 8, 9, 106, 417–418.

*See also* Apache

- authentication, setup for, 495–496
- web applications and, 419

**HTTP request cycle phase handlers**, 580–585, 612–630

- names, 582–584
- template, 581

**HTTP request phase cycle**, 565, 580, 581, 584, 590, 599,

- 601, 626, 630, 631
- visual diagram, 584–585

**http\_get()**, 106, 108–109, 110, 111, 741

**http\_get\_deinit()**, 109–110

**http\_get\_init()**, 106, 108

**Hypertable**, 13

**Hypertext Transfer Protocol.** *See* HTTP

**ibbackup, Hotbackup (InnoDB)**, 822–823

**identifying references**, 172–173

**ideographic languages**, 152

**idx\_delta**, 154, 155

**idx\_dist**, 154, 155

**idx\_part1**, 154, 155

**idx\_part2**, 154, 155

**IF()**, 68

`<IfDefine>`, 438

`<IfModule>`, 438

**IFNULL()**, 68

`<IfVersion>`, 438

**illusion of state**, 419

**implementation (object orientation)**, 212

**Include (directive)**, 441

**include path**, 194, 195, 636

**incr()**, 326

**increment function (memcache)**, 371

**Indexed Sequential Access Method (ISAM)**, 115. *See also* MyISAM

**indexer (program)**, 152

- section, 155

**indexer worker**, 748, 749, 759, 760, 764, 765

**indexes**, 31–32. *See also* distributed indexes; full-text indexing

- b-tree, 149
- clustered, 118
- `FULLTEXT`, 149–152
  - issues, 151–152
  - using, 150–151

- 64-bit, 153

- `sphinx.conf`, 154–155

**IndexIgnore**, 457

**indexing directives**, 454–457

**IndexOptions**, 455–456

**IndexOrderDefault**, 456

**index.pl**, 764

**India.** *See also* NULL

- Brahmasputha Siddhanta*, 30

- Buddhist philosophy, 30

- cities (table), 278, 279

- states (table), 278

- Vedic philosophy, 30

**information schema**, 33, 83–84

**informational functions**

- `libmemcached`, 372

- MySQL, 59–60

`INFORMATION_SCHEMA`, 33, 83

`information_schema`, **contact list and**, 293–294

**inheritance**, 212–213

- @ISA array and, 197, 221, 234, 235, 237, 239, 244

- data sources and, 154

- diagram, 213

- Felidea subclassing and, 234–240

- in Perl, 244

- reusable code and, 240

**initializations, contact list and**, 287–290

**initialize()**, 288, 289, 290, 295

**injection attacks**, 256, 257, 717

**in-line <Perl> Sections**, 699–700

**inner joins**, 44, 47

**InnoDB Oy**, 118, 822

**InnoDB**, 18, 112, 118–123. *See also* tablespace files

- configuration, 119–120

- features, 118–119

- under the hood, 120–121

- MyISAM v., 118–119

- table creation, 120

- transactional methods, 276–277

- transactions, 121–123

**InnoDB Hotbackup, ibbackup**, 822–823

**input parameters, binding**, 262–263

`INSERT`, 35–36, 97, 301–302

`INSERT IGNORE`, 37

**insert\_contact()**, 286, 299, 300, 301, 302, 303, 310

- testing, 305–306

**inserting contacts**, 285, 301–302, 305–306

**inserting data**, 35–38

`INSERT . . . ON DUPLICATE KEY UPDATE`, 57–58

**insertRow()**, 715

**inserts**

- bulk, 36–37

delayed, 37–38  
 low priority, 38  
**insertSession()**, 671, 672  
**insertUser()**, 542–543  
**Installation Wizard, MySQL**, 116, 795, 796  
**installations**  
   Apache, 424–432  
     on Apple OS X, 429  
     components, 424–425  
     on Redhat-based systems, 428–429  
     on Ubuntu/Debian-based Linux, 427–428  
     on UNIX, 429–432  
     on Windows, 425–427  
   Cache::Memcached::libmemcached, 365  
   DBD::mysql, 247  
   DBI, 247  
   Gearman, 741–742  
   Gearman MySQL UDFs, 741–742  
   gearmand, 741  
   libapreq2, 434–435  
   libmemcached, 360–361  
   memcached, 318–321  
     on CentOS, 318–319  
     from source, 320  
     on Ubuntu, 319–320  
   Memcached Functions for MySQL, 385–388  
     checking of, 387–388  
   Memcached::libmemcached, 365  
   mod\_perl  
     on Redhat-based systems, 428–429  
     from source, 433–434  
     on Ubuntu/Debian-based Linux, 427–428  
     on Windows, 426–427  
   modules (Perl) and, 201  
   MySQL, 793–810  
     directory structure and, 20  
     MySQL version and, 793–794  
     package type, 794  
     on RPM-based Linux systems, 804  
     from source, 807–809  
     on Ubuntu, 804–806  
     on UNIX, 807–810  
     on Windows, 795–803  
   Sphinx, 152–153  
   UDFs (Memcached Functions for MySQL), 386–387  
     install.pl and, 386–387  
     SQL script and, 386  
**install\_driver()**, 281  
 install.pl, 386–387  
**instance methods**, 214  
**instance variables**, 215  
**instantiation**, 213  
   Cache::Memcached and, 324–325  
   for precaching method calls, 348  
   user application, 334  
**interface**, 212  
   object-oriented, 373  
**Internet. See also computers; Web**  
   Internet-centric systems, 1  
   as necessity, 1  
   “network is the computer”, 1  
**INTERPOLATE**, 691, 728, 729  
**IO threads**, 134  
**IO::Prompt**, 289, 291  
**IP-based virtual hosting**, 460  
**IPC::Sharable**, 2, 315. *See also* memcached  
**is-a relation**, 197, 213, 244. *See also* @ISA array  
**@ISA array**, 197, 221, 234, 235, 237, 239, 244. *See also*  
   inheritance

**ISAM (Indexed Sequential Access Method)**, 115. *See also*  
   MyISAM  
**isAuthenticated()**, 669  
**is\_junk()**, 760  
**isolation**, 19, 122. *See also* ACID  
**is\_young()**, 89  
 items, 103  
**iteration**  
   hash, 179  
   for loop, 207  
   references and, 181  
   sort() and, 177

## J

**Java servlets**, 1  
**JavaScript**, 1, 2. *See also* Ajax  
   Ajax and, 707, 738  
   functions, 781  
     implementation of, 781–785  
   templating systems and \$ character in, 726  
   URLHandler and, 780–781  
**JavaScript Framework, Prototype**, 708, 732  
   Ajax example, 725–727  
   purpose of, 725  
**JavaScript Object Notation. See JSON**  
**job\_servers()**, 747, 761  
**join()**, 176–177  
**join predicate**, 44  
**joins**, 43–44  
   equi-, 44  
   explicit inner, 44  
   inner, 44, 47  
   LEFT, 45, 46, 47  
   natural, 44, 45  
   RIGHT, 45, 46  
   self-, 40, 47, 49  
   types of, 44  
**Joomla**, 8, 482  
**JSON (JavaScript Object Notation)**, 708  
   cache, 316  
   data structure, 717, 718, 719  
   Perl module, 713–717, 738, 787  
   search engine application and, 789–790  
**JSON to Perl**, 713  
**JSON::XS**, 713, 714, 716

## K

**Karsson, Lars**, 17  
**KeepAlive (directive)**, 446  
**KeepAliveTimeout**, 446  
**\$key\_field**, 273  
**keys**  
   foreign, 119  
   hashes and, 179, 180  
   K/n, 360  
   main, 353  
   primary, 31  
   stale, 353–355  
   statistics, 326  
**K/n keys**, 360  
**Koenig, Andreas**, 8  
**Kovrin, Alexy**, 353. *See also* dog-piling

## LAMMP (Linux, Apache, MySQL, memcached, Perl)

### L

**LAMMP (Linux, Apache, MySQL, memcached, Perl)**, 1–13.

See also Apache; Linux; memcached; mod\_perl; MySQL; Perl

Apache, 3–4  
 architectural implementation, 11–12  
 future of, 12–13  
 Gearman, 11, 13, 317  
 Linux, 2–3  
 memcached, 2, 5–8, 13  
 MySQL, 4–5  
 Perl, 8–10  
 Sphinx, 10

**LAMP stack**, 3, 5, 13

**languages**. See *specific languages*

**large file support, Apache**, 421

**Larsson, Allan**, 17

**last subscript value, of array**, 177–178

**LAST\_INSERT\_ID()**, 60

**last\_insert\_id**, 275

**“lazy” processing/caching**, 315, 316

**least recently used**. See **LRU**

**LEFT joins**, 45, 46, 47

**left shift operator**, 58

**LENGTH()**, 63

**Lerdorf, Rasmus**, 16

**less-than operator**, 41

**lexical scoping**, 190

**lexx**, 807

**libapreq2 (Apache request library)**, 424, 428, 486, 565, 566  
 a2enmod and, 428, 467, 480, 486  
 cookies and, 424, 565  
 enable, 428  
 installing, 434–435

**libcurl**, 106, 107, 109, 110, 162, 383

**-libdir**, 385

**libevent**, 318, 319, 320

**libmemcached**, 7, 198, 319, 359–381  
 behavioral functions, 366–369  
 Cache::Memcached v., 359  
 features, 360  
 informational functions, 372  
 installing, 360–361  
 Perl support for, 359, 364–365  
 project page for, 361  
 utility programs, 361–364  
 list, 360

**libmysql**, 246, 252, 256

**libncurses**, 807

**libtool**, 807

**life cycle**. See **Apache life cycle**

**LIKE operator**, 41, 63

**LIMIT**, 40, 50, 53

**<Limit>**, 438–439

**<LimitExcept>**, 439

**\$line\_ separator**, 265

**Linux**. See also **CentOS; Debian; Redhat-based systems; RPM; SUSE; Ubuntu; UNIX**  
 Apache and, 419  
 DBI/DBD: :mysql and, 247  
 LAMMP and, 2–3  
 MySQL and, 4  
 Perl modules and, 205  
 plug-ins, 419  
 RPM-based, MySQL install on, 804

**Lion class**, 212, 213, 235, 237, 238, 239, 240, 242, 244

**list\_contacts()**, 294–297

**Listen (directive)**, 441–442, 756

**listing contacts**, 285, 294–297

**lists, arrays v.**, 174

**LiveHTTPHeader**, 650–651, 705

**LiveJournal.com**, 4, 6, 7, 9, 11, 165, 315

**LoadModule (directive)**, 442

**localtime()**, 670

**<Location>**, 437

**<LocationMatch>**, 437

**locking, row-level**, 113, 119

**Log (mod\_perl request phase handler)**, 576

**logarithmic functions**, 62

**-log-bin**, 137

**-log-bin -logbin**, 137

**-log-bin-index**, 137

**LogFormat**, 447–448

**logging Apache2 Perl modules**, 586, 592

**logging directives**, 446–448

**Logical Volume Manager (LVM)**, 821–822

**loginForm()**, 666

**LoginHandler**  
 implementation, 663–667  
 testing, 673  
 WebApp class methods and, 667–670

**login\_user**, 91

**logs**  
 binary, 134, 141  
 relay, 26, 134, 138

**log-slave-updates**, 137, 142

**-log-slave-updates**, 137

**LogTestHandler**, 623, 626, 628, 631

**lookup, of contacts**, 285, 309–312

**low priority inserts**, 38

**LRU (least recently used)**, 313, 318, 353, 559. See also memcached

**Lutonen, Air**, 15

**LVM (Logical Volume Manager)**, 821–822

**LWP**, 205, 757

**Lynx class**, 212, 213, 235, 236

### M

**\$m**, 700, 701, 703

**-m**, 321

**MacEachern, Doug**, 8

**main()**, 191, 290, 517, 567, 635, 636, 637, 638, 641, 642

**main key**, 353

**mainform display function**, 507  
 interface for, 507–508

**mainform() subroutine**, 522–523, 641

**main() subroutine**, 517–518

**make install**, 110, 111

**\_makeInsertStatement()**, 341

**make\_label()**, 295

**makeSessionId()**, 670

**makeSound**, 231, 233, 239

**\_makeUpdateStatement()**, 341

**\_makeWhereClause()**, 336, 337

**Maki, Daisuke**, 365, 375

**MANIFEST file**, 204–205

**manual error handling**, 279–280

**map operator**, 175

**MapReduce**, 11, 13

**MaptoStorage**, 576

**Maria (storage engine)**, 13, 18, 19, 25, 26, 85, 112, 149

**MariaDB**, 13

**Mason (HTML::Mason)**, 686, 698–704  
 components and, 698, 700–702

## memc\_libmemcached\_version()

- initialization/cleanup, 702
- objects, 700
- reusable code and, 686
- syntax, 698–700
- underlisting page in, 703–704
- master (database instance)**
  - binary log, 134, 141
  - Blackhole storage engine and, 136
  - dual configuration, 135
  - replication
    - multiple, 139–141, 162
    - overview, 133–134
    - schemes, 134–137
  - ring configuration, 135, 136
- master-host**, 138
- master-password**, 138
- master-port**, 138
- master-ssl –master-ssl-ca=file\_name**, 138
- master-ssl-capath=directory\_name**, 138
- master-ssl-cert=file\_name**, 138
- master-ssl-cipher=cipher\_list**, 138
- master-ssl-key=file\_name**, 138
- master-user**, 138
- MATCH() ... AGAINST**, 149
- math operator**, 58
- MAX()**, 60
- MaxClients (directive)**, 444
- MAX\_CONNECTIONS\_PER\_HOUR** count, 82
- MaxKeepAliveRequests (directive)**, 446
- \$maxlen**, 265
- MAX\_QUERIES\_PER\_HOUR** count, 81
- MaxRequestsPerChild (directive)**, 445
- MaxRows**, 271, 272
- MaxSpareServers (directive)**, 444
- MAX\_UPDATES\_PER\_HOUR** count, 81
- MAX\_USER\_CONNECTIONS** count, 82
- Maya**, 30
- McCool, Rob**, 15, 418
- McPeak, Jeremy**, 707
- MD()**, 92
- md5**, 92, 93
- Mediawiki**, 8
- MED/SQL specification**, 125, 128
- memcached**, 313–357. *See also* libmemcached
  - basic concept, 313–315, 357
  - benefits of, 357
  - Cache::Memcached and, 323
  - complex data types and, 329–330
  - connection, contact list application and, 533–534
  - connection functions, 366
  - connector method, 344–345
  - decrement function, 371
  - delete function, 372
  - development of, 315
  - dog-piling and, 352–355, 357
  - examples, 328–331. *See also* user application
  - expiration and, 313, 325, 330, 353
  - Gearman and, 11, 13, 317
  - get functions, 370–371
  - increment function, 371
  - installing, 318–321
    - on CentOS, 318–319
    - from source, 320
    - on Ubuntu, 319–320
  - LAMMP and, 2, 5–8, 13
  - LRU, 313, 318, 353, 559
  - /MySQL, contact list application and, 503–504. *See also* contact list application
  - NULl and, 393, 394, 412
  - operations, 325
  - options, 321
  - scalar example and, 328–329
  - starting, 321–323
  - startup scripts, 322–323
    - Debian-based, 322
    - Redhat-based, 322–323
  - UNIX and, 322
  - value setting functions, 369–370
- memcached cluster**, 323
  - diagram, 314
  - replicating data to, 355–357
- memcached functions**
  - data retrieval (get), 370–371
  - increment/decrement/delete, 371–372
  - informational, 372
  - setting values, 369–370
  - utility, 372
- Memcached Functions for MySQL**, 7, 11, 105, 383–416
  - features, 384
  - functionality of, 384
  - functions
    - behavioral, 397–400
    - decrementing, 396–397
    - establish connection to memcached Server, 388–389
    - fetching, 295
    - incrementing, 395–396
    - setting values, 389–394
    - statistical, 400–401
    - version, 401–402
  - installation, 385–388
    - checking of, 387–388
    - overview of, 383–384
    - prerequisites for, 385
  - read-through caching, SELECT and, 412–415
  - single database handle (example), 403–409
    - changes to connection, 403–405
    - changes to getUser(), 405–407
  - source
    - compiling of, 386
    - configuration of, 385–386
    - triggers and, 403, 409–412
  - UDF install, 386–387
    - install.pl and, 386–387
    - SQL script and, 386
  - usage
    - examples, 402–412
    - function descriptions and, 388–402
- memcached UDFs**, 383. *See also* Memcached Functions for MySQL
- Memcached::libmemcached**, 359, 364
  - Cache::Memcached::libmemcached v., 365–366, 375
  - examples
    - object-oriented approach, 374–375
    - procedural usage, 373–374
  - installing, 365
- memc\_add()**, 391
- memc\_append()**, 393–394
- memcat**, 360, 361–362
- memc\_behavior\_get()**, 398
- memc\_behavior\_set()**, 398–399
- memc\_cas()**, 392
- \$MEMC\_CATALOGUE\_EXPIREY**, 531
- memc\_decrement()**, 396–397
- \$MEMC\_DEFAULT\_SERVERS**, 530
- memc\_delete()**, 394
- memc\_get()**, 395, 749, 763
- memc\_increment()**, 395–396, 749
- memc\_libmemcached\_version()**, 402

## memc\_list\_behaviors()

**memc\_list\_behaviors()**, 397  
**memc\_list\_distribution\_types()**, 400  
**memc\_list\_hash\_types()**, 399–400  
**memcp**, 360, 362  
**memc\_prepend()**, 393  
**memc\_replace()**, 391–392  
**memc\_server\_count()**, 389  
**memc\_servers\_set()**, 388–389, 759  
**memc\_set()**, 389–390  
**memc\_set\_by\_key()**, 390–391  
**memc\_stat\_get\_keys()**, 401  
**memc\_stat\_get\_value()**, 401  
**memc\_stats()**, 400–401  
**memc\_udf\_version()**, 402  
**memerror**, 364  
**memflush**, 360, 362  
**Memory/Heap**, 112  
**memrm**, 360, 363  
**memslap**, 360, 363–364  
**memstat**, 360, 362–363  
**Merge**, 112  
**meta-data**, 16, 675  
   BrowserMatch and, 454, 462  
   comment\_log.ARM, 124  
   contacts.CSM file, 131  
   CookieName, 454, 467  
    $\$dSn$  and, 249  
   Federated Servers and, 128  
 $\$method$ , 231, 254  
**methods. See also specific methods**  
   accessor. *See* accessor methods  
   binding, 261–264  
   Cache::Memcached, 325–328  
   class, 217  
   contact list application  
     caching methods, 549–560  
     database methods, 542–548  
     WebApp class methods, 529–542  
   database (mod\_perl), 651–662  
   database handle, 274–277  
   defined, 212, 214  
   fetch, 258–261  
   instance, 214  
   multi-step utility, 269–274  
   object, 214  
   Perl, 214  
   Sphinx::Search, 772–773  
   statement handle. *See* statement handles  
   static, 214, 215, 217  
   for storing session data, 670–675  
   transactional, 276–277  
   virtual, 217  
**mget**, 370, 371  
**Microsoft Windows. See Windows**  
**MIN()**, 60  
**mini SQL (mSQL)**, 16, 246  
**MinSpareServers (directive)**, 445  
**mod\_access**, 424  
**mod\_apreq2**, 435, 566  
**mod\_auth**, 424  
**mod\_auth\_dbm**, 424  
**mod\_authn\_alias**, 421, 424  
**mod\_authnz\_ldap**, 424  
**mod\_authz**, 424  
**mod\_authz\_host**, 424  
**mod\_autoindex**, 424  
**mod\_dav**, 423  
**mod\_deflate**, 423  
**mod\_headers**, 424

**mod\_imap**, 423  
**mod\_include**, 424  
**mod\_info**, 424  
**mod\_perl**, 1, 4, 8, 565–599, 633–705  
   Ajax and, 707–738  
   Apache2 Perl modules, 585–598  
   configuration, 566–569  
   contact list application and, 567–569  
   database methods, 651–662  
   development of, 8, 419  
   directives  
     configuration, 569–575  
     handler, 575–585  
   installation  
     on Redhat-based systems, 428–429  
     from source, 433–434  
     on Ubuntu/Debian-based Linux, 427–428  
     on Windows, 426–427  
   new features, 566  
   shared library, 566  
   templating and, 686–698  
**mod\_perl handlers**, 575, 601–631. *See also WroXHandler entries*  
   for Ajax examples  
     example 1, 710  
     example 2 (display returned HTML table), 711–712  
     example 3 (building table on client), 716  
     example 4 (MySQL Ajax client), 719–720  
     example 5 (reading raw POST data), 724–725  
   /ajax\_handler URL, 710, 711, 715, 724, 727, 732  
   Apache life cycle and, 8, 9, 576, 601, 612  
   categorization, 576  
   connection, 607–612  
   connection cycle, 578–579  
   conversion process  
     mod\_perl handler to ModPerl::Registry script, 641–643  
     ModPerl::Registry script to mod\_perl handler, 633–634, 635–641  
   directives, 575–585  
   file upload, 675–685  
   filter, 579–580  
     example, 627–630  
   HTML::Template and, 695–697  
   HTTP request cycle, visual diagram, 584–585  
   LoginHandler, 663–667, 677  
   ModPerl::Registry v., 633–634, 643  
   scope, 575  
   for search engine application, 766  
   server life cycle, 578  
     visual diagram, 584–585  
   stacked, 575  
   Template Toolkit and, 690–692  
   type, 575–576  
   using, 601–631  
**mod\_perl Search handler**, 768  
**ModPerl::Registry**, 9, 566, 599  
   conversion process  
     mod\_perl handler to ModPerl::Registry script, 641–643  
     ModPerl::Registry script to mod\_perl handler, 633–634, 635–641  
   mod\_perl handlers v., 633–634, 643  
   scripts, bomb alert, 643  
**ModPerl::RegistryLoader**, 634–635  
**mod\_proxy**, 423  
   directives, 481–483  
   enabling, 480  
   reverse proxying and, 478–480  
**mod\_proxy\_load\_balancer**, 423

- mod\_rewrite**, 434, 454, 468–478, 501
  - regular expressions and, 468, 472, 473
  - transparency and, 468
- mod\_ssl**, 423
- modules**
  - Apache, 419–420
    - version 1.3/2.2, 423–424
  - Apache2 Perl, 585–598
  - APR, 585
  - Perl, 193–206
    - CPAN and, 205–206
    - documenting, 197–201
    - Linux and, 205
    - making installable, 201
    - MANIFEST and, 204–205
    - reusable code and, 194, 197
    - tests added to, 201–204
    - writing, 194–197
- modulus operator**, 58
- mod\_version**, 423
- monitoring MySQL**, 823–824
  - Cacti, 648, 823
  - MySQL Enterprise Monitor, 824
  - Nagios, 436, 823, 824
- Moose**, 240–243, 244
  - Perl and, 240
  - project website, 240, 243
- more\_results()**, 277, 278, 279
- MPM Event**, 420
- MPM NT**, 420
- MPM Prefork**, 420
- MPM Worker**, 420
- MPMs (Multi-threaded Processing Modules)**, 420
- mSQL (mini SQL)**, 16, 246
- Multifunctional Database**, 799
- multiple memcached servers, replicating data to**, 355–357
- multiple-key data fetch**, 415
- multiple-master replication**, 139–141, 162
- multi\_set()**, 356
- multi-step utility methods**, 269–274
- multitenancy**, 13
- multi-threaded**, 18, 27, 420
- Multi-threaded Processing Modules. See MPMs**
- multi-version concurrency control (MVCC)**, 113
- MVCC (multi-version concurrency control)**, 113
- my.cnf**, 122, 812
  - sample file, 824–827. *See also* replication
- MYD file**, 115, 116, 117, 813, 821
- my\_func()**, 106
- my\_func\_deinit()**, 106
- my\_func\_init()**, 106
- MYI file**, 115, 116, 117, 813, 821
- my.ini**, 122, 812
- MyISAM**, 13, 18, 25, 35, 38, 85, 112, 115–118, 799
  - features, 115–116
  - FULLTEXT indexes and, 151
  - under the hood, 116–117
  - InnoDB v., 118–119
  - table
    - creation of, 116
    - maintenance, 117–118
- mysamchk**, 25–26, 117, 118
- mylvmbackup**, 822
- my\_realloc()**, 106, 107
- my\_reverse()**, 747
- my\_reverse\_fn()**, 747
- myschema**, 134, 139
- MySQL**, 15–84, 85–162. *See also* Memcached Functions for MySQL; Sphinx
  - access control, 75, 84
  - account management statements, 80–84
  - advanced features, 85–162
  - backups, 817–823. *See also* mysqldump
  - basic functionality, 16–17
  - C/C++ and, 18
  - clustering and, 18
  - command executor, 19
  - command options
    - basic, 813–815
    - InnoDB, 815–816
    - port setting, 814–815
    - socket setting, 815
    - tablespace, 815, 817
    - user, 814
  - configuration, 811–829
    - running for first time, 811
    - schema creation, 811, 812
    - server configuration file, 812–817
    - setting privileges, 812
  - connection, contact list application and, 532–534
  - core components, 19
  - database application. *See* contact list application
  - features of, 18–19
  - FULLTEXT indexes, 149–152
  - functions, 59–70. *See also* Memcached Functions for MySQL; UDFs
    - aggregate, 60–62
    - complete listing, 59
    - control flow, 68–70
    - date, 66–68
    - date/time, 68–70
    - informational, 59–60
    - numeric, 62
    - string, 62–66
  - genesis of, 17–18
  - global system user, 75–76
  - help facility, 59, 70–72
  - InnoDB Hotbackup, ibbackup, 822–823
  - installation, 793–810
    - directory structure and, 20
    - MySQL version and, 793–794
    - package type, 794
    - on RPM-based Linux systems, 804
    - from source, 807–809
    - on Ubuntu, 804–806
    - on UNIX, 807–810
    - on Windows, 795–803
  - LAMMP and, 4–5
  - Linux and, 4
  - /memcached, contact list application and, 503–504.
    - See also* contact list application
  - monitoring, 823–824
    - Cacti, 648, 823
    - MySQL Enterprise Monitor, 824
    - Nagios, 436, 823, 824
  - operators, 58–59
    - Bit, 58
    - Boolean, 58
    - comparison, 71–72
    - complete listing, 58
    - left shift, 58
    - less-than, 41
    - LIKE, 41
    - math, 58
    - modulus, 58

## MySQL (continued)

### MySQL, (continued)

- ranges with, 41–42
- right shift, 58
- optimizer, 16, 19, 126, 250
- parser, 16, 19
- Perl and, 245–283
- primer, 16–19
- privileges, 74–84, 812
- programs, 19–28
  - client, 20–25
  - utility, 25–27
- RDBMS, 4, 16, 18, 148, 245, 246, 293, 294
- reference manual, 33, 38, 59, 62, 78, 81
- replication. *See* replication
- SQL features and, 85–111
- startup utilities, 27–28
- SUSE and, 794, 804
- table handler, 16, 17, 19
- UNIX and, 19, 20, 21, 27
- usage of, by websites, 4, 5
- user account, 74–75
- user manual, 68
- user-defined variables, 72–74, 84, 85, 162
- versions, 793–794
  - online information, 794
  - WAMP and, 3
- mysql, 20–21
  - grant tables, 76–80, 81, 84
- mysql>, 21
- MySQL database-driven application. *See* contact list application**
- MySQL Enterprise Monitor**, 824
- MySQL Installation Wizard**, 116, 795, 796
- MySQL Server Instance Configuration Wizard**, 798
- mysqladmin**, 21–22, 246, 812, 813
- mysqladmin shutdown**, 282
- mysqlbinlog**, 26–27
- mysql\_config**, 27
- mysqld**, 27
- mysqld1**, 28, 139, 140, 141, 814, 825
- mysqld2**, 28, 139, 140, 141, 815, 826
- mysqld\_multi**, 28, 139–141. *See also* replication
- mysqld\_safe**, 27
- mysqldump**, 21, 25, 40, 132, 133, 246, 803, 812, 818–819
- mysqlhotcopy**, 821
- mysqlimport**, 22–23
- mysql\_insertid**, 267
- mysql\_is\_autoincrement**, 267
- mysql\_is\_blob**, 267
- mysql\_is\_key**, 267
- mysql\_is\_num**, 267
- mysql\_is\_pri\_key**, 267
- mysql\_length**, 267
- mysql.server**, 27
- mysqlshow**, 23–25
- mysql\_type**, 267
- mysql\_type\_name**, 267
- mysub()**, 187, 190

## N

- Nagios**, 436, 823, 824
- NAME**, 266
- name-based virtual hosting**, 460
- NAME\_hash**, **NAME\_lc\_hash**, **NAME\_uc\_hash**, 266
- NAME\_lc**, **NAME\_uc**, 266
- namespace**, 192

- NameVirtualHost**, 460
- naming convention (Perl), underscore and**, 215, 217, 218, 220
- naming file handles**, 182
- Narada Muni**, 99, 100, 270
- NASA –JPL**, 4
- National Center for Supercomputing Applications (NCSA)**, 418. *See also* Apache
- Native switch()**, 209
- natural join**, 44, 45
- NATURAL LANGUAGE MODE**, 149
- NATURAL LANGUAGE MODE WITH QUERY EXPANSION**, 150
- NCSA (National Center for Supercomputing Applications)**, 418. *See also* Apache
- NDB Cluster**, 33, 112
- Netbas**, 17
- Netframework database**, 18
- netmask**, 82
- “network is the computer,”** 1
- new()**, 215, 216, 217, 224, 531–532
- NEW**, 100, 101
- \$new\_values**, 298, 299, 302
- nfreeze()**, 403, 667, 671, 672
- NONDETERMINISTIC**, 88
- nondeterministic caching**, 318, 346
- NOT LIKE**, 63
- nothingness**, 30
- notifyUser() subroutine**, 735
- NOW()**, 66, 67
- NULL**, 30–31, 45, 64, 68, 93
  - concept of, 30–31
  - empty strings v., 30, 31
  - memcached and, 393, 394, 412
  - MyISAM and, 115
  - zero v., 30, 31
- NULLABLE**, 266
- NULLIF()**, 68
- numeric functions**, 62
- NUM\_OF\_FIELDS**, 266
- NUM\_OF\_PARAMS**, 266

## O

- object methods**, 214
- object-oriented interface**, 373. *See also* Memcached::libmemcached
- object-oriented Perl**, 9, 211–244. *See also* bless(); Perl Moose and, 240–243
- object-oriented programming**, 212–213. *See also* classes; encapsulation; inheritance; methods; reusable code
  - benefits of, 211
  - CamelCase and, 221
  - implementation and, 212
  - interface and, 212
  - Perl and, 9, 244
  - polymorphism, 213
  - references. *See* references
  - reusable code and, 212
  - virtual method and, 217
- objects**
  - defined, 212, 214
  - hierarchy, Feldiea class, 213, 235
- ODBC**, 19, 114, 246, 249, 250, 266
- OLD**, 100, 101
- OLTP (Online Transaction Processing)**, 800
- onChange()**, 767, 781, 782, 784
- onClick()**, 784

**onclick()**, 710  
**one-liner regular expressions**, 206–207  
**Online Transaction Processing (OLTP)**, 800  
**onQueue()**, 781, 782, 790  
**onreadystatechange**, 714  
**onreadystatechange**, 709  
**onSuccess**, 732, 784  
**open source model**, 2, 12  
**open source web development, future of**, 12–13  
**opendir()**, 185  
**operators**  
   map, 175  
   MySQL, 58–59. *See also specific operators*  
   Bit, 58  
   Boolean, 58  
   case, 68  
   comparison, 71–72  
   complete listing, 58, 71  
   greater than or equal, 72  
   left shift, 58  
   less-than, 41  
   LIKE, 41  
   math, 58  
   modulus, 58  
   ranges with, 41–42  
   right shift, 58  
**\$ops**, 288, 289, 291  
**optimizations (regex)**, 208  
   grouping optimization, 208  
   regex compilation, 208  
**optimizer**, 16, 19, 126, 250  
**\$OPTIONS**, 221, 231, 237  
**\$options**, 215, 231  
**Options (directive)**, 442–443  
**\$opt\_XXX variables**, 289  
**or operator**, 209  
**Oracle**, 48, 112, 118, 150, 246, 822  
   Innobase Oy and, 118, 822  
**Order (directive)**, 451–452  
**ORDER BY**, 42, 44, 50, 51, 103, 124  
**ORDER BY 'creation date' GROUP BY 'creation date'**, 103  
**ordering**, 42  
**OS X**. *See Apple OS X*  
**output parameters, binding**, 263–264  
**overridden**, 212, 213, 217

## P

**package scoping**, 190  
**packages**, 192–206. *See also modules*  
   creating, 193  
   reusable code and, 211  
**page caching**, 318  
**Pantherinae**, 213  
**Pantherinea**, 212, 234, 235, 236, 237  
**ParamArrays**, 266  
**ParamTypes**, 266  
**ParamValues**, 266  
**parent**, 46, 47  
**parentheses, subroutines and**, 187  
**parse\_links()**, 761, 763  
**parser**, 16, 19  
**parse\_title()**, 761  
**partial page caching**, 318  
**pass by reference**, 169, 518  
**\$password**, 253  
**Password.pm**, 532  
**“patchy” server**, 418. *See also Apache*  
**pattern matching**, 63, 64, 206, 471, 473, 475  
**PBXT (Primebase XT)**, 18, 85, 113  
**PCRE (Perl Compatible Regular Expressions)**, 421  
**PEARL**, 163  
**percentage sign (%)**, wildcards and, 41, 64, 139  
**Percona**, 13, 823  
**performance Apache2 Perl modules**, 586, 594, 595  
**performance comparisons**  
   Cache::Memcached and, 377–380  
   Cache::Memcached::libmemcached and, 376, 381  
   test script/tool for, 377–380  
**Perl**, 163–210. *See also mod\_perl*; object-oriented Perl  
   Ajax and. *See Ajax*  
   Apache2 modules, 585–598  
   arrays. *See arrays*  
   benefits of, 9  
   **bless()**, 211, 214, 215, 216, 219, 223, 227, 236, 238, 244, 334, 348, 404, 532, 661  
   CGI programs and, 8  
   client library for memcached. *See Cache::Memcached*  
   complex data structures, 180–182  
   CPAN and. *See CPAN*  
   data types, 165–168  
     file handles, 168, 182  
     hashes, 49, 167, 172  
     scalars, 165–167  
     subroutines, 168, 186–188  
     type globs, 168  
     usage, 168–192  
   driver, MySQL and, 246  
   Gearman and, 746–747  
   global and, 190  
   glue hash trickery, 49  
   history of, 163–164  
   install utility (`install.pl`), 386–387  
   LAMMP and, 8–10  
   libmemcached and, 359, 364–365. *See also libmemcached*;  
     Memcached::libmemcached  
   modules. *See modules*  
   MySQL and, 245–283  
   primer, 163–210  
   prototypes and, 187  
   public/private and, 213, 217, 218  
   references. *See references*  
   scope and, 164  
   6.0, 13, 208–209  
   strengths of, 164  
   WAMP and, 3  
   writing classes in. *See classes*  
**Perl 6.0**, 13, 208–209  
**Perl Compatible Regular Expressions (PCRE)**, 421  
**Perl Data Structures Cookbook**, 180  
**Perl filter handlers**. *See filter handlers*  
**Perl Gearman client**, 747  
**Perl Gearman workers**. *See Gearman workers*  
**Perl module, JSON**, 713–717, 738, 787  
**<Perl> sections**, 569–570  
   inline, 699–700  
**Perl to JSON**, 713  
**PerlAccessHandler**, 577, 583, 585, 588, 630  
   Apache life cycle overview and, 577  
   example, 612–614  
**PerlAddVar**, 572

## PerlAuthenHandler

- PerlAuthenHandler**, 573, 574, 577, 583, 585, 615, 619, 620, 621, 626, 630
  - Apache life cycle overview and, 577
  - example, 615–619
- PerlAuthzHandler**, 573, 577, 583, 585, 619, 621, 626, 631
  - Apache life cycle overview and, 577
  - example, 619–622
- PerlChildExitHandler**, 577, 578
- PerlChildInitHandler**, 577, 578
- PerlCleanupHandler**, 577, 584, 585
- PerlFixupHandler**, 577, 583, 585
- PerlHeaderParserHandler**, 577, 582, 585
- PerlInitHandler**, 577, 582, 585, 594, 606, 704
- PerlInputFilter**, 577, 580, 585
- PerlLoadModule**, 571
- PerlLogHandler**, 574, 577, 584, 585
  - Apache life cycle overview and, 577
  - example, 622–627
- PerlMapToStorageHandler**, 577, 582
- PerlModule**, 570
- PerlMyAdmin**, 281
- PerlOpenLogsHandler**, 577
- PerlOptions**, 573–574
- PerlOutputFilter**, 577, 580, 585, 629, 631
- PerlPassEnv**, 572
- PerlPostConfigHandler**, 577, 578
- PerlPostConfigRequire**, 573
- PerlPostRead**, 582
- PerlPostReadRequestFilter**, 577
- PerlPreConnectionHandler**, 577, 579, 607, 613, 614, 630
  - example, 608–612
- PerlProcessConnectionHandler**, 577, 579, 607
- PerlRequire**, 573
- PerlResponseHandler**, 573, 575, 577, 584, 585
  - Apache life cycle overview and, 577
  - example, 601–607
    - initial setup, 602
    - log level setting, 603
    - log messages with, 602–603
    - printing document header, 604–607
    - printing HTTP header, 603
    - redirection and, 603–604
- PerlSetEnv**, 571–572
- PerlSetVar**, 572
- PerlSwitches**, 574
- PerlTransHandler**, 577, 582, 585
- PerlTypeHandler**, 577, 583, 585
- Personal Home Page**, 16. *See also* PHP
- phases**. *See* request phases
- Phillips, George**, 15
- PHP**, 1, 2, 4, 8, 9, 15–16
  - Apache and, 417
  - application deployment model, 8–9
  - CGI and, 15–16
  - filtering and, 627
  - ProxyPresereHost and, 482–483
  - Web and, 15
- PI()**, 62
- ping()**, 275
- placeholders**, 256, 717
- Plain Old Documentation**. *See* POD
- Plesk**, 8
- plugin**, 385
- plug-ins**
  - Linux, 419
  - to Template Toolkit, 687
- .pm extension**, 194
- POD (Plain Old Documentation)**, 197–201, 233, 366, 518, 574–575, 746
- pod2man**, 198
- polymorphism**, 213
- pop()**, 176
- popup\_menu()**, 728, 769
- port setting (command option)**, 814–815
- PostgreSQL**, 4, 16, 48, 152, 427
- PostReadRequest**, 576
- Power of the SELECT**, 413
- Practical Extraction and Reporting Language**, 164
- pragmatic module**. *See also* 'use base'
  - strict, 192
  - warnings, 192
- precaching**, 346
  - cities, 347
  - method calls, 348
  - states, 347–348
- PRECISION**, 266
- preparation phase**, 422, 576
- prepare()**, 255, 256, 257, 264, 269, 270, 272, 273, 274, 301
- prepare\_cached()**, 255
- prepared statements**
  - emulated, 250, 256, 257
  - server-side, 250, 252, 255, 256, 257
- primary keys**, 31
- PrimeBase (PBXT)**, 18, 85, 113
- PrintError**, 254, 265, 280
- printing arrays**, 178
- PrintWarn**, 254, 265
- private/public (Perl)**, 213, 217, 218
- privilege columns**, 78
- privileges, MySQL**, 74–84, 812
- proactive caching**, 318
- procedures**. *See* stored procedures
- process file handles**, 185–186
- processForm()**, 731
- processing, lazy**, 315, 316
- processUrl()**, 781, 783, 784
- procs\_priv**, 76, 80
- Professional Ajax (Zaka, McPeak & Fawcett)**, 707
- program flow, contact list application**, 515–516
- protocol, stateless**, 417, 419
- Prototype JavaScript Framework**, 708, 732
  - Ajax example, 725–727
  - purpose of, 725
- prototypes**, 187
- <Proxy>**, 439
- proxying, reverse**. *See* reverse proxying
- <ProxyMatch>**, 440
- ProxyPass**, 481
- ProxyPassReverse**, 481–482
- ProxyPassReverseCookieDomain**, 482
- ProxyPassReverseCookiePath**, 482
- ProxyPreserveHost**, 482
  - PHP and, 482–483
  - usefulness of, 482
- ps command**, 141, 431
- pseudo-Data::Dumper**, 181. *See also* Data::Dumper
- public/private (Perl)**, 213, 217, 218
- push()**, 176
- Python**, 4, 152, 317, 360, 417, 480, 580, 687

## Q

- Query()**, 774
- \$query**, 336
- query cache**, 257
- query expansion, bling**, 150

query syntax, Boolean, 159  
 querying data, 38–50

## R

**\$r**, 699, 700  
**Rails**, 1, 353  
**RaiseError**, 254, 265, 280  
 raising number to power, function for, 62  
**RAND()**, 92  
**random character generator**, 92, 257  
**randomString()**, 670, 710, 727  
**ranges**, 41–42  
**raw disk partitions**, 119, 816  
**RDBMS (Relational Database Management System)**, 4, 16,  
 18, 148, 245, 246, 293, 294. *See also* MySQL  
 MySQL and, 85  
 server in, 245–246  
**read statements**, 258  
**readdir()**, 185  
**reading, from process file handles**, 185  
**read-through caching**, 315, 316, 357  
 with SELECT, 412–415  
**readyState**, 709  
**Redhat-based systems**  
 Apache configuration schemes on, 486–487  
 Apache installation on, 428–429  
 DBI/DBD: :mysql on, 247  
 memcached and, 322–323  
 mod\_perl installation and, 428–429  
 startup scripts and, 322–323  
**Redirect (directive)**, 443  
**ref()**, 172, 181  
**reference manual (MySQL)**, 33, 38, 59, 62, 78, 81  
**references**, 169–173  
 array, 170  
 hash, 171  
 identifying, 172–173  
 importance of, 213, 216, 283  
 iteration and, 181  
 object-oriented programming and, 216. *See also*  
 object-oriented Perl  
 scalar, 169–170  
 subroutine, 171–172  
**ref\_literate()**, 181  
**REG800**, 17  
**regex. See regular expressions**  
**REGEXP**, 63, 64  
**regions**, 332. *See also* user application  
**regular expressions (regex)**, 64–65, 206–208. *See also*  
 conditional pattern  
 Apache and, 421  
 compilation, 208  
 mod\_rewrite and. *See* mod\_rewrite  
 one-liner, 206–207  
 optimizations, 208  
 storing, in variables, 207–208  
**Relational Database Management System. See RDBMS**  
**relay logs**, 26, 134, 138  
**–relay-log**, 138  
**–relay-log-index**, 138  
**–relay-log-info-file**, 138  
**reload**, 282  
**remove()**, 326  
**REPAIR TABLE**, 118, 125  
**REPLACE()**, 63  
**replace()**, 326, 350  
**REPLACE**, 56–57  
**replacing data**, 56–58  
**replicate data, to multiple memcached servers**, 355  
**–replicate-do-db**, 138  
**–replicate-do-table = schemaname.tablename /**, 139  
**–replicate-do-wild-table = schemaname.tablename%**, 139  
**replicate-do-wild-table = webap%.%**, 139  
**–replicate-ignore-db**, 138  
**–replicate-ignore-table = schemaname.tablename**, 139  
**–replicate-ignore-wild-table = schemaname.tablename%**, 139  
**replicate-ignore-wild-table = webapps.t%**, 139  
**–replicate-rewrite-db=from\_name-> to\_name**, 139  
**replication**, 18, 85, 133–148. *See also* backups, MySQL  
 asynchronous, 133, 134  
 binary log and, 134, 141  
 cache, 318  
 command options, 137–139  
 adding, 141–143  
 filtering rules, 136  
 manually setting master, 147–148  
 multiple-master, 139–141, 162  
 my.cnf sample file, 824–827  
 mysqld\_multi and, 139–141  
 over SSL, 138  
 overview, 133–134  
 row-based, 133, 138  
 running, verification of, 144–147  
 schemes, 134–137  
 setting up, 139–148  
 statement-based, 133, 138, 139  
 uses of, 133  
**replication backup slave**, 817  
**–report-host**, 138  
**–report-port**, 138  
**request parsing phase**, 421–422, 576  
**request phase cycle. See also Apache life cycle**  
 Apache, 423, 461, 576, 584  
 HTTP, 565, 580, 581, 584, 590, 599, 601, 626, 630, 631  
**request phases**  
 Apache, 421–422, 576  
 handler, 422  
 preparation, 422  
 request parsing, 421–422  
 security, 422  
 mod\_perl, 576  
**RequestHandler**, 582  
**request/request record Apache2 Perl modules**, 586,  
 587–590  
**Require (directive)**, 452  
**require statement**, 195. *See also* use statement  
**\$REQUIRED\_COLS**, 530  
**resource Apache2 Perl modules**, 595  
**Response (mod\_perl request phase handler)**, 576  
**result\_cb()**, 106, 107–108  
**reusable code**  
 CPAN and, 164  
 functions and, 86  
 getUser() and, 537–538  
 inheritance and, 240  
 \_makeWhereClause() and, 336, 337  
 Mason and, 686  
 object orientation and, 212  
 packages and, 211  
 Perl modules and, 194, 197  
 stored procedures and, 86  
**reverse()**, 177  
**reverse proxying**  
 Apache and, 478–480  
 mod\_proxy and, 478–480  
 with two virtual hosts, 499–500

## REVOKE

REVOKE, 76, 80, 81, 82, 83  
**rewinddir()**, 185  
**rewrite directives**, 468–478  
**RewriteCond**, 468  
   flags and, 471–472  
**RewriteEngine**, 468  
**RewriteLog**, 477–478  
**RewriteLogLevel**, 478  
**RewriteMap extensions**, 470–471  
**RewriteRule**, 472  
   conditional pattern and, 472–473  
   substitution and, 473  
   flags, 473–477  
**rewriting URLs**, 468–478. *See also* mod\_rewrite  
**RIGHT joins**, 45, 46  
**right shift operator**, 58  
**ring configuration**, 135, 136  
**RitmarkFS**, 113  
**roach motel**, 132. *See also* Blackhole  
**rollback()**, 276, 277  
**ROLLBACK**, 121  
**ROLLBACK TO SAVEPOINT**, 122, 123  
**ROLLUP**, GROUP BY with, 61  
**Rolsky, Dave**, 698  
**root user**, 75, 205, 386, 431, 806, 811  
**Round Robin Database (RRDTool)**, 823  
**row-based replication**, 133, 138  
**row-level locking**, 113, 119  
**rows()**, 258, 264–265  
**RowsInCache**, 266  
**RPM**, 19, 20, 247, 361, 428, 794, 804. *See also* Yum  
**RRDTool (Round Robin Database)**, 823  
**RSS feeds**, 316. *See also* XML  
**Ruby**, 4, 19, 152, 240, 317, 353, 360, 417, 584

## S

**SaaS (Software-as-a-Service)**, 6, 13  
 --safe-updates option, 53, 121, 122  
**sakila**, 27, 150, 156, 161, 248, 828  
**Sanders, Tony**, 15  
**Sanskrit**, 30  
**saveFileInfo()**, 684  
**SAVEPOINT**, 122  
**saveUpload() subroutine**, 680–682  
**saveUrl()**, 781, 785  
**saveUser()**, 538–542  
**saveUserToCache()**, 549–550  
**say()**, 209  
**scalar()**, 177  
**scalar example, memcached and**, 328–329  
**scalar references**, 169–170  
**scalars**, 165–167. *See also* references  
   addition, 173  
   concatenation, 174  
   usage, 173–174  
**SCALE**, 266  
**scale-out v. scale-up**, 1  
**SCHEMA()**, 60  
**SCHEMA\_PRIVILEGES**, 83  
**schemas**. *See also* information schema; *specific schemas*  
   creating, 29  
   information, 32–33  
   modification, 33–34  
   MySQL configuration and, 811, 812  
**schemes, replication**. *See* replication

**Schutz, Jon**, 10  
**scope columns**, 78  
**scope/scoping**  
   dynamic, 190  
   lexical, 190  
   mod\_perl handlers, 575  
   package, 190  
   Perl and, 164  
   variable, 189–192  
     adherence, forcing of, 192  
     example, 191–192  
     strict pragmatic module and, 192  
**scrambled**, 78  
**ScriptAlias**, 458  
**ScriptAliasMatch**, 458  
**seamless\_rotate**, 756  
**search, apt-cache**, 247, 319, 427, 428, 742, 805  
**search (utility program)**, 152, 159–161  
   debugging and, 159  
   options, 160  
**search engine application (example)**, 739–792  
   components, 747–748  
   database tables, 749–751  
   database triggers, 751  
   flow of content in, 749  
   Gearman workers, 756–766  
   mod\_perl handlers for, 766  
   search application, 766–777  
   paginating, 776–777  
   search template, 766–768  
   using, 777, 778  
   Sphinx setup, 752–756  
   URL queue application, 777–778  
   URLHandler, 778–786  
   URLQueue interface, 790–792  
   URLQueueHandler, 778, 787–790  
**search engines**, 739. *See also* Sphinx::Search  
**search template**, 766–768  
**searchd**, 152, 156, 756  
**searching text**. *See* full-text searching  
**search() subroutine**, 771–775  
**Secure Sockets Layer**. *See* SSL  
**SECURITY**, 88  
**security phase**, 422, 576  
**seekdir()**, 185  
**SELECT**, 38–39  
   read-through caching with, 412–415  
**SELECT, Power of the**, 413  
**selectall\_arrayref()**, 270–272  
   arguments, 271  
   attribute usage, 271  
   attributes, 271  
**selectall\_hashref()**, 272–273  
**selectcol\_arrayref()**, 273  
**selectrow\_array()**, 273–274  
**selectrow\_arrayref()**, 274  
**selectrow\_hashref()**, 274  
**\$self**, 215, 217  
**self-joins**, 40, 47, 49  
**SELinux package**, 319  
**semicolons**, 30, 32, 88, 96, 97, 250  
**send()**, 710, 718  
**sendEmail()**, 735–736  
**SERVER**, 128  
**server, in RDBMS**, 245–246  
**server configuration Apache2 Perl modules**, 586, 592–594  
**server control, Apache**, 483  
**server life cycle phase handlers**, 578  
   visual diagram, 584–585

- server record Apache2 Perl modules**, 586, 591
- server tuning directives**, 444–446
- server-id**, 138
- ServerLimit (directive)**, 445
- ServerName (directive)**, 440
- ServerRoot (directive)**, 440–441
- server-side prepared statements**, 250, 252, 255, 256, 257
- ServerSignature**, 449
- session caching**, 318
- session management**, 662–675. *See also* user account
  - creation with Ajax
  - cookies and, 662–670
  - LoginHandler
    - implementation, 663–667
    - WebApp class methods and, 667–670
  - storing session data, methods for, 670–675
- sessionInfo()**, 665
- sessions ids**, 93, 643
- sessions table**, 91, 92, 93, 94, 670–671
- sessionVariablesTable()**, 665, 666
- SET**, 72, 73
- set()**, 325, 330, 331, 350
- SET NAMES, 129, 130
- SET PASSWORD, 81
- set\_<attribute>**, 226–227
- setCookie()**, 669
- set\_debug()**, 324
- SetHandler**, 461
- SetHandler modperl**, 571
- SetHandler perl-script**, 571
- SetLimits()**, 773
- SetMatchMode()**, 773
- setMemcUIDList()**, 556
- SetRankingMode()**, 772–773
- SetServer()**, 772
- SetSortMode()**, 773
- setUser()**, 339, 345, 350, 351
- setUserGeoParams()**, 340, 345, 350
- setWeight()**, 229
- sha1**, 90, 91
- shift()**, 176, 188
- shopping carts**, 318, 643, 662, 663, 666
- SHOW CREATE TABLE, 32, 103
- SHOW ENGINES, 113, 114, 115
- SHOW GRANTS, 83
- SHOW MASTER STATUS, 144, 145
- SHOW PRIVILEGES, 76
- SHOW SLAVE STATUS, 144, 145
- SHOW TABLE STATUS, 129, 130
- SHOW TABLES, 32, 294
- shutdown, 282
- sigils**, 164, 168, 170, 171, 177, 182, 726, 728
- simple database application. See contact list application**
- single database handle (example)**, 403–409
  - changes to connection, 403–405
  - changes to getUser(), 405–407
- 64-bit**
  - CPU, 318
  - indexes, 153
  - server, on 32-bit server, 403
  - ‘x86\_64’ and, 429
- skeletal frame, contact list application**, 515–516
- Slashdot.org**, 4, 7, 9, 165, 207, 411, 412, 634
- slaves (database instances)**
  - Blackhole storage engine and, 136
  - dual master configuration and, 135
  - replication
    - overview, 133–134
    - schemes, 134–137
    - ring configuration and, 135, 136
- Slice (attribute)**, 271, 272
- slices, array**, 178
- social websites**, 11, 315. *See also* LiveJournal.com
- socket setting (command option)**, 815
- Software-as-a-Service (SaaS)**, 6, 13
- Solaris**, 4
- sort()**, 177
- SOUNDEX()**, 63
- SOUNDS LIKE**, 63
- source code, Apache**, 424
- Sphinx (SQL Phrase Index)**, 10, 152–162
  - configuration, 153–158
  - full-text search engine, 152
  - FULLTEXT v., 152
  - installation, 152–153
  - main index, defining of, 157–158
  - search (utility program), 152, 159–161
  - search engine application and, 752–756
  - searching, 158–161
  - source code, 152
  - starting, 158
  - when to use, 161–162
- sphinx.conf (configuration file)**, 827–829
  - data sources, 153–154, 156
  - database connection options, 156
  - entire code for, 827–829
  - indexer section, 155
  - indexes, 154–155
  - searchd section, 156
- sphinx\_<counter>**, 752, 753
- Sphinx::Search**, 10, 11, 768, 769, 771, 772
  - methods, 772–773
- SPH\_MATCH\_ALL**, 158
- SPH\_MATCH\_ANY**, 158
- SPH\_MATCH\_BOOLEAN**, 158
- SPH\_MATCH\_EXTENDED**, 158
- SPH\_MATCH\_PHRASE**, 158
- split()**, 176–177
- SQL. See also MySQL**
  - features, MySQL and, 85–111
  - injection attacks, 256, 257, 717
  - /MED specification, 125, 128
  - storage engines and, 111. *See also* storage engines
  - stored procedures and, 87. *See also* stored procedures
  - threads, 134
- SQL Phrase Index. See Sphinx**
- sql\_db**, 156
- sql\_host**, 156
- sql\_pass**, 156
- sql\_port**, 156
- sql\_query**, 156, 753
- sql\_query\_info**, 156, 753
- sql\_sock**, 156
- sql\_user**, 156
- square brackets [ ]**, 87, 687
- square root**, 62
- src directory**, 107
- SSL (Secure Sockets Layer)**
  - Apache and, 417
  - directives, 463–466
  - GRANT and, 81
  - replication over, 138
  - virtual hosts and, 466
- SSLCertificateFile**, 464
- SSLCertificateKeyFile**, 464–465
- SSLEngine**, 463–464
- SSLMutex**, 465–466

## SSLRequireSSL

**SSLRequireSSL**, 464  
**SSLSessionCache**, 465  
**SSLSessionCacheTimeout**, 465  
**stacked handlers (mod\_perl)**, 575  
**stale key**, 353–355  
**Stallman, Richard**, 2  
**stampede, cache**, 352–355, 357  
**Starkey, Jim**, 18  
**StartServers (directive)**, 444  
**startup scripts**, 322–323  
   Debian-based, 322  
   Redhat-based, 322–323  
**startup utilities (MySQL)**, 27–28  
**state, illusion of**, 419  
**state caching**, 318  
**stateless protocol**, 417, 419  
**\$statement**, 270, 271, 273, 274  
**Statement (attribute)**, 266  
**statement handles**, 250, 255–261, 283  
   attributes, 265–269  
     MySQL-specific, 267  
     outputs, 268–269  
   methods. *See also* prepare()  
     dump\_results, 265  
     execute(), 255, 256, 257, 258  
     fetch methods, 258–261  
     rows(), 258, 264–265  
**statement-based replication**, 133, 138, 139  
**statements, prepared. See prepared statements**  
**states table**, 30, 332, 515, 720. *See also* contact list  
   application; data; user application  
   indexes, 30–31  
**static methods**, 214, 215, 217  
**statistics keys**, 326  
**stats()**, 326–328  
**stats**, 96, 97, 98  
**status Apache2 Perl modules**, 586, 596–598  
**STDDEV()**, 61  
**STDDEV\_POP()**, 61  
**STDERR**, 184–185  
**STDOUT**, 184–185  
**Stein, Lincoln**, 419, 516  
**Storable**, 329, 403, 667, 671  
**storage engines**, 85, 111–133. *See also specific storage engines*  
   abilities, 113  
   ACID-compliant, 18. *See also* ACID  
   external, 113  
   internal, 112  
   SQL and, 111  
   using, 113–115  
**stored procedures**, 5, 9, 15, 19, 68, 80, 86–94  
   benefits of, 86–87  
   DBI and, 277–279  
   examples, 88–94  
   functions and, 86–94  
   reusable code and, 86  
   SQL and, 87  
   syntax, 87–88  
**store\_page()**, 759  
**storeSession()**, 671  
**store\_urls()**, 762  
**storing files, in databases**, 675  
**storing session data**, 670–675  
**STRCMP()**, 63, 65  
**strict pragmatic module**, 192  
**string comparison functions**, 63–64  
**string functions**, 62–66

**subroutines**, 168, 171–172, 186–187. *See also* functions;  
   *specific subroutines*  
   dispatcher(), 519–522  
   handler(), 677–680, 732–734  
   header(), 518  
   main(), 517–518  
   mainform(), 522–523  
   notifyUser(), 735  
   parentheses and, 187  
   Perl data types, 168, 186–188  
   references, 171–172  
   saveUpload(), 680–682  
   search(), 771–775  
   table creation, 292–304  
   userform(), 525–528  
   userlist(), 523–525  
   viewuser(), 528–529  
   XS, 9, 246  
**subscriber, of last array member**, 177–178  
**substitution**  
   flags for, 473–474  
   special, 474–477  
   values for, 473  
**SUBSTR()**, 65, 92  
**SUBSTRING()**, 65  
**subtractNumbers()**, 203  
**SUM()**, 60  
**Sun Microsystems**, 1, 5, 7  
**Sunya**, 30  
**SUSE**  
   Apache configuration on, 487–489  
   MySQL and, 794, 804  
**Swartz, Jonathan**, 698  
**Swiss army knife**, 164  
**switch()**, 209  
**switch statement**, 172  
**symbol table**, 189–191, 229  
**System 5 (SYSV)**, 322, 466, 809  
**SYSV (System 5)**, 322, 466, 809

## T

**tab-delimited data**, 22  
**table creation subroutine (contact list application)**, 292–304  
**table handler**, 16, 17, 19  
**TABLE\_PRIVILEGES**, 83  
**tables. See also data; specific tables**  
   ALTER TABLE, 33–34, 55  
   Archive, 125  
   BigTable project, 13  
   Blackhole, 112, 132–133  
   CHECK TABLE, 117, 118  
   contact list application and, 286  
   corrupted, 25, 26, 117, 118, 125, 151  
   creating, 29–30  
   CSV, 130–131  
   dispatch, 172  
   Federated, 126–128  
   FLUSH TABLES, 25, 131, 132  
   FLUSH TABLES WITH READ LOCK, 820, 822  
   foo, 415–416  
   .frm files, 115, 116, 121, 129, 131, 813, 821  
   grant, 76–80, 81, 84  
   India cities, 278, 279  
   India states, 278  
   InnoDB, 120  
   inserting multiple values in, 257

## url\_fetch()

- MyISAM, 116, 117–118
  - REPAIR TABLE, 118, 125
  - sessions, 91, 92, 93, 94, 670–671
  - SHOW CREATE TABLE, 32, 103
  - SHOW TABLE STATUS, 129, 130
  - SHOW TABLES, 32
  - states. *See* states table
  - symbol table, 189–191, 229
  - users. *See* users table
  - virtual, 102
  - tablespace files**, 118, 119–120, 125, 799, 800, 809, 813, 814, 815, 816, 817
  - tables\_priv**, 76, 80
  - tags**, 694–695
    - HTMLesque, 694–695
  - Tcl**, 417
  - telldir()**, 185
  - Template Toolkit**, 686–693
    - Ajax example 2 and, 711, 713
    - caching templates, 693
    - features, 687
    - mod\_perl handler (example) with, 690–692
    - plug-ins to, 687
    - syntax, 687–689
  - templating systems**, 686–698
    - \$ characters in JavaScript, 726
    - approaches/philosophies, 686
    - HTML::Template, 686, 694–698
    - Mason, 686, 698–704
    - Template Toolkit, 686–693
  - TestFilter**, 579, 628, 629, 631
  - testing**
    - contact list application (CGI), 563–564
    - cookies, tools for, 649–651
    - delete\_contact(), 306–309
    - edit\_contact(), 300, 304–305
    - find\_contact(), 310–312
    - headers, tools for, 649–651
    - insert\_contact(), 305–306
    - LoginHandler, 673
    - modules (Perl) and, 201–204
  - Test::More**, 202
  - TestReponseHandler**, 607
  - TestString**, 468–469
  - TEXT, 115, 124, 748, 750
  - thaw()**, 403, 667
  - thingy**, 215
  - this**, 215
  - ThreadLimit (directive)**, 445
  - threads**. *See also* multi-threaded
    - IO, 134
    - SQL, 134
  - ThreadsPerChild (directive)**, 445–446
  - thundering herd**, 352–355, 357
  - Tiger class**, 212, 213, 235
  - time/date functions**. *See* date/time functions
  - Timeout (directive)**, 446
  - Tina**, 85, 130
  - TLS (Transport Layer Security)**, 417, 423, 463
  - tmpdir**, 814
  - Tomcat, Apache**, 4, 479
  - Torvalds, Linus**, 2
  - tr** DOM element, 730, 786
  - TraceLevel**, 254, 265
  - Trans**, 576
  - transactional methods**, 276–277
  - transactions**, 121–123
  - TransferLog**, 447
  - transparency**
    - Federated storage engine and, 125
    - mod\_proxy and, 478, 480, 481
    - mod\_rewrite and, 468
  - Transport Layer Security (TLS)**, 417, 423, 463
  - trigger\_fun.sql**, 411
  - triggers**, 5, 7, 15, 19, 68, 71, 74, 86, 94–102
    - database. *See* search engine application
    - examples, 95–102
    - limitations, 102
    - Memcached Functions for MySQL and, 403, 409–412
    - syntax, 95
  - Tripod project**, 482–483. *See also* ProxyPreserveHost
  - TRUNCATE, 53, 54
  - TYPE**, 266
  - Type (mod\_perl request phase handler)**, 576
  - type globs**, 165, 168, 189, 190, 230, 231, 244
- ## U
- u**, 321
  - Ubuntu**
    - Apache configuration on, 484–486
    - Apache installation and, 427–428
    - DBI/DBD: :mysql on, 247
    - memcached on, 319–320
    - mod\_perl installation and, 427–428
    - MySQL installation on, 804–806
  - UDF API**, 105, 106
  - UDFs (user defined functions)**, 5, 7, 86, 105–111. *See also* Gearman MySQL UDFs; Memcached Functions for MySQL
    - calling, 388
    - memcached, 383
    - required functions for, 106
    - writing, 105–106
  - UID**, 508
  - uid**, 512
  - underlisting page, in Mason**, 703–704
  - underscore(\_)**
    - make\_label() and, 295
    - naming convention (Perl) and, 215, 217, 218, 220
    - wildcards and, 64, 139
  - UNION, 48–50
  - UNIQUE, 29, 31, 32, 35, 37, 55, 115, 332, 514, 615
  - Unireg**, 17
  - UNIX**. *See also* Linux
    - Apache and, 417, 419
    - Apache installation on, 429–432
    - directory, 186
    - domain socket, 366
    - manpages, 197
    - memcached and, 322
    - MySQL and, 19, 20, 21, 27
    - MySQL installation on, 807–810
    - SYSV, 322, 466, 809
  - UNIX\_TIMESTAMP()**, 67
  - unsetCookie()**, 669
  - unshift()**, 176
  - UPDATE, 50–52, 96
  - update\_contact()**, 302–303
  - updateMemcUIDList()**, 556–558
  - updateUser()**, 341, 543–545
  - updating contacts**, 285, 302–303
  - updating data**, 50–52
  - upload handler**. *See* file upload mod\_perl handler
  - uploads directory, in Apache configuration**, 676–677
  - url()**, 517
  - URL queue application**, 777–778
  - url\_fetch()**, 761

## url\_fetch job

**url\_fetch job**, 765

**url\_fetcher\_worker.pl**, 761

**URLHandler**, 778–786

- JavaScript and, 780–781
- template implementation, 785–787

**url\_memc\_update**, 410

**url\_mem\_delete**, 410

**url\_mem\_insert**, 410

**url\_process()**, 762, 764

**URLQueue interface**, 790–792

**URLQueueHandler**, 778, 787–790

**URLs, rewriting**. *See* **rewriting URLs**

**urls index**, 752

**urls table**, 750

**urls\_blob table**, 750

**urls\_delta index**, 752, 755

**urls\_queue table**, 750

**urls\_queue\_insert**, 751

**urls\_queue\_update**, 751

**urls\_stored table**, 750

**usage()**, 289

'use base ', 236, 237, 238, 239, 244

**use statement**, 194, 195, 248

**UseDir**, 458–459

**user (command option)**, 814

**User (directive)**, 443

**user account (MySQL)**, 74–75

**user account creation with Ajax (example 7)**, 728–738

- account handler, 732–735
- account page mod\_perl handler, 728–729
- account page template, 729–732
- in action, 736–738
- sendEmail(), 735–736

**user application (memcached example)**, 357

- accessor methods, 338–339
- modifying to use cache, 348–350
- caching
  - implementation plan, 345
  - key scheme, 346
  - strategies, 318
  - where to add, 345–346
- connectDB(), 334
- data design, 332–333
- data modification methods, 339–342
- data retrieval methods, 335–337
- instantiation, 334
- Memcached Functions for MySQL and, 402–409, 416
- precaching, 346
  - cities, 347
  - method calls, 348
  - states, 347–348
- user data caching
  - get method modifications, 351–352
  - set method modifications, 350–351
- UserApp
  - caching in, 352
  - implementation, 334
  - using, 342–344

**user data caching**

- get method modifications, 351–352
- set method modifications, 350–351

**user defined functions**. *See* **UDFs**

**User Directory directives**, 458–459

**user manual (MySQL)**, 68

**user tracking, cookies and**, 643

**UserApp**

- caching in, 352
- implementation, 334
- using, 342–344

**USER\_COLS**, 530

**user-defined variables**, 72–74, 84, 85, 162

**userExists()**, 547–548

**userExistsInCache()**, 553–554

**userform display function**, 507, 510–513

**userform() subroutine**, 525–528, 641

**userList()**, 714

**userlist**, 509, 510, 513

**userlist**, 712, 713

**userlist\_ajax2.tt2**, 713

**userlist() subroutine**, 523–525

**username**, 253

**USER\_PRIVILEGES**, 83

**users table**, 30, 332, 514, 671. *See also* contact list

- application; data; session management; user application
- column data types, 31
- column modification, 33–34
- indexes, 31–32
- inserting data, 35–36
- stored procedures example, 88–89

**usertable**, 715, 723

**utility functions, libmemcached**, 372

**utility methods, multi-step**, 269–274

**utility programs**

- libmemcached, 360, 361–364
- search (Sphinx), 152, 159–161

## V

**value setting functions (memcached)**, 369–370

**VARCHAR**, 31, 115

**variable scope**. *See* **scope**

**variable types**. *See* **data types**

**variables**. *See also* **user-defined variables**; **specific variables**

- instance, 215
- regular expressions stored in, 207–208

**VARIANCE()**, 61

**Vedic philosophy**, 30

**VERSION**, 214

**views**, 86, 102–105

- read-only, 293

**viewuser display function**, 507, 513

**viewuser() subroutine**, 528–529, 641

**vim editor**, 186

**virtual hosting**, 459–460

- configuring (Apache task), 493–495
- IP-based, 460
- name-based, 460
- reverse prox setup with, 499–500
- SSL and, 466

**virtual method**, 217

**virtual tables**, 102. *See also* **views**

**virtual user**, 529, 530, 532

**<VirtualHost>**, 440, 459, 460

**-vv**, 321, 322, 329

## W

**Wall, Larry**, 8, 163, 164. *See also* **Perl**

**WAMP (Windows, Apache, MySQL, Perl)**, 3

**warn()**, 224, 529

**warnings pragmatic module**, 192

**Web**. *See also* **CGI**; **Perl**; **PHP**; **search engine application**

- Drizzle and, 13, 421
- Perl and, 8

PHP/CGI and, 15  
 search engine applications and, 739

**Web 2.0**, 1

**web applications**. *See also* search engine application  
 caching and, 1–2  
 data and, 313  
 as desktop applications, 707, 708. *See also* Ajax  
 future of, 12  
 HTTP and, 419  
 tools for building, 740

**web client worker**, 748, 749, 750, 757–762

**web development, future of**, 12–13. *See also* LAMMP

**WebApp class methods**  
 contact list application, 529–542  
 file upload mod\_perl handler and, 682–684  
 LoginHandler and, 667–670

**WebApp.pm**, 516, 517, 568, 602, 608, 609, 616, 617, 620, 651, 705, 776

**\$WebApp::REQUIRED\_COLS**, 520, 530, 531

**websites**  
 MySQL usage and, 4, 5  
 social, 11, 315

**webuser**, 76, 77, 78, 79, 80

**WHERE clause**, 40–41

**whereClause()**, 656–658

**Widenius, Monty**, 13, 16, 17, 18, 22, 23

**wildcards**  
 Apache and, 436, 437, 485  
 percentage sign (%) and, 41, 64, 139  
 underscore(\_) and, 64, 139

**Williams, John**, 698

**Williams, Ken**, 698

**Windows**  
 Apache installation on, 425–427  
 MySQL installation on, 795–803

**Windows, Apache, MySQL, Perl (WAMP)**, 3. *See also* Apache; LAMMP; MySQL; Perl

**wire-frame**, 286–287, 288, 290

**-with-mysql**, 385

**Wordpress**, 8, 482

**work()**, 747

**Worker, MPM**, 420

**worker process**, 11. *See also* Gearman workers

**World Wide Web**. *See* Web

**Writely.com**, 708

**write-through caching**, 315–317, 357

**writing, to process file handles**, 185

**WroxHandlers::AccessTestHandler**, 613, 614, 619, 621, 626, 630

**WroxHandlers::AuthZTestHandler**, 619, 620, 621, 626, 631

**WroxHandlers::ContactHandler**, 636, 642

**WroxHandlers::ContactListHandler**, 635, 636, 637

**WroxHandlers::DenierHandler**, 608, 611, 612, 613, 614, 630

**WroxHandlers::LogTestHandler**, 623, 626, 628, 631

**WroxHandlers::SearchHandler**, 768–771

**WroxHandlers::TestFilter**, 579, 628, 629, 631

**WroxHandlers::TestResponseHandler**, 607

## X

**XML**, 1, 21. *See also* Ajax  
 DOM and, 316, 708  
 feeds, 47, 67, 103, 316  
 parsing library, 434  
 pipe, 152  
 SGML and, 436  
 WIX and, 795

**XMLHttpRequest object**, 708, 709, 711, 714

**XS (eXternal Subroutine)**, 9, 246

## Y

**yacc**, 807

**Yast**, 742, 794, 804

**Yum**, 247, 320, 360, 428, 794, 804

## Z

**Zaka, Nicholas C.**, 707

**zero**, 30. *See also* NULL

**zlib**, 123, 403, 408, 491