

Index

- A**
- ab-normalization, 196–197. *See also* normalization
 - ACID properties, 81
 - Add Counters dialog box, 20
 - Address Windowing Extensions. *See* AWE
 - advanced tab, Performance Options window, 175–176
 - affinity
 - CPU, 181
 - I/O, 181
 - network I/O, 183–187
 - Affinity Mask, 48
 - AFTER trigger, 218, 219
 - aggregates, 295–298
 - Agile Database Techniques (Ambler)*, 197
 - Alerts, 18, 22
 - proactive, 32–33
 - <ALL INSTANCES>, 20
 - ALTER INDEX command, 54
 - ALTER SETTINGS permission, 173
 - Ambler, Scott, 197
 - Analyze Block Info report, 90
 - anomalies, 190
 - deletion, 190
 - insert, 190
 - redundancy, 190
 - update, 190
 - Application and System event logs, 17
 - application login, 95
 - application mode, 25, 57
 - ApplicationName, 324
 - approximate numerics data types, 199
 - Arithmetic Expression, 245
 - artificial keys, 209
 - Assert operator, 245
 - Assign operator, 248
 - association tables, 501
 - ASYNC_I/O_COMPLETION, 70
 - ATA disks, 154
 - atomicity, 81
 - Audit Add DB User Event, 475
 - Audit Add Login to Server Role Event, 474
 - Audit Add Member to DB Role Event, 475
 - Audit Add Role Event, 475
 - Audit Addlogin Event, 474
 - Audit Backup/Restore Event, 475
 - Audit Change Audit Event, 475
 - Audit Change Database Owner, 475
 - Audit Database Scope GDR Event, 474
 - Audit DBCC Event, 475
 - Audit Login, 97, 324
 - Audit Login Change Property Event, 474
 - Audit Login Failed, 474
 - Audit Login GDR Event, 474
 - Audit Logout, 97, 324
 - Audit Schema Object GDR Event, 474
 - Audit Schema Object Take Ownership Event, 475
 - Audit Server Alter Trace Event, 475
 - Audit Server Starts And Stops, 474
 - AuditUser section, 462
 - AUTO_CREATE_STATISTICS, 362
 - autogrow, 54, 161–162
 - disable, 162
 - AUTO_UPDATE_STATISTICS, 363
 - AUTO_UPDATE_STATISTICS_ASYNC, 363
 - Available Bytes, 26
 - Average Disk Bytes
 - /Read, 415
 - /Transfer, 415
 - /Write, 415
 - Average Disk Queue length, 20–21, 27, 415
 - Read, 415
 - Write, 415
 - Average Disk Seconds
 - /Read, 27, 415
 - /Transfer, 27, 415
 - /Write, 27, 415
 - AWE (Address Windowing Extensions), 150–152
 - API, 150–151
 - best practice, 151
 - when to use, 151–152
- B**
- backup and restore strategy, 526–527
 - balanced trees (b-trees), 232, 355–356
 - baselines, 29, 31, 38, 49, 65
 - disk bottlenecks and, 49
 - Profiler traces, data collection timeframe for, 102, 103
 - basic volumes, 160, 161
 - benchmark testing, 514–515
 - trace scenario for, 98
 - best practices
 - autogrow disabled, 162
 - AWE, 151
 - backup and restore strategy, 527
 - basis volume/dynamic volume, 161
 - bottlenecks, early detection, 524
 - change management, 530
 - configuration-based CPU bottlenecks, 48
 - correct edition of SQL Server, 525
 - CPU, 144
 - data cache, 265
 - database file placement, 158
 - database mirroring, 528
 - defragmentation, 534
 - designing for performance from start, 497–521
 - disk bottlenecks, 27, 28
 - Disk Queue Length, 414
 - HBA queue depth, 156
 - high availability, 526
 - hyper-threading disabled, 165, 178
 - index maintenance, 54
 - IOPS, 413
 - latency and MB/sec, 414
 - load testing, 529
 - memory, 144
 - multi-core processors, 163
 - NTFS Allocation Unit Size, 161
 - /PAE, 150
 - plan guides, 537
 - RAID10, 157
 - SANs, 156

best practices (continued)

best practices (continued)

SAS, 155
 SATA, 155
 sector alignment, 160
 server-side traces in production environment, 136
 Showplan options, 261
 64-bit processors, 165
 soft-NUMA, 170
 SQL Server additional features, 526
 SQL trace design, 94
 SQL trace events, 97
 SQLIO, 454
 SQLIOSim, 458–459
 SQLIOSim errors, 469
 storage subsystem, 470
 StorPort drivers, 157
 System Monitor, 31–33
 tempdb, 51, 52
 /3GB option, 149
Binary(n), 201
binary strings data types, 201
BinaryData, 325
Bit, 198
Bitmap operator, 245
.BLG file, 60
blocked process report, 90, 104, 105
blocked process threshold, 90, 105
the blocker script, 89–90
blocking events, SQL Profiler and, 104–107
blocking locks, 83, 231, 312–316. See also deadlocks
 monitoring, 89–91
 preventative measures, 313–316
 trace scenario for, 98
Blocking Scripts, 12
blocks, 159. See also sectors
 size, 413
blogs, performance tuning, 13
Bookmark Lookup operator, 245, 257
bookmark lookups, 285–288
Boost SQL Server priority check box, 41, 176
bottlenecks, 25–29, 42–54, 523–524
 CPU, 17, 18, 28–29, 46–49
 configuration-based, 48
 confirming, 47
 schema-based, 48–49
 disk I/O, 17, 18, 27–28, 49–54, 144
 baselines and, 49
 best practices, 27, 28
 configuration-based, 50–53

confirming, 50
 schema-based, 53–54
 early detection, 524
 memory, 17, 18, 25–26, 43–46, 144
 configuration-based, 45–46
 confirming, 44–45
 DMVs and, 26
 external memory pressure, 43–44
 internal memory pressure, 44
 types, 43–46
 storage, 450–452
 trigger, 220
b-trees. See balanced trees
budget constraints, 4
Buffer Manager, 26
buffer pool, 46, 263
bulk update locks, 230
BulkUpdateUser section, 462

C

cache, 166–168. See also processor(s)

data, 263–265
 best practice, 265
 freeing pages in, 264
 in-memory data pages, 263
 monitoring, 264–265
 plan, 262–263
 procedure, 265–268
 best practices, 267–268
 freeing, 266
 in-memory plans, 265
 internal storage, 265
 zero cost plans, 265

caching effects, minimizing, 269

caching issues, query plan, 309–311

candidate keys, 209

capturing

blocking events, 104–107
 deadlock graphs, 110–114
 Showplan XML data, 107–109
 workloads, 321–330

CASCADE, 211

case scenarios,

workload-replaying, 337–351

multiple workloads on single server, 337, 346–351
 validating performance improvement, 337, 340–343
 workload generation, 338–340
 workload in different environments, 337, 343–346

catalog views, 472, 533

cdb debugger, 12

change management, 529–530

Char(n), 199

character data types, 199–200

check constraints, 212

CHECKPOINT command, 269

checkpoint process, 264

child operator, 254

ClientProcessID, 325

CLR (common language runtime), 215

stored procedures, 214, 215

Clustered Index

Delete operator, 245
 Insert operator, 245
 Scan operator, 245, 257
 Seek operator, 245, 257
 Update operator, 245

clustered indexes, 233–235, 357–359

importance, 291

clustering, 527

Coalesce statement, 289, 290

Codd, Edgar F., 189

Collapse operator, 245

columns, 197–214

computed, 205–206
 data types, 197–205
 default values, 213–214
 identity, 357
 included, 359–360
 nullable, 288–290
 options, 206–214
 single-use, 507–509

command line

counter list tool, 33–34
 SQLIOSim from, 467

commands. See specific commands

committed memory, 44

common language runtime. See CLR

common table expressions. See CTEs

common waits, 69–70

communicating performance issues, 519–521

compilation/recompilation, excessive, 49

complex views, 279–280

Compute Scalar operator, 245, 275, 276, 281

computed columns, 205–206

Concatenation operator, 245

concurrency, 79–80, 227

optimistic, 79
 pessimistic, 79–80
 side effects, 228

configuration settings, SQL Server, 171–187

CPU, 175–176
 inspecting

deadlocks

- advanced settings, 174–175
 - considerations before, 171–172
 - with scripts, 173–175
 - with SQL Server Management Studio, 172–173
 - configuration-based**
 - CPU bottlenecks, 48
 - disk bottlenecks, 50–53
 - memory bottlenecks, 45–46
 - performance problems, 41
 - consistency**, 81
 - Constant scan operator**, 245, 281
 - constraints**
 - check, 212
 - foreign key, 210–212
 - primary key, 209–210
 - unique, 212
 - controller/disk contention**, 53
 - Convert (Database Engine) operator**, 248
 - corporate user scale**, 498
 - Correlated Subqueries**, 275
 - Derived Tables and, 298–302
 - UDF v., 275
 - cost-based optimization**, 240
 - costly queries**. *See* **queries**
 - counter list tool**
 - command line, 33–34
 - GUI, 33–34
 - Counter Logs**, 18
 - counters**, 16, 38. *See also* **specific counters**
 - adding, 20, 22, 38
 - default, 18–19
 - missing, 33–34
 - Perfmon, 414–417
 - covered indexes**, 276, 359
 - CPUs**. *See also* **processor(s)**
 - affinity, 181
 - best practice, 144
 - bottlenecks, 17, 18, 28–29, 46–49
 - configuration-based, 48
 - confirming, 47
 - schema-based, 48–49
 - cache, 166–168
 - memory's impact, 144
 - parallel processing, 180–181
 - performance, 145–146
 - problems, 57–59
 - system architecture, 168–170
 - CREATE INDEX**, 362
 - CREATE STATISTICS**, 363
 - CREATE TABLE statement**, 226
 - CTE-based paging**, 303–305
 - CTEs (common table expressions)**, 295–298
 - CTRLIST.EXE utility**, 33–34
 - Current Waiting Requests**, 488
 - Cursor Catchall operator**, 248
 - Cursor data type**, 202
 - cursor operators**, 248
 - cursor usage**, 217
 - incorrect, 504
 - CursorClose**, 324
 - CursorExecute**, 324
 - CursorOpen**, 324
 - CursorPrepare**, 324
 - CursorUnprepare**, 324
 - CXPACKET**, 70, 76
- D**
- DAC**. *See* **Direct Administrators Connection**
 - DAS**. *See* **direct-attached storage**
 - data**
 - analysis, 7
 - log data, 35–38
 - patterns/trends, 32
 - anomalies, 190
 - capture, 23
 - right timing/duration, 23
 - collection, 6–7
 - columns. *See* **columns**
 - duplicates in, 290–291
 - generation, System Monitor, 23–24
 - log files and
 - autogrowth, 51
 - placement, 51, 56
 - pages, 221–223
 - in-memory, 263
 - performance, schema, 221–237
 - quality, schema, 189–220
 - rows, 221–222
 - structure, 222, 223
 - data cache**, 263–265
 - best practice, 265
 - freeing pages in, 264
 - in-memory data pages, 263
 - monitoring, 264–265
 - Data Definition Language triggers**. *See* **DDL triggers**
 - Data Dude**, 12
 - Data File Auto Grow**, 474
 - Data File Auto Shrink**, 474
 - Data Manipulation Language triggers**. *See* **DML triggers**
 - data types**, 197–205. *See also* **specific data types**
 - approximate numerics, 199
 - binary strings, 201
 - character, 199–200
 - date/time, 199
 - exact numerics, 198–199
 - large value, 203–205
 - miscellaneous, 201–202
 - non-persisted, 202
 - unicode character, 200
 - database(s)**. *See also* **workloads**
 - backup and restore, 526–527
 - file placement, 157–158
 - best practice, 158
 - mirroring performance, 54–55, 527–528
 - OLAP, 189, 196
 - OLTP, 27, 189, 196
 - People, 353–355
 - Database Consistency Checks (DBCCs)**, 26
 - SQLIOSim configuration for, 465–467
 - database deployment life cycle**, 523–525
 - Database Mirroring**, 54–55, 527–528
 - State Change, 475
 - Database Tuning Advisor (DTA)**, 39, 53, 532
 - index tuning, 365–393
 - individual queries, 366–374
 - too many indexes, 389–391
 - for updates, 374–386
 - partitioned indexes/tables and, 407
 - workload tuning, 392–393
 - DatabaseID**, 325
 - DatabaseName**, 325
 - Datetime**, 199
 - date/time data types**, 199
 - DBCC FREEPROCCACHE**, 269
 - DBCC FREESYSTEMCACHE**, 269
 - DBCC SHOW_STATISTICS**, 241
 - DBCCs**. *See* **Database Consistency Checks**
 - DDL (Data Definition Language) triggers**, 218, 219
 - Deadlock Graph event class**, 110
 - deadlocks**, 83, 110, 231–232, 312–316. *See also*
 - locks/locking
 - conditions for, 312
 - graphs, 88, 89
 - SQL Profiler and, 110–114
 - “How to resolve a deadlock” bulletin, 114
 - monitoring, 86–89
 - output, 87–88
 - preventative measures, 313–316
 - “System Deadlocks,” 312
 - trace scenarios for, 98

debuggers

debuggers

- cdb, 12
- ntsd, 12
- windbg, 12
- Decimal (p, s)**, 198
- Declare operator**, 248
- default trace**, 473–477
- default values, column**, 213–214
- defragmentation, 533–534. See also fragmentation**
 - best practice, 534
 - online, 534
- Delete (Database Engine) operator**, 245
- Deleted Scan operator**, 246
- deletion anomaly**, 190
- denormalization, 196–197, 510–512. See also normalization**
- density**, 233
- deployment strategies**, 523–537
- Derived Tables, Correlated Subqueries and**, 298–302
- Direct Administrators Connection (DAC)**, 11–12
- direct recursion**, 220
- direct-attached storage (DAS)**, 155, 156. *See also* SANs
- dirty reads**, 80, 228
- disabling**
 - autogrow, 162
 - hyper-threading, 165, 178
 - indexes, 360
 - parallelism, 77
- disaster recovery**, 526
- disconnected composite activities**, 504
- Disk Bytes/Sec**, 415
 - Read, 415
 - Write, 415
- % Disk Time**, 27, 415
 - Idle, 415
 - Read, 415
 - Write, 415
- disk Input/Output. See disk I/O**
- disk I/O (disk Input/Output)**, 18, 153, 154–162, 182
 - bottlenecks, 17, 18, 27–28, 49–54, 144
 - baselines and, 49
 - best practices, 27, 28
 - configuration-based, 50–53
 - confirming, 50
 - schema-based, 53–54
 - memory's impact on, 144
- Disk Queue Length**, 414
- Disk Reads/Sec**, 415
 - Transfers, 415
 - Writes, 415

- Diskpar**, 159, 160
- Diskpart**, 160
- disks**
 - array, 155, 156
 - ATA, 154
 - configuration, 158–161
 - /controller contention, 53
 - FC, 155
 - fragmentation, 54, 161–162
 - external, 54, 161, 162
 - index/table, 54
 - internal, 54, 161, 162
 - logical, 415–417
 - paging to, high CPU and, 48
 - partition provisioning, 158
 - PATA, 154
 - performance, 145–146
 - problems, 56
 - physical, 415–417
 - SAS, 154–155
 - SATA, 154–155
 - SCSI, 154
 - sector alignment, 159–160
 - storage design, 154
 - terminology, 159
 - types, 154–155
 - volume type, 160–161
- Display Estimated Query Plan**, 243
- DISTINCT statements**, 288, 290
- Distribute Streams operator**, 248
- DML (Data Manipulation Language) triggers**, 218–219
- DMVs. See Dynamic Management Views**
- DMVStats**, 72–73, 494
- Dorr, Bob**, 469
- Draper, Gert**, 13
- DSS simulation, SQLIOSim configuration for**, 464–465
- DTA. See Database Tuning Advisor**
- duplicates, in data**, 290–291
- durability**, 81
- Duration column configuration**, 102, 104
- dynamic locking**, 231
- Dynamic Management Views (DMVs)**, 11, 472–473
 - memory bottlenecks and, 26
 - Object_ids, 515
- dynamic volumes**, 160–161

E

- Eager Spool operator**, 246, 255
- EndTime**, 325
- Error data column**, 325
- ErrorLog**, 474

- estimated CPU Cost**, 249
- estimated I/O cost**, 249
- Estimated Number of Rows**, 249
- estimated row size**, 249
- estimated Subtree Cost**, 249
- Ethereal**, 17
- EULA.txt**, 423
- event categories**, 95
- event classes, 95, 96. See also specific event classes**
- event names**, 97
- event numbers**, 96, 97
- EventClass data column**, 324
- EventClass Ids**, 270
- EventSequence**, 324
- exact numerics data types**, 198–199
- excessive compilation/recompilation**, 49
- exclusive locks**, 83, 230
- Exec Prepared SQL**, 324
- ExistingConnection**, 324
- explicit conversion costs**, 282
- extent pages**, 223–224
- external fragmentation**, 54, 161, 162
- external memory pressure**, 43–44
- external waits**, 68

F

- fabric**, 155, 156
- failover clustering**, 527
- Fetch Query operator**, 248
- fiber mode**, 181
- fibers**, 181
- file groups**, 53, 226
 - partitions, 226
- files**, 53
- fill factor**, 356
- Filter (Database Engine) operator**, 246
- filters**, 97
- First Normal Form**, 191
- flagBits**, 222
- Float(n)**, 199
- foreign keys**, 210–212
 - constraints, 210–212
- forms**, 191
 - First Normal, 191
 - Fourth Normal, 191
 - Second Normal, 191
 - Third Normal, 191
- 4-gigabyte tuning (4GT)**, 149
- 4GT. See 4-gigabyte tuning**
- Fourth Normal Form**, 191
- fragmentation**, 533–534
 - defragmentation, 533–534
 - disk, 54, 161–162

issue statements

- external, 54, 161, 162
 - index/table, 54
 - internal, 54, 161, 162
 - index
 - monitoring, 394–395
 - removing, 395–397
 - freeCnt**, 222
 - freeData**, 222
 - Freedman, Craig**, 13
 - FT:Crawl Aborted**, 475
 - FT:Crawl Started**, 475
 - FT:Crawl Stopped**, 475
 - functional dependencies**, 191
 - functions, in views**, 512–513
- G**
- Garden, Euan**, 13
 - gatekeeper user role**, 499
 - Gather Streams operator**, 248
 - GetRangeThroughConvert**, 281
 - Globally Unique Identifiers. See GUIDs**
 - gpedit.msc**, 182
 - graphical query plans**, 243–245
 - graphical user interface. See GUI**
 - GUI (graphical user interface)**, 172
 - GUI counter list tool**, 33–34
 - GUIDs (Globally Unique Identifiers)**, 357–358
- H**
- hard page fault**, 148
 - hardware, choosing/configuring**, 143–170
 - hardware NUMA**, 168, 178
 - hash joins**, 249, 252–254
 - Hash Match operator**, 246
 - Hash Match Root operator**, 246
 - Hash Match Team operator**, 246
 - Hash Warning event**, 474
 - HBA. See host bus adapter**
 - heaps**, 359
 - high CPU, paging to disk and**, 48
 - high-availability**, 526
 - historical tables**, 502
 - host bus adapter (HBA)**, 155, 156
 - drivers, 157
 - queue depth, 156
 - HostName**, 325
 - “How to resolve a deadlock” bulletin**, 114
 - Huffman, Clint**, 36
 - hyper-threading**, 48, 164–165, 176–178. *See also processor(s)*
 - disabled, 165, 178
- I**
- IA64 processors. See processor(s)**
 - ID-based partitioning**, 510
 - identity column**, 357
 - If operator**, 248
 - image data type**, 201, 203
 - implicit conversion issues**, 280–282
 - Include Actual Execution Plan menu option**, 242
 - Include Actual Query Plan button**, 244
 - included columns feature**, 359–360
 - Index Scan operator**, 257
 - Index Seek operator**, 257
 - Index Tuning Wizard**, 365. *See also Database Tuning Advisor*
 - indexes**, 42, 232–236, 353–407
 - best practice, 54
 - clustered, 233–235, 357–359
 - importance, 291
 - complex access pattern, 514
 - covered, 276, 359
 - disabling, 360
 - evaluating, for performance, 513–514
 - fragmentation, 54
 - monitoring, 394–395
 - removing, 395–397
 - inappropriate/missing, 53
 - maintenance, 393–397
 - monitoring usage, 516
 - Nonclustered, 236, 359–360
 - Delete, 246
 - Insert, 246
 - Scan, 246
 - Seek, 246
 - Spool, 246
 - Update, 246
 - no-thought/default, 513–514
 - partitioned, 53–54, 224–227, 397–407, 530–532
 - benefits, 397–398
 - creating, 226–227, 398–407
 - deleting data from, 403–406
 - DTA and, 407
 - file groups, 226
 - function, 226
 - horizontal, 225
 - for performance, 225–226
 - prerequisites, 398
 - production system, 406–407
 - scheme, 226
 - for subsets of data, 225
 - utilization, 226
 - vertical, 224–225
 - performance tuning, 353–407
 - with DTA, 365–393
 - prerequisite, 270
 - review, 355–364
 - selectivity, 233
 - simple access pattern, 514
 - sp_helpindex for, 242
 - statistics, 241–242, 360–364
 - creating, 362–363
 - updating, 363–364
 - structure, 232
 - T-SQL tuning for, 285–291
 - views, 364
- indexId**, 222
- indirect recursion**, 220
- infinite user scale**, 498
- in-memory data pages**, 263
- inner table**, 250
- Input/Output. See I/O**
- Input/Output operations per second. See IOPS**
- insert anomaly**, 190
- Insert (Database Engine) operator**, 246
- Inserted Scan operator**, 246
- INSTEAD OF trigger**, 218, 219
- Int data type**, 198
- intent locks**, 83, 230
- intermediate level pages**, 232, 356
- internal fragmentation**, 54, 161, 162
- internal memory pressure**, 44
- Intrinsic operator**, 248
- I/O (Input/Output), 175. See also disk I/O; network I/O**
 - affinity, 181
 - performance measurements, 414–422
- I/O_COMPLETION**, 70
- IOMeter**, 412, 454
- IOPS (Input/Output operations per second)**, 413
- Is Null statement**, 289, 290
- isolation**, 81
- isolation levels**, 81–82, 228–229
 - read committed, 81, 82, 229
 - read uncommitted, 81, 82, 229
 - repeatable read, 81, 82, 229
 - serializable, 82, 229
 - snapshot, 82, 229
- issue statements**, 94
 - information in, 94
 - well-prepared, 94

IsSystem

IsSystem, 325

iterative approach, 4, 17, 23, 42

Iterator Catchall operator, 246

J

JetStress, 454

join operators, 249–254

joins

hash, 249

merge, 249

Jose, Ian, 13

K

kernel mode, 25, 57, 149, 150

Key Lookup operator, 286

key-range locks, 230

keys

artificial, 209

candidate, 209

foreign, 210–212

natural, 209, 358

primary, 209–210, 358

surrogate, 209, 358

Keyset operator, 248

L

Language Element Catchall, 248

large result sets, 504

large value data types, 203–205

location matrix, 204

latency, 413–414

Lazy Spool operator, 246, 255

lazywriter process, 264

LAZYWRITER_SLEEP, 69

LCK_M_*, 70

leaf pages, 232, 356

level, 222

lightweight pooling, 181

load testing, 528–529

LoadSim, 454

Lock Pages in Memory option,
26, 151, 182

Lock:Deadlock, 110

Lock:Deadlock Chain, 110

**Locking/Blocking Performance
Dashboard report**, 488–490

**locks/locking, 82–83,
229–232. See also
deadlocks**

in action, 83–85

blocking, 83, 231, 312–316

monitoring, 89–91

preventative measures,
313–316

trace scenario for, 98

bulk update, 230

compatibility, 230

dynamic, 231

escalation, 82–83, 230–231

exclusive, 83, 230

granularity, 230

intent, 83, 230

key-range, 230

modes, 83, 230

problems, 83

schema, 230

shared, 83, 230

update, 83, 230

Log File Auto Grow, 474

Log File Auto Shrink, 474

Log Row Scan operator, 246

log shipping, 528

logical disks, 415–417

logical operators, 249, 255

logical processors, 28

Logical Unit Numbers. See LUNs

LoginName, 324

LogMan, 8, 35, 439

LogParser, 36

logs, System Monitor, 59

analyzing, with SQL Server,
59–64

correlated with SQL Profiler

traces, 30–31, 64,

137–139

data analysis, 35–38

imported into SQL Server,
60–64

management tools, 34–35

retaining, 32

**long-running queries. See
queries**

lookup operators, 257

lost updates, 80, 228

lsn, 222

LUNs (Logical Unit Numbers),

27, 51, 53, 158

M

machine architecture, 168–170

management, change, 529–530

master data tables, 501

max degree of parallelism, 48,
180, 181

Max Worker Threads, 48

MAX1 statement, 292, 293,
295

MAX2 statement, 292, 293,
295

MB/Sec, 413

measuring workloads, 330–336

memory, 144, 147–153

best practice, 144

bottlenecks, 17, 18, 25–26,
43–46, 144

configuration-based, 45–46

confirming, 44–45

DMVs and, 26

external memory pressure,
43–44

internal memory pressure, 44

types, 43–46

committed, 44

configuration, 149–153, 182

AWE, 150–152

/PAE option, 149–150,

151–152

64-bit processors, 151, 153

32-bit processors, 149–152

/3GB option, 149, 151–152

/USERVA option, 149

CPU impact, 144

I/O impact, 144

Min/Max server memory

options, 46

parameters, 182

performance, 145–146

problems, 56–57

physical, 147

pressure, 532

external, 43–44

internal, 44

target, 44

VAS, 26, 147

physical memory v., 46, 147

VMM, 147–148

memory pages/second counter,
26

MemToLeave area, 46

Merge Interval operator, 246

Merge Join operator, 246

merge joins, 249, 251–252

merge replication, 204

**Microsoft Management Console
(MMC)**, 18

minimizing caching effects, 269

Min/Max server memory

options, 46

mirroring, database, 54–55,

527–528

miscellaneous data types,

201–202

Missing Column Statistics, 474

missing counters, 33–34

Missing Indexes Report, 491

Missing Join Predicate, 474

mixed extents, 223

MMC. See Microsoft

Management Console

modes. See specific modes

mom and pop single user scale,

498

Money data type, 199

monitoring. See also System

Monitor

blocking locks, 89–91

deadlocks, 86–89

patterns/trends, data analysis

multi-core processors, 163. See also processor(s)
multiple executions, costly queries with, 130
multiple triggers, 219

N

natural keys, 209, 358
Nchar(n), 200
Nested Loop Joins, 249, 250–251
Nested Loops operator, 246
nested triggers, 219
.NET CLR, 154, 215
NetBIOS, 31
Netcap tool, 12
Netmon. See Network Monitor
network I/O, 153–154, 182–187
 network connection affinity, 183–187
 network packet size, 183
Network Monitor (Netmon), 12
network monitoring, 17, 18
nextPage, 222
NO ACTION, 211
NOCOUNT, 216
Nonclustered Index Spool, 255
nonclustered indexes, 236, 359–360
 Delete, 246
 Insert, 246
 Scan, 246
 Seek, 246
 Spool, 246
 Update, 246
non-persisted data types, 202
non-repeatable reads, 80, 228
Non-Uniform Memory Access (NUMA), 168–170
 hardware, 168, 178
 soft, 169–170, 179–180
normalization, 189–197
 ab-, 196–197
 anomalies, 190
 denormalization, 196–197, 510–512
 example, 190–195
 forms, 191
 rules, 191–195
NOT EXISTS statements, 270–271
NOT IN statements, 270–271
NTDomainName, 325
ntext, 200, 203
NTFS Allocation Unit Size, 161
ntsd debugger, 12
NTUserName, 325
nullability, 212–213
nullable columns, 288–290

nullable fields, 509–510
nulls, 212, 213
NUMA. See Non-Uniform Memory Access
Numeric (p, s), 198
Nvarchar(max), 200, 203
Nvarchar(n), 200

O

Object:Altered, 475
Object:Created, 474
Object:Deleted, 474
Object_id, 515
object-relational impedance mismatch, 197
objects. See specific objects
objID, 222
Ok, Slava, 13
OLAP (online analytical process) databases, 189, 196
OLTP (online transactional process) databases, 27, 189, 196
 simulation, SQLIOSim and, 462–463
ON clause, 227
On DELETE CASCADE, 211
ON DELETE SET DEFAULT, 211
ON DELETE SET NULL, 211
ON UPDATE CASCADE, 211
ON UPDATE SET DEFAULT, 212
ON UPDATE SET NULL, 211
online analytical process. See OLAP
online defragmentation, 534
Online Index Insert, 246
online transactional process. See OLTP
Open Source Project Hosting, 36
operators, 245–257. See also specific operators
 child, 254
 cursor, 248
 details/descriptions, 249
 join, 249–254
 logical, 249, 255
 lookup, 257
 parallelism, 248
 parent, 254
 physical, 249, 255
 in query plans, 245–257
 scan, 256–257
 seek, 256–257
 spool, 254–256
optimistic concurrency, 79
ORDER BY statements, 288
outer table, 250

P

/PAE (Physical Address Extensions), 149–150
 best practice, 150
 when to use, 151–152
page row offset arrays, 222–223
paged data, 148
pageID, 222
PAGEIOLATCH_*, 70, 78
pages, 221–224
 data, 221–223
 extent, 223–224
 headers, 221
 content details, 222
 intermediate level, 232, 356
 leaf, 232, 356
 root, 232, 356
paging, 148, 302–309
 CTE-based, 303–305
 to disk, high CPU and, 48
 ROW_COUNT-based, 305–306
 temp table, 307–309
 TOP@X, 306–307
Paging File counters, 26
PAL. See Performance Analysis of Logs
parallel processing, 180–181
parallel table scan, 76
 details, 77
 improvements, 78
parallelism, 76–78
 disabling, 77
 max degree of, 48, 180, 181
parallelism operators, 248
Parameter Table Scan operator, 246
Param.txt, 423
parent operator, 254
partition functions, 530–532
partition schemes, 531
partition switching, 530
partitioned tables/indexes, 53–54, 224–227, 397–407, 530–532
 benefits, 397–398
 creating, 226–227, 398–407
 deleting data from, 403–406
 DTA and, 407
 file groups, 226
 function, 226
 horizontal, 225
 for performance, 225–226
 prerequisites, 398
 production system, 406–407
 scheme, 226
 for subsets of data, 225
 utilization, 226
 vertical, 224–225
partitioning, ID-based, 510
patterns/trends, data analysis, 32

People database

People database, 353–355

Perf Stats Script, 12

PerfDash.chm file, 492, 494

Perfmon, 10, 18. *See also*

System Monitor

counters, 414–417

Performance Analysis of Logs (PAL), 36–38

Performance Console, 18

Performance Counter Library, rebuilding, 34

performance counters. See counters

Performance Dashboard reports,

10–11, 58, 73, 471–494

architecture, 485–487

Current Waiting Requests, 488

Help file, 492

installing, 478–481

limitations, 492–494

Locking/Blocking, 488, 490

Missing Indexes report, 491

online information, 494

prerequisites, 477

Reporting Services, 487

running, 481–485

security and, 487

System CPU Utilization, 488, 489

Performance Library DLLs, 16

performance logs

analyzing, with SQL Server, 59–64

correlated with SQL Profiler traces, 30–31, 64, 137–139

data analysis, 35–38

imported into SQL Server, 60–64

management tools, 34–35

retaining, 32

Performance Logs and Alerts,

18, 22

Performance Monitor. See

System Monitor

performance objects, 16. *See also specific performance objects*

Performance Options window, advanced tab, 175–176

performance problems, 40–42.

See also bottlenecks

configuration-based, 41

CPU, 57–59

disk, 56

memory, 56–57

schema-based, 41–42

typical, 55–59

performance testing, 410

performance thresholds, XML templates for, 36

performance tuning

blogs, 13

indexes, 353–407

with DTA, 365–393

schema, 189–237

science, 3–4

SQL Server configuration settings, 171–187

steps, 4–7

data analysis, 7

data collection, 6–7

examples, 7–10

iterative approach, 4, 17, 23, 42

plan of attack, 6

problem statement, 4–6

tools, 10–13

trace scenario, 98

T-SQL, 239–318

PERSISTED option, 205

personas. See user roles

pesimistic concurrency, 79–80

phantom reads, 80, 228

Physical Address Extensions.

See /PAE

physical disks, 415–417

physical memory, 147

VAS v., 46, 147. *See also* memory

physical operators, 249, 255

physical processors, 28

PhysicalDisk

-Avg. Disk Queue Length, 50

-Avg. Disk sec/Read, 50

plan cache, retrieving query plan from, 262–263

plan guides, 534

best practice, 537

example, 534–537

plan of attack, 6

Pminlen, 222

polling interval. See sampling interval

Population Query operator, 248

preemptive scheduling, 68

Prepare SQL, 324

prerequisite indexes, 270

pressure, memory, 532

external, 43–44

internal, 44

preventative measures, 13–14

prevPage, 222

primary keys, 209–210, 358

constraints, 209–210

Priority Boost, 41, 176

% Privilege Time, 28, 57

proactive use

Alerts, 32–33

System Monitor, 29

problem statement, 4–6

procedure cache, 265–268

best practices, 267–268

freeing, 266

in-memory plans, 265

internal storage, 265

zero cost plans, 265

process object, 28

Processor

%Privilege Time, 47

%User Time, 47

%User Time - sqlservr, 47

processor(s), 28. *See also CPUs*

cache, 166–168

logical, 28

object, 28

parallel processing, 180–181

physical, 28

64-bit IA64

configuring, 180

memory configuration, 151, 153

System Monitor and, 29–30

x64 v., 166

system architecture, 168–170

32-bit x64, 162, 166

32-bit x86

configuring, 180

hyper-threading, 48,

164–165, 176–178

memory configuration, 149–152

multi-core, 163

% Processor time, 29, 57

Process-Working Set, 44

production environment, server-side traces in, 100–101, 136

Professional SQL Server 2005

CLR Programming with

Stored Procedures,

Functions, Triggers,

Aggregates, and Types

(Wrox), 215, 274

Profiler. See SQL Profiler

Profiler90.exe, 99

PSS SQL Server Engineers

website, 12

PSSDiag, 12

Q

queries

costly

with multiple executions, 130

with single execution, 130

SQL Profiler/server-side trace, 122–137

DTA tuning of individual, 366–374

gathering facts on, 243–268

long-running

serial-SCSI disks

- identified with SQL Profiler, 114–122
 - trace scenario, 98
 - rewriting, pushing predicates deeper, 271–273
 - Query Opt team**, 13
 - query optimizer**
 - components, 240–242
 - gameplan/function, 239–242
 - query plans**, 243–268
 - caching issues, 309–311
 - display modes, 258–261
 - essentials, 244–245
 - graphical, 243–245
 - interpreting, 257–258
 - operators in, 245–257
 - recompiling, 266
 - retrieving, from plan cache, 262–263
 - simulated stress testing for, 316–317
 - text-based, 243
 - queue waits**, 68
- R**
- RAID**, 154, 157
 - RAID0+1**, 157
 - RAID1**, 157
 - RAID5**, 157
 - RAID10**, 157
 - Random Read performance**, 451–452
 - Random Write performance**, 452
 - RandomUser section**, 462
 - RDBMS**. *See* Relational Database Management System
 - read committed isolation level**, 81, 82, 229
 - read uncommitted isolation level**, 81, 82, 229
 - ReadAheadUser section**, 462
 - read-oriented tables**, 232
 - reads**
 - dirty, 80, 228
 - non-repeatable, 80, 228
 - phantom, 80, 228
 - Real data type**, 199
 - real-life performance testing**, 410–411
 - rebinds**, 254
 - rebuilding Performance Counter Library**, 34
 - recompilation/compilation, excessive**, 49
 - recursion**
 - direct, 220
 - indirect, 220
 - recursive triggers**, 220
 - redundancy anomaly**, 190
 - Refresh Query operator**, 248
 - Relational Database Management System (RDBMS)**, 81
 - Reliability and Performance Monitor**, 18
 - Relog**, 35
 - Remote Delete operator**, 246
 - Remote Insert operator**, 246
 - Remote Query operator**, 246
 - Remote Registry service**, 31
 - Remote Scan operator**, 246
 - remote servers, System Monitor from**, 32
 - Remote Update operator**, 247
 - Repartition Streams operator**, 248
 - repeatable read isolation level**, 81, 82, 229
 - replaying workloads**, 337–351
 - replication, merge**, 204
 - Replication, SQL Server**, 528
 - report user role**, 499
 - Reporting Services, Performance Dashboard reports and**, 487
 - Reserved**, 97
 - reservedCnt**, 222
 - resource utilization**, 24–25
 - resource waits**, 68
 - RESOURCE_SEMAPHORE**, 69
 - RESOURCE_SEMAPHORE_QUERY_COMPILE**, 70
 - Result operator**, 248
 - retaining performance logs**, 32
 - rewinds**, 254
 - RID Lookup**, 247, 257, 286
 - rollover trace files**, 126–127
 - root pages**, 232, 356
 - Row Count Spool operator**, 247, 255
 - row density**, 233. *See also* density
 - row versioning**, 80, 85–86
 - ROW_COUNT-based paging**, 305–306
 - RPC Output Parameter**, 324
 - RPC:Completed**, 97, 108, 110, 115, 323, 324
 - RPC:Starting**, 97, 324
- S**
- sample database server disk configuration**, 21
 - sampling interval**, 21, 23, 24
 - SANs (storage area networks)**, 155–156
 - best practice, 156
 - vendors, 27, 49
 - SAS (serial-SCSI) disks**, 154–155
 - SATA (serial-ATA) disks**, 154–155
 - scan operators**, 256–257
 - schedulers**, 68, 69
 - runnable, 68, 69
 - running, 68, 69
 - suspended, 68, 69
 - schema**, 189–237
 - bottlenecks
 - CPU, 48–49
 - disk, 53–54
 - memory, 46
 - data performance, 221–237
 - data quality, 189–220
 - locks, 230
 - normalization, 189–197
 - anomalies, 190
 - denormalization, 196–197, 510–512
 - example, 190–195
 - forms, 191
 - rules, 191–195
 - OLAP database, 189, 196
 - OLTP database, 27, 189, 196
 - performance problems, 41–42
 - tuning, 189–237
 - science, of performance tuning**, 3–4
 - scopes**, 219
 - scripts**, 172
 - SQL Server settings inspected with, 173–175
 - SCSIPort drivers**, 157
 - searchable arguments**, 282–285
 - searchable predicate performance issues**, 283
 - Second Normal Form**, 191
 - sectors**, 159
 - alignment, 159–160
 - best practice, 160
 - security, Performance Dashboard reports and**, 487
 - seek operators**, 256–257
 - Segment operator**, 247
 - SELECT ***, 276–280
 - Select Distinct memory_node_id**, 169
 - SELECT statements**, 274
 - singleton, 292–295
 - UDFs in, 274–276
 - Sequence operator**, 247
 - SequenceProject operator**, 247
 - serial-ATA disks**. *See* SATA disks
 - serializable isolation level**, 82, 229
 - serial-SCSI disks**. *See* SAS disks

server

- server**
 - hardware configuration, 143–170
 - resources, 145–146
- server instance**, 94
- Server Memory Change**, 474
- Server Processes Trace Data**, 102
- Server Property window**, 172, 173
- ServerName**, 325
- server-side traces**. *See* **traces, server-side**
- SET DEFAULT**, 211
- SET NULL**, 211
- SET TRANSACTION ISOLATION LEVEL T-SQL command**, 82
- set-up role**, 499
- shared locks**, 83, 230
- Showplan**
 - _All, 259
 - best practices, 261
 - Statistics Profile Event Class, 276
 - _Text, 259
 - _XML, 258–259
 - XML event class, 107–108
 - SQL Profiler and, 107–109
- ShrinkUser section**, 462
- signal wait times**, 68, 76
- simple select views**, 278–279
- simulated stress testing, for query plans**, 316–317
- single execution, costly queries with**, 130
- single-responsibility principle**, 505, 506
- singleton SELECT statements**, 292–295
- single-use columns**, 507–509
- single-use tables**, 506–507
- 64-bit processors**. *See* **processor(s)**
- sizing**, 524–525
 - incorrect, 532–533
- SLEEP_BPOOL_FLUSH**, 69
- sliding windows**, 398, 406
- slotCnt**, 222
- smalldatetime**, 199
- Smallint**, 198
- smallmoney**, 199
- snapshot isolation level**, **82, 229**. *See also* **row versioning**
- Snapshot operator**, 248
- soft NUMA**, 169–170, 179–180
- software application core architecture**, 504–505
 - performance unknowns, 505
- Sort operator**, 247
- Sort Warnings event**, 474
- SOS_SCHEDULER_YIELD**, 70, 76
- sp_blocker_pss**, 89, 104
- sp_configure ‘clr enabled,’** 174
- sp_configure command**, 105, 182
- sp_configure ‘show advanced options,’** 174–175
- sp_helpindex**, 242
- SPID**. *See* **system server process ID**
- Split operator**, 247
- spool operators**, 247, 254–256
- SP:StmtCompleted**, 115
- sp_trace_create**, 133
- sp_trace_setevent**, 134
- sp_trace_setfilter**, 134, 135
- sp_trace_setstatus**, 136
- SQL BP website**, 13
- SQL CAT website**, 13
- SQL Engine Tips**, 13
- SQL Manageability**, 13
- SQL Profiler**, 10, 17, 65, 93–139, 532
 - capturing
 - blocking events, 104–107
 - deadlock graphs, 110–114
 - Showplan XML data, 107–109
 - costly queries, 122–137
 - improvements, 39
 - long-running queries, 114–122
 - analysis/tuning, 119–122
 - issue analysis, 115
 - scenario, 115
 - setup, 115–118
 - tracing, 118–119
 - traces. *See* **traces, SQL Profiler workloads**
 - capturing, 321–330
 - measuring, 330–336
 - replaying, 337–351
- “SQL Server 2000 I/O Basics,”** 469
- SQL Server 2005**
 - additional features, 525–526
 - analyzing performance logs with, 59–64
 - Books Online, 91
 - configuration settings, 171–187. *See also* **configuration settings, SQL Server**
 - editions, 525
 - importing System Monitor logs into, 60–64
 - Performance Dashboard reports, 10–11, 58, 73, 471–494
 - architecture, 485–487
 - Current Waiting Requests, 488
- Help file, 492
- installing, 478–481
- limitations, 492–494
- Locking/Blocking, 488, 490
- Missing Indexes report, 491
- online information, 494
- prerequisites, 477
- Reporting Services, 487
- running, 481–485
- security and, 487
- System CPU Utilization, 488, 489
- service pack, 494
- supportability, 39, 422–427
- Waits and Queues, 494
- SQL Server Best Practices toolbox**, 11
- SQL Server Best Practices website**, 11
- SQL Server Buffer Pool**, 26
- SQL Server Import and Export Wizard**, 61–64
- “SQL Server I/O Basics Chapter 2,”** 469
- SQL Server Management Studio**, 10
 - SQL Server settings configuration, 172–173
- SQL Server Operating System (SQLOS)**, 40
- SQL Server performance problems**. *See* **performance problems**
- SQL Server Replication**, 528
- SQL Server storage team website**, 13
- SQL Trace**. *See* **Trace, SQL**
- SQL:BatchCompleted**, 97, 108, 110, 115, 323, 324
- SQL:BatchStarting**, 97, 324
- SQLCMD**, 11
- SQLDiag**, 12
- SQLIO**, 412, 423–454
 - advanced tests, 435–439
 - results, 439–450
 - best practices, 454
 - intermediate tests, 427–432
 - results, 432–434
 - running, 422–424
 - basic I/O test, 425
 - parameter files, 425–427
 - summary results from different storage subsystems, 453–454
- Sqllo.exe**, 423
- SQLIOSim**, 409, 412, 454–459
 - best practice, 458–459
 - command line, 467

table(s)

- configuring, 461–467
 - for DBCS simulation, 465–467
 - for DSS simulation, 464–465
 - for OLTP simulation, 462–463
 - errors, best practice, 469
 - interpreting results, 467–469
 - Sqlsim.hwcache.cfg.ini**, 460
 - Sqlsim.nothrottle.cfg.ini**, 460
 - Sqlsim.seqwrites.cfg.ini**, 460–461
 - Sqlsim.sparse.cfg.ini**, 461
 - SQLIOStress**, 412, 454
 - SQLOS**. *See* **SQL Server Operating System**
 - SQLServer:Access Methods**
 - Full scans/sec, 50
 - Page Splits/sec, 50
 - SQLServer:Buffer Manager**
 - Buffer cache hit ratio, 45
 - Checkpoint pages/sec, 45
 - Lazy writes/sec, 45
 - Page Life Expectancy, 44–45
 - Stolen pages, 45
 - Target Pages, 44
 - Total Pages, 44
 - SQLServer:Database Mirroring**
 - Log Bytes Received/sec, 55
 - Log Bytes Sent/sec, 55
 - Log Send Queue KB, 55
 - SQLServer:General Statistics**
 - Temp Tables Creation Rate, 52
 - Temp Tables For Destruction, 52
 - SQLServer:Memory Manager, Memory Grants Pending**, 45
 - SQLServer:SQL Statistics**
 - SQL Compilations/sec, 47, 49
 - SQL Re-Compilations/sec, 47, 49
 - SQLServer:Wait Statistics**
 - Lock Waits, 55
 - Memory Grant Queue Waits, 55
 - Page I/O Latches, 55
 - SQLTRACE_BUFFER_FLUSH**, 69
 - Sql_variant**, 201
 - StartTime**, 325
 - statistics, index**, 241–242, 360–364
 - creating, 362–363
 - updating, 363–364
 - Statistics IO option**, 261
 - Statistics Profile option**, 260
 - Statistics Time option**, 261
 - Statistics XML option**, 259–260
 - storage area networks**. *See* **SANs**
 - storage subsystem**
 - best practice, 470
 - mapping, 451
 - performance, 409–470
 - bottlenecks, 450–452
 - measuring tools, 412
 - metrics, 413–414
 - reliability, 454
 - “Storage Top 10 Best Practices,”** 469
 - stored procedures**, 214–218
 - advantages/disadvantages, 214
 - CLR, 214, 215
 - database access, 216–217
 - implementing, 215–218
 - naming convention, 217–218
 - T-SQL, 214, 215
 - StorPort drivers**, 157
 - Stream Aggregate operator**, 247
 - stress testing**, 410
 - supportability, SQL Server 2005**, 39, 422–427
 - surrogate keys**, 209, 358
 - swapped data**, 148
 - Switch operator**, 247
 - switches**, 155, 156
 - symmetric multi-processing (SMP)**, 168
 - sys.dm_exec_requests**, 70–71
 - sys.dm_os_waiting_tasks**, 71
 - sys.dm_os_wait_stats**, 71–72
 - example, 74–75
 - sys.dm_db_index_physical_stats**, 54
 - sys.dm_db_index_usage_stats**, 53, 533
 - sys.dm_io_backup_tapes**, 418
 - sys.dm_io_cluster_shared_drives**, 418
 - sys.dm_io_pending_io_requests**, 418
 - output, 419
 - sys.dm_io_virtual_file_stats**, 418–419
 - output, 420–421, 422
 - sys.dm_os_waiting_tasks**, 312
 - sys.dm_os_wait_stats**, 312
 - Sysname**, 201
 - system architecture**, 168–170
 - System CPU Utilization**, 488, 489
 - “System Deadlocks,”** 312
 - System Monitor**, 10
 - best practices, 31–33
 - configuration, log directly to database, 60
 - data generation, 23–24
 - impact, 22–23
 - logs, 59
 - analyzing, with SQL Server, 59–64
 - correlated with SQL Profiler traces, 30–31, 64, 137–139
 - data analysis, 35–38
 - imported into SQL Server, 60–64
 - management tools, 34–35
 - retaining, 32
 - monitoring
 - database mirroring
 - performance, 54–55
 - remote servers, 31, 32
 - server resources, 15–38
 - SQL Server resources, 39–65
 - wait types, 55
 - need for, 15–16
 - overview, 16–18
 - proactive use, 29
 - 64-bit systems, 29–30
 - tasks, 16–17
 - when to use, 16
 - system server process ID (SPID)**, 325, 327
 - system tables**, 502
 - System-Processor Queue Length**, 47
 - sys.traces**, 99–100
- T**
- table(s)**
- association, 501
 - data type, 202
 - fragmentation, 54
 - historical, 502
 - inner, 250
 - large, 530–532
 - master data, 501
 - outer, 250
 - partitioned, 53–54, 224–227, 397–407, 530–532
 - benefits, 397–398
 - creating, 226–227, 398–407
 - deleting data from, 403–406
 - DTA and, 407
 - file groups, 226
 - function, 226
 - horizontal, 225
 - for performance, 225–226
 - prerequisites, 398
 - production system, 406–407
 - scheme, 226
 - for subsets of data, 225
 - utilization, 226
 - vertical, 224–225
 - read-oriented, 232
 - single-use, 506–507
 - system, 502
 - temp, 217, 273–274
 - paging, 307–309
 - table variables v., 274, 518
 - temporary, 217
 - throughput, 515
 - transactional detail, 502

table(s) (continued)

- table(s) (continued)**
 transactional header, 501
 type, 501
 workflow, 502
 write-oriented, 232
- Table Delete operator**, 247
- Table Insert operator**, 247
- Table Scan operator**, 247, 256
- Table Spool operator**, 247, 255
- Table Update operator**, 247
- table variables**, 217, 273, 274
 temp tables v., 274, 518
- Table-valued Function operator**, 247
- target memory**, 44
- temp tables**, 217, 273–274
 paging, 307–309
 table variables v., 274, 518
- tempDB**, 28, 51–52, 157, 158
 best practices, 51, 52
 contention within, 52–53
 usage, 516–517
- templates**, 97
- testing**
 benchmark, 514–515
 trace scenario for, 98
 load, 528–529
 performance, 410
 real-life performance, 410–411
 simulated stress testing, for
 query plans, 316–317
 stress, 410
- text data type**, 200, 203
- TextData**, 324
- Third Normal Form**, 191
- 32-bit processors**. *See* **processor(s)**
- threads**, 164. *See also* **hyper-threading**
- /3GB option**, 149
 when to use, 151–152
- throughput**, 503
 use-case, 503–504
- Timestamp**, 202
- Tinyint**, 198
- tools, performance tuning**, 10–13
- Top operator**, 247
- TOP statement**, 292, 293, 295
- TOP@X paging**, 306–307
- turnBits**, 222
- _TOTAL**, 20, 27
- % Total Processor time**, 28
- trace events**, 473–477
- trace files**, 97
 rollover, 126–127
- trace flag** 1222, 86, 88
- Trace Logs**, 18
- trace scenarios**, 97–98
 benchmark testing, 98
 deadlock causes, 98
 long-running queries, 98
 performance tuning, 98
- Trace, SQL**, 93
- trace tables**, 97
- traces**, 96
 default, 473–477
- traces, server-side**, 99
 costly queries, 122–137
 analysis of data, 126–135
 execution, 126
 generation, 123–126
 in production environment,
 100–101
 best practices, 136
 SQL Profiler traces v., 99
- traces, SQL Profiler**, 30, 99
 capturing
 blocking events, 104–107
 deadlock graphs, 110–114
 Showplan XML data,
 107–109
 correlated with System Monitor
 data, 30–31, 64,
 137–139
 data collection timeframe, 102
 for baseline purposes, 102,
 103
 design, best practice, 94
 events, best practice, 97
 long-running queries, 118–119
 options, 99–102
 output data, configuring,
 101–102
 preparations for setup, 93–98
 server-side traces v., 99
 terminology, 94–98
 where to start, 102, 103
- tracks**, 159
- transactional detail tables**, 502
- transactional header tables**, 501
- transactional user roles**, 499
- transactions**, 81, 227
 long-running, 504
- Transact-SQL (T-SQL)**, 59, 60,
 65
 operators, 248
 performance tuning, 239–318.
See also query plans
 endgame, 268–317
 essential tools, 269–270
 with hints, 309–311
 for indexes, 285–291
 patterns/idioms, 292–309
 predicate, 280–285
 process flow, 268–269
 query optimizer
 components, 240–242
 gameplan/function, 239–242
 stored procedures, 214, 215
- trends/patterns, data analysis**, 32
- triggers**, 218–220
 AFTER, 218, 219
 bottlenecks, 220
 chain, 219
 DDL, 218, 219
 DML, 218–219
 example, 220
 INSTEAD OF, 218, 219
 multiple, 219
 nested, 219
 recursive, 220
- T-SQL**. *See* **Transact-SQL**
- tuning**. *See* **performance tuning**

U

- UDFs**. *See* **user-defined functions**
- UDX operator**, 247
- unicode character data types**, 200
- uniform extents**, 223
- unique constraints**, 212
- uniqueidentifier data type**, 202,
 203–205, 357–358
 advantages, 204–205
 limitations, 204
- update anomaly**, 190
- update locks**, 83, 230
- Update (Database Engine) operator**, 247
- % Usage**, 26
- % Usage Peak**, 26
- user**
 location, 500
 login, 95
 mode, 25, 149, 150
 process, 264
 response time requirements,
 500
 scale, 498
 corporate, 498
 infinite, 498
 mom and pop, 498
 segmentation, 500
 uptime requirement, 500–501
- user roles**, 498–499
 access requirements, 500
 gatekeeper, 499
 report, 499
 set-up, 499
 transactional, 499
- % User Time**, 28, 57
- user-defined functions (UDFs)**, 274
 Correlated Subquery v.,
 275
 in SELECT statements,
 274–276

XML templates, performance thresholds

user-mode address space,
149

/USERVA option, 149
Using SQLIO.rtf, 423

V

Varbinary(max), 201, 203

Varbinary(n), 201, 203

Varchar(max), 200, 203

Varchar(n), 199, 203

VAS. See virtual address space

views

catalog, 472, 533

complex, 279–280

DMVs, 11, 472–473

memory bottlenecks and, 26

functions in, 512–513

indexed, 364

simple select, 278–279

virtual address space (VAS), 26,
147

physical memory v., 46, 147

**Virtual Memory Manager
(VMM),** 147–148

**Visual Studio 2005 Team Edition
for Database Professionals,**
12

**volumes, 160–161. See also
disks**

basic, 160, 161

dynamic, 160–161

W**Wait Stats**

in action, 73–79

report, 72, 73

WAITFOR, 69

**waits, 55, 67–91. See also
specific waits**

architecture, 68–69

common, 69–70

description of all, 69

external, 68

queue waits, 68

resource, 68

tracking, 70–72

warm server, 44

While operator, 248

windbg debugger, 12

**Windows Advanced User Right
Lock Pages in Memory
option,** 151

Windows Failover Cluster,
161

Windows fibers, 181

Windows Network Monitor, 17

Windows Resource Kit, 33

Windows-On-Windows mode, 29

workflow tables, 502

workloads

capturing, 321–330

characterizing, 322–326

DTA tuning, 392–393

measuring, 330–336

modifying, 327–330

performance reference,
334–336

replaying, 337–351

multiple workloads on single

server, 337, 346–351

requirements for, 324–326

scenarios, 337–351

validating performance

improvement, 337,

340–343

workload generation,

338–340

workload in different

environments, 337,

343–346

subsets, 327–329

WRITELOG, 70, 77

reduced, 78, 79

write-oriented tables, 232

X

**x64 processors. See
processor(s)**

**x86 processors. See
processor(s)**

xactreserved, 222

Xaio, Wei, 13

Xml data type, 201

**XML templates, performance
thresholds,** 36