



Inside Microsoft® SQL Server® 2008: T-SQL Querying

*Itzik Ben-Gan, Lubor Kollar,
Dejan Sarka, Steve Kass (Solid
Quality Mentors)*

To learn more about this book, visit Microsoft Learning at
<http://www.microsoft.com/MSPress/books/12804.aspx>

9780735626034

Microsoft®
Press

© 2009 Itzik Ben-Gan; Dejan Sarka; Lubor Kollar; Steve Kass.

Index

Symbols and Numbers

#CachedPages sample table, 592–94

.NET

CLR database code, 476–77

reference types, 484

∈ (set membership operator), 44–45

@expression argument, 610–11

@length argument, 610–11

@myOD variable, 605

@offset argument, 610–11

A

abstraction

Accumulate method, 482–83

Actual Execution Plan, 653

acyclic graphs, 660

ad hoc paging, 350–51

ad hoc queries, 136

Add Outer Rows phase, 5, 11

adjacency list model, 99–100

AFTER triggers, 110

aggregate functions

OVER clause, 29

subqueries, 14

aggregate product, pivoting, 475

aggregate window functions, 454

aggregation, 445. *See also* pivoting

bitwise operations specialized solution, 490–94

cumulative, 453–57

custom, 473–99

duration by query, 155–57

OVER clause, 445–48

PIVOT operator, 24

product specialized solution, 488–90

query signature, 157–59

running, 451–52

sliding, 457–59

specialized solutions, 487–99

string concatenation specialized solution, 487–88

tiebreakers, 448–51

top wait isolation, 137–38

user-defined aggregates (UDA), 476–82

year-to-date, 459–60

Aldous, David, 292

algebra, relational, 90–104

algorithms, 43–44, 277–79. *See also* complexity

binary search, 282

joins, 421–29

linear complexity, 133–34

LISLP problem solution, 292

O(n log n), 288–89

quadratic sorting, 288

running time comparisons, 286

scale, 279–82

sorting, 287–89

swapping, 289

ultra sort, 289

aliases

column, 319–20, 322

reuse, 16

SELECT list, 14–15

table, 606

ALL predicate, 316–18

all-at-once operations, 14–15

allocation order scans, 192, 208–19

allocation units, 189

alphabetical order, 43, 57–58

ALTER DATABASE option, 647

ALTER INDEX statement, 258

ALTER TABLE SWITCH, 645–46

An Introduction to Database Systems (Date), 83, 125

Analysis Services, wait analysis, 140

analytical ranking functions, 330–32

NTILE, 354–59

RANK and DENSE RANK, 352–54

ROW_NUMBER, 332–52

tile number, 354–59

ancestors, iteration/recursion, 681–84

Anchor Member, 328–30

anchor rows, 549–50, 592

And operator, 68–70

ALL predicate, 317

logical transformations, 556–59

ANSI SQL, 1

aggregate window functions, 454

constraints, 105

cross joins, 396–97

cursors, 17

INSERT VALUES clause, 562

join logical processing order, 409

join syntax, 389–90

nonsupported joins, 401

NULL values, 111

ORDER BY clause, 16

outer joins, 399

OVER clause subclauses, 459

relations, 103

semicolon termination, 322

set operations, 436

two-valued logic, 623

anti-semi joins, 415–16

antisymmetric relation properties, 75–76

ANY predicate

ANY predicate, 316–18
 APPLY operator, 21–22, 527, 535–36
 TOP n for each group, 537–43
 arguments
 common table expressions, 323
 derived tables, 320–21
 arrays, 287
 separating elements, 429–35
 assembly creation and deployment, 482–87
 assignment SELECT, 612–14
 assignment UPDATE, 614–16
 asterisk, 306
 asymptotic complexity, 283
 atomic types, 86
 attributes
 pivoting, 460–64
 relations, 85–87
 scalar, 86–87
 tuples, 84
 types, 86
 AUTO_CREATE_STATISTICS property, 228
 auxiliary table of numbers, 359–62
 average fragmentation in percent, 256–57
 AVG aggregate, 453–57

B

bag theory, 64–65
 balanced trees, 191
 base columns, 14
 bcp.exe, 565
 BEFORE triggers, 109
 benchmarks, row numbering, 344–48
 Ben-Gan, Gabriel, 757
 Ben-Gan, Itzik, 44
 Bernoulli sampling algorithm, 268
 BETWEEN predicate, 651–52
 Big Oh notation, 283–84
 BigNumSeq table, 364
 bill of materials (BOM) example, 663–66
 bin packing problem, 281
 binary search algorithm, 282
 bitmap filters, 426–28
 Bitmap operator, 426–28
 bitmap pages, 190–91
 bitwise AND operation, 490–94
 bitwise operations specialized solution, 490–94
 bitwise OR operation, 490–94
 bitwise XOR operation, 490–94
 BLOBs, 290
 block sequence values, 597–98
 blocking sequences, 596
 BOM sample table, 663–66
 Boolean algebra, 74
 Boolean expressions, 65–66
 restriction expression, 91–92
 T-SQL, 67

Boolean operators, 90–91
 Boyce-Codd normal form, 117–19
 braces, 45
 B-trees, 189. *See also* subtrees; trees
 INSERT SELECT statement scenarios, 578–89
 BULK INSERT statement, 567–68
 BULK rowset provider, 565–67
 bushy plans, 411–14

C

C# code
 UDA creation, 477–82
 user-defined functions, 160–61
 cache
 clearing, 171–72
 query execution plans, 171
 calculus, relational, 90–104
 candidate keys, 105–06
 cardinal numbers, 59–60
 cardinality
 notation, 56
 sets, 56–57
 Cartesian Product phase, 3, 7–8
 Cartesian products, 53–54
 cross joins, 390–91
 Cascade implementation, 107–08
 CASE expressions
 aggregate product specialized solution, 489
 EXISTS predicate, 310–12
 NTILE function, 355
 outer joins, 400
 PIVOT operator, 24
 pivoting, 462–63
 unsupported logical phrases, 442
 characteristic function definition, 55
 CHARINDEX function, 432
 CHECK constraints, 108–09, 670
 MERGE statement, 632
 CHECKSUM, 554
 Chen, Peter, 87
 chiasitic relationships, 410–11
 Cities sample table, 666–70
 CLR (Common Language Runtime). *See* Common Language Runtime (CLR)
 Clustered Index Scan operator. *See* index scans; clustered indexes
 Clustered Index Seek operator.
 See index seek; clustered indexes
 clustered indexes, 191–95
 index seek + ordered partial scan, 250
 index tuning, 169–70
 ordered scan, 202–04
 seek + ordered partial scan, 233–38
 unordered index scan, 245
 unordered scan, 198–201
 clustering key, 196
 CMEMTHREAD wait, 136

- COALESCE expression, 312
- COALESCE function, 475
- concurrency
 - wait analysis, 137
- Codd, Edgar F., 1
- code revision, query tuning, 269–76
- collation, 57–58
- columns
 - aliases, 14–16, 319–20, 322
 - base, 14
 - Boyce-Codd normal form, 117–19
 - copy generation, 26–27
 - extraction of elements, 27
 - fifth normal form, 120–21
 - first normal form, 113–15
 - foreign key, 106–08
 - fourth normal form, 119–20
 - identity, 595–96
 - IDENTITY property, 110
 - included nonkey, 237
 - key, 105–06
 - nonunique sort column method,
 - with tiebreaker, 337–38
 - nonunique sort column method,
 - without tiebreaker, 338–40
 - pivoting. *See* pivoting
 - second normal form, 115–16
 - SELECT list ordering, 17
 - set operations, 32
 - spreading, 24
 - third normal form, 116–17
 - unique sort column method, 335–37
- Common Language Runtime (CLR)
 - database code, 476–77
 - user-defined data type, 188
 - user-defined functions, 160–61
- common table expressions (CTEs),
 - 321–22
 - arguments, 323
 - auxiliary table of numbers, 362
 - column aliases, 322
 - data modification, 324–25
 - DELETE statement, 606
 - EmpsPaths, 722–26
 - EmpsRn, 722–26
 - inline function definitions, 325–26
 - level limiting, 680
 - multiple, 323
 - multiple references, 324
 - recursive, 327–30
 - Tiles, 495
 - unsupported logical phrases, 442
 - UPDATE statement, 608
 - views, 325–26
 - WITH keyword, 322
- compatibility mode, 398
- compatibility views, 171
- Completed event classes, 150
- complexity, 277–79. *See also* algorithms
 - asymptotic, 283
 - best- and worst-case, 283
 - Big Oh notation, 283–84
 - comparisons, 285–86
 - constant, 283
 - exponential and superexponential,
 - 134–35
 - linear, 133–34
 - polynomial and nonpolynomial, 284–85
 - sublinear, 282
 - technical definitions, 283
- composable DML, 636–38
- composite joins, 397
- Concatenation operator, 330
- connected graphs, 660
- consistency vs. correctness, 105
- constant complexity, 283
- Constant Scan operator, 643–45, 651–52
 - parallelism, 652–57
- constraints, 104–05
 - check. *See* CHECK constraints
 - declarative, 105–09
 - join dependency, 121
 - order of enforcement, 110
- context, mathematics and, 41–43
- contrapositives, 71
- control-of-flow statements, 65–66
- correctness vs. consistency, 105
- correlated subqueries, 297–98, 302
 - EXISTS predicate, 305–14
 - tiebreaker, 302–06
- COUNT aggregate, 466
- COUNT(*), 14, 30, 655
- COUNT(O.orderid), 14
- COUNT(val), 502
- covering indexes, 201
- CREATE AGGREGATE command, 485
- CREATE ASSEMBLY command, 485
- CREATE CLUSTERED INDEX statement, 645
- CREATE INDEX command, 548, 647–48
- CREATE STATISTICS command, 645, 647
- CREATE SYNONYM command, 360
- CROSS APPLY operator, 21, 536
- Cross Join phase. *See* Cartesian Product phase
- cross joins, 7, 390–95. *See also*
 - Cartesian Product phase
- CTEs (common table expressions). *See*
 - common table expressions (CTEs)
- CUBE subclass, 506
- CUBE subclass, 511–12
- cumulative aggregation, 453–57
- cursors, 17
 - custom aggregations, 473
 - gaps solution, 374
 - islands solution, 383–84
 - query tuning, 268–76
 - row number calculation, 341–42

custom aggregation, 473–99

 pivoting, 474–99

custom sequences, 596–600

CustomerData sample table, 567

Customers sample table, 306, 308

 cross joins, 390–95

 hash joins, 425–26

 merge joins, 424–25

 MERGE statement, 616–17

 multiple joins, 408–11

 triggers, 627

 UPDATE statement, 607

CustomersDim sample table, 629

CustomersStage sample table, 616–17

CXPACKET wait, 136, 145

cycles, iteration/recursion, 691–94

D

Dafni, Adi, 757

DAG (directed acyclic graph).

 See directed acyclic graph (DAG)

data

 aggregation. *See* aggregation

 bad, domains and, 47–48

 collection, 187

 deletion, 601–06

 duplicate, removal, 601–03

 insertion, 561–601

 integrity, 104–11. *See also* constraints

 large value type updates, 610–11

 maintenance, materialized path,
 695–701

 merging, 616–28

 model, 83

 modification, CTEs, 324–25

 modification, TOP option, 531–33

 OUTPUT clause, 628–38

 preparation, sample, 259–65

 processing, 83

 schema, 83

 structure, 277, 279

 temporal, 122

 trend identification, 291

 type. *See* types

 updating, 606–16

data collector, 187

data definition language (DDL), 460

 partitioned views and tables, 640

 triggers, 109

data integrity

 domain, 108–09

 enforcing, 109–11

 entity, 105–06

 referential, 106–08

Data Manipulation Language (DML), 460

 composable, 636–38

 constraints, 105

 relations, 103–04

 triggers, 109

Data Modeling Essentials (Simsion and Witt),
 111–12

database

 data integrity. *See* data integrity

 FULL recovery model, 571–74

 generalization, 124–25

 I/O analysis, 145–48

 ID, 256

 non-FULL recovery mode, 574–75

 NULL values, 110–11

 relational model. *See* relational database model

 schema, 104

 specialization, 124–25

Database Design for Smarties (Muller), 112

Database Engine Tuning Advisor, 187

DATE type, 48

 binary string conversion, 450

date values, 42

Date, C. J., 83, 122, 125

DATEADD function, 368, 373, 392

DATEDIFF function, 373

DATETIME type, 417

 accuracy level, 458

 binary string conversion, 450

DBCC DROPCLEANBUFFERS, 118

DBCC FLUSHPROCINDB, 171

DBCC FREEPROCCACHE, 171

DBCC FREESYSTEMCACHE, 171–72

DBCC IND, 213–14

dbo.Customers table, 5–7

dbo.EmpYearValues table, 24–28

dbo.Orders table, 5–7

DDL (data definition language).

See data definition language (DDL)

DecToBase function, 491

definitions, 38–39

 cardinality, 56

 Cartesian products, 54

 characteristic function of a set, 55

 complexity, 283

 logical operators, 69

 ordered pairs and tuples, 53

 propositions and predicates, 66

 set complement, 62

 set difference, 63

 set partition, 63

 subsets, 61

 undefined terms, 39

 union and intersect, 62–63

defragmentation utilities, 258

Degree of Parallelism event, 653

DELETE statement, 103–04, 601

 OUTPUT clause, 630–32

 TOP option, 531–33

DELETE trigger, 627–28

DeMorgan, Augustus, 70

DeMorgans laws, 70
 denormalization, 122–24
 DENSE_RANK function, 352–54, 383
 derived tables, 318–19
 arguments, 320–21
 column aliases, 319–20
 multiple references, 321
 nesting, 320–21
 Designing Database Solutions
 (Sarka, Leonard, Loria, and Wiernik), 122
 determinism, 333–34
 RANK and DENSE_RANK functions, 353
 TOP option, 529–30
 Diaconis, Persi, 292
 Difference operator, 93–94
 direct subordinates, 717–18
 directed acyclic graph (DAG), 666
 transitive closure, 740–45
 directed graphs, 659–60
 Discard Results option, 344
 Disk Usage collection set, 148
 Disk Usage Summary report, 148
 DISTINCT clause, 15–16, 369,
 371, 742–43
 DISTINCT COUNT, 299
 DISTINCT phase, 5
 DISTINCT predicate, 414
 Distribute Streams Parallelism operator, 652–57
 Divide operator, 95–97
 dividend relation, 95–97
 divisor relation, 95–97
 dm_db_index_operational_stats, 256
 dm_db_index_usage_stats, 256
 dm_db_index_physical_stats, 257
 DMFs (Dynamic Management Functions).
 See specific DMFs
 DML (Data Manipulation Language).
 See Data Manipulation Language (DML)
 DMOs (Dynamic Management Objects), 172.
 See *also* specific DMOs
 DMVs (Dynamic Management Views).
 See specific DMVs
 domain integrity, 108–09
 domain-key normal form, 122
 domains, 84
 bad data, 47–48
 calculus, 102–03
 check constraint, 108–09
 modeling, 49
 DROP statistics command, 645
 DROP TABLE statement, 601
 dta.exe command-line utility, 187
 Dynamic Management Functions (DMFs).
 See specific DMFs
 Dynamic Management Objects (DMOs), 172.
 See *also* specific DMOs
 Dynamic Management Views (DMVs).
 See specific DMVs
 dynamic pivoting, 487–88

E

edges, 99–100
 Element Of operator, 90–91
 elements, separating, 429–35
 ellipsis, 45
 employee organization chart example, 661–63
 Employees sample table, 661–63
 cross joins, 390–95
 self joins, 402–04
 TOP n, 539–42
 EmpOrders sample table, 451–52
 empty sets, 54–55, 315
 encapsulated types, 86
 English-to-mathematics translation, 35–44
 entity
 defined, 87
 primitive, 124
 Entity Attribute Value (EAV), 460–61
 entity integrity, 105–06
 enumeration, sets, 45
 equality, 39
 Equals operator, 90–91
 equi-joins, 94, 402–03
 errors
 composite joins, 397
 duplicate key, 312
 ORDER BY table expressions, 18–19
 partitioned views updates, 640
 subqueries, 314–16
 Estimated Execution Plan, 644
 Estimated Subtree Cost, 178
 Evaluate Expressions phase, 5
 EXCEPT DISTINCT operation, 437–38
 EXCEPT operation, 31–32, 435–39
 excluded middle, law of, 68
 exclusive locks, 257–58
 Exclusive or, 70
 execution plan, 2
 analysis, 174–85
 cached, 169–71
 graphical, 174–85
 EXISTS predicate
 asterisk use, 306
 correlated subqueries, 305–14
 minimum missing values, 309–12
 semi joins, 414–16
 vs. IN predicate, 307
 expand-collapse technique, 404
 exponential complexity, 134–35
 expressions
 logical transformations, 556–59
 TOP option, 530–31
 Extend operator, 98
 T-SQL support, 103–04
 extents, 188–89
 external column aliasing, 319–20
 external fragmentation, 256–57
 external sorting, 287

F

factorial function, 281–82
 faithfulness, 49–51
 FALSE values, 9
 FAST_FORWARD cursor, 268–69
 Fermats Last Theorem, 110
 fifth normal form, 120
 fillfactor, 194, 257
 filtering
 bitmap filters, 426–28
 indexes, statistics and, 239–42
 filters, 8. *See also* specific filters
 first normal form, 113–15
 bitwise operations, 490
 first page request, 548–49
 FLOAT data type, 41
 fn_dblog function, 569
 fn_trace_gettable function, 149, 155
 FOR keyword, 464
 FOR XML PATH option, 214
 FOR XML query option, 487–88
 FORCE ORDER hint, 406
 bushy plans, 413
 foreign keys, 106–08
 nested loops, 423
 forests, 661
 format file, 565
 Format.Native property, 484
 Format.UserDefined property, 484
 forwarding pointers, 191
 fourth normal form, 119–20
 fragmentation, 256–58
 logical index, 233–34
 logical scan, 192–93
 Freedman, Craig, 429
 FROM clause
 derived tables, 318
 MERGE statement, 618
 TABLESAMPLE, 265
 FROM phase, 3, 7
 FULL keyword, 397–401
 FULL recovery model, 571–74
 FULLSCAN, 647–48
 functional dependencies
 multivalued dependency, 120
 normal forms, 112
 functions, 43. *See also* specific functions
 aggregate. *See* aggregate functions
 aggregate window, 454
 analytical ranking, 330–32
 inline definitions, CTEs, 325–26
 fuzzy logic, 75

G

Galindo-Legaria, Cesar, 273
 gaps, 363–86
 Gather Stream operator, 653–57

generalization
 database, 124–25
 relational database model, 124–26
 GetAncestor method, 717–18
 GetDescendant method, 711–12
 GetFirstRows, 591–94
 GetLevel method, 708
 GetNextPage, 549–51
 GetNextRows, 591–94
 GetPrevPage, 551–52
 GetReparentedValue, 712–14
 GetSequence procedure, 598
 GetTopProducts sample table, 535
 Global Aggregation operator, 655
 globally unique identifiers (GUIDs), 600–01
 random, 212
 temporary tables, 216
 graph theory, 99–100
 graphical execution plans, 174–85
 graphs, 659–60. *See also* specific graphs
 GROUP BY ALL, inner joins, 395–97
 GROUP BY clause
 derived tables, 319
 grouping sets, 506–07
 relational division, 299
 self joins, 404
 subclasses, 506
 GROUP BY phase, 5, 12–13
 grouping factor, 503–05
 GROUPING function, 524
 grouping sets, 12–13, 506–07
 algebra, 514–18
 CUBE subclass, 511–12
 GROUPING SETS subclass, 508–10
 GROUPING_ID function, 518–21
 materialize, 521–23
 PIVOT operator, 23
 ROLLUP subclass, 512–14
 sample data, 507
 sorting, 524
 GROUPING SETS subclass, 506
 GROUPING SETS subclass, 508–10
 addition, 517–18
 division, 515–17
 multiplication, 514–15
 GROUPING_ID function, 506, 518–21
 Groups sample table, 473–74
 median, 554
 GUIDs (globally unique identifiers).
 See globally unique identifiers (GUIDs)

H

Halpin, Terry, 88, 111–12
 hash algorithm, 428
 Hash Match operator, 426
 hash tables, 425–28
 HAVING clause, 80
 cumulative aggregations, 455–57

- HAVING phase, 5, 13–14
 - heaps, 189–91
 - INSERT SELECT statement scenarios, 575–78
 - Heisenberg Uncertainty Principle, 149
 - Heisenberg, Werner, 149
 - hierarchies, 99–100, 661
 - HIERARCHYID data type, 719
 - list sorting, 726–30
 - materialized path, 706–14
 - normalizing, 719–23
 - parent-child conversion, 724–26
 - hints, 185–86
 - joins, 407–14
 - histograms, 499–503
 - HOBT, 189
 - Hungarian notation, 89
- I**
- I/O subsystem
 - AND logic costs, 558–59
 - current and previous occurrence matching, 545
 - OR logic costs, 558–59
 - performance analysis, 145–48
 - query costs, 224, 229
 - reads, index seek cost, 193
 - STATISTICS IO option, 172–73
 - TOP n costs, 539, 542
 - wait analysis, 136–37, 143, 145
 - IBinarySerialize interface, 484
 - identity, 39
 - IDENTITY function, 342–44
 - IDENTITY property
 - inserting values, 110
 - SELECT INTO statement, 564
 - sequence mechanisms, 595–96
 - IF EXISTS, 626
 - IF keyword, 65–66
 - if.then statements, 70–72
 - ijk dialect, 40
 - IN predicate
 - vs. EXISTS predicate, 307
 - IN_ROW_DATA allocation units, 189
 - Include Actual Execution Plan, 654
 - INCLUDE clause, 548
 - filtered indexes, 240–41
 - included nonkey columns, 237
 - increasing subsequences, 291
 - Index Allocation Map (IAM) pages, 190–91
 - allocation order scans, 192
 - index ID, 256
 - index keys
 - updates, 219–23
 - index order scans, 204, 208
 - Index Scan operator, 205
 - allocation order scans, 208–12
 - index order scans, 219
 - index scans, 544
 - allocation order scans, 192, 208–19
 - APPLY operator, 546
 - index order scans, 204, 208
 - ordered clustered index, 202–04
 - ordered covering nonclustered index scan, 204–07
 - Storage Engine, 207–23, 256
 - strategy analysis, 244–56
 - unordered clustered index, 198–201, 245
 - unordered covering index scan, 245–46
 - unordered covering nonclustered index, 201–02
 - index seek, 193, 544
 - clustered index seek + ordered partial scan, 233–38, 250
 - covering nonclustered index seek + ordered partial scan, 251
 - nonclustered index seek + ordered partial scan + lookups, 223–28, 247–50
 - partition elimination, 649–50
 - subtree removal, 700–01
 - TOP n, 539
 - unordered nonclustered index scan + lookups, 228–33, 246–47
 - Index Seek operator, 223–26
 - indexed views, 242–44
 - indexes
 - access methods, 197–239.
 - See also index scans; index seek
 - clustered. See clustered indexes
 - costs, 238
 - covering, 201
 - covering index seek + ordered partial scan, 251
 - filtered, statistics and, 239–42
 - fragmentation, 192–93, 256–58
 - index seek + ordered partial scan + lookups, 247–50
 - intersection, 238–39
 - joins and, 421–23
 - level calculations, 193–95
 - nonclustered index seek + ordered partial scan + lookups method, 223–28
 - on a clustered table, 196–97
 - on a heap, 195–96
 - ordered covering scan, 204–07
 - pages and extents, 188–89
 - partitioning, 258–59
 - performance monitoring, 256
 - rebuilding, 257–58
 - rebuids, 648
 - reorganizing, 251
 - strategy analysis, 244–56
 - tuning, 169–70, 188–97. See index tuning
 - unordered covering scan, 201–02, 245–46
 - unordered index scan + lookups, 246–47
 - unordered nonclustered index scan + lookups, 228–33
 - INDEXPROPERTY function, 193
 - induced order, 59

Information Modeling and Relational Databases
 (Halpin and Morgan), 88, 111–12

Information Principle, 83

Init method, 482–83

inline column aliasing, 319–20

inline function definitions, CTEs, 325–26

inner joins, 395–97

- sliding total sample, 417–20
- strategy forcing, 428–29

input expressions

- TOP option, 530–31

INSERT EXEC statement, 590–94

INSERT loop, 360

INSERT SELECT FROM OPENROWSET

- statement, 566
 - minimal logging, 567–68

INSERT SELECT statement

- CASE expression, 310–12
- minimal logging, 567–68
- minimal logging summary, 590
- TABLOCK hint, heap, B-tree, TF-610, key range scenarios, 575–89

INSERT statement, 103–04

- auxiliary table of numbers, 360–62
- MERGE statement, 617–21
- OUTPUT clause, 629–30
- TOP option, 531–33

INSERT TOP, 532

INSERT trigger, 627–28

INSERT VALUES statement, 562

insertion sort, 288

Inside Microsoft SQL Server 2008, 105, 109, 122, 127, 318

INSTEAD OF triggers, 109

instructions, 43–44

integrity

- domain, 108–09
- entity, 105–06
- referential, 106–08

interchangeability, principle of, 88

internal fragmentation, 257

INTERSECT operation, 435–36, 439–40

- precedence, 440

Intersect operator, 93, 31–32

- T-SQL support, 103–04

intersect, set, 62–63

IntervalWaits function, 139–40

INTO clause, 441

intractable problems, 285

IP address, 704–30

irreflexive relation properties, 75–76

IsDescendantOf method, 715–16

IsInvariantToDuplicates property, 484–85

IsInvariantToNulls property, 484–85

IsInvariantToOrder property, 484–85

islands, 363–86

- variation, 384–86

IsNullIfEmpty property, 484–85

isolation levels, 211–12

iteration/recursion, 670

- ancestors, 681–84
- cycles, 691–94
- sorting, 688–91
- subgraph/subtree with path enumeration, 685–88
- subordinates, 671–81

iterative/procedural query tuning vs. set-based approaches, 268–76

J

Jensen, Clifford, 757

join hints, 185–86

JOIN keyword, 185, 428–29

Join operator, 94

- T-SQL support, 103–04

joins, 389

- algorithms, 421–29
- anti-semi, 415–16
- composite, 397
- cross, 390–95
- DELETE statement, 603–06
- dependency constraints, 121
- equi-, 94, 402–03
- hash, 428
- hints, 407–14
- inner, 395–97
- logical evaluation order, 408–11
- logical processing phase, 390
- many-to-many, 423
- merge, 423–25
- multiple, 405–06
- nested loops, 422–23
- nonsupported, 401
- old vs. new style, 389–403
- outer, 397–401
- self, 402–04
- semi, 98
- semi joins, 414–16
- theta, 94
- UPDATE statement, 606–10

K

Kass, Steve, 35, 267–68, 277

Kelly, Andrew J., 127

key lookups, 196–97

key-range, INSERT SELECT statement scenarios, 579–89

keys. *See also* foreign keys; primary keys

- Boyce-Codd normal form, 117–19
- duplicate, 312
- entity integrity, 105–06
- first normal form, 113–15
- natural vs. surrogate, 106

NULL values, 106
 second normal form, 115–16
 third normal form, 116–17
 uniqueness and applicability, 106
 Kogan, Eugene, 487
 k-tuples, 53

L

L_SUPPKEY, 648
 large object (LOB) data, 565–67
 large value type updates, 610–11
 LargeOrders sample table, 533, 630
 LastDay CTE, 523
 latch waits, 137
 law of excluded middle, 68
 LCK waits, 137
 leaf level, 191–95
 split pages, 192–93
 leaf nodes, 718, 738–39
 leaf_row_size, 193
 left input, 20–21
 APPLY operator, 21–22
 LEFT keyword, 397–401
 LEFT OUTER join, 543
 left semi joins, 414–16
 Leonard, Andy, 122
 LIKE condition, 702
 LIKE predicate, 232, 727
 linear complexity, 133–34
 LINEITEM sample table, 641–45
 LINEITEMPART sample table, 641–45
 lists, 287
 LOB_DATA allocation units, 189
 locks
 exclusive, 257–58
 index rebuilds, 257–58
 shared, 219, 257–58
 wait analysis, 137
 LOG function, 489–90
 logging
 analysis, 569–71
 minimally logged operations, 567–90
 testing insert scenarios, 571–89
 logic. *See also* fuzzy logic; predicate logic
 puzzles, 757–77
 three-valued, 9, 74
 two-valued, 623
 logical equivalence, 70
 logical index fragmentation, 233–34
 allocation order scans, 208–19
 logical operators, 68–70. *See also*
 specific operators
 logical query processing, 1–2
 OVER clause, 29–31
 phases, 2–5, 7–20. *See also* specific phases
 phases, joins and, 390

sample query, 5–7
 set operators, 31–32
 table operators, 20–28
 logical reads, 251–52
 logical scan fragmentation, 192–93, 256–57
 logical transformations, 556–59
 longest increasing subsequence length problem
 (LISLP), 291–95
 lookups
 cost, 196
 key, 196–97
 RID, 196
 Loria, Javier, 122

M

Machanic, Adam, 757
 magnetic tape storage, 287
 Management data warehouse, 187
 manual partitioning, 88
 materialize grouping sets, 521–23
 materialized path, 694–95
 data maintenance, 695–701
 querying, 701–06
 materialized path, HIERARCHYID data type,
 706–08
 data maintenance, 708–14
 querying, 715–19
 mathematics
 context, 41–43
 conventions, 39–40
 definitions, 38–39
 equality, identity, and sameness, 39
 functions, parameters, and values, 43
 graph theory, 99–100
 grouping sets algebra, 514–18
 instructions and algorithms, 43–44
 median, 494–97, 554–56
 mode, 497–99
 numbers, 41
 relational algebra and calculus, 90–104
 set S, 35–37
 well-definedness, 37–38
 Matrix sample table, 468–69
 MAX(order date), 302
 MAX(ordered), 302–05
 tiebreaker, 448–51
 MAX(requireddate), 302–05
 MaxByteSize property, 484
 MAXDOP hint, 257
 MAXRECURSION hint, 329–30, 680–81
 MDX (Multidimensional Expressions), 507
 median, 494–97
 TOP option, 554–56
 memory, wait analysis, 143
 merge algorithm, 423–25
 Merge Interval operator, 351

MERGE INTO clause, 618
 Merge method, 482–83
 MERGE predicate, 617–18
 MERGE statement, 103–04, 294, 617–21

- multiple WHEN clauses, 623–24
- OUTPUT clause, 634–36
- predicate addition, 621–23
- triggers, 627–28
- values, 626–27

 Messages sample table, 632
 metadata table queries, 648
 Microsoft SQL Server Customer Advisory Team, 158
 minimally logged operations, 567–90
 minimum missing values

- EXISTS predicate, 309–12
- outer joins, 400–01

 Minus operator, 93–94

- T-SQL support, 103–04

 missing values

- EXISTS predicate, 309–12
- outer joins, 400–01

 mode, 497–99
 modeling, 111–12

- domains, 49
- Object-Role Modeling (ORM), 111–12
- relational databases, 88

 modifications

- TOP option, 531–33

 modus ponens, 70
 MonthlyOrders sample table, 417–20
 Moran, Brian, 149
 Morgan, Tony, 88, 111–12
 Muller, Robert J., 112
 Multidimensional Expressions (MDX), 507
 multipage access, 351–52
 multiple joins, 405–06
 multiple references

- common table expressions, 324
- table expressions, 321

 multiset theory, 64–119
 multivalued dependencies, 120
 multivalued subqueries, 297–98
 mutator operators, 86
 MyGroupingSets sample table, 521–22
 MyOrders sample table, 557

N

naming conventions, 49–51. *See also* notation

- Hungarian notation, 89
- relational database model, 89

 National Institute of Standards and Technology (NIST), 659
 Natural Join operator, 94
 natural keys, 106
 natural numbers, 86
 nave set theory, 52
 nested loop algorithm, 422–23
 Nested Loops operator, 544

- parallel query plans, 654–57
- partition elimination, 649

 nested sets

- left and right value assignment, 731–36
- querying, 737–39

 nesting, derived tables, 320–21
 network waits, 145
 NEWID function, 553, 601
 NEWSEQUENTIALID function, 601
 next page request, 549–51
 next pointers, 204
 NIST (National Institute of Standards and Technology), 659
 No Action implementation, 107–08
 NOCOUNT option, 618–19
 nodes, 99–100
 NOEXPAND hint, 244
 NOLOCK hint

- allocation order, 215–19
- index order scan, 223

 non_leaf_row_size, 194
 nonblocking sequences, 598–600
 non-equi-join joins

- sliding total sample, 417–20

 non-FULL recovery mode, 574–75
 nonpolynomial complexity, 284–85
 nonscalar types, 86–87
 nonunique sort column method

- with tiebreaker, 337–38
- without tiebreaker, 338–40

 NORECOMPUTE option, 647
 normal forms

- additional, 122
- Boyce-Codd, 117–19
- domain-key, 122
- fifth, 120
- first, 113–15
- fourth, 119–20
- functional dependencies, 112
- higher, 119–22
- second, 115–16
- sixth, 122
- third, 116–17

 normalization, 111–22.

- See also* normal forms

 normalizing

- HIERARCHYID data type, 719–23

 Not Equals operator, 90–91
 NOT EXISTS predicate, 742

- semi joins, 415–16
- vs. NOT IN predicate, 307–09

 NOT IN predicate

- semi joins, 415–16
- vs. NOT EXISTS predicate, 307–09

 Not operator, 68–70

notation

- Big Oh, 283–84
- cardinality, 56
- Hungarian, 89
- ordered pairs and tuples, 53
- set theory, 45–46
- set-builder, 45–46
- sets, 45–46
- shorthand, 56
- NP switch, 428
- NTILE function, 354–59
- NULL values, 9, 48
 - @expression, @length, and @value arguments, 611
- aggregate product specialized solution, 489
- COALESCE function, 475
- EXCEPT DISTINCT operation, 437
- filtered indexes, 239
- GROUP BY phase, 13
- GROUPING SETS subclause, 509
- GROUPING_ID function, 518–21
- HIERARCHYID data type, 710
- in databases, 110–11
- IN predicate, 307
- INTERSECT operation, 439
- key constraints, 106
- multiple joins, 408
- NOT EXISTS and NOT in predicate, 307–09
- ORDER BY clause, 19–20
- outer joins, 399
- pivoting, 462–63
- ranking functions, 336
- row removal, UNPIVOT operator, 28
- set operations, 32
- specialization, 124–25
- UNIQUE constraint, 241–42
- UNPIVOT operator, 471
- NULLIF, 489
- num_leaf_pages, 194
- num_rows, 193
- numbers
 - cardinal, 59–60
 - mathematics and, 41
 - natural, 86
 - ordinal, 59–60
 - whichth, 60–61
- numerical order, 57
- Nums sample table, 131, 359–62
 - cross joins, 390–95
 - missing values, returning, 375–83
- NumSeq table, 363–64
- NVARCHAR data type, 188–89
- NVARCHAR(MAX) data type, 189
- NVARCHAR(MAX) type
 - updating, 610–11

O

- O(n log n)
 - LISLP problem, 292
 - sorting algorithms, 288–89
- object ID, 256
- Object-Role Modeling (ORM), 88, 111–12
- order
 - trichotomy, 58–59
- offline index rebuilding, 257–58
- OLEDDB wait, 137
- OLTP (online transaction processing).
 - See online transaction processing (OLTP)
- ON clause
 - bushy plans, 413
 - inner joins, 395–97
 - MERGE statement, 618
 - multiple joins, 409–11
- ON filter, 3
 - OUTER JOIN clause, 12
- ON filter phase, 8–10
- online index rebuilding, 257–58
- online transaction processing (OLTP)
 - MERGE statement, 616
 - wait analysis, 136
- open schema, 460–62
- OPENROWSET function, 565
- OpenSchema sample table, 461–62
- operators. *See also* specific operators
 - Boolean, 90–91
 - Codds, 91–97
 - cost percentages, 178
 - mutator, 86
 - relational algebra, 98–102
 - relations. *See* relations
 - relations and tuples, 90–91
 - selector, 86
 - set, 31–32
 - table, 20–28
 - ToolTip information, 179–85
 - type, 86
- optimization. *See also* query optimizer
 - indexing strategies analysis, 244–56
 - nested loops, 421–22
 - partitioned views and partitioned tables, 640
- optimized bitmap filters, 426–28
- optimizer. *See* query optimizer
- OPTION clause, 185
- Or operator, 68–70
 - IN predicate, 316–17
 - logical transformations, 556–59
- order, 57
 - alphabetical, 57–58
 - induced, 59
 - numerical, 57
 - sets, 57–61
 - total, 59

ORDER BY clause, 205
 cross joins, 393
 derived tables, 318–19
 ranking function, 331, 334
 TOP option, 16, 527, 534–35

ORDER BY list, 353

ORDER BY operation, 91, 436

ORDER BY phase, 5, 16–20
 OVER clause, 30–31

Order property, 208–12
 index order scans, 219

OrderDetails sample table, 465
 TOP n, 537–38

OrderDups sample table, 602–03

ordered pairs, 53–54

Ordered property, 204–05
 allocation order vs. index order scans, 207–08

Orders sample table, 131, 269–76, 306–07, 507
 data aggregation, 466–68
 hash joins, 425–26
 merge joins, 424–25
 multiple joins, 408–11
 TOP n, 537–38

OrdersArchive sample table, 631

ordinal numbers, 59–60

ORM (Object-Role Modeling), 88, 111–12

orthogonal design, 125–26

OUTER APPLY operator, 21, 536

OUTER JOIN clause, 12

outer joins, 11, 397–401
 filters, 12
 sliding total sample, 417–20

OUTER keyword, 398

OVER clause, 29–31
 aggregation, 445–70
 ranking functions, 331
 subclauses, 459

Ozer, Stuart, 158

P

Pack operator, 100–01

Page Free Space (PFS) pages, 191

page splits, 191
 allocation order scans, 208–12

page_density, 194

PAGEIOLATCH_SH wait, 142

pages, 188–89

paging
 multipage access, 351–52
 row numbers, 349–52
 TOP option, 547–52

parallel queries, 228–31

parallel query plans, 136
 wait analysis, 145

Parallelism operators, 228, 652–57
 parallelism, partitioning and, 652–57

parameters, 43

parent-child representation conversion, 724–26

parentheses, 322, 528
 chiasitic relationships, 410–11

Partial Aggregation operator, 655

PARTITION BY clause, 30
 OVER clause, 447–48
 ranking functions, 331–32
 Segment operator, 333

partition ID, 256

partitioned row numbers, 344

partitioned tables, 639–40
 partition elimination, 649–52
 query plans, 641–45
 statistics, 645–48
 vs. partitioned views, 640

partitioned views, 639–40

partitioning. *See also* partitioned tables
 manual, 88
 parallelism, 652–57
 partitioned views, 639–40
 ranking functions, 334
 subqueries, 340–41

partitions sets, 63–64

Parts sample table, 663–66

Pascal, Fabian, 119

path enumeration, 685–88

PATH mode, 487–88

path queries, 716–17

penguin dialect, 39–40

PERCENT keyword, 265

PERCENT option, 528, 555

performance
 row number calculation, 344–49
 selectivity and query cost, 253–55
 tracing effects on, 149–50
 tuning methodology, 131–34
 workload tracing, 150–55

performance counters, 143–44

Performance sample database, 127–31
 join algorithms, 421

performance testing
 data preparation, 259–65
 TABLESAMPLE, 265–68

PerfWorkloadTraceStart procedure, 151

physical query processing, 2

PIVOT operator, 22–24, 463–64, 466, 470
 phases, 23–24

pivoting, 460
 aggregate product, 475
 attributes, 460–64
 custom aggregation, 474–99
 data aggregation, 466–70
 dynamic, 487–88
 relational division, 465–66
 string concatenation, 475
 unpivoting, 470–73

- PivotTables
 - wait analysis, 140–42
 - plan guides, 124
 - plan handles, 168
 - plan hash, 168
 - point queries, 233–34
 - Poletti, Marcello, 757
 - polynomial complexity, 284–85
 - pool cache, 171–72
 - POWER function, 489–90
 - Practical Issues in Database Management (Pascal), 119
 - precedence
 - set operations, 440
 - predicate logic, 35, 65
 - alternatives, 73–75
 - DeMorgans laws, 70
 - generalizations, 73–75
 - implications, 70–72
 - law of excluded middle, 68
 - logical equivalence, 70
 - operators, 68–70
 - predicates. *See* predicates
 - programming languages, 65–66
 - propositions, 66–68
 - quantification, 72–73
 - relations, 75–80
 - predicates, 66–68
 - MERGE statement additions, 621–23
 - proposition creation from, 67–68
 - quantified, negating, 73
 - relations and, 87–88
 - truth value, 68
 - uncommon, subqueries, 316–18
 - preserved tables, 11
 - previous page request, 551–52
 - previous pointers, 204
 - primary keys, 105–06
 - nested loops, 423
 - primitive entities, 124
 - principle of interchangeability, 88
 - process-level analysis, 148–50
 - performance workload tracing, 150–55
 - query statistics, 167–69
 - trace data analysis, 155–67
 - product aggregate specialized solution, 488–90
 - Product operator, 92–93
 - T-SQL support, 103–04
 - Profiler, 186
 - programming languages
 - dialects, 40
 - fourth-generation, 277
 - predicate logic, 65–66
 - Project operator, 92
 - proof by contradiction, 68
 - proof by contrapositive, 71
 - propositional functions, 35
 - propositions, 66–68
 - creation from predicates, 67–68
 - relations and, 87–88
 - proto-tuple, 103
 - PvtCustOrders sample table, 470
- ## Q
- quadratic scaling, 280
 - quadratic sorting algorithms, 288
 - quantification
 - multiple, 73
 - predicate logic, 72–73
 - quantified statements, 72
 - multiple, 73
 - negating, 72–73
 - queries. *See also* query optimizer; query plan; query tuning
 - ad hoc, 136
 - aggregation, 156–67
 - compilation, 640
 - cost and performance statistics, 253–55
 - cost percentages, 178–79
 - execution plan. *See* execution plan
 - filters. *See* specific filters
 - HIERARCHYID data type, 715–19
 - materialized path, 701–06
 - nested sets, 737–39
 - ORDER BY clause, 31
 - parallel, 228–31
 - partitioned tables. *See* partitioned tables
 - path, 716–17
 - plan guides, 124
 - point, 233–34
 - processing. *See* logical query processing; physical query processing
 - range, 233–34
 - recursive, CTEs, 327–30
 - run time measurement, 173–74
 - S set sample application, 77–80
 - sample, 5–7
 - selectivity vs. query cost, 253–55
 - set operations. *See* set operations
 - set-based, 268–76
 - signature, 157–67
 - statistics, 167–69
 - subqueries. *See* subqueries
 - wait analysis. *See* wait analysis
 - query hash, 168
 - query hints, 185–86
 - query optimizer, 2
 - bitmap filter, 427
 - Database Engine Tuning Advisor, 187
 - hash table, 425–28
 - hints, 185–86
 - join hints, 407–14
 - join strategy forcing, 428–29
 - joins, 412–13

query optimizer

- query optimizer (*continued*)
 - logical transformations, 556–59
 - merge joins, 423–25
 - paging, 350–52
 - relational algebra operators, 101–02
 - scan order, 273–76
 - semi joins, 415
- query plans
 - parallel, 136, 145
 - parallelism, 652–57
 - partitioned tables, 641–45
- query processing. *See* logical query processing; physical query processing
- Query Statistics History report, 167
- query tuning, 127
 - course of action determination, 145
 - database/file level analysis, 145–48
 - index tuning, 187–259. *See also* index tuning
 - indexes and queries, 169–70
 - methodology, 131–34
 - process level analysis, 148–69
 - sample data, 127–31
 - set-based vs. iterative/procedural approaches, 268–76
 - tools, 171–87
 - wait analysis, 134–43
 - wait correlation with queues, 143–44
- queues, wait correlation, 143–44
- quick sort, 289
- QUOTENAME function, 488
- quotient relation, 95–97

R

- RAND function, 552–54
- random vs. sequential, 193
- Range Expression, 650–52
- ranges, 108–09
 - missing and existing, 363–86
 - queries, 233–34
- RANK function, 352–54
 - mode, 498
- ranking functions, 60–61
 - analytical. *See* analytical ranking functions
 - gaps solution, 372–73
 - NULL values, 336
- RDBMS (relational database management systems), 1, 83. *See also* relational database model
- read committed isolation level, 219
- Read method, 484
- read uncommitted isolation, 219
- READPAST hint, 633
- real numbers, 41, 51
- recursion. *See* iteration/recursion
- recursive common table expressions, 327–30
- Recursive Member, 328–30
- Redistribute Streams operator, 653–57
- references, multiple
 - common table expressions, 324
 - table expressions, 321
- referential integrity, 106–08
- reflexive relation properties, 75–76
- RegexReplace function, 160–61
- relational algebra, 90–104
 - operators, 98–102
 - T-SQL support, 103–04
 - relational calculus, 90–104
 - T-SQL support, 103–04
- relational database management systems (RDBMS), 1, 83
- relational database management systems (RDMBS), 83. *See also* relational database model
- relational database model, 83
 - algebra and calculus, 90–104
 - data integrity, 104–11
 - denormalization, 122–24
 - generalization and specialization, 124–26
 - naming conventions, 89
 - normalization, 111–22
 - relations, tuples and types, 84–89
 - summary, 89–90
 - views, 88–89
- relational division, 312–14
 - pivoting, 465–66
- relations
 - attributes, 85–87
 - divisor, dividend, and quotient, 95–97
 - operators, 90–91
 - properties of, 75–76
 - propositions and predicates, 87–88
 - relational database model, 84–89
 - universe, 76
 - virtual, 88–89
- relvar, 126
- Rename operator, 98
 - T-SQL support, 103–04
- Repartition Streams operator, 228
- REPEATABLE clause, 266
- REPLACE function, 433–34
- representation, faithful, 49–51
- Resource Governor, 171–72
- Restrict operator, 91–92
 - T-SQL support, 103–04
- restriction expression, 91–92
- Results to Text output mode, 435
- reverse logic, 72
 - relational division problems, 312–14
- RID lookup operation, 196
- right input, 20–21
 - APPLY operator, 21–22
- RIGHT keyword
 - outer joins, 397–401
- right semi joins, 414–16
- Rincon, Eladio, 127
- RNBenchmark table, 344–48
- Road System example, 666–70
- Roads sample table, 666–70
- ROLLUP subclass, 506
- ROLLUP subclass, 512–14

- root pages, 193
- roots node, 738–39
- ROUND function, 570
- row number calculation
 - benchmarks, 348–49
 - cursors, 341–42
 - IDENTITY-based, 342–44
 - nonpartitioned, 343
 - partitioned, 344
 - performance considerations, 344–49
 - subqueries, 335–41
- row numbers
 - benchmarks, 348–49
 - calculation. *See* row numbers calculation
 - paging, 349–52
- row overflow pages, 188
- ROW_NUMBER function, 330–52, 433–34
 - benchmarks, 348–49
 - cross joins, 392–93
 - current and previous occurrence matching, 546–47
 - median, 495
 - TOP n, 542
- ROW_OVERFLOW_DATA allocation units, 189
- ROWMODCTR, 647
- rows
 - anchor, 549–50, 592
 - copy generation, 26–27
 - current and previous occurrence matching, 543–47
 - duplicate, 15
 - duplicate data removal, 601–03
 - foreign key, 106–08
 - grouping, 23
 - index levels, 193–95
 - keys, 105–06
 - NULL values removal, 28
 - pivoting. *See* pivoting
 - random, TOP option, 552–54
 - removal, 28
 - set operations, 31–32
 - size limits, 188–89
 - TOP option, 16
 - value constructors, 607–08
- ROWS keyword, 265
- ROWS option, 266
- rows_per_leaf_page, 194
- rows_per_non_leaf_page, 194
- rowsets, 88
- RPCCompleted event class, 150
- running aggregation, 451–52
- Russell, Bertrand, 52
- Russell's Paradox, 52, 96, 110–11
- Rys, Michael, 487

S

- S set, 46
 - sample application, 77–80
- Sales sample table, 330–31
- Sales.MyShippers sample table, 314–16
- Sales.Orders sample table, 497
- SalesRN CTE, 350
- sameness, 39
- sample data. *See also* specific sample tables
 - grouping sets, 507
 - Performance database, 127–31
 - preparation, 259–65
 - TABLESAMPLE, 265–68
- Sarka, Dejan, 44, 122, 757
- Scalar operator, 650
- scalar subqueries, 297–98
- scalar types, 86–87
- scale, algorithms, 279–82
- SCOPE_IDENTITY function, 629
- second normal form, 115–16
- Segment operator, 207, 333
- SELECT clause, 331
- SELECT INTO statement, 216, 563–64
 - FULL recovery model, 571–74
 - minimal logging, 567–68
 - non-FULL recovery mode, 574–75
- SELECT list
 - aliases, 14–15
 - asterisk use, 306
 - bushy plans, 414
 - column order, 17
 - DATEADD function, 392
 - derived tables, 319
 - DISTINCT clause, 16, 369, 371
 - pivoting, 462
 - self joins, 404
 - unpivot operator, 471
- SELECT phase, 5, 14–16
 - ORDER BY clause, 29–30
- SELECT query, 278
 - partition elimination, 649–52
 - TOP option, 527–35
- SELECT statement, 103–04
 - assignments, 611–14
 - NOLOCK hint, 216
 - showplan, 643–45
- SELECT TOP, 528–29
- SELECT INTO statement, 343–44
- selection sort, 288
- selectivity, 224, 251
 - logical reads and, 251–52
 - performance statistics and query cost, 253–55
 - point determination, 248–49
 - vs. logical reads, 252
 - vs. query cost, 253–55
- selector operators, 86
- self joins, 402–04
- self-contained subqueries, 297–302
- semi joins, 98, 414–16
- semicolons, 322
- Semijoin operator, 98
- SEQUEL, 1
- sequence mechanisms
 - custom sequences, 596–600
 - IDENTITY property, 595–96

Sequence Project operator

- Sequence Project operator, 333
- sequential access, 287
- Serializable attribute, 484
- Server Activity collections, 148
- Server Activity History report, 139
- Server Actual History report, 148
- server instance
 - partitioned view, 639
 - wait analysis, 134–37
- Server Management Objects (SMO), 187
- Server Management Studio (SMSS)
 - cross joins, 396–97
 - Discard Results option, 329–44
- Sessions sample table, 260–65
- Set Default implementation, 107–08
- SET FORCEPLAN ON statement, 406
- Set Null implementation, 107–08
- set operations, 31–32, 435–36
 - EXCEPT, 437–39
 - INTERSECT, 439–40
 - INTO clause, 441
 - NULL values, 32
 - precedence, 440
 - UNION, 436–37
 - unsupported logical phrases, circumventing, 441–42
- set operators, 31–32, 56–63
- set S. *See* S set
- SET STATISTICS IO option, 351
- set theory, 35, 44. *See also* sets
 - domains of discourse, 46–49
 - faithfulness, 49–51
 - generalizations, 64–65
 - multiset theory, 64–65
 - nave, 52
 - notation, 45–46
 - ordered pairs, tuples, and Cartesian products, 53–54
 - Russell's Paradox, 52
 - set membership operator definition, 44–45
 - set U, 46
 - empty sets, 54–55
- set-based query tuning vs. iterative/procedural approaches, 268–76
- sets. *See also* set operations; set operators; set theory
 - cardinality, 56–57
 - characteristic function, 77–80
 - characteristic function definition, 55
 - complement, 62
 - difference, 63
 - empty, 54–55, 315
 - enumeration, 45
 - membership operator definition, 44–45
 - nested. *See* nested sets
 - notation, 45–46
 - operators. *See* set operators
 - order, 57–61
 - partitions, 63–64
 - set-builder notation, 45–46
 - subsets, 61–62
 - union and intersection, 62–63
 - universe. *See* U set
 - well-definedness, 46
- shared locks, 219, 257–58
- Shippers sample table, 269–76, 566
- SHOWPLAN_XML option, 186
- SIMPLE recovery model, 575
- Simsion, Graeme, 111–12
- Singh, Simon, 110
- single sequence values, 596–97
- SINGLE_BLOB type, 566
- SINGLE_CLOB type, 566
- SINGLE_NCLOB type, 566
- sixth normal form, 122
- sliding aggregation, 457–59
- sliding total, previous year, example, 417–20
- sliding window scenario, 642
- SMO (Server Management Objects), 187
- SMSS (Server Management Studio). *See* Server Management Studio (SMSS)
- Solid Quality Mentors, 127
- SOME predicate, 316–18
- Sort operator, 286, 509–10
- SORT_IN_TEMP_DB option, 257
- sorting
 - algorithms, 285–86
 - external, 287
 - grouping sets, 524
 - HIERARCHYID data type, 726–30
 - insertion and selection, 288
 - iteration/recursion, 688–91
 - O(n log N) algorithms, 288–89
 - quadratic algorithms, 288
 - quick sort, 289
 - running time comparisons, 285–86
 - swapping, 289
 - ultra sort, 289
- source code, 43–44
- sp_autostats, 647
- sp_configure, 653
- sp_create_plan_guide, 124
- sp_get_query_template procedure, 157
- sp_updatestats, 142–43
- specialization
 - database, 124–25
 - relational database model, 124–26
- specialized solutions, 487–99
 - bitwise operations, 490–94
 - product, 488–90
 - string concatenation, 487–88
- spread by element, 464
- spreading, PIVOT operator, 24
- SPStmtCompleted event class, 150
- SQL
 - pronunciation origin, 1
 - relations, 103
- SQL handle, 168

- SQL Server 2005
 - partitioning, 641–45
 - query plans, parallelism, 654–57
 - showplan, 649–52
- SQL Server 2008
 - CLR database code, 476–77
 - constraints, order of enforcement, 110
 - data collection and Management
 - data warehouse, 187
 - hash joins, 425–26
 - hints, 185
 - partitioning, 639–57
 - query plans, parallelism, 654–57
 - showplan, 649–52
 - Timestamp type, 109–10
 - tracing, 149–50
 - triggers support, 109
 - XML type, 109
- SQL Server Magazine, 757
- SQL_VARIANT data type, 188–89, 461
- UNPIVOT operator, 473
- SQLBatchCompleted event class, 150
- SQLStmtCompleted event class, 150
- SqlUserDefinedAggregate attribute, 484–85
- statistics
 - automatic maintenance, 142–43
 - cloning, 187
 - filtered indexes, 239–42
 - partitioned tables, 645–48
 - queries, 167–69
- statistics cloning, 187
- STATISTICS IO, 172–73
- STATISTICS IO option, 172–73
- STATISTICS TIME option, 173–74
- STATISTICS XML option
- Storage Engine, 207–23
- stored procedures, 109
- Stream Aggregate operator, 509–10, 655
- string concatenation, 449
 - aggregate specialized solution, 487–88
 - pivoting, 475
- StringBuilder class, 484
- StringConcat function, 487
- strings, searching, 289–90
- StructLayoutAttribute, 485
- STUFF function, 610–11
- subgraph/subtree, with path enumeration, 685–88
- sublinear complexity, 282
- subordinates
 - direct, 717–18
 - iteration/recursion, 671–81
- subqueries, 297–98
 - aggregate functions, 14
 - correlated. *See* correlated subqueries
 - gaps solution 1, 366–69
 - gaps solution 2, 369–71
 - misbehaving, 314–16
 - multivalued, 297–98
 - partitioning, 340–41
 - RANK and DENSE_RANK functions, 352–54
 - row number calculation, 335–41
 - scalar, 297–98
 - self joins, 404
 - self-contained, 297–302
 - table-valued, 297–98
 - uncommon predicates, 316–18
- subsequences, increasing, 291
- Subset Of operator, 90–91
- subsets, 61–62
- SUBSTRING function, 214, 431
 - mode, 499
- subtrees
 - cost, 178
 - moving, 697–99, 712–14
 - querying, 715–17
 - removal, 700–01
- subtypes, 124
- SUM aggregate, 453–57
- SUM function, 489–90
- SUM(qty) function, 468
- superexponential complexity, 134–35
- Superset Of operator, 90–91
- supertypes, 124
- surrogate keys, 106
- swapping algorithms, 289
- SWITCH command, 645–46
- SWITCH OUT command, 647
- symmetric relation properties, 75–76
- syntax, joins, 389–90
- sys.assemblies, 486
- sys.assembly_modules, 486
- sys.dm_db_missing_index_columns, 232
- sys.dm_db_missing_index_details, 232
- sys.dm_db_missing_index_group_stats, 232
- sys.dm_db_missing_index_groups, 232
- sys.dm_exec_cached_plans, 171
- sys.dm_exec_plan_attributes, 171
- sys.dm_exec_query_plan, 168, 171
- sys.dm_exec_query_stats, 167–69
- sys.dm_exec_sql_text, 168, 171
- sys.dm_io_virtual_file_stats, 145–48
- sys.dm_os_performance_counters, 143–44
- sys.dm_os_wait_stats, 134–37
- sys.syscacheobjects, 171
- sys.system_internals_allocation_units, 189–90
- SYSDATETIME function, 173–74
- SYSTEM keyword, 265
- SYSTEM method, 265–66
- system types, 87
- System.Object class, 484

T

- table expressions, 318
 - common (CTEs), 321–30. *See also* common table expressions (CTEs)
 - derived tables, 318–21
 - interchangeability, 89
 - left and right input, 20–21
 - ORDER BY clause, 18–20
 - TOP option, 18–20
- table hints, 185–86
- table operators, 20–28. *See also* specific operators
 - processing order, 11
- table scan, 198–201, 245, 557
- Table Scan operator, 643–45
- Table Spool operator, 263
- tables
 - aliases, 606
 - auxiliary table of numbers, 359–62
 - clustered, nonclustered indexes, 196–97
 - constraints, 109–10
 - derived. *See* derived tables
 - foreign key, 106–08
 - heaps, 189–91
 - joins. *See* table joins
 - key, 105–06
 - metadata query, 648
 - normalization. *See* normal forms; normalization
 - organization, 189
 - parent and child relations, 106–08
 - partitioned. *See* partitioned tables
 - partitioning, 258–59
 - pivoting. *See* pivoting
 - preserved, 11
- TABLESAMPLE, 265–68
- table-valued subqueries, 297–98
- TABLOCK hint, 211–12, 215, 566, 568
 - INSERT statement heap, B-tree, TF-610, key range scenarios, 575–89
 - minimal logging summary, 590
- Talmage, Ron, 757
- Tchernitsky, Nicolay, 757
- TClose operator, 99–100
- temp db database, 137, 148
- temporal data, 122
- TempSeq table, 364–65
- Terminate method, 482–83
- testing, insert scenarios, 571–89
- TF-610, INSERT SELECT statement scenarios, 579–89
- theta joins, 94
- third normal form, 116–17
- three-valued logic, 9, 74
- tiebreaker, 302–06
 - aggregation, 448–51
 - determinism, 334
 - median, 496
 - mode, 498–99
 - nonunique sort column method, 337–38
 - TOP option, 529–30
- tile number functions, 354–59
- Tiles CTE, 495
- TOP n for each group, 537–43
- Top operator, 207, 308
- TOP option, 16, 527
 - determinism, 529–30
 - input expressions, 530–31
 - matching current and previous occurrences, 543–47
 - median, 554–56
 - modifications, 531–33
 - on steroids, 534–35
 - paging, 547–52
 - random rows, 552–54
 - table expressions, 18–20
 - TOP n for each group, 537–43
- TOP PERCENT option, 554
- TOP phase, 5
- ToString method, 484, 711
- total order, 59
- TPC-H benchmark, 641–45
- tracing, 149–50, 186
 - data analysis, 155–67
 - performance workload, 150–55
- transaction log, wait analysis, 136, 148
- transactions, 105
- Transact-SQL. *See* T-SQL
- transitive closure, 99–100, 740
 - directed acyclic graphs, 740–45
 - undirected cyclic graphs, 745–54
- transitive relation properties, 75–76
- translation, English to mathematics, 35–44
- tree diagrams, 99–100
- trees, 660–61. *See also* subtrees
 - left and right values assignment, 731–36
- trend identification, 291
- trend marker practical application, 290–92
- trichotomy, 58–59
- triggers, 109
 - denormalization, 123–24
 - MERGE statement, 627–28
- TRUE values, 9
- true/false expressions. *See* Boolean expressions
- TRUNCATE TABLE statement, 600–01
- truth value, 68
- T-SQL, 1–2
 - Boolean expressions, 67
 - cycle detection, 691
 - HIERARCHYID data type, 707
 - joins logical processing order, 409
 - joins, nonsupported, 401
 - LISLP problem solution, 292–95
 - MAX attribute, 449
 - relational algebra and calculus support, 103–04
 - semicolon termination, 322
 - statement assignments, 611
 - UPDATE syntax, 596
 - vs. CLR, function implementation, 159

- tuples, 53–54
 - attributes, 84
 - calculus, 102–03
 - header, 103
 - heading, 84
 - operators, 90–91
 - properties, 84
 - relational database model, 84–89
 - sets. *See* relations
- two-valued logic, 623
- types
 - atomic, 86
 - constraints, 109
 - defined, 85
 - encapsulated, 86
 - operators, 86, 90–91
 - relational database model, 84–89
 - scalar vs. nonscalar, 86–87
 - subtypes and supertypes, 124
 - system, 87
 - user-defined, 87
 - vs. domains, 84

U

- U set, 46
 - empty sets, 54–55
- UDAs (user-defined aggregates), 476–82
- ultra sort, 289
- undefined terms, 39
- undirected cyclic graphs, 745–54
- undirected cyclic weighted graphs, 670
- undirected graphs, 659–60
- UNION ALL operation, 437
- UNION ALL operator, 31–32
- UNION DISTINCT operation, 437
- UNION operation, 31–32, 435–37
- Union operator, 92–93
 - T-SQL support, 103–04
- union, set, 62–63
- UNIQUE constraint
 - NULL values, 241–42
- unique sort column method, 335–37
- UNIQUEIDENTIFIER value, 600
- uniquifier, 191, 196
- UNKNOWN values, 9, 74
 - EXISTS predicate, 305–06
 - IN predicate, 307
- Unpack operator, 100–01
- UNPIVOT operator, 24–28, 471–73
 - phases, 25–28
- unpivoting, 470–73
- UPDATE statement, 103–04
 - assignments, 614–16
 - joins, 606–10
 - MERGE statement, 617–21
 - OUTPUT clause, 632–34
 - TOP option, 531–33

- UPDATE STATISTICS command, 645, 647
- UPDATE trigger, 627–28
- updating data, 606–16
- updating, partitioned views, 639–40
- UPDLOCK hint, 710
- USE PLAN hint, 122–24
- user-defined aggregates (UDAs), 476–82
- user-defined functions
 - auxiliary table of numbers, 362
 - inline, CTEs, 325–26
- user-defined types, 87
- USING clause
 - MERGE statement, 618

V

- vacuous truths, 71–72
- values, 43
- VALUES clause, 472–73, 561–84
- VARBINARY data type, 188–89
- VARBINARY(MAX) data type, 189
 - updating, 610–11
- VARCHAR data type, 188–89
- VARCHAR(MAX) data type, 189, 484, 688
 - updating, 610–11
- variables
 - functional dependencies, 112
 - types, 86
- vertices, 99–100
- views, 88–89
 - common table expressions, 325–26
 - compatibility, 171
 - indexed, 242–44
 - updatable, 109
- Visual Studio 2008, assembly creation and deployment, 482–87

W

- wait analysis
 - instance level, 134–37
 - top wait isolation, 137–38
 - wait information collection, 139–43
- weighted graphs, 666
- well-definedness, 37–38
 - sets, 46
- WHEN clause, 623–24
- WHEN MATCHED clause, 624
 - MERGE statement, 621
 - multiple, 623–24
- WHEN MATCHED THEN clause, 618–20
- WHEN NOT MATCHED BY SOURCE
 - clause, 624
- WHEN NOT MATCHED clause
 - MERGE statement, 623, 624
- WHEN NOT MATCHED THEN clause
 - MERGE statement, 618–20

WHERE clause

WHERE clause

inner joins, 395–97

outer joins, 399–401

WHERE filter, 399. *See also* WHERE phase

OUTER JOIN clause, 12

WHERE phase, 5, 11–12

whichth number, 60–61

Wiernik, Adolfo, 122

window-based calculations, 29, 445

WITH clause, table hints, 185

WITH CUBE option, 506, 511–12

WITH keyword, CTEs, 322

WITH ROLLUP option, 506, 514

WITH statement, multiple CTEs, 323

WITH TIES option, 16

TOP option, 530

Witt, Graham, 111–12

WRITE method, 484, 610–11

WRITELOG wait, 136

X

XML

showplans, 185–86

triggers and validations, 109

Y

YEAR(orderdate), 468

year-to-date aggregation, 459–60