



Index

Symbols

- [] (square brackets), 9
- { } (Braces), 10
- | (vertical bar), 10
- < > (angle brackets), 9
- 1NF (First normal form), 196-198, 202-203, 455
- 2NF (Second normal form), 198-200, 203-205, 455
- 3NF (Third normal form), 200-202, 456

A

- Accents
 - character sorts and, 56
 - dictionary sorts and, 54
- Access. *See* Microsoft Access, client model
- Access plans, 444, 456
- ADO (ActiveX Data Objects), 456
- Aggregate functions. *See* Set functions
- >ALL subquery, 126
- ALTER INDEX, 225-226
- ALTER TABLE, 173
- AND operators, 23-27
- Angle brackets (< >), 9
- ANSI SQL. *See* ANSI support
- ANSI style joins, 106-107

ANSI support

- bit data types, 160
- character data types, 137
- character string operations, 138-139
- decimal data types, 154
- float data types, 151-154
- GROUP BY, 71-72
- integer data types, 148
- isolation levels, 401
- large object data types, 163-167
- NOT NULL declarations, 260
- ORDER BY, 47-50
- outer joins, 108
- serial data types, 158
- set operators, 127
- subqueries, 112-113
- temporal data types, 143
- temporal ranges, 144
- APIs (Application Programming Interfaces)
 - comparing DBMSs' use of, 301
 - defined, 456
- Applets, Java, 456
- Application-DBMS communication (MS Query example), 307-321
- Attribute functions, ODBC, 307

- Auto-commit
 - vs. COMMIT, 383
 - data-change statements and, 327
 - Java connection methods, 335-337
- AVG function, 79, 83
- B**
- B+-tree, 214
- B*-tree, 214
- B-trees
 - characteristics of, 214-216
 - defined, 457
 - external searches and, 213
 - fragmentation and, 223-225
 - indexes and, 211
 - key deletion, 222-223
 - key insertion, 219-222
 - leaves, 215
 - levels, 217, 227
 - rebuilding, 225-227
 - searching, 217-219
 - summary, 227-228
- Back compression, 253, 456
- Backus-Naur Form (BNF) notation, 457
- Balanced trees. *See* B-trees
- Batch updates, 364
- BDB (Berkeley DB) tables, 396, 456
- “the Big Eight”. *See also* DBMS support
 - free page space attributes of, 192
 - installation parameters, 5-7
 - overview of, 3-5
 - portability, 8
 - selection criteria, 4
 - test results, 7-8
- Big-endian, 53, 456
- BINARY data type, 161-162
- BINARY LARGE OBJECT (BLOB) data type, 163-167
- Binary sorts
 - Character sort types and, 52
 - defined, 456
 - support for, 56-57
- BIT data type, 160-163
- BIT VARYING data type, 160-163
- Bit vectors, 247, 456
- Bitmap indexes
 - defined, 456
 - locks and, 417
 - overview of, 246-249
- Bitmaps, 456
- BLOB (BINARY LARGE OBJECT) data type, 163-167
- Block fetch, 355, 456
- Blocking units. *See* Pages
- Blocks. *See* Pages
- BNF (Backus-Naur Form) notation, 457
- Bookmarks, 238, 242, 457, 471
- BOOLEAN data type, 160-163
- Braces ({}), 10
- Buffer pool, 457
- Bulk INSERT, 372-374, 457
- Bytecode, 457
- C**
- Cache
 - cache access vs. disk access, 182
 - stored procedures and, 298
- Calculated fields, 207
- Call groups, ODBC, 305-307
- CALL statements, 296
- Capitalization. *See* Case sensitivity
- Cardinality, optimizers, 444
- Cartesian explosion, 88, 457
- Cartesian joins, 457
- Cartesian product, 87-88, 457
- CASE
 - portability, 35, 377
 - searches and, 34-35
- Case sensitivity
 - character sorts and, 56
 - defined, 54
 - dictionary sorts and, 457-458
 - searches and, 20-21
- CAST, 38, 82-83
- Catalog functions
 - JDBC, 349-350
 - ODBC, 307, 328-329
- CBOs (cost-based optimizers)
 - cost factors, 443-444
 - defined, 458
 - EXPLAIN statements, 450
 - hints, 450-451
 - list of, 442
 - multi-column statistics, 449
 - point count and, 13-14
 - vs. RBO (rule-based optimizer), 441-443
 - statistics analysis, 448-449

- statistics and histograms, 445-448
 - terminology, 444
- CGI (Common Gateway Interface), 289, 458
- CHAR (CHARACTER) data type
 - character data types and, 137-138
 - fixed size columns and, 133
 - vs. VARCHAR, 139-141
- Character data types
 - length specification, 141
 - overview, 137-139
 - string operations, 138-139
 - summary, 142-143
 - support for, 137
 - VARCHAR vs. CHAR, 139-141
 - variant character sets, 141-142
- CHARACTER LARGE OBJECT (CHAR) data type, 163-167
- Character sorts, 52-61
 - Binary sorts, 52
 - collations, 57-61
 - Dictionary sort with tie-breaking, 52, 54
 - Dictionary sorts, 52-54
 - summary, 61-62
 - support for, 55-57
- CHECK constraints, 262-265
 - compared with NOT NULL, 258, 262-263
 - defining on columns, 264
 - functions of, 263
 - not included in, 263-264
 - portability, 263
- Client applications, 298
- Client/server architecture, 427-440
 - clients, 435-439
 - middleware, 428-430
 - overview of, 427-428
 - separateness vs. parallelism, 433-434
 - servers, 430-435
 - system vs. per-client processes, 431
- Client validations, 281-282
- Clients
 - client model, 436-438
 - overview, 435-436
 - summary, 438-439
- CLOB (CHARACTER LARGE OBJECT) data type, 163-167
- close() method, 359
- Clustered indexes, 236-244
 - Cluster keys and, 237
 - defined, 458
 - weak-clustered vs. strong-clustered, 237-238
- Clustered keys
 - choice of, 240-241
 - clustered indexes and, 237
 - defined, 458
 - monotonically sequential, 246
 - sort-merge joins and, 97
- Clusters, 194-196
 - defined, 458
 - function of, 195
 - Oracle's implementation of, 195
- Codd, Dr., 10
- Code for points, 12-14
- COLLATE clause, 48, 72, 85
- Collations
 - character set dependence and, 59
 - defined, 458
 - overview, 57
 - support for, 58-59
- Column constraints. *See* Constraints
- Columns, 131-172
 - avoiding redundant, 72-73
 - bit data types and, 160-163
 - character data types, 142-143
 - character data types and, 137-139
 - decimals, 154-158
 - fixed size vs. variable length, 133-136
 - floats, 151-154
 - integers, 147-151
 - large object data types, 163-167
 - length specification, 141
 - NULLs and, 167-169
 - number data types, 147, 160
 - order within rows, 170-171
 - overview of, 131
 - serials, 158-159
 - shifts, impact on performance, 135-136
 - size, impact on performance, 131-137
 - temporal data types, 143-147
 - VARCHAR vs. CHAR, 139-141
 - variant character sets, 141-142
- Com units, 443
- COMMIT statements
 - data changes and, 382-385
 - Java connection methods, 336-337
 - optimistic locking and, 419-423
 - transaction-ends and, 388

- Common Gateway Interface (CGI), 289, 458
 - Composite index. *See* Compound indexes
 - Composite tables, 101-103
 - Compound indexes
 - defined, 458
 - functions of, 229-231
 - Compression, 252-253, 458
 - Concatenation, 140
 - CONCUR_READ_ONLY, concurrency type, 354, 356
 - Concurrency, 459
 - Concurrency type setting, JDBC, 354
 - Concurrent transactions, 403-405. *See also* isolation levels
 - defined, 388, 459
 - problems, 403-404
 - CONCUR_UPDATEABLE, concurrency type, 354
 - Connection pooling, 333-335, 459
 - Connections, JDBC
 - connection pooling, 333-335
 - DBMS info and, 338-346
 - overview of, 332-333
 - settings, 335-338
 - summary of, 346
 - Constant folding, 19-20
 - Constant propagation, 14-17, 459
 - Constraints, 257-284
 - CHECK, 262-265
 - client validation, 281-282
 - disabling, 279-281
 - FOREIGN KEY, 265-268
 - NOT NULL, 257-262
 - portability, 257, 263, 270, 274, 278
 - PRIMARY KEY, 268-272
 - redundant SELECT clauses and, 282-283
 - triggers, 273-277
 - UNIQUE, 272-273
 - Contention, 183
 - Control intervals. *See* Pages
 - Cost, 443-444, 459
 - Cost-based optimizers. *See* CBOs (cost-based optimizers)
 - COUNT function, 78-82
 - Covering indexes, 232-233, 459
 - CPU units, 443
 - CREATE INDEX statements, 212
 - CREATE PROCEDURE statements, 290
 - CREATE TABLE statements
 - column order and, 170-171
 - physical storage and, 173
 - CREATE VIEW column, 71
 - Cross joins, 457
 - CUBE columns, 72
 - CUBE keyword, 83-84
 - Cursors
 - defined, 459
 - ODBC, 306
 - sensitive vs. insensitive, 353
 - Cyclic reference problems, 269-270
- D**
- Data blocks. *See* Pages
 - Data change statements, 326-328, 367-386
 - auto-commit mode and, 327
 - COMMIT and ROLLBACK and, 382-385
 - defined, 388, 459
 - DELETE, 378
 - FETCH and, 381-382
 - fetch loops and, 327
 - INSERT, 369-374
 - JDBC and, 363-364
 - logs, 368-370
 - overview of, 367-368
 - passing multiple data-change statements, 326-327
 - portability, 270, 326
 - reads and writes and, 132
 - ugly updates and, 379-381
 - UPDATE, 374-378
 - Data Definition Language (DDL), 383, 459
 - Data Manipulation Language (DML), 383, 461
 - Data, retrieving with getXXX methods, 357-358
 - Data source, 333, 459
 - Data types
 - conversion of, 140
 - matching Java methods to, 360-361
 - subqueries and, 113
 - truncation of, 141
 - Data warehousing, 184-185
 - Database Administrators (DBAs), 2, 459
 - Database Management Systems. *See* DBMS (Database Management Systems)
 - DatabaseMetaData class, 349
 - DataSource class, 333
 - DATE data type, 143-147

- DB2. *See* IBM
- DBAs (Database Administrators), 2, 459
- dbc, 459
- DBMS (Database Management Systems)
- APIs used by, 301
 - “the Big Eight”, 3-5
 - DBMS-specific optimizations, 19
 - defined, 459
 - ensuring use of correct, 18-19
 - information tables, 338-346
 - SQL and, 1
 - TOP equivalents in, 125
- DBMS information tables, 338-346
- DBMS optimizer, 104-105
- DBMS support
- ALTER INDEX/REBUILD, 226
 - APIs, 301
 - auto-creation of indexes, 235
 - bit data types, 161
 - built-in collations, 60-61
 - CBOs (cost-based optimizers), 442
 - character data types, 137
 - character sorts, 55
 - character string operations, 138-139
 - decimal data types, 154
 - EXCEPT, 34
 - float data types, 151
 - get Calls, 340
 - GROUP BY, 71-72
 - integer data types, 148
 - is Calls, 346
 - isolation levels, 401
 - join plans, 89
 - large object data types, 163
 - lock granularity, 395
 - lock modes, 398
 - NOT NULL, 260
 - nulls Calls, 343
 - ORDER BY, 47-50
 - outer joins, 108
 - pages, 176
 - read groups, 181
 - ROWID equivalents, 188
 - serial data types, 158
 - set operators, 127
 - stores Calls, 344
 - subqueries, 112-113
 - supports Calls, 344
 - temporal data types, 143
 - temporal ranges, 144
- dbspace. *See* Tablespaces
- DDL (Data Definition Language), 383, 459
- Dead code elimination, 17-18
- Deadlocks, 411, 412-414, 459
- Decimal data types, 154-158
- Decision Support System (DSS), 461
- Declarative constraints, 273-274. *See also* Constraints
- DEFERRABLE constraint, 278
- Deferred updates, 379
- Deferring constraints, 278-281
- Degree, optimizer terminology, 444
- DELETE statements
- B-trees and, 222-223
 - data changes and, 363, 367, 388
 - format of, 378
- DELETE triggers, 275, 277
- DeMorgan's Theorem, 27
- Denormalization, 103, 459
- Density (indexes or tables), 105, 460
- Dependence
- defined, 460
 - rules, 198
 - transitively dependent, 200
- desc functions, 306, 460
- DESC values, IBM, 250
- Deterministic functions, 287-288, 460
- Diagnostic functions, ODBC, 306
- Dictionary sort with tie-breaking, 52, 54, 460
- Dictionary sorts, 52-54, 55-56, 460
- Dirty Read problems, 403, 460
- Disabling constraints, 279-281
- Disk access, vs. cache access, 182
- DISTINCT
- as alternative to GROUP BY, 75
 - subqueries, 120-121
- Distributive Law, 25
- DML (Data Manipulation Language), 383, 461
- Double INs, 123-125
- DOUBLE PRECISION, 151-154
- Driven table. *See* Inner tables
- Driver tables, 461
- DriverManager class, 332

- Drivers
 JDBC, 331, 347
 in joins, 80
 Driving table. *See* Outer tables
 DROP (indexes), 226
 Dropping constraints, 280
 DSS (Decision Support System), 461
 Dummy queries, 352
- E**
- Elevator seeking, 191–192, 461
 Enterprise Bean, 461
 env, 461
 Equals operator, 89
 Equijoins, 89, 461
 Escalation
 defined, 461
 locks, 396–397
 {Escape}, 349
 EXCEPT
 portability, 34, 127
 search tuning, 34
 Exclusive locks, 391–392, 461
 EXISTS subquery, 121–123
 Expanding UPDATES, 189–190, 461
 EXPLAIN statements, 450
 Expressions
 GROUP BY clause and, 84–85
 join expressions, 92
 ORDER BY clause and, 51
 restrictive expressions, 92
 Expressions column, 71
 Extents, 179–181
 allocation and, 179–180
 defined, 179, 461
 read groups and, 180–181
 storage hierarchy and, 175
 External searches, B-trees, 213
- F**
- FETCH, 381–382
 Fetch loops, 323–326
 data-change statements and, 327
 defined, 461
 improvements to, 324–326
 prototype for, 323–324
 FetchSize setting, JDBC queries, 354–355
 Files, 181–182
 defined, 181, 461
 functions of, 181–182
 storage hierarchy and, 175
 FILLFACTOR. *See* Free page space
 Filters. *See* Restrictive expressions
 First normal form (1NF), 196–198, 202–203, 455
 Fixed decimal numbers. *See* decimal data types
 Fixed size columns, vs. variable length columns,
 133–136
 Flattened subquery processing, 112, 462
 Flattening, 116–118, 462
 FLOAT columns, 81–82
 FLOAT data types, 151–154
 Floating Point Unit (FPU), 152, 462
 Fold functions, 21
 FOR BIT DATA attribute, 161–162
 FOR UPDATE transactions, 410–412
 FOREIGN KEY constraints, 265–268
 function of, 265
 matching to PRIMARY KEY constraints, 266
 portability, 270
 FPU (Floating Point Unit), 152, 462
 Fragmentation, 190–192
 B-trees and, 223–225
 defined, 462
 elevator seeking and, 191–192
 identifying, 191
 signs of, 190–191
 Fragments. *See* Partitions
 Free page space, 192–194
 attribute for, 192–193
 avoiding migration and, 193–194
 defined, 461, 468
 Freelist, Oracle, 462
 FROM keyword, 378
 Front compression, 252–253, 462
 Full text indexing, 251
 Functionally dependent
 defined, 462
 normalization and, 198
 Functions, deterministic vs. nondeterministic,
 287–288
 Functions keys, 250
- G**
- GAIN:x/8, 7, 14
 get Calls, DBMS information tables, 340–343
 getBestRowIdentifier, 349–350

getBoolean method, 360
 getByte method, 360
 getDate method, 360
 getShort method, 359-360
 getString method, 358-359
 getXXX methods, JDBC, 357-360
 Global Unique Identifier (GUID), 419, 462
 Granularity (of locks), 394-396, 462
 GROUP BY clauses, 69-86

- alternatives to, 75
- avoiding joins/grouping on same tables, 74
- avoiding redundant columns, 72-73
- COUNT function, 80-82
- expressions and, 84-85
- HAVING clauses vs. WHERE clauses, 74-75
- indexes and, 77-79
- MAX function, 84
- multiple aggregation levels, 83
- overview, 69-72
- reducing row counts before expanding, 73
- set functions and, 79-86
- sorting, 76-77, 79
- SUM function, 81-83
- summary, 75-76

 Grouping queries, 69. *See also* GROUP BY clauses
 GUID (Global Unique Identifier), 419, 462

H

Hash joins, 98-99, 462-463
 Hashes, 98, 250-251, 462
 HAVING clauses, 74-75
 Heap-organized tables. *See* Heap storage structures
 Heap storage structures

- defined, 463
- fragmentation, 190-192
- free page space, 192-194
- migration, expanding UPDATE, 189-190
- overview, 187
- ROWIDs, 187-189
- summary, 194

 "Hello-world" example, ODBC, 304-305
 Heuristic, optimizer terminology, 444
 Hints

- cautions relative to use of, 19
- CBOs and, 450-451

 Histograms, 445-448, 463
 Hoisting, 321-323, 463
 Host programs, 463

Hot spots

- avoiding, 271
- defined, 463
- sequences and, 418-419

I

I/O parallelism, 433
 I/O units, 443
 IBM

- ALTER INDEX/REBUILD, 226
- APIs, 301
- auto-creation of indexes, 235
- "the Big Eight", 4
- bit data types, 161
- built-in collations, 60-61
- CBOs (cost-based optimizers), 442
- character data types, 137
- character sorts, 55
- character string operations, 138-139
- decimal data types, 154
- EXCEPT, 34
- float data types, 151
- get Calls, 340
- GROUP BY, 71-72
- installation parameters, 5
- integer data types, 148
- is Calls, 346
- isolation levels, 401
- join plans, 89
- large object data types, 163
- lock granularity, 395
- lock modes, 398
- logging, 370
- NOT NULL, 260
- nulls Calls, 343
- ORDER BY, 47-50
- outer joins, 108
- pages, 176
- parallelism, 434
- read groups, 181
- ROWID equivalents, 188
- serial data types, 158
- set operators, 127
- stores Calls, 344
- subqueries, 112-113
- supports Calls, 344
- temporal data types, 143
- temporal ranges, 144

- IEEE floats, 153
- Impedance, 358-360, 463
- IN/NOT IN subquery, 119-120
 - double INs, 123-125
 - encouraging early break out, 114
 - EXISTS vs., 122-123
 - flattening vs. not flattening, 116-117
- IN operators, 28-29
- In-place updates, 463
- In-to-out subquery processing, 112, 119, 463
- Index locks, 416-418. *See also* Locks
- Index-organized tables. *See* Clusters
- Indexes, 211-255
 - as alternative to sorts, 64-65
 - bitmap indexes, 246-249
 - clustered indexes, 236-244
 - compound indexes, 229-231
 - covering indexes, 232-233
 - DBMS variations, 250-251
 - GROUP BY clauses and, 77-79
 - key values, 251-254
 - overview of, 211-213
 - qualities of good indexes, 91
 - redundant, 235
 - sorting, 55
 - structure of. *See* B-trees
 - types of, 229, 244-246
 - unique indexes, 233-236
- Informix
 - ALTER INDEX/REBUILD, 226
 - APIs, 301
 - auto-creation of indexes, 235
 - “the Big Eight”, 4
 - bit data types, 161
 - built-in collations, 60-61
 - CBOs (cost-based optimizers), 442
 - character data types, 137
 - character sorts, 55
 - character string operations, 138-139
 - decimal data types, 154
 - EXCEPT, 34
 - float data types, 151
 - get Calls, 340
 - GROUP BY, 71-72
 - installation parameters, 5
 - integer data types, 148
 - is Calls, 346
 - isolation levels, 401
- join plans, 89
- large object data types, 163
- lock granularity, 395
- lock modes, 398
- MATCHES compared with SIMILAR, 32
- NOT NULL, 260
- nulls Calls, 343
- ORDER BY, 47-50
- outer joins, 108
- pages, 176
- read groups, 181
- ROWID equivalents, 188
- serial data types, 158
- set operators, 127
- stored procedures, 287
- stores Calls, 344
- subqueries, 112-113
- supports Calls, 344
- temporal data types, 143
- temporal ranges, 144
- Ingres
 - ALTER INDEX/REBUILD, 226
 - APIs, 301
 - auto-creation of indexes, 235
 - “the Big Eight”, 4
 - bit data types, 161
 - built-in collations, 60-61
 - CBOs (cost-based optimizers), 442
 - character data types, 137
 - character sorts, 55
 - character string operations, 138-139
 - decimal data types, 154
 - EXCEPT, 34
 - float data types, 151
 - get Calls, 340
 - GROUP BY, 71-72
 - installation parameters, 5-6
 - integer data types, 148
 - is Calls, 346
 - isolation levels, 401
 - join plans, 89
 - large object data types, 163
 - lock granularity, 395
 - lock modes, 398
 - nonsupport of SUBSTRINGs, 30
 - NOT NULL, 260
 - nulls Calls, 343
 - ORDER BY, 47-50

- outer joins, 108
- pages, 176
- read groups, 181
- ROWID equivalents, 188
- serial data types, 158
- set operators, 127
- stores Calls, 344
- subqueries, 112-113
- supports Calls, 344
- temporal data types, 143
- temporal ranges, 144
- Inner tables
 - defined, 463
 - good indexes and, 91
 - selection criteria for, 93
 - smaller table as, 90
- INSERT statements, 369-374
 - B-trees and, 219-222
 - bulk INSERT, 372-374
 - data changes and, 363, 367, 388
 - examples of, 371
 - format of, 369
 - hot spots and, 418
 - performance improvements for, 371-372
 - summary of, 374
- INSERT triggers, 275, 277
- Installation parameters, 5-7
- INT (INTEGER) data type, 147-151
- Integers, 22, 147-151
- Intent locks, 397, 463
- InterBase
 - ALTER INDEX/REBUILD, 226
 - APIs, 301
 - auto-creation of indexes, 235
 - “the Big Eight”, 4
 - bit data types, 161
 - built-in collations, 60-61
 - CBOs (cost-based optimizers), 442
 - character data types, 137
 - character sorts, 55
 - character string operations, 138-139
 - decimal data types, 154
 - EXCEPT, 34
 - float data types, 151
 - get Calls, 340
 - GROUP BY, 71-72
 - installation parameters, 6
 - integer data types, 148
 - is Calls, 346
 - isolation levels, 401
 - join plans, 89
 - large object data types, 163
 - lock granularity, 395
 - lock modes, 398
 - nonsupport of SUBSTRINGs, 30
 - NOT NULL, 260
 - nulls Calls, 343
 - ORDER BY, 47-50
 - outer joins, 108
 - pages, 176
 - read groups, 181
 - ROWID equivalents, 188
 - serial data types, 158
 - set operators, 127
 - stores Calls, 344
 - subqueries, 112-113
 - supports Calls, 344
 - temporal data types, 143
 - temporal ranges, 144
 - use of versioning by, 392
- INTERSECT
 - portability, 34, 73, 127
 - search tuning, 34
- INTERVAL data type, 143-147
- INTO keyword, 371
- Invariant assignment, hoisting, 321-323
- is Calls, DBMS information tables, 346
- ISO SQL. *See* ANSI support
- Isolation levels, 399-416
 - abbreviations, 401
 - concurrency problems and, 403-405
 - deadlocks and, 412-414
 - defined, 464
 - descriptions, 400
 - Java connection methods and, 338
 - lock modes and, 401-402
 - overview, 399
 - READ COMMITTED transactions, 406
 - READ ONLY or FOR UPDATE transactions, 410-412
 - READ UNCOMMITTED transactions, 405-406
 - REPEATABLE READ transactions, 407-408
 - SERIALIZABLE transactions, 408-410
 - summary, 414-416
 - support, 401
- ITL slots, 389-390, 464

J

Java Database Connectivity. *See* JDBC (Java Database Connectivity)

Java Naming and Directory Interface (JNDI), 333, 464

Java Virtual Machines (JVMs), 365, 464

JDBC (Java Database Connectivity), 331-365

- connections. *See* connections, JDBC
- data changes, 363-364
- defined, 464
- driver functions, 331
- driver types, 347
- vs. ODBC, 362
- program example, 332
- queries. *See* queries, JDBC
- result sets. *See* result sets, JDBC
- support for, 4

JDBC-ODBC bridges, 347

JNDI (Java Naming and Directory Interface), 333, 464

JOIN, 73

Join expressions, 92, 464

Join indexes, 100-101, 464

Joins

- avoiding, 74, 99-100, 103
- composite tables and, 101-103
- defined, 464
- driver in, 80
- hash, 98-99
- join indexes, 100-101
- nested-loop, 89-95
- old-style vs. ANSI style, 106-107
- outer, 107-109
- overview of, 87-88
- plans, 88-89
- portability, 106
- sort-merge, 95-98
- subqueries vs., 115-116, 118
- three way and beyond, 103-105
- views of, 208-209

Journals, Oracle, 369

JVM (Java Virtual Machines), 365, 464

K

Key values, indexes, 251-254

L

Large object. *See* LOB (large object)

Latches, 388-389, 464

Law of Transitivity, 14-15

Least-Recently-Used (LRU), 465

Leaves (of an index), 215, 464

LIKE joins, 95

LIKE patterns, 29-31

Little-endian, 53, 464

Loads, 372-374

- defined, 464

LOB (large object)

- data types, 163-167

- defined, 464

- LOB pages, 179

- storage of, 165

Lock mode, 390, 465

Lock records, 390, 469

Lock type. *See* Lock mode

Locking, 388

Locks. *See also* Index locks; Isolation levels; Logs;

- Versioning

- concurrent transactions and, 388

- defined, 464

- escalation of, 396-397

- granularity of, 394-396

- Index locks, 416-418

- intent locks, 397

- isolation levels and, 401-402

- modes of, 390-394

- optimistic locking, 419-423

- overview, 387-388

- summary, 397-399

- types of, 388-390

Logs. *See also* Locks; Versioning

- IBM, 370

- turning off during data changes, 368-369

Lost Update problems, 403, 465

LRU (Least-Recently-Used), 465

M

Manual partitioning, 185

Marks on the wall, 389-390, 417, 465

Matches, 89

MATCHES (Informix), compared with SIMILAR, 32

Materialization, 120, 465

Materialized views, 208, 465

Max Bytes column, 47-48, 72

Max Columns column, 47, 72

MAX function, 78, 84, 85

Merge joins. *See* Sort-merge joins

Merge scans. *See* Sort-merge joins

- Metadata, 296
- Methods, JDBC, 466
- Microsoft
 - anti-NULL bias in databases, 169
 - ODBC support by, 302-303
- Microsoft Access, client model, 436-438
- Microsoft SQL Server 2000
 - ALTER INDEX/REBUILD, 226
 - APIs, 301
 - auto-creation of indexes, 235
 - “the Big Eight”, 4
 - bit data types, 161
 - built-in collations, 60-61
 - CBOs (cost-based optimizers), 442
 - character data types, 137
 - character sorts, 55
 - character string operations, 138-139
 - decimal data types, 154
 - EXCEPT, 34
 - float data types, 151
 - get Calls, 340
 - GROUP BY, 71-72
 - installation parameters, 6
 - integer data types, 148
 - is Calls, 346
 - isolation levels, 401
 - join plans, 89
 - large object data types, 163
 - lock granularity, 395
 - lock modes, 398
 - NOT NULL, 260
 - nulls Calls, 343
 - ORDER BY, 47-50
 - outer joins, 108
 - pages, 176
 - read groups, 181
 - ROWID equivalents, 188
 - serial data types, 158
 - set operators, 127
 - stored procedures, 287
 - stores Calls, 344
 - subqueries, 112-113
 - supports Calls, 344
 - temporal data types, 143
 - temporal ranges, 144
- Middleware, client/server architecture, 428-430
 - functions of, 428-429
 - separation of tiers and, 429-430
 - SQL statements and, 429
 - summary, 430
- Migrated row, 466
- Migration, 189-190, 193-194
- MIN function, 78, 85
- Monotonics, 418
- MS Query example, 307-321
- Multi-column statistics, 449
- Multi Version Concurrency Control (MVCC), 392-394, 466. *See also* Locks
- Multicolumn index. *See* Compound indexes
- Multiple aggregation levels, 83
- Multithread architecture, 432-433
- MVCC (Multi Version Concurrency Control), 392-394, 466. *See also* Locks
- MySQL
 - ALTER INDEX/REBUILD, 226
 - APIs, 301
 - auto-creation of indexes, 235
 - “the Big Eight”, 4
 - bit data types, 161
 - built-in collations, 60-61
 - CBOs (cost-based optimizers), 442
 - character data types, 137
 - character sorts, 55
 - character string operations, 138-139
 - decimal data types, 154
 - EXCEPT, 34
 - float data types, 151
 - get Calls, 340
 - GROUP BY, 71-72
 - installation parameters, 6
 - integer data types, 148
 - is Calls, 346
 - isolation levels, 401
 - join plans, 89
 - large object data types, 163
 - lock granularity, 395
 - lock modes, 398
 - nonsupport for subqueries, 115
 - nonsupport for UNION, 33
 - nonsupport for views, 206
 - NOT NULL, 260
 - nulls Calls, 343
 - ORDER BY, 47-50
 - outer joins, 108
 - pages, 176
 - read groups, 181

MySQL *continued*

- ROWID equivalents, 188
- serial data types, 158
- set operators, 127
- stores Calls, 344
- subqueries, 112-113
- supports Calls, 344
- temporal data types, 143
- temporal ranges, 144

N

- NaN (Not a Number), 157, 466
- National-language character sets, 141-142
- Native APIs, 302
- NCHAR (NATIONAL CHARACTER) data type, 137-138
- NCHAR VARYING (NATIONAL CHARACTER VARYING) data type, 137-138
- NCLOB (NATIONAL CHARACTER LARGE OBJECT) data type, 163-167
- Nested-loop joins, 89-95
 - defined, 466
 - inner/outer selection, 90-93
 - vs. nested loop subqueries, 113-114
 - restrictive expressions and, 92
 - summary, 95
- Nested-loop subqueries, 113-114
- Net drivers, JDBC, 347
- Niladic functions, 466
- NO WAIT/WAIT, 390
- Non-clustered indexes, 466
- Non-repeatable Read problem, 404, 467
- Non-volatile datum, 441-442
- Nondeterminism, 288, 466
- Normalization, 10
 - breaking rules of, 202-206
 - defined, 467
 - first normal form (1NF), 196-198, 202-203
 - overview of, 196
 - second normal form (2NF), 198-200, 203-205
 - summary of, 206
 - third normal form (3NF), 200-202
- Not a Number (NaN), 157, 466
- NOT EXISTS subquery, 121-123
- NOT IN subquery. *See* IN/NOT IN subquery
- NOT NULL, 257-262
 - compared with CHECK constraints, 258
 - impact on storage, 258-260

- NULL indicators and, 262
- NOT operators, 27-28
- NTFS (NT File System), 467
- NULLs. *See also* NOT NULL
 - columns and, 167-169
 - concepts behind, 257-258
 - indicators, 262
 - lack of standard for sorting, 51
 - nonconformed use of, 260
 - sort speed of, 49
 - storage requirements of, 167-168
- NULLs Sort column, 48
- Number data types
 - decimals, 154-158
 - floats, 151-154
 - integers, 147-151
 - overview, 147
 - serials, 158-159
 - summary, 160
- NUMERIC, decimal data types, 154-158

O

- ODBC (Open Database Connectivity), 301-330
 - APIs used by DBMS, 301
 - application-DBMS communication, 307-321
 - call groups, 305-307
 - catalog functions, 328-329
 - data-change statements, 326-328
 - defined, 303, 467
 - fetch loops, 323-326
 - future of, 302-303
 - “hello-world” example, 304-305
 - vs. JDBC, 362
 - resource hierarchy, 304
 - SQLPrepare, 321-323
 - support for, 4
- Old-style joins, 106-107
- OLTP (Online Transaction Processing), 86, 467
- Online Transaction Processing (OLTP), 86, 467
- Open Database Connectivity. *See* ODBC (Open Database Connectivity)
- Optimistic locking, 419-423, 467
- Optimistic stored procedures, 293
- Optimizers. *See also* CBOs (cost-based optimizers)
 - DBMS optimizer and, 104-105
 - rule-based vs. cost-based, 441-443
- OR operators, 24-25

Oracle

ALTER INDEX/REBUILD, 226
 AND optimization, 24
 APIs, 301
 auto-creation of indexes, 235
 “the Big Eight”, 4
 bit data types, 161
 built-in collations, 60-61
 CBOs (cost-based optimizers), 442
 character data types, 137
 character sorts, 55
 character string operations, 138-139
 clusters, 195
 decimal data types, 154
 EXCEPT, 34
 float data types, 151
 get Calls, 340
 GROUP BY, 71-72
 installation parameters, 6
 integer data types, 148
 is Calls, 346
 isolation levels, 401
 join plans, 89
 Journals, 369
 large object data types, 163
 lock granularity, 395
 lock modes, 398
 locking, 420
 NOT NULL, 260
 nulls Calls, 343
 numeric formats, 157
 optimizers, 443-444
 OR optimization, 25
 ORDER BY, 47-50
 outer joins, 108
 pages, 176
 read groups, 181
 ROWID equivalents, 188
 serial data types, 158
 set operators, 127
 stored procedures, 287
 stores Calls, 344
 subqueries, 112-113
 supports Calls, 344
 temporal data types, 143
 temporal ranges, 144
 versioning, 392-394
 ORDER BY clause. *See also* Character sorts

alternatives to, 62-67
 collations, 57-61
 defined length vs. actual length, 44-45
 duplicates and, 43
 general considerations, 39-40
 INTEGERS vs. CHARs, 45-46
 INTEGERS vs. SMALLINTs, 45
 portability, 50
 presorting, 44
 sets vs. multisets, 46
 speed tests, 40-46
 support, 47-50
 syntax, 40-46
 when not to apply, 50
 ORDER BY Expression column, 48
 Out-of-place updates, 379, 467
 Out-to-in subquery processing, 112, 468
 Outer joins, 107-109
 Outer tables, 93, 467

P

Packed decimal storage, Oracle, 157, 468
 Page reads, 468
 Pages, 174-179
 characteristics of, 174
 DBMS support, 176
 defined, 456, 468
 fragmentation, 190-192
 functions of, 177
 LOB pages, 179
 rows and, 176
 rows/page recommendation, 178
 storage hierarchy and, 175
 Parallelism
 partitions and, 183
 types of, 433
 Parameter values, 292-293
 Partitioning, 468
 Partitions, 182-185
 data warehousing and, 184-185
 defined, 182, 468
 manual, 185
 pros/cons of, 183-184
 separateness and, 433
 two-partition system, 183
 Pass Through Query (PTQ), 437, 469
 Pcode, 290
 PCTFREE. *See* Free page space

- Performance
 enhancing, 2-3
 reads/writes and, 132
- Persistent Stored Modules (PSMs), 286, 469. *See also*
 Stored procedures
- Pessimistic locking, 468
- Phantom problem
 concurrent transactions and, 404
 defined, 468
- PL/SQL (Procedure Language extensions to SQL),
 286, 468. *See also* Stored procedures
- Point counts, 12-14
- Portability
 bulk INSERT, 372-373
 CASE, 35, 377
 CAST, 38, 82
 CHECK constraints, 263
 COMMIT and ROLLBACK, 384
 data change statements, 326
 deferring constraints, 278
 EXCEPT, 34, 127
 floats, 152
 FOREIGN KEY constraints, 270
 INTERSECT, 34, 73, 127
 joins, 106
 INTO keyword, 371
 FROM keyword, 378
 ORDER BY, 50
 overview of, 8
 select lists, 35
 SIMILAR and MATCHES, 32
 stored procedures, 285
 subqueries, 74, 109, 115, 124
 SUBSTRINGs, 30
 triggers and constraints, 257, 274
 UNION, 33, 84, 109, 124
 updates, 270
 views, 206
- Positioned delete, 469
- Positioned update, 469
- Precompilers, 290-292, 469
- Prepared statements, 469
- Presorting, ORDER BY clause, 44, 66
- PRIMARY KEY constraints, 268-272
 cyclic reference problems and, 269-270
 function of, 268
 rule of thumb for defining, 271
- Procedure Language extensions to SQL (PL/SQL),
 286, 468. *See also* Stored procedures
- Procedures
 overview of, 285
 parameter passing, 292
- Processes
 per-client processes, 431-434
 summary, 435
 system processes, 431
 vs. threads, 430
- Programmers, vs. administrators, 2
- Programs, writing, 2
- Projects, 469
- Pseudocolumns, 189, 469
- PSM (Persistent Stored Modules), 286, 469. *See also*
 Stored procedures
- PTQ (Pass Through Query), 437
 defined, 469
- Q**
- Queries
 defined, 469
 materialized views and, 208
 views of, 206-207
- Queries, JDBC
 concurrency type setting, 354
 dummy queries, 352
 FetchSize, 354-355
 getBestRowIdentifier, 349-350
 ResultSetMetaData, 350-352
 Scroll Type settings, 353-354
 SELECT statement for, 348-349
 settings, 353
 summary of, 355-356
 syntax, 349
- Query parallelism, 434
- R**
- RAM records, 390, 469
- RBO (rule-based optimizer), 441-443, 469, 471
- READ COMMITTED transactions
 defined, 470
 isolation and, 400
 lock modes and read/write permissions, 402
 overview of, 406
- Read groups, 180-181
- Read-only transactions, 388, 410-412

- READ UNCOMMITTED transactions
 - defined, 470
 - isolation and, 400
 - lock modes and read/write permissions, 402
 - overview of, 405-406
 - Reads/writes
 - page reads, 468
 - performance and, 132
 - REAL, float data types, 151-154
 - REBUILD INDEX, 225-226
 - RECREATE (indexes), 226
 - Reference books, 453-454
 - Remote Procedure Call (RPC), 293, 471
 - REPEATABLE READ transactions
 - defined, 470
 - isolation and, 400
 - lock modes and read/write permissions, 402
 - overview of, 407-408
 - Response time, 432, 470
 - Restrict filters, 470
 - Restrictive expressions
 - defined, 470
 - processing after join clauses, 107
 - types of expressions, 92
 - Result sets
 - cursors and, 306
 - defined, 470
 - size of, 470
 - Result sets, JDBC, 356-361
 - closing, 360-361
 - getXXX methods, 357-360
 - how many rows in, 356-357
 - summary, 361-363
 - ResultSetMetaData, 350-352
 - Reverse keys, 250
 - RID (row identifier), 470. *See also* row identifiers
 - RLE (Run-Length Encoding), 470-471
 - ROLLBACK statements
 - data changes and, 382-385
 - transaction-ends and, 388
 - ROLLUP columns, 72
 - ROLLUP keyword, 83-84
 - Root node, indexes, 471
 - Row count, 51
 - Row groups. *See* Pages
 - row identifiers. *See also* ROWIDs (row identifiers)
 - RID, 470
 - ROWID, 187-189
 - tid, 188
 - Row locators, 238, 242, 457, 471
 - Row subqueries, 112-113
 - ROWIDs (row identifiers), 187-189
 - defined, 471
 - inner table order and, 96
 - matching rows and, 89
 - parts of, 187-188
 - pros/cons of, 188-189
 - serialized transactions and, 189
 - Rows
 - column order within, 170-171
 - normalization and, 206
 - size effect on performance, 132
 - Rowset size, 471
 - RPC (Remote Procedure Call), 293, 471
 - rs.close() method, 359
 - Rule-based optimizer (RBO), 441-443, 469, 471
 - Rules, for style, 36-37
 - Run-Length Encoding (RLE), 470-471
- S**
- SARG (Search ARGument), 21-23
 - SCN (System Change Number), 392-393, 471
 - Scroll Type settings, JDBC queries, 353-354
 - Search ARGument (SARG), 21-23
 - Search conditions
 - defined, 471
 - point counts and, 12-14
 - SQL syntax optimization, 11
 - structure of, 22
 - Searches, internal and external, 213. *See also* Simple searches
 - Second normal form (2NF), 198-200, 203-205, 455
 - Select list, 35, 472
 - SELECT statement
 - JDBC queries, 348-349
 - redundant clauses, 282-283
 - stored procedures and, 295-296
 - subqueries. *See* Subqueries
 - SELECT triggers, 275-277
 - Selectivity (of an index)
 - defined, 472
 - join plans and, 88
 - Sequences, 418-419, 472

- SERIALIZABLE transactions
 - defined, 472
 - isolation and, 400
 - lock modes and read/write permissions, 402
 - overview of, 408-410
- Serialized transactions, 189, 472
- Serials, 158-159
- Servers, client/server architecture, 430-435
 - processes and threads, 430
 - separateness vs. parallelism, 433-434
 - system vs. per-client processes, 431
- Set dependent
 - defined, 472
 - normalization and, 198
- Set functions, 79-86
 - AVG function, 79, 83
 - COUNT function, 80-82
 - defined, 472
 - list of, 70
 - MAX function, 84
 - MIN function, 78, 85
 - multiple aggregation levels, 83
 - SUM function, 81-83
- Set operators
 - defined, 472
 - EXCEPT, 34
 - INTERSECT, 34
 - replacing joins with, 73
 - subqueries and, 126-127
 - support for, 127
 - UNION, 32-33, 113, 124, 208-209
- SET TRANSACTION statement, 399
- Shared locks, 391-392, 472
- Shifts (of rows), 473
- Shrinking updates, 473
- SIMILAR patterns, 31-32
- Simple searches, 11-38
 - overview of, 11-12
 - style notes, 36-37
- Simple searches, general tuning
 - case-insensitive searches, 20-21
 - code for points and, 12-14
 - constant folding, 19-20
 - constant propagation, 14-17
 - dead code elimination, 17-18
 - sargability, 21-23
 - using right DBMS, 18-19
- Simple searches, specific tuning
 - AND operators, 23-24
 - AND plus OR, 25-27
 - CASE, 34-35
 - EXCEPT, 34
 - IN operators, 28-29
 - INTERSECT, 34
 - LIKE patterns, 29-31
 - NOT operators, 27-28
 - OR operators, 24-25
 - SIMILAR patterns, 31-32
 - UNION, 32-33
- Skews, 473
- SMALLINT data type, 147-151
- Sort keys, 62-63, 66-67, 473
- Sort LOBs column, 48
- Sort-merge joins, 95-98, 473
- Sorting, GROUP BY clauses, 76-77, 79
- Sorts. *See* ORDER BY clause
- Speed, sorts, 40, 51
- SPL (Stored Procedure Language), 286, 473. *See also*
 - Stored procedures
- Splitting (an index)
 - B-trees and, 220-221, 228
 - defined, 473
 - hot spots and, 418
- SQL calls
 - characteristics of, 303
 - ODBC groups, 305-306
 - SQL.LOG and, 318
- SQL-data-change statements. *See* Data change statements
- SQL query statement, 348-349
- SQL Server 2000. *See* Microsoft SQL Server 2000
- SQL Standard. *See* ANSI support
- SQL terminology and concepts, 8-9
- SQL:1999. *See* ANSI support
- SQL.LOG, 318
- SQLPrepare, 321-323, 321-323
- Square brackets ([]), 9
- standard deviation (STDEV), 70
- Statistics
 - analysis, 448-449
 - defined, 473
 - formulas, 447-448
 - histograms and, 446-447
 - multi-column, 449
 - recorded by CBOs, 445-446
- STDEV (standard deviation), 70

- stmt, 473
- Storage hierarchy
 - extents, 179-181
 - files, 181-182
 - pages, 174-179
 - partitions, 182-185
 - tables, 174-175, 186
 - tablespaces, 185-186
- Stored Procedure Language (SPL), 286, 473
- Stored procedures, 285-299
 - advantages of, 288-289
 - comparing dialect syntaxes, 287
 - deterministic functions, 287-288
 - improvements to, 294-295
 - JDBC data changes and, 363
 - metadata and, 296
 - optimistic example, 293
 - overview of, 285-287
 - parameter values and, 292-293
 - portability, 285
 - precompilation and, 290-292
 - SELECT procedures, 295-296
 - traffic reduction, 289-290
- stores Calls, DBMS information tables, 344
- Strong-clustered indexes, 238-240
 - characteristics of, 245
 - Clustered keys and, 237
 - defined, 473
 - secondary indexes to, 242-244
- Style notes, 36-37
- Subqueries
 - >ALL, 126
 - defined, 473
 - DISTINCT, 120-121
 - double INs, 123-125
 - EXISTS, 121-122
 - flattening, 116-118
 - IN, 119-120
 - IN vs. EXISTS, 122-123
 - joins. vs., 115-116, 118
 - nested-loop plans, 113-114
 - plans for processing, 112
 - portability, 74, 109, 115, 124
 - set operations and, 126-127
 - support for, 112-113
 - syntax choices, 128
 - TOP, 125
- Subselect. *See* Subqueries
- SUBSTRINGs, compared with LIKE, 30
- SUM function, 79, 81-83
- Summary aggregates. *See* Set functions
- supports Calls, DBMS information tables, 344-345
- Sybase
 - ALTER INDEX/REBUILD, 226
 - anti-NULL bias in, 169
 - APIs, 301
 - auto-creation of indexes, 235
 - “the Big Eight”, 4
 - bit data types, 161
 - built-in collations, 60-61
 - CBOs (cost-based optimizers), 442
 - character data types, 137
 - character sorts, 55
 - character string operations, 138-139
 - decimal data types, 154
 - EXCEPT, 34
 - float data types, 151
 - get Calls, 340
 - GROUP BY, 71-72
 - installation parameters, 6-7
 - integer data types, 148
 - is Calls, 346
 - isolation levels, 401
 - join plans, 89
 - large object data types, 163
 - lock granularity, 395
 - lock modes, 398
 - NOT NULL, 260
 - nulls Calls, 343
 - ORDER BY, 47-50
 - outer joins, 108
 - pages, 176
 - read groups, 181
 - ROWID equivalents, 188
 - serial data types, 158
 - set operators, 127
 - stored procedures, 287
 - stores Calls, 344
 - subqueries, 112-113
 - supports Calls, 344
 - temporal data types, 143
 - temporal ranges, 144
 - Total Density statistic, 105
- Synchronized (method), 473
- Syntax
 - defined, 474

- Syntax *continued*
 JDBC queries, 349
 optimization based on, 11
 subquery choices for, 128
- System Change Number (SCN), 392-393, 471
- System processes/threads, 431
- T**
- Table scans, 424, 474
- Table subqueries, 112-113
- Tables
 clusters, 194-196
 denormalizing, 103
 extents, 179-181
 files, 181-182
 first normal form (1NF), 196-198, 202-203
 fragmentation, 190-192
 free page space, 192-194
 heap storage structures, 187, 194
 inner/outer selection criteria, 93
 inner tables, 90-91
 migration, expanding UPDATE, 189-190
 normalization, 196, 202-206
 pages, 174-179
 partitions, 182-185
 queries of views, 206-208
 ROWIDs, 187-189
 second normal form (2NF), 198-200, 203-205
 storage hierarchy, 174-175, 186
 tablespaces, 185-186
 third normal form (3NF), 200-202
 views, 206-209
- Tablespaces, 185-186
 defined, 185, 474
 functions of, 186
 storage hierarchy and, 175
- Tear-down functions, ODBC, 305-306
- Temporal data types, 143-147
- Third normal form (3NF), 200-202, 456
- Threads
 per-client threads, 431-434
 vs. processes, 430
 summary of, 435
 system threads, 431
- Three way joins, 103-105
- Throughput, 1, 433, 474
- Tid, 188. *See also* row identifiers
- TIME data type, 143-147
- TIMESTAMP data type, 143-147
- TOP subquery, 125
- Total Density statistic, 105
- Traffic reduction, 289-290
- Transaction-end statements, 388
- Transactions, 388, 474
- Transforms, 474
- Transitively dependent, 200, 474
- Triggers, 273-277
 vs. declarative constraints, 273-274
 portability, 257, 274
 steps in execution of, 275
 types of, 275
- Truncation of data types, 141
- Tuning database performance, 1
- Tuning simple searches. *See* Simple searches, general
 tuning; Simple searches, specific tuning
- Two-tier drivers, JDBC, 347
- Type 1 (JDBC-ODBC bridges) drivers, JDBC, 347
- Type 2 drivers, JDBC, 347
- Type 3 (net) drivers, JDBC, 347
- Type 4 (two-tier) drivers, JDBC, 347
- TYPE_FORWARD_ONLY, scroll type, 353, 356
- U**
- UDT (user-defined data type), 168, 474
- Unicode
 big-endian vs. little endian, 53
 character data types and, 142
 defined, 474
- Uniform Resource Locator (URL), 333, 475
- UNION
 portability, 33, 84, 109, 124
 subqueries and, 113, 124
 tuning simple searches, 32-33
 views of, 208-209
- UNIQUE clauses, 272-273
- Unique indexes, 233-236, 474-475
- Unique key constraints, 272-273
- UNIQUE keyword, 212
- Uniquifier, 474
- Update locks, 391-392, 475
- UPDATE statements, 374-378
 B-trees and, 219
 batching, 376-377
 data changes and, 363, 367, 388
 dependent, 375-376
 examples, 374-375

expanding, 189-190
format of, 374
summary of, 377-378
UPDATE triggers, 275, 277
Updates. *See* Data change statements
URL (Uniform Resource Locator), 333, 475
User-defined data type (UDT), 168, 474

V

Validation, clients, 281-282
VARBINARY data type, 161-162
VARCHAR (CHARACTER VARYING) data type
vs. CHAR, 139-141
character data types, 137-138
variable columns and, 133-134
Variable length columns, 133-136
Variant character sets, 141-142
Vectors, 247, 475

Versioning, 392-394, 475. *See also* Locks;

Logs

Vertical bar (|), 10

Views, 206-209

portability, 206

queries of, 206-207

queries on materialized views, 208

unions or joins, 208-209

Volatile data, 441-442

W

WAIT/NO WAIT, 390

Weak-clustered indexes, 246, 458

Western European character sets, 141-142

WHERE clauses

avoiding redundant conditions, 18

vs. HAVING clauses, 74-75

Wildcards, 31

