

Index

• Symbols and Numerics •

* (asterisk) for all columns in table, 140, 314

1NF (first normal form), 132–133

2NF (second normal form), 133–134

3NF (third normal form), 55, 134–135

4NF (fourth normal form), 135

5NF (fifth normal form), 135

• A •

abnormal form, 136

ABS numeric value function, 180

ABSOLUTE option, FETCH statement, 371–372

Access 2003 Power Programming with VBA (Taylor and Andersen), 323

Access For Dummies (Ulrich-Fuller and Cook), 323

access to database, controlling. *See* privileges, user; security

ACID database, 307

ActiveX controls, 335

adding data user privilege, 282

adding new data, 146–150. *See also* INSERT statement

ad-hoc query, defined, 24–25

aggregate (set) functions, 71–72, 167–170, 219–220

alias (correlation name), defined, 232

ALL predicate, WHERE clause, 207–210

ALL qualifier, nested queries, 255–257

ALTER command, 55

ALTER statement, 65

ALTER TABLE statement, 104

altering table data, multitable database, 125

altering table structure, 92–94, 104, 125

AND logical connective, 70–71, 217, 417

Anderson, Virginia (author), *Access 2003 Power Programming with VBA*, 323

ANSI (American National Standards Institute) (Web site), 26

ANY predicate, WHERE clause, 207–210

ANY qualifier, nested queries, 255–257

API (application programming interface), 332

applet, defined, 337

application layer, ODBC, 331

application programming interface (API), 332

applications, working SQL with

about, 313–314

application defined, 298

asterisk for all columns in table, 314

CAST as access tool for tables in applications, 192–193

compound SQL statement

about, 373–374

assignment, 380

atomicity, 374–375

conditions, 376–380

cursors, 376

exception handling with, 401

flow of control statement, 380–386

privileges, 387–388

stored functions, 387

stored modules, 388–389

stored procedures, 386

variables, 375–376

embedded SQL method, 317–320

historical overview, 83

JDBC function in, 336–337

Microsoft Access, 97–105, 323–326

Microsoft Visual Studio, 322–323

module language method, 320–322, 392–393

ODBC function in, 329–335

applications (*continued*)
 problems with, 316
 RAD tool
 about, 84
 altering table structure, 92–94
 creating a table, 85–92
 data to track, deciding on, 84–85
 deleting a table, 96–97
 index, creating, 94–96
 object-oriented RAD tools, 322–323
 SQL compared to application, 23–25,
 83, 315
 XML data
 about, 339–340
 array mapping, 358
 conclusion, 359
 distinct UDT mapping, 356–357
 domain mapping, 355–356
 functions compatible with, 347–351
 mapping from/to SQL, 342–346, 355–359
 multiset mapping, 358–359
 predicates compatible with, 352–353
 ROW mapping, 357
 table conversion to SQL, 354–355
 XQuery language, 38–39
 approximate numerics, 31–32
 ARRAY data type, 41–42
 array mapping, XML data and SQL, 358
 ASENSITIVE keyword, DECLARE CURSOR
 statement, 368
 assertion, 48, 129
 assignment, compound SQL statement, 380
 asterisk (*) for all columns in table,
 140, 314
 ATOMIC keyword, compound SQL
 statement, 375
 atomic (scalar) property of value,
 158, 307, 374–375
 attribute (column). *See also* constraints
 adding data to, 148
 adding or deleting, 125
 asterisk (*) as wildcard value, 140, 314
 column-name join, 235–236
 creating view from modified, 144–145

 defined, 9
 defining for multitable database, 109–110
 domains
 mapping XML data and SQL, 355–356
 multitable database, 113, 121–122
 overview, 19
 user privileges, 278, 285–286
 identifying for multitable database,
 108–109
 keys, 114–116, 133–134. *See also*
 primary key
 reference for, 162–163
 relational function in table, 15
 in table design, 88–92
 transferring data, 149–150
 UDT, 43
 AUTHORIZATION clause, MODULE
 statement, 321
 authorization identifier, defined, 321
 AVG function, 72, 169

• B •

backing up data, 305–306, 418
 bad input data, data integrity problem, 126
 base table, 141
 basic join, 229–231
 BEGIN keyword, compound SQL statement,
 374
 beta testing of database design, 414
 BETWEEN predicate, WHERE clause, 202–203
 BIGINT data type, 29
 BINARY data type, 34–35
 BINARY LARGE OBJECT data type, 35
 binary string, 34–35
 BINARY VARYING data type, 35
 BLOB data type, 35
 blocks of rows, adding to table, 148–150
 BOOLEAN data type, 35, 68–69
 Boyce-Codd normal form (BCNF), 132, 135
 Buxton, Stephen (author), *Querying XML*,
 347



- CALL statement, 386
- CARDINALITY numeric value function, 179–180
- Cartesian product, defined, 199
- CASCADE keyword, REVOKE statement, 290–291
- cascading deletions, 123, 264
- CASE conditional expression, 183–191
- CASE . . . END CASE statement, 381–382
- CAST data-type expression, 191–193, 316
- catalog, creating, 63–64
- catalog-qualified name, 63
- CEIL or CEILING numeric value function, 181
- chain of dependency in user privileges, 291
- CHARACTER data type, 33
- CHARACTER LARGE OBJECT data type, 33
- character set, 113, 278, 285–286, 342–343
- character string, 32–34, 67–68, 178–179
- CHARACTER VARYING or VARCHAR data type, 33
- CHARACTER_LENGTH numeric value function, 178
- client extensions in Web-based databases, 334
- client role in Internet/intranet database environment, 50–51
- client-driven database design, pitfalls of, 411–412
- client/server database environment, 48–50, 332
- CLOB data type, 33
- cluster, 54, 63
- COALESCE conditional expression, 190–191, 244–246
- Codd, E. F. (IBM database pioneer), 13, 40, 132
- collation (collating sequence), 65, 113, 278, 285–286
- collection data types, 41–42, 69, 179–180, 358–359
- column (attribute). *See also* constraints
 adding data to, 148
 adding or deleting, 125
 asterisk (*) as wildcard value, 140, 314
 column-name join, 235–236
 creating view from modified, 144–145
 defined, 9
 defining for multitable database, 109–110
 domains
 mapping XML data and SQL, 355–356
 multitable database, 113, 121–122
 overview, 19
 user privileges, 278, 285–286
 identifying for multitable database, 108–109
 keys, 114–116, 133–134. *See also*
 primary key
 reference for, 162–163
 relational function in table, 15
 in table design, 88–92
 transferring data, 149–150
 UDT, 43
- column reference, 162–163
- column-name join, 235–236
- comma-delimited values, 146
- COMMIT statement, 73, 298–299, 304
- comparison operators, 70, 207–210, 252–257, 259–261
- comparison predicate, WHERE clause, 201
- complex query, 25
- composite key, 115, 133–134
- compound SQL statement
 about, 373–374
 assignment, 380
 atomicity, 374–375
 conditions, 376–380
 cursors, 376
 exception handling with, 401
 flow of control statement, 380–386
 privileges, 387–388
 stored functions, 387
 stored modules, 388–389
 stored procedures, 386
 variables, 375–376
- concatenation operator, 67–68, 164
- conceptual view (schema)
 creating, 62–63
 overview, 19
 owner of, 74
 XML, 339, 344, 346
- concurrent access threat to data, 295–296

- condition handler, 378–379
 - condition join, 234–235
 - conditional expressions. *See also* queries
 - CASE statement, 183–191
 - CASE...END CASE statement, 381–382
 - COALESCE, 190–191, 244–246
 - flow of control statements, 380–386
 - FOR...DO...END FOR statement, 385
 - IF...THEN...ELSE...ENDIF statement, 381
 - ITERATE statement, 385–386
 - LEAVE statement, 383
 - LOOP...ENDLOOP statement, 382–383
 - NULLIF, 189–190
 - overview, 166
 - REPEAT...UNTIL...END REPEAT statement, 384
 - WHILE...DO...END WHILE statement, 384
 - conditions in compound SQL statement, 376–380
 - consistency of database property, 307
 - console, launching ad-hoc query from, 24–25
 - constant values, 158–160
 - constraints
 - assertion, 48, 129
 - column (attribute), 19–20
 - data integrity, maintaining, 77–78, 104, 127–129
 - identity-constraint-option in VALID predicate, 353
 - NOT NULL, 59
 - overview, 47–48
 - in transactions, 307–311
 - violations of, 398–400
 - constructor function, structured UDT, 44
 - containment hierarchy, 54, 63–64. *See also* schema; table
 - CONTENT predicate (XML), 352
 - contention threat to data, 295
 - CONTINUE action, WHENEVER clause, 393
 - CONVERT string value function, 176
 - converting data or variables using CAST, 191–193
 - Cook, Ken (author), *Access For Dummies*, 323
 - core SQL:2008 statement, 26–27
 - correlation name (alias), defined, 232
 - COUNT function, 71, 168–169
 - CREATE DOMAIN statement, 285–286
 - CREATE INDEX statement, 103
 - CREATE MODULE statement, 388–389
 - CREATE statement, 55–57, 64–65
 - CREATE TABLE statement, 100–103
 - CREATE TRIGGER statement, 404–408
 - CREATE TYPE statement, 288
 - CROSS JOIN operator, 233
 - CURRENT_DATE datetime value function, 182
 - CURRENT_TIME datetime value function, 182
 - CURRENT_TIMESTAMP datetime value function, 182
 - CURRENT_USER variable, 162
 - cursor
 - about, 363–364
 - closing, 372
 - compound SQL statement, 376
 - declarations, 364–368
 - defined, 363
 - fetching data from single row, 370–372
 - opening, 369–370
- D •
- data
 - access control. *See* privileges, user
 - backing up, 305–306, 418
 - criteria for storage and retrieval, 8
 - defined, 9
 - dirty read, defined, 300
 - manipulating
 - about, 139
 - adding new data, 146–150. *See also* INSERT statement
 - Data Manipulation Language (DML), 53, 66–73
 - deleting obsolete data, 155–156. *See also* DELETE statement
 - retrieving data, 140–141. *See also* queries
 - transferring data, 149–150, 154–155

- updating existing data, 151–153. *See also* UPDATE statement
 - views, 141–145
- protecting. *See* integrity, data
- sorting with indexes, 117–118
- types. *See* data types
- values. *See* values
- Data Control Language (DCL). *See also*
 - privileges, user
 - about, 53, 73, 278
 - delegating security, 78–79
 - referential integrity constraints, 77–78
 - transactions, 73–74
 - user access levels, 278–280
- Data Definition Language (DDL)
 - about, 53, 54–55
 - catalog, 63–64
 - indexes, 103, 105
 - schemas, 62–63
 - statement types, 64–66
 - tables, 55–57, 100–103, 104
 - views, 57–62
- data dictionary, defined, 9
- Data Manipulation Language (DML)
 - about, 53, 66
 - logical connectives, 70–71
 - predicates, 69–70
 - set functions, 71–72
 - subqueries, 72–73
 - value expressions, 66–69
- data redundancy, data integrity problem, 126–127
- data source layer, ODBC, 331
- data types
 - about, 28
 - binary string, 34–35
 - Boolean, 35, 68–69
 - character string, 32–34, 67–68, 178–179
 - collection, 41–42, 69, 179–180, 358–359
 - datetime, 35–37, 68, 165–166, 370
 - defining new, permission for, 278
 - interval, 37, 68
 - mapping to/from XML, 344
 - numerics, 29–32, 67
 - REF, 42
 - ROW, 40, 69, 357
 - SQL compared to procedural languages, 316
 - summary, 45–46
 - user-defined, 42–45, 69
 - XML, 37–39, 340–342
- database
 - access control. *See* privileges, user; security
 - client/server environment, 48–50, 332
 - cluster, 54, 63
 - creating
 - about, 53, 54–55
 - catalog, 63–64
 - schemas, 62–63
 - statement types, 64–66
 - tables, 55–57, 85–92, 100–103, 104
 - views, 57–62
 - design pitfalls to avoid, 411–414
 - desirable properties of, 307
 - Internet/intranet environment, 50–51, 332–337
 - models, 12–13
 - multitable. *See* multitable database
 - objects. *See* schema; table; views
 - relational. *See* relational database
 - structure as value of, 11
 - verifying structure prior to retrieval, 415
 - Web-based environment, 50–51, 332–337
- database administrator (DBA), 278–279
- Database Development For Dummies* (Taylor), 55
- database management system (DBMS)
 - avoiding becoming a slave to favorite, 413
 - defined, 10
 - exceeding capacity of, 127
 - form-based entry variations, 146–147
 - interaction with SQL on client/server system, 48–50
 - Microsoft Access tool, 97–105, 323–326
 - object-relational, 20–21
 - platform instability problem, 294
 - programming function of, 24
 - proprietary tools for manipulation, 141
 - RAD tool
 - about, 84
 - altering table structure, 92–94

- database management system, RAD tool
 - (*continued*)
 - creating a table, 85–92
 - data to track, deciding on, 84–85
 - deleting a table, 96–97
 - index, creating, 94–96
 - object-oriented RAD tools, 322–323
 - relational database advantages for, 13–14
 - SQL function in, 1
- database object owner, 279
- data-manipulation code, flat file, 12
- Datasheet view, Microsoft Access, 85–87
- DATE data type, 36
- datetime data types, 35–37, 68, 165–166, 370
- datetime value functions, 182
- day-time interval, 37, 68, 166
- DBA (database administrator), 278–279
- DCL (Data Control Language). *See also* privileges, user
 - about, 53, 73, 278
 - delegating security, 78–79
 - referential integrity constraints, 77–78
 - transactions, 73–74
 - user access levels, 278–280
- DDL (Data Definition Language)
 - about, 53, 54–55
 - catalog, 63–64
 - indexes, 103, 105
 - schemas, 62–63
 - statement types, 64–66
 - tables, 55–57, 100–103, 104
 - views, 57–62
- DECIMAL data type, 30
- declarations
 - cursor, 364–368
 - host variables in embedded SQL, 319–320
 - module, 321
 - temporary-table clause, 321
- DECLARE CURSOR statement, 364–368
- default SQL transaction, 299–300
- DEFERRABLE designation for constraint, 308
- DEFERRED setting for constraint, 308, 310
- defining new data types, permission for, 278
- defining objects, multitable database, 108–113. *See also* DDL (Data Definition Language)
- delegating user privilege, 289–290
- DELETE statement
 - about, 75
 - with cursor, 372
 - function of, 155–156
 - with nested queries, 263–264
 - trigger, relationship to, 404, 406
 - user access, 278
- deleting obsolete rows user privilege, 284
- deletion anomaly, 130
- departmental database, defined, 9
- Design view, Microsoft Access, 87–92
- detail area, diagnostics, 396–398
- diagnostics areas, error handling, 394–400
- DIAGNOSTICS SIZE keyword, SET TRANSACTION statement, 299
- dirty read for data, defined, 300
- DISTINCT predicate, WHERE clause, 211
- distinct types, UDT, 43, 356–357
- DK/NF (domain-key normal form), 132, 135–136
- DLL (dynamic link library), defined, 331
- DML (Data Manipulation Language)
 - about, 53, 66
 - logical connectives, 70–71
 - predicates, 69–70
 - set functions, 71–72
 - subqueries, 72–73
 - value expressions, 66–69
- DOCUMENT predicate (XML), 352
- documenting database development, 414
- domain-key normal form (DK/NF), 132, 135–136
- domains
 - mapping XML data and SQL, 355–356
 - multitable database, 113, 121–122
 - overview, 19
 - user privileges, 278, 285–286
- DOUBLE PRECISION data type, 31
- driver DLL layer, ODBC, 331
- driver manager layer, ODBC, 331
- DROP DOMAIN statement, 286
- DROP INDEX statement, 105

DROP statement, 55, 65–66
 DROP TABLE statement, 104
 durability of database property, 307
 dynamic link library (DLL), defined, 331

• E •

editing (manipulating) data
 about, 139
 adding new data, 146–150. *See also*
 INSERT statement
 deleting obsolete data, 155–156. *See also*
 DELETE statement
 DML, 53, 66–73
 retrieving data, 140–141. *See also* queries
 transferring data, 149–150, 154–155
 updating existing data, 151–153. *See also*
 UPDATE statement
 views, 141–145
 ELSE keyword in conditional expression,
 183
 embedded SQL, 162, 316, 317–320
 END keyword, compound SQL statement,
 374
 ENDIF keyword in conditional expression,
 184
 enterprise database, 9, 13
 entity integrity, multitable database,
 120–121
 equi-join, 231–233, 234
 equipment failure threat to data, 294–295
 equiwidth partitioning, 181
 error handling
 about, 391
 CASE expression as tool for, 186
 diagnostics areas, 394–400
 exception conditions, 401–402
 preparing for, 418
 SQLSTATE host variable, 391–393
 WHENEVER clause, 393–394
 exact numerics, 29–30
 EXCEPT operator, 228–229
 exception avoidance, defined, 186
 exception conditions, error handling,
 401–402

EXCEPTION statement, 401–402
 EXEC SQL directive, 317, 319
 EXECUTE keyword, permission to use,
 278, 388
 existence test, nested queries, 257–259
 EXISTS predicate, WHERE clause, 210–211,
 257–258
 EXP (exponent) numeric value function,
 180
 extended SQL:2008 statement, 26–27
 extensions to SQL standard, 105, 329
 EXTRACT numeric value function, 178, 341

• F •

Fagin, Ronald (normalization researcher),
 132
 FETCH statement, cursor, 370–372
 field, defined, 158
 field properties (Access), 88–92
 fifth normal form (5NF), 135
 first normal form (1NF), 132–133
 FIRST option, FETCH statement, 371
 flat files, 11–12
 FLOAT data type, 32
 FLOOR numeric value function, 181
 flow of control statements, 380–386
 FOR READ ONLY clause, DECLARE
 CURSOR statement, 367
 FOR . . . DO . . . END FOR statement, 385
 foreign data file, copying data from,
 148–149
 foreign key, 115–116, 122, 214
 form-based data entry, 146–147
 fourth normal form (4NF), 135
 FROM clause, 198, 199
 FULL keyword, MATCH predicate, 215–216
 FULL OUTER JOIN operator, 239
 function call, 265, 330, 387
 functional dependence, 133–134, 268
 functions
 datetime value, 182
 defined, 157
 recursive, 265–268
 set, 71–72, 167–170, 219–220

functions (*continued*)
 stored, 387
 string value, 170–176
 structured UDT, 44
 value, 170–182
 XML compatibility, 347–351

• G •

GET DIAGNOSTICS statement,
 397, 399–400
 GO TO address action, WHENEVER
 clause, 393
 GRANT statement
 about, 75–76, 280–281
 across levels, privileges, 287–289
 character set access, 285–286
 collation access, 285–286
 delegating privileges, 78–79, 289–290
 deleting obsolete rows, 284
 domain access, 285–286
 inserting data, 282
 looking at data, 282–283
 modifying table data, 283
 referencing related tables, 284–285
 with REVOKE, 292
 translation access, 285–286
 trigger permission, 287
 Greenwich Mean Time/GMT (universal
 mean time), defined, 36
 GROUP BY clause, 198, 218–221, 416–417

• H •

hardware failure threat to data, 294–295
 HAVING clause, 198, 220–221, 261–262
 header area, diagnostics, 395–396
 helper application, 335
 hierarchical database model, 12, 287
 holdability, declaring cursor, 365
 host variables (parameters), 161, 319–320,
 391–393

• I •

IBM and history of SQL, 25
 IDE (integrated development
 environment), 83
 identifiers, mapping to/from XML, 343
 identity-constraint-option in
 VALID predicate, 353
 IF keyword in conditional expression, 183
 IF...THEN...ELSE...ENDIF statement,
 381
 IMMEDIATE setting for constraints, 308
 impedance mismatch, defined, 42
 implementation, defined, 26
 implicit transaction-starting statement, 303
 IN predicate
 nested queries introduced with,
 250–251, 259
 WHERE clause, 203
 indexed sequential access method (ISAM),
 331
 indexes, 94–96, 103, 105, 116–119
 information schema, 63
 INNER JOIN operator, 236, 242
 INSENSITIVE keyword, DECLARE CURSOR
 statement, 368
 INSERT statement
 about, 75
 basics of adding new data, 147–150
 with nested queries, 263–264
 row value expression, 194
 trigger, relationship to, 404, 406
 user access, 278
 inserting data user privilege, 282
 insertion anomaly, 131
 INTEGER data type, 29
 integrated database, defined, 9
 integrated development environment
 (IDE), 83
 integrity, data
 about, 119–120
 altering table data, 125
 constraints, 77–78, 104, 127–129
 DCL tool, 53, 73–79, 278–280
 defined, 120

domain integrity, 121–122
 entity integrity, 120–121
 form-based data entry issues, 146–147
 good design as prerequisite for, 415
 potential problem areas, 125–127
 protecting
 about, 293
 constraints within transactions, 307–311
 SQL transactions, using. *See* transaction processing
 threats overview, 293–297
 referential integrity, 77–78, 104, 122–125, 212–216
 interactive SQL mode of operation, 313
 International Standards Organization (ISO), 7, 21
 Internet/intranet database environment, 50–51, 332–337
 INTERSECT DISTINCT operator, 227
 INTERSECT operator, 226–228
 interval data type, 37, 68, 166
 ISAM (indexed sequential access method), 331
 ISO (International Standards Organization), 7, 21
 isolation levels, SQL transactions, 300–302, 304
 isolation of database property, 307
 ITERATE statement, 385–386

• J •

JDBC (Java Database Connectivity), 336–337
 join (JOIN) operators
 about, 229
 basic join, 229–231
 checking for valid queries, 416
 ON clause with, 246
 column-name join, 235–236
 condition join, 234–235
 CROSS JOIN, 233
 equi-join, 231–233, 234
 INNER JOIN, 236

natural join, 234
 OUTER JOIN, 236–240
 UNION JOIN, 240–246
 WHERE clause with, 246

• K •

key
 composite, 115, 133–134
 creating, 114–116
 defined, 114
 DK/NF (domain-key normal form), 132, 135–136
 foreign, 115–116, 122, 214
 normalization role, 133–134
 primary
 avoiding update anomalies, 125
 database planning, 55
 entity integrity, 120–121
 importance of indexing, 95
 in multitable database, 114–115
 parent-child table relationships, 122
 referential integrity rules, 214

• L •

LANGUAGE clause, MODULE statement, 321
 LAST option, FETCH statement, 371
 last-in-first-out (LIFO) stack, 394
 leaf type, structured UDT, 287
 LEAVE statement, 383
 LEFT OUTER JOIN operator, 237–239
 legacy systems, advantages of relational database for, 13
 LIFO (last-in-first-out) stack, 394
 LIKE predicate, WHERE clause, 204–206
 literal values, 158–160
 LN (logarithm) numeric value function, 180
 locking database objects, 305
 logical compared to physical schema, 62–63
 logical connectives, 70–71, 216–218, 417

login for DBA, 279
 looking at data user privilege, 282–283
 LOOP . . . ENDLOOP statement, 382–383
 LOWER string value function, 175

• M •

malice, data integrity problem, 126
 manipulating data
 about, 139
 adding new data, 146–150. *See also*
 INSERT statement
 Data Manipulation Language (DML),
 53, 66–73
 deleting obsolete data, 155–156. *See also*
 DELETE statement
 retrieving data, 140–141. *See also* queries
 transferring data, 149–150, 154–155
 updating existing data, 151–153. *See also*
 UPDATE statement
 views, 141–145
 mapping XML from/to SQL, 342–346,
 355–359
 MATCH predicate, WHERE clause, 212–216
 MAX function, 71, 169–170
 mechanical failure, data integrity problem,
 126
 Melton, Jim (author), *Querying XML*, 347
 MERGE statement, 154
 metadata, 9, 19, 63
 method in procedural language, defined,
 322
 Microsoft Access, 97–105, 323–326. *See also*
 RAD (rapid application development)
 tool
 Microsoft Visual Studio, 322–323
 MIN function, 72, 170
 MOD (modulus) numeric value function, 180
 modification anomalies, 130
 modified attribute, creating view from,
 144–145
 modifying table data user privilege, 283
 module language method, 320–322,
 392–393
 MODULE statement, 321–322

modules, stored, 388–389
 MULTISET data type, 42
 multiset mapping, XML data and SQL,
 358–359
 multitable database
 about, 107
 character set, 113
 collation, 113
 data integrity, maintaining
 about, 119–120
 altering table data, 125
 constraints, 127–129
 domain integrity, 121–122
 entity integrity, 120–121
 potential problem areas, 125–127
 referential integrity, 122–125
 design
 defining objects, 108–113
 identifying tables and columns, 108–109
 overview, 107–108
 domains, 113
 indexes, working with, 116–119
 keys, creating, 114–116
 normalization, 130–136
 view, 58–62
 mutator function, structured UDT, 44

• N •

NAMES ARE clause, MODULE statement, 321
 NATIONAL CHARACTER data type, 34
 NATIONAL CHARACTER LARGE OBJECT
 data type, 34
 NATIONAL CHARACTER VARYING data
 type, 34
 native driver, defined, 332
 natural join, 234
 nested queries (subqueries)
 about, 247–248
 ALL qualifier, 255–257
 ANY qualifier, 255–257
 comparison operators with, 259–261
 defined, 72–73, 247
 DELETE statement with, 262–264
 DML, 72–73
 existence test, 257–259

- EXISTS predicate, WHERE clause, 210–211
 - HAVING clause with, 261–262
 - INSERT statement with, 263–264
 - NOT IN predicate with, 251–252
 - IN predicate, introduced with, 250–251, 259
 - rows, returning sets of, 249–252
 - SELECT statement in, 247–248
 - single value, returning, 252–255
 - SOME qualifier, 255–257
 - UNIQUE predicate, WHERE clause, 211
 - UPDATE statement with, 262–263
 - WHERE clause in, 210–211, 248
 - network database model, 12
 - NEXT option, FETCH statement, 371
 - nonprocedural language, SQL as, 23–25
 - nonrepeatable read problem, defined, 301
 - normal forms, 40, 130–136
 - normalization, 56, 130–136, 247
 - NOT DEFERRABLE designation for constraint, 308
 - NOT EXISTS predicate, WHERE clause, 258–259
 - NOT IN predicate
 - nested queries with, 251–252
 - WHERE clause, 203
 - NOT LIKE predicate, WHERE clause, 204–206
 - NOT logical connective, 71, 218, 417
 - NOT NULL constraint, 59
 - NULL predicate, WHERE clause, 206–207
 - null value, 47, 345
 - NULLIF conditional expression, 189–190
 - NUMERIC data type, 29–30
 - numeric data types, 29–32, 67, 165
 - numeric literal, defined, 67
 - numeric value functions, 176–181
 - observer function, structured UDT, 44
 - OCCURRENCES_REGEX numeric value function, 177–178
 - OCTET_LENGTH numeric value function, 179
 - ODBC (Open Database Connectivity), 329–335
 - ODBC layers, 331
 - ON clause with joins, 246
 - one-to-many relationship among tables, 60, 112–113
 - online application processing (OLAP), 181
 - OOP (object-oriented programming), 42
 - Open Database Connectivity (ODBC), 329–335
 - OPEN statement, cursor, 369
 - operand, defined, 172
 - operator error, data integrity problem, 126
 - operators
 - about, 223
 - comparison, 70, 201–210, 252–257, 259–261
 - concatenation, 67–68, 164
 - EXCEPT, 228–229
 - INTERSECT, 226–228
 - joins (JOIN)
 - about, 229
 - basic join, 229–231
 - checking for valid queries, 416
 - ON clause with, 246
 - column-name join, 235–236
 - condition join, 234–235
 - CROSS JOIN, 233
 - equi-join, 231–233, 234
 - INNER JOIN, 236
 - natural join, 234
 - OUTER JOIN, 236–240
 - UNION JOIN, 240–246
 - WHERE clause with, 246
 - numeric value expression, 67
 - UNION, 150, 223–226
 - XML, 347–351
 - OR logical connective, 71, 217–218, 417
 - Oracle Corporation, 13, 25
 - ORDER BY clause, 198, 199, 221–222, 365–367
- 0 ●
- object compared to relational model, 20
 - object-oriented programming (OOP), 42
 - object-oriented RAD tools, 322–323
 - object-relational model, 20–21
 - objects, database. *See* schema; table; views

OUTER JOIN operators, 236–240
 OVERLAPS predicate, WHERE clause, 212
 OVERLAY string value function, 174
 owner of schema, 74

• p •

parameters (host variables),
 161, 319–320, 391–393
 parent-child table relationship,
 122, 124, 212–214
 PARTIAL keyword, MATCH predicate,
 215–216
 password, default DBA, 279
 permissions. *See* privileges, user
 personal database, defined, 9
 phantom read problem, defined, 301–302
 physical compared to logical schema,
 62–63
 platform instability threat to data, 294
 pointer, 42
 portability considerations, 105
 POSITION numeric value function, 176–177
 POSITION_REGEX numeric value
 function, 178
 POWER numeric value function, 180
 precision of a number, defined, 29
 predicates. *See also* WHERE clause
 defined, 33
 DML, 69–70
 nested query, 210–211, 250–252, 259
 XML-compatible, 352–353
 preprocessing of programs for SQL,
 317, 319
 primary key
 avoiding update anomalies, 125
 database planning, 55
 entity integrity, 120–121
 importance of indexing, 95
 in multitable database, 114–115
 parent-child table relationship, 122
 referential integrity rules, 214
 PRIOR option, FETCH statement, 371
 private attributes of UDT, 43

privileges, user
 across levels, 287–289
 adding data, 282
 character sets, 285–286
 collations, 285–286
 compound SQL statement, 387–388
 delegating, 78–79, 289–290
 deleting obsolete rows, 284
 domains, 278, 285–286
 granting, 280–281, 292
 hierarchy of, 280
 importance of using controls, 418
 looking at data role, 282–283
 modifying table data role, 283
 overview, 74–77
 referencing related tables, 284–285
 revoking, 75–76, 290–291, 292
 roles, 281–282
 translations, 285–286
 trigger events, 287
 procedural language. *See* applications,
 working SQL with
 PROCEDURE statement, 322
 proprietary API, 332
 protected attributes of UDT, 43
 protecting data. *See* integrity, data
 public, the, 280
 public attributes of UDT, 43
 publishing, database, 333

• q •

quantified comparison operators,
 207–210, 252, 255–257
 queries. *See also* operators
 about, 197
 ad-hoc, 24–25
 checking for validity, 416
 complex type, 25
 cursor operations, 365
 defined, 24
 modifying clauses
 FROM, 198, 199
 about, 197–199
 GROUP BY, 198, 218–221

- HAVING, 198, 220–221
 - logical connectives, 216–218
 - ORDER BY, 198, 199, 221–222
 - WHERE. *See* WHERE clause
 - nested (subqueries)
 - about, 247–248
 - ALL qualifier, 255–257
 - ANY qualifier, 255–257
 - comparison operators with, 259–261
 - defined, 72–73, 247
 - DELETE statement with, 263–264
 - DML, 72–73
 - existence test, 257–259
 - EXISTS predicate, WHERE clause, 210–211
 - HAVING clause with, 261–262
 - INSERT statement with, 263–264
 - NOT IN predicate with, 251–252
 - IN predicate, introduced with, 250–251, 259
 - rows, returning sets of, 249–252
 - SELECT statement in, 247–248
 - single value, returning, 252–255
 - SOME qualifier, 255–257
 - UNIQUE predicate, WHERE clause, 211
 - UPDATE statement with, 262–263
 - WHERE clause in, 210–211, 248
 - recursive, 265–274
 - Querying XML* (Melton and Buxton), 347
- R •**
- RAD (rapid application development) tool
 - about, 84
 - altering table structure, 92–94
 - creating a table, 85–92
 - data to track, deciding on, 84–85
 - deleting a table, 96–97
 - index, creating, 94–96
 - object-oriented RAD tools, 322–323
 - READ COMMITTED transaction isolation level, 301
 - READ UNCOMMITTED transaction isolation level, 300–301
 - read-write permissions, controlling, 305
 - REAL data type, 31–32
 - record (tuple), 9, 15
 - recursive queries
 - about, 265–268
 - defined, 268
 - purposes, 268–274
 - syntax, 271–272
 - redundancy strategy to avoid data loss/corruption, 294
 - REF data type, 42, 69
 - REFERENCES privilege, 78, 278
 - REFERENCING old_or_new_value_alias_list phrase, 407–408
 - referencing related tables, user privileges, 284–285
 - referential integrity, 77–78, 104, 122–125, 212–216
 - relation (table), defined, 14. *See also* table
 - relational database. *See also* database
 - about, 7–8
 - advantages, 13–14
 - components, 14–16
 - constraints, 19–20
 - defined, 9, 13
 - design considerations, 21
 - domains, 19
 - flat files, compared to, 11–12
 - object-relational model, 20–21
 - schemas, 19
 - size and complexity range, 9
 - SQL function, 1
 - views, 16–18
 - RELATIVE option, FETCH statement, 371–372
 - REPEATABLE READ transaction isolation level, 301–302
 - REPEAT...UNTIL...END REPEAT statement, 384
 - reserve words, 28, 419–422
 - RESIGNAL action, 380
 - RESTRICT keyword, REVOKE statement, 290–291
 - retrieving data, 8, 140–141. *See also* queries
 - returnability, declaring cursor, 365
 - REVOKE statement, 75–76, 290–291, 292

RIGHT OUTER JOIN operator, 239
 roles, defined, 282
 ROLLBACK statement, 73, 298–299, 304, 306
 row compared to set operations methods, 316
 ROW data type, 40, 69, 357
 row value expression, 193–195
 row values, defined, 158
 rows, returning sets of, nested queries, 249–252

• S •

SAN (storage area network), 10
 sandbox, defined, 337
 SAVEPOINT statement, 306
 scalable DBMS, 10
 scalar (atomic) property of value, 158, 307, 374–375
 scale of a number, defined, 30
 schema
 creating, 62–63
 overview, 19
 owner of, 74
 XML, 339, 344, 346
 SCHEMA clause, MODULE statement, 321
 scope of database project, establishing accurate, 412
 scripts and client extensions, 335
 SCROLL keyword, DECLARE CURSOR statement, 368
 scrollability, declaring cursor, 365
 search conditions, CASE expression with, 184–186
 searched CASE statement, 382
 searching the database. *See* queries
 second normal form (2NF), 133–134
 security
 about, 277
 DCL, 53, 73–79, 278–280
 privileges, user
 across levels, 287–289
 adding data, 282
 character sets, 285–286

collations, 285–286
 compound SQL statement, 387–388
 delegating, 78–79, 289–290
 deleting obsolete rows, 284
 domains, 278, 285–286
 granting, 280–281, 292
 hierarchy of, 280
 importance of using controls, 418
 looking at data role, 282–283
 modifying table data role, 283
 overview, 74–77
 referencing related tables, 284–285
 revoking, 75–76, 290–291, 292
 roles, 281–282
 translations, 285–286
 trigger events, 287
 working with views to enhance, 145
 SELECT statement, 140–141, 247–248, 278
 selected columns, adding data to, 148
 selection condition, creating view from, 143–144
 self-consistency of columns, 15
 self-describing aspect of database, defined, 9
 sensitivity, declaring cursor, 365, 367–368
 serial access to data, advantage of, 297
 SERIALIZABLE transaction isolation level, 302
 server extensions in Web-based databases, 333–334
 server role in database environment, 48–51
 session, client, defined, 162
 SESSION_USER variable, 162
 set compared to row operations methods, 316
 set (aggregate) functions, 71–72, 167–170, 219–220
 SET TRANSACTION statement, 299, 303–304
 SIMILAR predicate, WHERE clause, 206
 simple CASE statement, 381–382
 SIMPLE keyword, MATCH predicate, 215–216
 single value, returning, nested queries, 252–255
 single-table view, 57–58

- SMALLINT data type, 29
- SOME predicate, WHERE clause, 207–210
- SOME qualifier, nested queries, 255–257
- sorting data with indexes, 117–118
- source type, UDT, 43
- special variables, 162
- specialized character set, establishing, 113
- SQL. *See also* applications, working SQL
 - with; *specific topics*
 - about, 1–4
 - in client-server system, 48–50
 - common mistakes in using, 411–414
 - constraints, 47–48
 - core SQL:2008 statement, 26–27
 - as data sublanguage, 183
 - default transaction, 299–300
 - defined, 23–25
 - embedded SQL, 162, 316, 317–320
 - extended SQL:2008 statement, 26–27
 - history, 7, 25–26
 - on Internet/intranet, 50–51, 332–337
 - limitations of, 183
 - Microsoft Access, using with, 97–99
 - name origin, 25
 - null value, 47
 - portability considerations, 105
 - procedural languages, compared to, 83, 315–316
 - reserve words, 28, 419–422
 - statement overview, 26–27
 - Web-based use of, 50–51, 332–337
- SQL module, 320–322
- SQL/DS relational database (RDBMS), 25
- SQL/PSM, 375–380
- SQLSTATE class values, 376–380
- SQLSTATE host variable, 391–393
- SQRT (square root) numeric value function, 181
- standard API, 332
- standards, SQL, 25–27, 373
- statement overview, 26–27, 64–66. *See also individual statements*
- storage area network (SAN), 10
- stored procedures, compound SQL statement, 386
- stored routines, defined, 387
- string value expression, 67–68, 164
- string value functions, 170–176
- structured types, UDT, 44–45, 287
- sublanguage, SQL as data, 24
- subqueries (nested queries)
 - about, 247–248
 - ALL qualifier, 255–257
 - ANY qualifier, 255–257
 - comparison operators with, 259–261
 - defined, 72–73, 247
 - DELETE statement with, 263–264
 - DML, 72–73
 - existence test, 257–259
 - EXISTS predicate, WHERE clause, 210–211
 - HAVING clause with, 261–262
 - INSERT statement with, 263–264
 - NOT IN predicate with, 251–252
 - IN predicate, introduced with, 250–251, 259
 - rows, returning sets of, 249–252
 - SELECT statement in, 247–248
 - single value, returning, 252–255
 - SOME qualifier, 255–257
 - UNIQUE predicate, WHERE clause, 211
 - UPDATE statement with, 262–263
 - WHERE clause in, 210–211, 248
- subselect, 153, 248, 416
- SUBSTRING SIMILAR string value function, 172
- SUBSTRING string value function, 171–172
- SUBSTRING_REGEX string value function, 173
- subtransaction, 306
- subtypes, structured UDT, 44, 287
- SUM function, 72, 170
- Sun Microsystems, 336
- super user, 278–279
- supertypes, structured UDT, 44, 287
- system administrator, 278–279
- system tables, 63
- SYSTEM_USER variable, 162

• T •

table. *See also* multitable database; virtual table (relational database)

- access control, 283, 284–285
- altering structure, 92–94, 104, 125
- attribute (column). *See* attribute (column)
- base type, 141
- blocks of rows, adding, 148–150
- CAST as access tool, 192–193
- conversion from XML to SQL, 354–355
- creating, 55–57, 85–92, 100–103, 104
- deleting, 96–97, 104
- mapping to/from XML, 344–345
- one-to-many relationship, 60, 112–113
- parent-child relationship, 122, 124, 212–214
- system, 63
- temporary-table declaration, 321
- translation, 65
- union-compatibility, 223
- views, 57–58, 142–143

table constraint, 128–129

Taylor, Allen G. (author)

- Access 2003 Power Programming with VBA*, 323
- Database Development For Dummies*, 55

temporary-table declaration clause, 321

termination condition, 267

test database querying, 416

THEN keyword in conditional expressions, 183

third normal form (3NF), 55, 134–135

TIME WITH TIME ZONE data type, 36–37

TIME WITHOUT TIME ZONE data type, 36

TIMESTAMP WITH TIME ZONE data type, 37

TIMESTAMP WITHOUT TIME ZONE data type, 36

transaction processing

- about, 297–298
- ACID database, 307
- backing up data, 305–306
- COMMIT statement, 304
- concurrent access threat to data, 295–296

- constraints on, 307–311
- DCL, 73–74
- default transaction, 299–300
- defined, 295
- implicit transaction-starting statement, 303
- isolation levels, 300–302, 304
- locking database objects, 305
- ROLLBACK statement, 304
- savepoints and subtransactions, 306
- serial access to data, advantage of, 297
- SET TRANSACTION statement, 299, 303–304

transferring data, 149–150, 154–155

transitive dependency, 134–135

TRANSLATE string value function, 176

TRANSLATE_REGEX string value function, 173–174

translation of data, 149, 278, 285–286

translation table, 65

trigger execution context, 406

TRIGGER statement, 278, 287, 403–408

TRIM string value function, 175

troubleshooting (error handling)

- about, 391
- CASE expression as tool for, 186
- diagnostics areas, 394–400
- exception conditions, 401–402
- preparing for, 418
- SQLSTATE host variable, 391–393
- WHENEVER clause, 393–394

tuple (record), 9, 15

two-dimensional array, relation as, 14–15

type hierarchy, 287–289

• U •

UDT (user-defined data type), 42–45, 69

Ulrich-Fuller, Laurie (author), *Access For Dummies*, 323

UNDER keyword, permission for using, 278

UNION ALL operator, 225

UNION CORRESPONDING operator, 226

UNION DISTINCT operator, 225

UNION JOIN operator, 240–246

UNION operator, 150, 223–226
 union-compatibility of tables, 223
 unique (primary) key
 avoiding update anomalies, 125
 database planning, 55
 entity integrity, 120–121
 importance of indexing, 95
 in multitable database, 114–115
 parent-child table relationship, 122
 referential integrity rules, 214
 UNIQUE predicate, WHERE clause,
 211, 214–216
 universal time (Greenwich Mean Time/
 GMT), defined, 36
 update anomalies, 122–125
 UPDATE statement
 about, 75
 basics of updating data, 151–153
 CASE expression in, 185–186
 with cursor, 367, 372
 with nested queries, 262–263
 trigger, relationship to, 404, 406
 user access, 278
 updating table structure, consequences
 of, 92–93
 UPPER string value function, 174–175
 USAGE keyword, permission for, 278
 user name, defined, 281–282
 user privileges. *See* privileges, user
 user-defined data type (UDT), 42–45, 69

• U •

VALID predicate (XML), 353
 value expressions
 about, 163–164, 183
 Boolean, 68–69
 CAST data type, 191–193, 316
 collection, 69
 conditional, 166, 183–191
 datetime, 68, 165–166
 DML, 66–69
 interval, 68, 166
 numeric, 67, 165
 reference, 69
 row, 69, 193–195

 string, 67–68, 164
 user-defined, 69
 values. *See also* functions
 about, 157
 asterisk (*) as wildcard, 140, 314
 atomic property of, 158, 307, 374–375
 CASE expression with, 186–188
 comma-delimited, 146
 constants, 158–160
 null value, 47, 345
 row, defined, 158
 SQLSTATE class, 376–380
 types of, 157–163
 VARBINARY data type, 35
 variables
 basics of using, 160–162
 compound SQL statement, 375–376
 constants, compared to, 158
 host, 161, 319–320, 391–393
 VIEW command, 57–62
 views, 16–18, 57–62, 141–145
 virtual table (relational database). *See also*
 database
 about, 7–8
 advantages, 13–14
 components, 14–16
 constraints, 19–20
 defined, 9, 13
 design considerations, 21
 domains, 19
 flat files, compared to, 11–12
 object-relational model, 20–21
 schemas, 19
 size and complexity range, 9
 SQL function, 1
 views, 16–18

• W •

Web-based database environment,
 50–51, 332–337
 WHENEVER clause, error handling, 393–394
 WHERE clause
 about, 200–201
 ALL predicate, 207–210
 ANY predicate, 207–210

- WHERE clause (*continued*)
- comparison predicate, 201
 - DISTINCT predicate, 211
 - EXISTS predicate, 210–211, 257–258
 - function, 198
 - with joins, 246. *See also* join (JOIN) operators
 - LIKE predicate, 204–206
 - MATCH predicate, 212–216
 - in nested queries, 210–211, 248
 - NOT EXISTS predicate, 258–259
 - NOT IN predicate, 203
 - NOT LIKE predicate, 204–206
 - NULL predicate, 206–207
 - ON versus WHERE, 246
 - OVERLAPS predicate, 212
 - BETWEEN predicate, 202–203
 - IN predicate, 203
 - retrieving data with SELECT, 140
 - SIMILAR predicate, 206
 - SOME predicate, 207–210
 - UNIQUE predicate, 211, 214–216
 - with UPDATE, 151, 153
 - WHILE...DO...END WHILE statement, 384
 - WIDTH_BUCKET numeric value function, 181
 - wildcard character (*), 140, 314
 - WITH GRANT OPTION clause, GRANT statement, 78–79, 289–290, 291
 - WORK keyword, COMMIT statement, 304
 - workgroup database, defined, 9
 - XML (CONTENT (XMLSCHEMA)) data type, 39
 - XML (DOCUMENT (ANY)) data type, 39
 - XML (DOCUMENT (UNTYPED)) data type, 39
 - XML (SEQUENCE) data type, 38
 - XML (eXtensible Markup Language) data
 - about, 339–340
 - array mapping, 358
 - conclusion, 359
 - distinct UDT mapping, 356–357
 - domain mapping, 355–356
 - functions compatible with, 347–351
 - mapping from/to SQL, 342–346, 355–359
 - multiset mapping, 358–359
 - predicates compatible with, 352–353
 - ROW mapping, 357
 - table conversion to SQL, 354–355
 - XML data type, 37–39, 340–342
 - XMLAGG operator, 349
 - XMLCAST operator, 351
 - XMLCOMMENT operator, 350
 - XMLCONCAT operator, 348–349
 - XMLDOCUMENT operator, 347
 - XMLELEMENT operator, 347
 - XMLEXISTS predicate (XML), 352–353
 - XMLFOREST operator, 348
 - XMLP1 operator, 350
 - XMLPARSE operator, 350
 - XMLQUERY operator, 351
 - XMLTABLE pseudo-function (XML), 354–355
 - XQuery, 38–39, 347



- XML (CONTENT (ANY)) data type, 38
- XML (CONTENT (UNTYPED)) data type, 38–39



- year-month interval, 37, 68, 166