



# Index

## A

abbreviations

- in field names, 22
- in table names, 31

*Access*. *See under* Microsoft

acronyms

- in field names, 22
- in table names, 31

aggregate functions, 74, 375–377, 416–428.

*See also* AVG; COUNT; COUNT(\*);  
MAX; MIN; SUM

- column name in, 419
- in filters, 428–430
- grouping of, 442–444 (*See also* GROUP BY)
- imbedding of, 428
- multiple, 427–428
- Nulls and, 417, 416*f*
- overview, 416–418
- sample statements, 431–438
- syntax, 416, 416*f*, 595*f*
- uses of, 416
- WHERE clause in, 419, 430

aliases

- for column names, 131
- for tables, 252–254, 252*f*

ALL

- in subqueries, 386–389, 387*f*, 393–394,  
406–407
- syntax, 487*f*, 598*f*
- in UNIONS, 228–230, 341, 343, 345, 348

Allaire, *Cold Fusion*, 6

American National Standards Institute  
(ANSI), 57. *See also* SQL Standard

American Standard Code for Information  
interchange (ASCII), 107

analytic databases, defined, 4

AND, 179, 180*f*

- null values and, 195–196

with OR, 183–184

sample statements, 204–205

Ansa Software, *Paradox*, 5

ANSI (American National Standards  
Institute), 57

ANSI/ISO standard. *See* SQL Standard

ANSI NCITS-H2, 62

“ANSI X3.168-1989 Database Language  
Embedded SQL,” 58

“ANSI X3.135-1986 Database Language SQL”  
(SQL/86), 57–58

ANY, 386–389, 387*f*, 400, 598*f*

*approximate numeric* data type, 108–109

archiving of data, 547–549, 554–555

deleting data after archiving, 574–575, 579

artificial primary keys, 41–42, 41*f*

AS, 130–131, 252–253, 252*f*

ASC, 91, 96, 98–99, 100–101

ASCII (American Standard Code for Informa-  
tion interchange), 107, 157–159

Ashton Tate, *dBase*, 5

asterisk shortcut, 83–84, 83*f*

ASYMMETRIC BETWEEN, 165

attribute, defined, 6, 215. *See also* fields

AUTO\_INCREMENT, 543

AutoNumber (*Access*), 543

AVG, 416*f*, 423–424, 432, 435, 595*f*.

*See also* aggregate functions

## B

base tables, defined, 9

Begg, Carolyn, 50

BETWEEN, 154, 164–167, 164*f*, 191–193

ASYMMETRIC, 165

SYMMETRIC, 165

BIGINT data type, 108

BINARY data type, 108

BINARY LARGE OBJECT data type, 108

- BIT data type, 108, 109  
 BIT\_LENGTH, 595f  
 BIT VARYING data type, 108  
 BLOB data type, 108  
 Boolean data type, 109
- C**
- calculated columns, naming of, 129–131, 130f  
 calculated data. *See also* expressions  
   in fields  
     disadvantages of, 24, 25  
     resolving of, 33  
     uses of, 514–516  
   in SELECT statements, 74  
 Call-Level Interface (CLI) specification, 61–62  
 Cartesian product, 248, 297  
 cascade deletion rule, 45, 46f  
 case sensitivity  
   in collation of character string literals,  
     157–160  
   in comparison of character string literals,  
     171  
 CAST, 110–112, 110f  
   for datetime literal compatibility, 117,  
     119–120, 141, 142–143, 144–145  
   for mathematical expression compatibility,  
     123–124  
   syntax, 110f, 594f  
 Chamberlin, Donald, 54  
 changing data. *See* DELETE; INSERT; UPDATE  
 CHARACTER (CHAR) data type, 107  
 CHARACTER (CHAR) LARGE OBJECT data  
   type, 107  
 CHARACTER (CHAR) VARYING data type,  
   107  
 character string functions, syntax, 594f  
 character string literals  
   collating sequence of, 157–160  
   concatenating, 118–119, 118f  
   less than and greater than comparisons,  
     162  
   LIKE and, 169–173, 170f  
   MAX and, 424  
   range condition predicates, 166  
   specifying, 112–114, 113f  
   syntax, 113f, 492f  
 CHAR\_LENGTH, 595f
- CLI (Call-Level Interface) specification,  
   61–62  
 client/server computing, introduction of, 5, 65  
 CLOB data type, 107  
 Codd, Edgar F., 4–5, 54  
 Cold Fusion, 6  
 collating sequence  
   of character string literals, 157–160  
   ORDER BY clause and, 88–89  
 columns  
   added, SELECT statement and, 84  
   aliases for, 131  
   calculated, naming of, 129–131, 130f  
   column references, 245–246, 246f, 249,  
     593f  
   deleted, SELECT statement and, 84  
   ordering of, 82–83  
   as term, 72  
 COMMIT, 506  
 comparison predicates, 156–164  
   comparison operators, 154  
   equality and inequality, 160–161  
   exclusive, 161  
   inclusive, 161  
   less than and greater than, 162–164  
   NOT operator and, 185–186  
   string values collating sequence, 157–160  
   syntax, 157f, 597f  
   types of, 156–157, 157f  
 composite primary keys (CPK)  
   defined, 8, 39  
   use of, 39  
 Computer Associates International, Inc., 56  
 concatenation expressions, 117–120, 118f  
   length limits for returns, 119  
   in SELECT clause, 128–129  
   storing as data, 24  
   syntax, 118f  
 concatenation operators, 117–118, 118f  
 Connolly, Thomas, 50  
 copying of data, 547–549  
 correlation names (aliases)  
   for column names, 131  
   for tables, 252–254, 252f  
 CORRESPONDING clause, 343  
 COUNT, 375–376, 420–421. *See also*  
   aggregate functions

- DISTINCT and, 420, 469
- HAVING COUNT trap, 481–485
- Null values and, 420
- sample statements, 427–428, 431, 435
- syntax, 376*f*, 416*f*, 595*f*
- COUNT(\*), 375–376, 418–420
  - DISTINCT and, 421
  - Null values and, 417
  - sample statements, 395, 397, 399, 437, 438
  - syntax, 376*f*, 416*f*, 595*f*
  - WHERE clause of, 419
- CPK. *See* composite primary keys
- D**
- data. *See also* data types
  - defined, 75
  - dynamic, defined, 4
  - static, defined, 4
  - vs.* information, 75–77, 76*f*
- Database 2 (DB2). *See under* IBM
- database design. *See also* database structure
  - books on, 16–17
  - importance of understanding, 19–20
  - methodology, good
    - analysis of database with, 50
    - value of, 16
  - vs.* database theory, 16
  - vs.* implementation, 16
- Database Design for Mere Mortals* (Hernandez), 5, 16, 42, 50
- database implementation, *vs.* database design, 16
- database models
  - relational, history of, 4–6
  - types of, 4
- databases
  - examples
    - downloadable version of, 93
    - schema of, 601–605
    - types of, 3–4
    - Web-centric, 6, 65
- database structure
  - fine-tuning
    - fields, 21–29
    - tables, 30–42
    - thorough design process analysis, 50
  - sound, importance of, 20
- Database Systems: A Practical Approach* (Connolly and Begg), 50
- database theory
  - importance of understanding, 16
  - vs.* database design, 16
- data types, 107–109
  - changing types, 110–112, 110*f*
  - extended, 109
- Date, C. J., 16, 58
- DATE data type, 109
- date expressions, 124–126, 125*f*
  - leap year adjustments, 144–145
  - sample expressions, 141, 142–143, 144–145
  - in SELECT clause, 132–133
- date functions
  - in specific software, 607–614
  - syntax, 594*f*
- DATE keyword, 116, 117
- date literals
  - BETWEEN and, 165–166, 167
  - specifying, 115–116
  - syntax, 115*f*, 592*f*
- datetime arithmetic expressions, 124–127
- DATETIME data type. *See also entries under*
  - date; time; timestamp
  - concatenation of, 119–120
  - overview of, 109
- datetime functions
  - in specific software, 607–614
  - syntax, 594*f*
- datetime literals
  - specifying, 115–117, 115*f*
  - syntax, 115*f*, 592*f*
- dBase*, 5
- DB2 (Database 2). *See under* IBM
- DEC data type, 108
- DECIMAL data type, 108
- DEFAULT, 540
- degree of participation
  - defined, 48
  - enforcement of, 49–50
  - setting, 48–50, 49*f*
- DELETE
  - of all rows, 569–570
  - of data after archiving, 574–575, 579
  - overview, 567–568

DELETE (*cont.*)

- recovering lost data, 506–507, 570
- sample statements, 576–583
- of some rows, 571–575
- syntax, 568–569, 568*f*, 599*f*
- transactions and, 506–507, 570
- transforming SELECT statement into, 572, 572*f*
- uses of, 568, 575–576
- verifying accuracy of target materials, 571–572
- WHERE clause in, 568, 568*f*, 571–575
  - sample statements, 576–583
  - subqueries, 573–575

## deletion rule

- defined, 44
- establishing, 44–45, 46*f*
- types of, 45, 46*f*

## delimited identifiers, 22, 32

derived tables, in INNER JOIN, 254–256, 255*f*

## DESC keyword, 90–91, 100

diagrams. *See* syntax diagramsdifference, 222–227. *See also* EXCEPT

- commercial systems' support, 233
- limitations of, 227–228
- by OUTER JOIN with Null test, 295, 300, 318
- between result sets, 224–227
- set diagrams of, 225–227, 225*f*, 227*f*
- in set theory, 222–224
- uses of, 215, 227–228

## DISTINCT

- AVG and, 424
- COUNT/COUNT(\*) and, 376, 420, 469
- in JOINS, 233, 247*f*, 252*f*, 255*f*, 296*f*, 302*f*, 305*f*, 358, 384
- MAX and, 425–426
- MIN and, 426–427
- in SELECT statements, 84–86, 85*f*, 97, 102
- in UNIONS, 348
- “does not apply,” *vs.* “is not applicable,” 137–138

## DOUBLE PRECISION data type, 108–109

## duplicate fields

- disadvantages of, 24–25, 36
- identification of, 33
- resolving of, 33–38

duplicate rows, eliminating, 84–86, 85*f*

## dynamic data, defined, 4

## E

## EBCDIC (Extended Binary Coded Decimal Interchange Code), 158–159, 197

## embedded SQL, specification for, 58–59

## enforcement, of participation degree, 49–50

## equality comparison predicates, 160–161

## escape characters, 172

ESCAPE option, of LIKE predicate, 169*f*, 172–173

## Euler, Leonard, 219

Euler diagrams, 219. *See also* set diagramsevents, defined, 7, 7*f*, 32–33*exact number* data type, 108

## EXCEPT, 222–227

- alternatives to, 295
- syntax, 236–239, 238*f*

## execution, of query

- defined, 93
- methods, 93

EXISTS, 389–392, 393–394, 402–403, 598*f*expressions. *See also* value expressions

- column, subqueries as, 372–377
  - aggregate functions for, 375–377, 376*f*
  - GROUP BY in, 452–453, 461–463
  - sample statements, 395–499
  - syntax, 372–375
  - uses, 392

## data types

- changing of, 110–112, 110*f*
  - (*See also* CAST)
- overview of, 107–109

## defined, 106

## literal values, specifying, 112–117

- character string literals, 112–114, 113*f*
- datetime literals, 115–117, 115*f*
- numeric literals, 114*f*

naming of, 129–131, 130*f*

## overview of, 106

## in SELECT clause, 128–135

- concatenation expressions, 128–129
- date expressions, 132–133
- mathematical expressions, 131–132
- sample statements, 139–147
- value expressions, 135, 135*f*

- types of, 117-127
  - concatenation, 117-120, 118*f*
    - length limits for returns, 119
    - in SELECT clause, 128-129
    - storing as data, 24
    - syntax, 118*f*
  - date and time, 124-127
    - leap year adjustments, 144-145
    - sample expressions, 141, 142-143, 144-145
    - in SELECT clause, 132-133
  - mathematical, 121-124, 121*f*
    - nulls in, 138-139, 139*f*
    - sample statements, 140, 143, 145, 147
    - in SELECT clause, 131-132
  - uses of, 128
- Extended Binary Coded Decimal Interchange Code (EBCDIC), 107, 158-159
- extended data types, 109
- extensions to SQL standard, 60-61, 73
- EXTRACT, 595*f*
- F**
- false, value of, 507-508
- Federal Information Processing Standard (FIPS), 61
- fields
  - duplicate
    - disadvantages of, 24-25, 36
    - identification of, 33
    - resolving of, 33-38
  - fine-tuning of, 21-29
  - multipart
    - identification of, 24, 25, 25*f*, 26, 27*f*
    - resolving of, 25-27, 33
  - multivalued
    - identification of, 24, 27-28, 27*f*
    - resolving of, 27-29, 28*f*, 33
  - naming of, 21-22
  - overview of, 7-8
  - structure, fine-tuning of, 23-25, 23*f*
    - as term, 72
- filters. *See also* HAVING clause; WHERE clause
  - aggregate functions in, 428-430
  - subqueries as, 377-392
    - predicate keywords, 380-392
    - sample statements, 400-409
    - syntax, 378-380, 378*f*
    - uses, 393-394
    - WHERE *vs.* HAVING, uses of, 478-485
- FIPS (Federal Information Processing Standard), 61
- FIPS PUB 127, 61
- FK (foreign keys), 9, 9*f*
- FLOAT data type, 108-109
- foreign keys (FK), 9, 9*f*
- FROM clause
  - in FULL OUTER JOIN, 314*f*
  - in INNER JOIN, 247, 247*f*; 249-250
    - correlation names for tables, 252-254, 252*f*
    - embedded JOIN in, 256-261, 257*f*, 260*f*
      - sample statements, 270-272, 276-277, 280-282, 284-288
    - embedded select statement in, 254-256, 255*f*
      - sample statements, 278-282, 284-288
  - in OUTER JOIN, 296-300, 296*f*
    - embedded JOIN in, 304-314, 305*f*, 306*f*, 308*f*
      - sample statements, 320-324, 326-331, 333-334
    - embedded select statements in, 301-304, 302*f*
      - sample statements, 320-322, 326-327, 329-331
  - in SELECT, 74, 74*f*
  - in UNION JOIN, 318*f*
- FULL OUTER JOIN, 314-317, 314*f*
  - alternatives to, 316, 334
  - on non-key values, 317
- function, defined, 73, 92
- G**
- greater than* comparison predicates, 162-164
- GROUP BY, 444-458
  - aggregate functions in, 452, 455, 457-458
  - mixing columns and expressions, 450-452
  - nonaggregate column references, 450, 452, 455-456
  - sample statements, 459-470
    - as SELECT DISTINCT alternative, 453-454

GROUP BY (*cont.*)

- in SELECT statement, 74*f*, 75
- in subquery, 452–453, 461–463
- syntax, 445–450, 445*f*
- uses for, 458
- without aggregate functions, 453–454

## H

- HAVING clause, 74*f*, 75
  - HAVING COUNT trap, 481–485
  - restrictions on, 476
  - sample statements, 487–496
  - syntax, 476, 476*f*
  - uses of, 474–478, 486–487
  - vs.* WHERE, uses of, 478–485
- HAVING COUNT trap, 481–485
- Hernandez, Michael J., 5, 16, 42, 50

## I

## IBM

- DB2 (Database 2)
  - concatenation in, 118
  - data entry in, 538
  - date and time functions in, 607–608
  - history of, 5, 56, 57
  - not equal to* operator, 161
- SAA (Systems Application Architecture), 61
- SQL/Data System (SQL/DS), 56
- SQL development, 54–55
- string values collating sequence, 158–159
- System R*, 5, 54–55

## identifiers

- delimited, 22, 32
- regular, 22, 32

## inequality comparison predicates, 160–161

## information

- defined, 75
- vs.* data, 75–77, 76*f*

*Informix-SE*, 65, 118*INGRES*, 5, 56, 57, 118IN keyword, 380–386, 381*f*, 393–394, 404

## INNER JOIN, 244–262

- ON clause, 247–250, 247*f*
- column references, 245–246, 246*f*, 249
- correlation names for tables, assigning, 252–254, 252*f*

- embedded JOINS in, 256–261, 257*f*, 260*f*
  - sample statements, 270–272, 276–277, 280–282, 284–288
- embedded SELECT statements in, 254–256, 255*f*

## sample statements, 278–282, 284–288

## JOIN-eligible data types, 244–245

## knowledge of tables and, 261–262

## overview of, 244

## sample statements, 263–288

syntax, 246–262, 247*f*, 257*f*of three or more tables, 256–261, 257*f*, 260*f*, 270–277, 347*f*of two tables, 247–251, 247*f*, 257*f*, 264–269, 339–340, 340*f*

## uses for, 262–263

USING clause, 247, 247*f*, 250–251IN predicate, 154, 167–169, 168*f*, 598*f*

## INSERT

## column references in, 540

individual values and rows, 539–544, 539*f*

## overview, 537–539

## primary keys, automatic generation of, 542–544

## sample statements, 552–563

with SELECT expression, 544–550, 544*f*, 545*f*, 599*f*

## sample expressions, 554–555, 556–559, 561–562

sets of data, 544–550, 544*f*, 545*f*

## transactions and, 506

## uses for, 550–551

## with VALUES keyword, 539–544

## sample statements, 552–553, 556, 559–560

syntax, 539*f*, 599*f*

## INTEGER (INT) data type, 108, 109

International Organization for Standardization (ISO), 58. *See also* SQL StandardINTERSECT, syntax, 234–236, 236*f*intersection, 216–222. *See also* INTERSECT

## commercial systems' support, 233

## limitations of, 221

set diagrams of, 218–221, 219*f*, 220*f*, 235*f*

## of set results, 217–221

## in set theory, 216–217

## uses of, 215, 221–222

## INTERVAL data type, 109, 125

- interval literals, 116, 593*f*  
*An Introduction to Database Systems*  
 (Date), 16  
 “is not applicable,” *vs.* “does not apply,”  
 137–138
- IS NULL predicate, 154, 173–175, 174*f*, 197,  
 205, 295, 300, 318  
 syntax, 174*f*, 598*f*
- “ISO 9075-1987 Database Language SQL,”  
 58
- “ISO 9075:1989 Database Language SQL with  
 Integrity Enhancements” (SQL/89),  
 58
- “ISO/IEC 9075:1992 Database Language SQL,”  
 59–60
- ISO (International Organization for Standard-  
 ization), 58. *See also* SQL standard
- J**
- JOIN. *See also* INNER JOIN; OUTER JOIN  
 alternatives to, 250  
 commercial systems’ support, 250  
 default mode, 247  
 NATURAL, 251, 301  
 overview of, 243–244  
 UNION, 317, 318*f*  
 uses of, 221–222
- K**
- keys  
 composite primary (CPK)  
   defined, 8, 39  
   use of, 39  
 foreign (FK), 9, 9*f*  
 overview of, 8–9  
 primary (PK)  
   artificial, 41–42, 41*f*  
   automatic generation of, 542–544  
   composite, 8, 39  
   criteria for, 39–41, 40*f*, 41*f*  
   defined, 6, 8, 9*f*, 39  
   functions of, 8, 33  
   simple, 39
- L**
- largest value. *See* MAX
- LEFT OUTER JOIN  
 defined, 295–296
- sample statements, 319–334  
 syntax, 296–314, 296*f*
- less than* comparison predicates, 162–164
- LIKE, 154, 169–173, 169*f*  
 ESCAPE option, 169*f*, 172–173  
 sample pattern strings, 170*t*  
 sample statements, 206  
 syntax, 169*f*, 598*f*  
 wildcard characters, 169–170
- linking tables  
 advantages of, 14–15  
 defined, 13  
 defining of, 13–14  
 in resolving duplicate fields, 37–38, 37*f*, 38*f*  
 in resolving multivalued fields, 28, 29*f*
- literals. *See also* character string literals; date  
 literals; datetime literals; numeric  
 literals; time literals; timestamp  
 literals  
 interval, 116, 593*f*  
 keywords for, 116, 117  
 specifying values for  
   character string literals, 112–114, 113*f*  
   datetime literals, 115–117, 115*f*  
   numeric literals, 114*f*  
 types  
   changing, 110–112, 110*f*  
     (*See also* CAST)  
   overview, 107–109
- lost data, recovering, 570. *See also* transactions
- M**
- mandatory participation, 46–47, 48*f*  
 many-to-many relationship, 13–15, 13*f*, 14*f*,  
 43–44, 44*f*
- mathematical expressions, 121–124, 121*f*  
 nulls in, 138–139, 139*f*  
 sample statements, 140, 143, 145, 147  
 in SELECT clause, 131–132
- MAX, 376–377, 424–426. *See also* aggregate  
 functions  
 in multiple aggregate functions, 427  
 sample statements, 396, 398, 434  
 syntax, 376*f*, 416*f*, 595*f*
- mean values. *See* AVG
- members, of set. *See also* IN predicate  
 characteristics of, 216–217  
 defined, 215

- MEMO data type, 107
- Microrim, 5
- Microsoft
  - Access, 29, 65, 92
    - AutoNumber in, 543
    - concatenation in, 118
    - COUNT DISTINCT and, 470
    - date and time functions in, 609–610
    - date entry in, 537–538
    - DELETE in, 570, 570*f*
    - GROUP BY in, 456
    - JOINS in, 320, 334
    - LIKE predicate in, 170
    - subqueries, aggregate functions in, 521
    - transactions in, 506
    - UPDATE
      - JOIN clause in, 511–512, 521
  - subquery UPDATE expressions, 514
  - ODBC (Open Database Connectivity)
    - specification, 62
  - SQL Server
    - concatenation in, 118
    - date and time functions in, 610
    - FULL OUTER JOINS in, 316
    - history of, 5–6, 65
    - Identity data type, 543
    - not equal to* operator, 161
    - OUTER JOIN in, 299
    - TOP keyword in, 92
    - UPDATE, JOIN clause in, 511–512
  - Visual Studio, 6
- MIN, 416*f*, 426–427, 433, 595*f*. *See also* aggregate functions
- MONEY/CURRENCY data type, 109
- multipart fields
  - identification of, 24, 25, 25*f*, 26, 27*f*
  - resolving of, 25–27, 33
- multiple column retrieval, 81–83, 81*f*
- multivalued fields
  - identification of, 24, 27–28, 27*f*
  - resolving of, 27–29, 28*f*, 33
- MySQL
  - AUTO\_INCREMENT in, 543
  - concatenation in, 118
  - date entry in, 538
  - date and time functions in, 611–613
  - datetime specification in, 116, 117
  - embedded SELECT statements in, 255
  - history of, 6
  - JOINS in, 334
- N**
- names. *See also* aliases
  - for columns, 130–131
  - of fields, 21–22
  - of tables, 30–32
- NATIONAL CHARACTER (CHAR) data type, 107–108
- NATIONAL CHARACTER (CHAR) LARGE OBJECT data type, 108
- NATIONAL CHARACTER (CHAR) VARYING data type, 108
- National Committee for Information Technology Standards (NCITS), 62
- National Institute of Standards and Technology (NIST), 61
- NATURAL JOIN, 251, 301
- NCHAR data type, 107–108
- NCHAR VARYING data type, 108
- NCITS-H2, 62
- NCITS (National Committee for Information Technology Standards), 62
- NCLOB data type, 108
- NEXTVAL, 543
- NIST (National Institute of Standards and Technology), 61
- not equal to* operator, 161
- NOT operator
  - in comparison conditions, 185–186
  - double, 186–187
  - as first keyword, 184–187, 185*f*
  - within predicate, 175–178, 176*f*
  - sample statements, 186–187
- NTEXT data type, 108
- Nulls, 135–139. *See also* IS NULL predicate
  - COUNT and, 420
  - COUNT(\*) and, 417
  - defined, 136–137
  - in mathematical expressions, 138–139, 139*f*
  - and multiple condition searches, 193–197, 194*f*, 195*f*, 196*f*
  - uses of, 137–138
- NUMERIC data type, 108



numeric functions, syntax, 595*f*

numeric literals

BETWEEN and, 164–165

specifying, 114*f*

syntax, 114*f*, 592*f*

## O

objects, defined, 7, 32

ODBC (Open Database Connectivity)

specification, 62

*Office Access 2007*, 29

ON clause, 247–250, 247*f*, 296*f*, 297–301

one-to-many relationship, 12, 12*f*, 43, 43*f*

one-to-one relationship, 11–12, 12*f*, 42–43, 43*f*

Open Database Connectivity (ODBC)

specification, 62

operational databases, defined, 4

optional participation, 46–47, 48*f*

*Oracle*

concatenation in, 118

data entry in, 538

date and time functions in, 613–614

GROUP BY in, 456

history of, 5, 55–56

NEXTVAL in, 543

OUTER JOIN in, 299

ORDER BY

ASC keyword, 91, 96, 98–99, 100–101

column name, specifying, 90–91, 351–352, 351*f*

column number, specifying, 351–352, 351*f*

DESC keyword, 90–91, 100

order of precedence in, 88–89

in SELECT statements, 87–92, 87*f*

sample statements, 96, 98–99, 100–101, 146, 200–202

in UNIONS, 351–352, 351*f*

ordering

of columns, 82–83

of rows, 87–92, 87*f*

order of precedence

in mathematical expressions, 121–123

in multiple search conditions, 182

default order, 187–188

and efficiency of search, 190–191

prioritizing of conditions, 188–190

in ORDER BY clause, 88–89

OR operator, 180–182, 181*f*

with AND, 183–184

null values and, 196, 196*t*

sample statements, 203

orphaned records

avoiding, 9, 44–45

defined, 44

OUTER JOIN

ON clause, 296*f*, 297–301

commercial systems' support, 299

embedded JOINS in, 304–314, 305*f*, 306*f*,

308*f*

sample statements, 320–324, 326–331, 333–334

embedded SELECT statements in, 301–304,

302*f*

sample statements, 320–322, 326–327, 329–331

FULL, 314–317, 314*f*

alternatives to, 316, 334

on non-key values, 317

LEFT

defined, 295–296

sample statements, 319–334

syntax, 296–314, 296*f*

overview, 293–295

RIGHT

defined, 295–296

syntax, 296–314, 296*f*

sample statements, 319–334

syntax, 296–314, 296*f*

of three or more tables, 304–314, 306*f*, 308*f*

of two tables, 296–301, 296*f*, 305*f*

uses of, 228, 318–319

USING clause, 296*f*, 297, 300–301

overlapping ranges, checking for, 191–193, 192*f*

## P

*Paradox*, 5

parent-child relationship, 311

parentheses

for combining search conditions, 184

in embedded JOINS within JOINS,

257–258, 257*f*, 304–305, 306*f*, 308*f*

in embedded SELECT statements within

JOINS, 254–255, 255*f*

in mathematical expressions, 122–123

- parentheses (*cont.*)  
 in prioritizing of conditions, 189–190  
 for subqueries, 374
- participation degree  
 defined, 48  
 enforcement of, 49–50  
 setting, 48–50, 49f
- participation type, setting, 46–47, 47f, 48f
- pattern-matching condition. *See* LIKE
- PK. *See* primary keys
- POSITION, 595f
- predicates. *See also* search conditions  
 quantified, 386–389, 387f  
 in WHERE clause (*See also* BETWEEN;  
 comparison predicates; IN  
 predicate; IS NULL predicate; LIKE)  
 efficiency of, 190–191  
 expressing in different ways, 197–198  
 INTERSECT, 234–236, 236f  
 multiple conditions, 178–197  
 NOT operator, 175–178, 176f, 184–187,  
 185f  
 nulls, evaluation of, 193–197, 194f, 195f,  
 196f  
 AND operator, 179, 180f, 183–184,  
 195–196, 204–205  
 order of precedence. *See* order  
 of precedence  
 OR operator, 180–182, 181f, 184–185,  
 196, 196t, 203  
 AND and OR together, 183–184  
 overview, 152–154  
 sample statements, 198–206
- primary keys (PK)  
 artificial, 41–42, 41f  
 automatic generation of, 542–544  
 composite, 8, 39  
 criteria for, 39–41, 40f, 41f  
 defined, 6, 8, 9f, 39  
 functions of, 8, 33  
 simple, 39
- primary tables, defined, 11
- Q**  
 quantified predicates, 386–389, 387f  
 QUEL (Query Language), 56, 57  
 query. *See also* SELECT query  
 defined, 10, 73, 92–93  
 execution methods, 93  
 saved, defined, 10  
 Query Language (QUEL), 56, 57  
 query optimizers, 190–191, 256, 308–309,  
 320, 354
- quotes  
 for character string literals, 112–113, 113f  
 single, embedded, 186
- R**  
 range, finding data within. *See* BETWEEN  
*R:BASE*, 5  
 RDBMSs. *See* relational database management  
 systems  
 REAL data type, 108–109
- records  
 orphaned  
 avoiding, 9, 44–45  
 defined, 44  
 overview of, 8  
 as term, 72
- recovering lost data, 570. *See also*  
 transactions
- regular identifiers, 22, 32
- relation, 6. *See also* tables
- relational database management systems  
 (RDBMSs)  
 collating sequences of, 88–89  
 history of, 55–56, 64–65  
 products, 5–6
- relational databases  
 anatomy of, 6–15  
 history of model, 4–6  
 importance of understanding, 15–16
- relational database software. *See* relational  
 database management systems
- “A Relational Model of Data for Large Shared  
 Databanks” (Codd), 5
- Relational Software, Inc., 55–56
- Relational Technology, Inc., 56
- relationships between tables. *See also* linking  
 tables  
 degree of participation  
 defined, 48  
 enforcement of, 49–50  
 setting, 48–50, 49f  
 deletion rule  
 defined, 44

- establishing, 44–45, 46*f*
    - types of, 45, 46*f*
    - relationship characteristics, establishing, 44–50
    - type of participation, setting, 46–47, 47*f*, 48*f*
    - unresolved, defined, 13, 13*f*
  - Request/Translation/Clean UP/SQL
    - technique, 77–81
  - restrict deletion rule, 45, 46*f*
  - result set, defined, 77
  - RIGHT OUTER JOIN
    - defined, 295–296
    - syntax, 296–314, 296*f*
  - ROLLBACK, 506
  - ROUND, 521, 522
  - rows
    - counting of. *See* COUNT(\*)
    - ordering of, 87–92, 87*f*
    - as term, 72
  - row subqueries, 370–371
  - row value constructors, 370–371
- S**
- SAA (Systems Application Architecture), 61
  - sample database schema, 601–605
  - saved query, defined, 10
  - saving, of SELECT statements, 92–93
  - scalar subqueries, 370, 372, 373–375
  - search conditions. *See also* HAVING clause;
    - ON clause; predicates; WHERE clause
  - combining with parentheses, 184
  - expressing in different ways, 197–198
  - order of precedence, 182
    - default order, 187–188
    - and efficiency of search, 190–191
    - prioritizing of conditions, 188–190
  - syntax diagram, 178*f*, 185*f*, 597*f*
  - in WHERE clause, 152–153
  - secondary tables, defined, 11
  - SELECT clause
    - expressions in, 128–135
      - concatenation expressions, 128–129
      - date expressions, 132–133
      - mathematical expressions, 131–132
      - sample statements, 139–147
      - value expressions, 135, 135*f*
    - overview of, 74, 74*f*
  - SELECT expression
    - defined, 342
    - INSERT using, 544–550, 544*f*, 545*f*
    - as part of SELECT operation, 72
    - syntax, 591*f*
    - in value expression, 373, 373*f*
  - SELECT operation
    - overview, 72
    - parts of, 72 (*See also* SELECT expression; SELECT query; SELECT statement)
  - SELECT query, 87–92, 87*f*
    - defined, 87
    - as part of SELECT operation, 72
    - sample statements, 96, 98–99, 100–101, 146, 200–202
    - sorting rows with, 87–92, 87*f*
    - syntax, 87–88, 87*f*, 591*f*
  - SELECT statement
    - added or deleted columns and, 84
    - all columns, retrieval of, 83–84, 83*f*
    - clauses of, 73–75, 74*f* (*See also* FROM clause; GROUP BY; HAVING; SELECT clause; WHERE clause)
    - defining, 73
    - duplicate rows, eliminating, 84–86, 85*f*
    - embedded
      - in INNER JOIN, 254–256, 255*f*
      - sample statements, 278–282, 284–288
      - in OUTER JOIN, 301–304, 302*f*
      - sample statements, 320–322, 326–327, 329–331
    - keywords, 73
    - multiple columns, retrieval of, 81–83, 81*f*
    - ordering of columns, 82–83
    - ordering of rows, 87–92, 87*f*
    - overview of, 73
    - as part of SELECT operation, 72
    - sample statements, 93–102
    - saving, 92–93
    - subqueries in, 372–377
      - aggregate functions for, 375–377, 376*f*
      - GROUP BY in, 452–453, 461–463
      - sample statements, 395–499
      - syntax, 372–375
      - uses, 392
    - syntax, 74*f*, 77–78, 78*f*, 81*f*, 135*f*, 372*f*
    - transforming into DELETE statement, 572, 572*f*

- SELECT statement (*cont.*)  
 transforming into UPDATE statement, 505, 506*f*  
 translating requests into SQL, 77–81
- SEQUEL (Structured English Query Language), 54
- SEQUEL-XRM, 54
- SERIAL/ROWID data type, 109
- set(s)  
 defined, 214–215  
 members of (*See also* IN predicate)  
 characteristics of, 216–217  
 defined, 215  
 operations on, 215 (*See also* difference; intersection; union)
- set diagrams  
 difference, 225–227, 225*f*, 227*f*  
 intersection, 218–221, 219*f*, 220*f*, 235*f*  
 union, 231–232, 232*f*
- set identifier columns, 348, 357, 391
- set theory  
 difference in, 222–224  
 intersection in, 216–217  
 union in, 228–230
- simple primary keys, 39
- smallest value. *See* MIN
- SMALLINT data type, 108
- SOME, 386–389, 387*f*, 393–394, 598*f*
- spaces, in expression names, 130
- Specifying Queries as Relational Expressions (SQUARE), 55
- SQL  
 early vendor implementations, 55–56  
 future of, 65  
 history of, 53–65  
 importance of learning, 65–66  
 pronunciation of, 54
- SQL3, 62, 62*t*–64*t*
- SQL/86, 57–58, 59
- SQL/89, 58, 59
- SQL/92, 59–60
- SQL:1999, 65
- SQL:2003, 65
- SQL:2007, 65
- SQL Access group, 61
- SQL/Data System (SQL/DS), 56
- SQL/DS (SQL/Data System), 56
- SQL Server. *See under* Microsoft
- SQL Standard  
 alternative standards, 61–62  
 database models and, 4  
 data types in, 107–109  
 datetime expressions and, 127  
 development of, 56–58  
 evolution of, 58–64  
 extensions to, 60–61, 73  
 identifiers, types of, 22, 32  
 ORDER BY clause in, 88  
 parts of, 62, 62*t*–64*t*
- SQUARE (Specifying Queries as Relational Expressions), 55
- START TRANSACTION, 506
- static data, defined, 4
- Stonebraker, Michael, 56
- stored procedure, defined, 73, 92
- Structured English Query Language (SEQUEL), 54
- subqueries  
 for automatic primary keys generation, 543–544  
 as column expressions, 372–377  
 aggregate functions for, 375–377, 376*f*  
 GROUP BY in, 452–453, 461–463  
 sample statements, 395–499  
 syntax, 372–375  
 uses, 392  
 defined, 370  
 in DELETE statement, 573–575  
 as filters, 377–392  
 predicate keywords, 380–392  
 sample statements, 400–409  
 syntax, 378–380, 378*f*  
 uses, 393–394  
 row, 370–371  
 scalar, 370, 372, 373–375  
 within subqueries, 382–383  
 subquery UPDATE expressions, 514–516  
 table, 370, 371–372  
 types of, 370–372  
 in UPDATE statement, 508–514  
 uses for, 392–394  
 as value expression of aggregate function, 428  
 WHERE clause of, 373, 380

- SUBSTRING, 594f
- SUM, 416f, 421–423, 427–428, 433, 434, 595f  
     *See also* aggregate functions
- Sybase Enterprise Application Studio*, 6, 161
- SYMMETRIC BETWEEN predicate, 165
- syntax diagrams
- IN, 168f, 381f, 598f
  - aggregate functions, 416f, 595f
  - ALL, 487f, 598f
  - ANY, 487f, 598f
  - asterisk shortcut, 83f
  - AVG, 416f, 595f
  - CAST, 110f, 594f
  - character string functions, 594f
  - character string literals, 113f, 592f
  - column references, 246f, 593f
  - comparison predicates, 157f, 597f
  - concatenation expression, 118f
  - correlation names for table, 252f
  - COUNT/COUNT(\*), 376f, 416f, 595f
  - date expression, 125f
  - date functions, 594f
  - datetime functions, 594f
  - datetime literals, 115f, 592f
  - DELETE, 568f, 599f
  - DISTINCT, 85f
  - EXCEPT, 238f
  - EXISTS, 598f
  - filtering with comparison predicate, 378f
  - GROUP BY, 445f
  - HAVING, 476f
  - INSERT
    - using SELECT expression, 544f, 545f, 599f
    - with VALUES, 539f, 599f
  - INTERSECT, 236f
  - interval literals, 593f
  - IS NULL, 174f, 598f
  - JOIN, 596f
    - FULL OUTER, 314f
    - INNER
      - of three or more tables, 257f, 260f, 347f
      - of two tables, 247f, 257f, 305f
    - of more than two tables in alternating sequence, 260f, 308f
    - OUTER
      - of three or more tables, 306f, 308f
      - of two tables, 296f, 305f
      - using SELECT statements, 302f
      - of three or more tables, 257f, 260f, 306f, 308f, 347f
      - of two tables, 247f, 257f, 296f, 305f
    - UNION, 318f
  - LIKE predicate, 169f, 598f
  - literals, 113f, 114f, 115f, 592f
  - mathematical expressions, 121f
  - MAX, 376f, 416f, 595f
  - MIN, 416f, 595f
  - naming of expression, 130f
  - NOT, 176f, 185f
  - numeric functions, 595f
  - numeric literals, 114f, 592f
  - ORDER BY, 351f
    - BETWEEN predicate, 164f, 597f
    - quantified predicate, 387f
    - search conditions, 178f, 185f, 597f
  - SELECT expression, 373f, 591f
  - SELECT query, 87f, 591f
  - SELECT statement, 74f, 78f, 81f, 135f, 372f
    - with all clauses, 476f
    - embedded in INNER JOIN, 255f
    - OUTER JOIN using, 302f
    - UNION of two, 342f
  - SOME, 487f, 598f
  - subquery with IN predicate, 381f
  - SUM, 416f, 595f
  - time expressions, 126f
  - time functions, 594f
  - TIMESTAMP functions, 594f
  - UNION, 241f, 340f
    - ORDER BY clause, 351f
    - of three tables, 349f
    - of two SELECT statements, 342f
  - UPDATE, 502f, 599f
  - value expression, 134f, 373f, 592f
  - WHERE clause, 153f
- System R*, 5, 54–55
- Systems Application Architecture (SAA), 61
- T**
- tables
- base, defined, 9
  - correlation names (aliases), 252–254, 252f
  - derived, in INNER JOIN, 254–256, 255f

tables (*cont.*)

- fine-tuning of, 30–42
- INNER JOIN of, 247–251, 247*f*
- linking
  - advantages of, 14–15
  - defined, 13
  - defining of, 13–14
  - in resolving duplicate fields, 37–38, 37*f*, 38*f*
  - in resolving multivalued fields, 28, 29*f*
- logical, defined, 244
- naming of, 22, 30–32
- overview of, 6–7, 6*f*
- primary, defined, 11
- relationships between. *See* relationships between tables
- secondary, defined, 11
- structure, fine-tuning of, 32–33

table subqueries, 370, 371–372

TEXT data type, 107

TIME data type, 109

time expressions, 126–127, 126*f*

time functions
 

- in specific software, 607–614
- syntax, 594*f*

TIME keyword, 116, 117

time literals
 

- specifying, 116
- syntax, 115*f*, 592*f*

TIMESTAMP data type, 109

TIMESTAMP functions
 

- in specific software, 607–614
- syntax, 594*f*

TIMESTAMP keyword, 116

timestamp literals
 

- specifying, 116
- syntax, 115*f*, 592*f*

TINYINT data type, 108, 109

TOP, 92

totals. *See* SUM

transactions, 506–507, 554, 570

triggers, 42, 516

TRIM, 594*f*

true, value for, 507–508

tuple, 6. *See also* records

type of participation, setting, 46–47, 47*f*, 48*f*

## U

## UNION

- alternatives to, 352–353, 357–358
- column names in, 345
- commercial systems' support, 241
- of complex SELECT statements, 345–348, 347*f*
- CORRESPONDING clause, 343
- data type computability, 341–342
- multiple, 349–350, 349*f*
- overview, 39–342
- sample statements, 353–365
- set identifier columns and, 348, 357
- set requirements, 341
- of simple SELECT statements, 342–345, 342*f*
- sorting of, 351–352, 351*f*
  - sample statements, 354–355, 362–363
- syntax, 239–241, 241*f*, 340–341, 340*f*
- uses of, 215, 232, 233, 352–353

UNION ALL, 228–230, 341, 343, 345, 348

UNION JOIN, 317, 318*f*

union (set operation), 228–233
 

- combining result sets with, 230–232
- concept, 228–230
- limitations of, 233
- set diagrams, 231–232, 232*f*
- uses of, 215, 232–233

University of California, Berkeley, 5, 56

UNIX, SQL standards for, 61

unresolved relationships, defined, 13, 13*f*

UPDATE
 

- DISTINCT keyword and, 86
- JOIN in, 508–509, 511–512
- multiple columns, 507–508
- overview, 501–502
- sample statements, 503–504, 517–533
- selected rows, 504
- subquery UPDATE expressions, 514–516
- syntax, 502–503, 502*f*, 599*f*
- transactions and, 506–507, 554
- transforming SELECT statement into, 505, 506*f*
- translation process, 503
- uses for, 516–517
- verifying accuracy of target materials, 505

- WHERE clause, 502, 502*f*, 504–505, 506*f*, 507–508  
 subquery filters, 508–514
- USE, 538
- USING clause  
 alternatives to, 301  
 INNER JOIN, 247, 247*f*, 250–251  
 OUTER JOIN, 296*f*, 297, 300–301
- V**
- value expressions, 133–135, 134*f*  
 components of, 134  
 defined, 134  
 in SELECT clause, 135, 135*f*, 372–373, 373*f*  
 sample statements, 139–147  
 syntax for, 134, 134*f*, 373, 373*f*, 592*f*
- VALUES, INSERT statements with, 539–544  
 sample statements, 552–554, 556, 559–560  
 syntax, 539*f*, 599*f*
- VARCHAR data type, 107
- Venn, John, 219
- Venn diagrams, 219. *See also* set diagrams
- views  
 defined, 73, 92  
 overview of, 9–10, 11*f*
- Visual Studio*, 6
- W**
- “what if?” questions, 106
- WHERE clause, 151–156  
 in aggregate functions, 419, 430  
 alternatives to, 352–353, 357–358  
 composition of, 154–156  
 in DELETE statement, 568, 568*f*, 571–575  
 sample statements, 576–583  
 subqueries, 573–575  
 in multiple UNIONS, 350
- predicates (filters) in (*See also* BETWEEN; comparison predicates; IN predicate; IS NULL predicate; LIKE)  
 efficiency of, 190–191  
 expressing in different ways, 197–198  
 INTERSECT, 234–236, 236*f*  
 multiple conditions, 178–197  
 NOT operator, 175–178, 176*f*, 184–187, 185*f*  
 nulls, evaluation of, 193–197, 194*f*, 195*f*, 196*f*  
 AND operator, 179, 180*f*, 183–184, 195–196, 204–205  
 order of precedence. *See* order of precedence  
 OR operator, 180–182, 181*f*, 184–185, 196, 196*t*, 203  
 AND and OR together, 183–184  
 overview, 152–154  
 sample statements, 198–206  
 in UPDATE statements, 508–514  
 in SELECT statement, 74*f*, 75  
 of subquery, 373, 380  
 syntax of, 152, 153*f*  
 in UPDATE statement, 502, 502*f*, 504–505, 506*f*, 507–508  
 subquery filters, 508–514  
 uses of  
 generally, 151–152  
*vs.* HAVING, 478–485
- Wong, Eugene, 56
- X**
- X3, 57, 62  
 “X3.135-1992 Database Language SQL,” 59–60  
 X3H2, 57  
 X/OPEN standard, 61