

Index

Boldface text or page numbers indicate illustrations.

A

- Abend, 185
- Absolute value (%Abs), 191, 198
- Access paths, 56
- Accessing database files, 259–271
 - composite keys in, 267–270
 - end-of-file in, 260
 - File specification for, 282–284
 - first-in first-out access in, 259–265
 - key data structure (%Kds) function in, 269–270, 288
 - keys in, 259–265
 - legacy code and, 282–287
 - random access in, 265–267
 - random retrieval (Chain) in, 265–267
 - read a record (Read), 260–261
 - read equal key (Reade), 262–263
 - read prior equal record in (Readpe), 264–265
 - read prior record in (Readp), 264–265
 - result data structures in, 261
 - sequential access in, 259–265
 - set greater than (Setgt) in, 263–264, 287
 - set lower limit for (Setll), 261–262, 287
 - write to output file after (Write), 270–271
- Accumulation, 184
- Activation groups, 483–490, 491
 - caller's, 485
 - choosing the best, 486, **487**
 - default, 483–484
 - Delete Override (DLTOVR) and, 489
 - file open and, 489–490
 - file overrides and, 488–489, 491
 - open data paths and, 489–490
 - Open Database File (OPNDBF) and, 490
 - Original Program Model (OPM) and, 483
 - Override with Database File (OVRDBF) and, 488
 - shared opens and, 490
 - structure of, **487**
 - system-named, 484–485
 - user-named, 484
- ADD, 194
- Add duration (Adddur), 252, 253
- Add mode, in interactive applications, 392–395, 392–394, 399**
- Add Physical File Constraint (ADDPFCST), 391
- Addition, 182–183
 - to file, 283
 - sizing results for, 186
- Addmode subroutine, 399
- ADDPFCST, 391
- AIX, 5
- Algorithms, 14
- ALTER TABLE, 58
- Alternating format (Alt) keyword, 334–335
- Alwcpydta option, 312–313
- American Standard Code for Information Interchange. See ASCII*
- AND, 149, 389
- Application programming interfaces (APIs), 545–564
- Call a Prototyped Procedure or Program (Callp) operation in, 545
- calls, passing, and modular programming, 421
- CGI header for, 562
- Common Gateway Interface (CGI) and, 559
- Convert CGI input to DDS-formatted buffer (QtmhCvtDb), 559
- Copy compiler directive and copybooks for, 553
- Create User Space (QUSCRTUS) for, 548–550
- Display Command Line Window (QUSCMDLN), 546–547, **546**
- Execute Command (QCMDEXC), 547–548
- Get environment variable (QtmhGetEnv), 559
- List Database File Members (QUSLMBR), 548
- List Spooled File (QUSLSPL), 545
- parameters and parameter passing in, 545–546
- Parse QUERY_STRING (QzhhCgiParse), 559
- pointers in, 550
- Put environment variable (QtmhPutEnv), 559
- Read from Stdin (QtmhRdStin), 559
- Retrieve Pointers to User Space (QUSPTRUS), 550–553
- Retrieve Spooled File Attributes (QUSRSPLA), 545

- user space for, 548–553
 - web page creation using, with
 - HTTP, 559–563
 - wrapper procedures for, 554–559
 - Write to Stdout (QtmhWrStout),
 - 559, 560–562
 - Arguments, 36
 - Arithmetic*, 181–200. *See also Numeric data/data types*
 - arrays and, 339–340
 - date/time data and, 242–243, 255
 - Arrays, 323–371
 - Alternating format (Alt) keyword and, 334–335
 - arithmetic with, 339–340
 - assignment in, 337–339
 - calculations with, 337–350
 - Compile-time data (Ctdata) and, 331
 - compile-time, 331–334
 - crossfooting, 346, 369
 - as data structures, 325–329, 353–354, 368
 - defining, 324, 325
 - elements in, values, 324
 - externally described files and, 329–332
 - Fetch (SQL) and, 350–353, 368
 - get number of elements in (%Elem), 328, 369
 - indexes and, 336
 - initializing, 325
 - keyed, 368
 - keys for, keyed array data structure (*) and, 354–355
 - legacy code and, 355–368
 - look up an element in (%Lookup), 340–343, 355–358, 368, 369
 - multidimensional, 353–354
 - multiple-occurrence data structures (MODS) and, 366–368
 - number of elements in, using Dimension (Dim) keyword, 324
 - per record (Perrcd) keyword for, 332
 - preruntime, 358–359
 - related, defining, 334–336
 - runtime, 325, 358–359
 - separator lines in, 331
 - set/get portion of (%Subarra), 346–350, 369
 - sort elements of (Sorta), 345–346, 369
 - SQL and, 350–353, 368
 - as standalone variable as, 324, 368
 - sum elements of (%Xfoot), 346, 369
 - tables and, 323–324, 359–365
 - changing values in, 365
 - lookups in, 361–362
 - range, 364–365
 - related tables, 363–364
 - unequal %Lookup searches in, 343–345
 - values in, from a file, 324, 327–328
 - values in, not from a file, 324, 328–329
 - Arrival sequence, 56
 - AS/400, 4–5
 - ASCII, 67
 - Assign corresponding subfields (Eval-corr), 203–205
 - Automatic execution, INZSR
 - subroutine for, 164
- B**
- Batch processing, 3, 33, 373
 - Begin subroutine (Begsr), 162, 178
 - BIGINT, 62, 71–72, 121
 - Binary data, 65–66
 - Bind by copy, 452, 465, 472
 - Bind by reference, 470, 472, 490
 - Bind Service Program (BNDSRVPGM), 473
 - Binder language, 477, 478–482
 - Binding, 17, 19, 439
 - activation groups and*, 483–490. *See also Activation groups*, 483
 - bind by copy, 452, 465, 472
 - bind by reference and, 470, 472, 490
 - Bind Service Program (BNDSRVPGM) in, 473
 - binder language in, 477, 478–482
 - binding directories for, 455–457, 465, 471
 - Binding Directory (BNDDIR) in, 473
 - Create Binding Directory (CRTBNDDIR) in, 456, 466
 - Create Bound RPG Program (CRTBNDRPG) in, 17, 39, 50
 - Create Program (CRTPGM) in, 17
 - End Program Export List (ENDPGMEXP) in, 478–481
 - Export a Program Symbol (EXPORT) in, 478–481
 - procedures and, 451–457, 465
 - service programs and, 469–475, 491
 - Start Program Export List (STRPGMEXP) in, 478–481
 - static, 438
 - superglobal definition in, 482
 - unresolved imports and, 456, 466
 - Work with Binding Directory Entries (WRKBNDDIRE) in, 456, 466
 - Binding directories, 455–457, 465, 471
 - Binding Directory (BNDDIR) in, 473
 - Bits, 1, 66–67
 - Blanks/blank lines, 49, 50, 117–118
 - BNDDIR, 473
 - BNDSRVPGM, 473
 - Boolean data type, 31, 120
 - Bugs, 16
 - Building a program, 38
 - Built-in functions (BIFs), 36, 51, 182
 - arithmetic, 190–194
 - character data and, 209–219, 228, 229
 - data conversion, 192
 - date/time data and, 243–252
 - By-reference parameter passing, 414–415
 - By-value parameter passing, 457–459, 465
 - Byte size (%Size), 222, 229
 - Bytes, 66
- C**
- Calculation specification, 40, 45–47, 173
 - Call a Program (Call), 424
 - Call a Prototyped Procedure or Program (Callp) operation, 406, 409–410, 434, 435, 437, 465, 466, 545
 - Call interface, 407
 - Call stack**, 406, 406, 434
 - Calling external programs, 421
 - Calling parameters. See Calls, passing, and modular programming*
 - Calling programs, 405
 - Calls, passing, and modular programming**, 4, 451–457, 452, 453
 - application programming interfaces (APIs) in, 421
 - binding directories in, 455–457, 465
 - by-reference parameter passing in, 414–415
 - by-value parameter passing in, 457–459, 465
 - Call a Prototyped Procedure or Program (Callp) in, 406, 409–411, 434, 435, 437, 465, 466

- call interface for, 407
- call stack in, 406, **406**, 434
- compiler directives and, 460–466
- conditional compiler directives and, 461–463
- Const keyword for parameters in, 414
- Control Language (CL) or High Level Language (HLL) programs in, 421
- Copy and Include, 460–461, 465–466
- copybooks for, 460–461, 465–466
- Create Bound RPG Program (CRIBNDRPG) and, 451–457, 466
- Create Program (CRTPGM) and, 451–457, 465, 466
- Create RPG Module (CRTRPGMOD) and, 451–457, 465, 466
- data areas for, 421–424, 434–435. See also Data areas*
- Declare Parameters (Dcl-parm) in, 408
- Declare procedure interface (Dcl-pi) in, 410–411, 435
- Declare Prototype (Dcl-pr) in, 407–408, 435
- dynamic program calls in, 437
- example of, 418–420
- external program calls in, 421
- externally described files and, 459–460
- Extpgm for, 408
- file I/O, subprocedures and, 459–460, 465
- fixed-format legacy code and, 424–428, 463–464
- flow of control in, 406
- Inlr in, 416, 434
- Integrated Language Environment (ILE) and, 421, 438
- linear main programs in, 416
- literals or expressions as parameters in, Eval and, 414–415
- local data area (LDA) in, 422, 434
- LR to end program in, 417, 434
- maintaining modular programs and, with Update Program (UPDPGM), 457, **457**, 466
- naming the prototype for, 408
- Nomain modules and, 449–451, 465
- numeric data conversion example for, 421
- parameter descriptions in, 408
- parameters (arguments) in, 407–409, 434
- passing parameters and changing values in, 412–415
- procedure interface for, 410–412, 434
- procedures in. See Procedures*
- prototyping the call interface for, 407–409
- read-only reference parameter passing in, 413–415, 434
- recursive calls, recursion in, 406–407
- resolution process in, 437
- Return in, executing the program with, 38, 416–418, 435, 466
- reuse of code in, 405
- sample program showing, 428–434
- service programs and. See Service programs*
- templates for, 461
- using the modular approach in, 420–421
- variable scope in, 407
- wrapper procedures in, 554–559
- Case logic, 152–153
- Case sensitivity, 27
- CGI header, 562
- Chain, 265–267, 287, 288, 399
 - operation extender for, 276
- Change mode, in interactive applications, 392–395, 392–394, 399**
- Change Physical File (CHGPF), 81
- CHAR, 62, 68, 119–120
- Character data/data types, 65, 67–68, 201–231
 - Assign corresponding subfields (Eval-corr) in, 203–205
 - built-in functions for, 209–218, 228, 229
 - byte size (%Size) of, 222, 229
 - check characters (%Check, %Checkr) in, 217–218, 229
 - concatenation in, 206–207
 - continuation character (+) for, 201–202
 - conversion errors in, 503–505
 - convert other to (%Char), 218–219, 229, 246, 251, 256
 - convert to date (%Date), 246, 247–248, 256
 - convert to time (%Time), 246, 249–250, 256
 - convert to timestamp (%Timestamp), 246, 250, 256
 - current length in, 208
 - data-conversion functions for, 218–221
 - decimal positions (%Decpos) in, 224, 229
 - declared length in, 207–208
 - edit with edit code (%Editc), 219–221, 229
 - edit with edit word (%Editw), 221, 229
 - Eval and, 201–205, 228
 - Evaluate expression, right adjust (Evalr) in, 203–205
 - figurative constants and, 205
 - fixed- vs. variable-length strings of, 207–208
 - fixed-format legacy code and, 224–228
 - get/set length (%Len) of, 222–224, 229
 - get/set substrings in (%Subst), 211–212, 229
 - literals in, 202
 - move left (Move, Movel), 224–227
 - relational comparisons in, 147
 - replace character string in (%Replace), 212–213, 229
 - resulting indicators in, 228
 - retrieving data properties from, 222–224
 - scan and replace characters in (%Scanrpl), 213–214, 229
 - scan string (%Scan) in, 215–217, 229
 - simple assignment of values to, with Eval, 201–205
 - substrings in, 211–212
 - translate (%Xlate) in, 214–215, 229
 - trim leading/trailing blanks (%Trim), 209–211, 229
 - variable-length variables for, 207–208
- Character literals, 115
- CHECK, 386, 388
 - validity checking codes in, 387
- Check characters (%Check, %Checkr), 217–218, 229
- CHGPF, 81
- Close, 277, 288, 307–308, 319
- Closqlscr, 313
- CMD, 7
- Coding, 14, 16
- Colon (:): host variable indicator, 295
- Columns
 - compiled, 24
 - constraints on, 62–63
 - data type definition for, 62, 65–73
 - fullselect and, 74–75

- null values in, 62–63
 - SQL and, 57
 - Combined files, 378
 - Command Attention (CAnn), 383
 - Command Function (CFnn), 383
 - Command line, Display Command Line Window (QUSCMDLN), 546–547, 546**
 - Commands, 7
 - Execute Command (QCMDEXC), 547–548
 - Comments/commenting, 16, 24, 26, 27, 28, 49, 50**
 - divider lines in, 28
 - TEXT keyword and, 77
 - Commit, 313
 - Common Gateway Interface (CGI), 559
 - Common User Access (CUA), 400, 401
 - COMP, 386
 - Compile-time data (Ctdata), 331
 - Compile unit, 17
 - Compile-time arrays, 331–334
 - Compiler directives, 460–461, 465–466
 - for SQL, 317–318
 - Compilers and compiling, 2, 4, 16, 17, 19, 102
 - arrays and, compile-time, 331
 - bind by copy and, 452, 465
 - binding, 39
 - case insensitive nature of, 27
 - columns included in, 24
 - compiler directives for, 317–318, 460–461, 465–466
 - Copy directive for, 460–461, 465–466, 553
 - Create RPG Module (CRTRPGMOD) in, 17
 - externally described data structures and, 129–131
 - Include directive for, 460–461, 465–466
 - precompiler in, 313–314
 - preruntime arrays and, 358–359
 - procedures and, 439, 451–457
 - service programs and, 470–471
 - SQL and, 313–314
 - superglobal definition in, 482
 - Composite keys, 77
 - accessing database files and, 267–270
 - Compound operators, 184–185
 - Concatenation, 206–207
 - Concise operators, 38
 - Condition-controlled loops, 157
 - Conditional compiler directives, 461–463
 - Conditioning indicators, in interactive applications, 388–390, 401
 - Const keyword, 414
 - Constants, 99–100
 - figurative, 117–118
 - named*. See *Named constants*
 - Constraints, 62–63
 - interactive applications and, 391, 401
 - Continuation character (+), 201–202
 - Control break logic, 164–173, 165, 166, 177**
 - Control Language (CL), 421
 - Control options, 23, 26, 27, 28, 50, 438**
 - device name, 29
 - file, 29
 - file usage and Usage keyword in, 30
 - filenames in, 29
 - fixed-format programming and, 42
 - overflow indicator in, 30
 - standalone variable, 30–32
 - Convert CGI input to DDS-formatted buffer (QtmhCvtDb), 559
 - Convert other data to date (%Date), 246, 247–249, 256
 - Convert other data to time (%Time), 246, 249–250, 256
 - Convert to character (%Char), 218–219, 229, 246, 251, 256
 - Convert to decimal data (%Dec), 246, 251–252, 256
 - Convert to packed decimal (%Dec), 192–193, 198
 - Convert to packed decimal with half-adjust (%Dech), 193, 198
 - Convert to unsigned integer (%Uns/Unsh), 198
 - Copy directive, 553, 460–461, 465–466
 - Copybooks, 553
 - Count-controlled loops, 157
 - Counting, 184
 - CR to denote negative numbers, 13, 85–86
 - Create Binding Directory (CRTBNDDIR), 456, 466
 - Create Bound RPG Program (CRTBNDRPG), 17, 39, 50, 451–457, 466
 - Create Data Area (CRTDTAARA), 422
 - Create Display File (CRTDSPF), 374
 - Create Index, 75
 - CREATE INDEX*, 58. See also *Data Definition Language (DDL) statements*.
 - Create Logical File (CRTLF), 81, 102
 - Create Physical File (CRTPF), 81, 102
 - Create Printer File (CRTPRTF), 82
 - Create Program (CRTPGM), 17, 451–457, 465, 466
 - Create RPG Module (CRTRPGMOD), 17, 451–457, 465, 466
 - Create Service Program (CRTSRVPGM), 470–471, 471, 490**
 - Create SQL ILE RPG Object (CRTSQLRPGI), 313, 319
 - CREATE TABLE, 58, 61
 - Create User Space (QUSCRTUS), 548–550
 - CREATE VIEW, 58
 - Create Views, 73–75
 - CRTBNDDIR, 456, 466
 - CRTBNDRPG, 17, 39, 50, 451–457, 466
 - CRTDSPF, 374
 - CRTDTAARA, 422
 - CRTLF, 81, 102
 - CRTPF, 81, 102
 - CRTPGM, 17, 451–457, 465, 466
 - CRTPRTF, 82
 - CRTRPGMOD, 17, 451–457, 465, 466
 - CRTSQLRPGI, 313, 319
 - CRTSRVPGM, 470–471, 471, 490**
 - Ctl-opt, 51
 - Currency, 10, 13
 - fixed vs. floating symbols in, 13
 - Current length, of character data, 208
 - Current schema, 65
 - Cursors, SQL, 303–308, 318
 - Close, 307–308
 - Closqlcsr in, 313
 - Declare, 303
 - DELETE and, 308
 - Fetch, 306–307
 - Open, 306
 - positioned updates in, 308
 - Read Only clause and, 304, 305
 - scrollable type, 305–306
 - SELECT and, 304
 - serial type, 305
 - UPDATE and, 304, 308
 - Cycle main program, 32, 50
 - procedures and, 443–446, 465
 - Cycle of program development, 13–16, 19
- D**
- Data areas, 7, 434–435
 - calls, passing, and modular programming, 421–424
 - Create Data Area (CRTDTAARA) in, 422
 - data structures for, 422–423, 427–428, 435

- exclusive locks in, 423
- In, Out, and Unlock in, 423–424, 435
- local (LDA), 422, 434
- locking in, 423
- Submit Job (SBMJOB) and, 422
- Data-conversion functions, 192
 - character data and, 218–221
 - date/time data and, 246–252
- Data Definition Language (DDL) statements, 58
- Data Description Specifications (DDS), 26, 57, 75–81, 102
 - composite keys and, 77
 - data types and, 77
 - date/time data and, 234–235
 - display files and, 374
 - externally described printer files and, 82–85
 - field reference files in, 87–88, **87**
 - file creation with, 81
 - interactive applications and, 382–390, 401
 - join fields in, 80–81
 - keywords and, 76
 - logical files and, 78–81
 - output editing and, 85–87
 - physical files and, 76–78
 - relational operators in, 80
 - SQL and, 294
- Data files, 7
- Data item name, 44, 118, 139
- Data Manipulation Language (DML)*, 58, 291–294, 318. *See also SQL*
- Data passing, 4
- Data paths
 - service programs and, 489–490
 - shared open
- Data structures
 - arrays as*, 353–354, 368. *See also Arrays*
 - data area, 422–423, 427–428, 435
 - date/time data and, 235–236
 - declarations and, 107, 124–132, 142
 - error management and, 505–507
 - externally described, 129–131
 - indicator, 380–381
 - initializing, 126–127
 - interactive applications and, 380–381
 - key, (%Kds) function in, 269–270, 288
 - keyed array, 354–355
 - Like, Likeds, Likerec keywords in, 132–135, 142
 - multiple-occurrence (MODS), 366–368
 - overlapping subfields in, 127–129
 - program status, for error control, 508–509, 513
 - qualified, 131–132
 - return codes and, 300–303
 - SQL Communication Area (SQLCA), 300
 - subfields and, 124, 125–126
 - variable organization and, 124
- Data Studio for SQL, 60**
- Data types, 4, 7, 10, 11, 18, 24, 31–32, 65–73, 102**
 - ASCII and, 67
 - Boolean, 31, 120
 - character*, 65, 67–68. *See also character data/data types*
 - character literals and, 115
 - columns and, 62, 65–73
 - Data Description Specifications (DDS) and, 77
 - date, 65
 - date and time*, 65, 72–73, 122. *See also Date/time*
 - declarations and, 119–122
 - Definition specification and, 140–141
 - EBCDIC and, 66–67, 69
 - fixed-format programming and, 45, 94–95
 - hexadecimal notation and, 66–67
 - high- vs. low-order bits in, 66
 - indicator (Boolean), 31, 120
 - indicators and, 120
 - Like, Likeds, Likerec keywords in, 132–135, 142
 - numeric*, 65, 69–72. *See also numeric data/data types*
 - numeric literals and, 114
 - packed, 70–71, 120–121
 - standalone variables and, 119–122
 - timestamp*, 122. *See also Date/time*
 - typed literals and, 115
 - unsigned integers in, 121
 - zoned, 70–71, 120–121
- Data validation. See Validating data*
- Data-area data structures, 422–423, 427–428, 435
- Databases/database files, 3, 7, 8, 55–56
 - accessing. See Accessing database files*
 - adding records to, using Except (Calculation time output) in, 284–286
 - Data Manipulation Language (DML) in*, 291–294, 318. *See also SQL*
 - file I/O resulting indicators, 286–287
 - I/O operations. *See Updating database files*
 - List Database File Members (QUSLMBR), 548
 - locking files and records in, 275–276
 - maintenance of, 272
 - open/close file considerations in, 276–277
 - SQL and. See SQL updating files in. See Updating database files*
 - composite keys in, 77
 - Data Definition Language (DDL) statements in, 58
 - Data Description Specifications (DDS) in, 57, 75–81
 - data dictionary and, 88
 - Data Manipulation Language (DML) statements in, 58
 - data types in, 65–73, 77
 - Embedded SQL Host Language statements in, 58
 - externally described files in, 56
 - externally described printer files for, 82–85
 - field reference files in, 87–88, **87**
 - fields in, 55, 57
 - files in, 55
 - files in, with DDS, 81
 - IBM i operating system and, 55–56
 - indexes in, 75
 - join fields in, 80–81
 - keywords and, 76
 - legacy data/programs and. See Fixed-format and legacy programming*
 - level checking and, 81
 - library in, 57
 - logical files in, 57, 78–81
 - naming conventions and, 64
 - output editing in, 85–87
 - physical files in, with DDS, 76–78
 - program-described files in, 55–56, 88–91
 - qualified names in, 64
 - record format names in, with RCDFMT, 63–64
 - records in, 55, 57, 76
 - relational operators in, 80
 - SQL and*, 57–65. *See also SQL*
 - SQL development tools for, 58–60
 - SQL Procedural Language (SPL) statements in, 58
 - SQL vs. IBM i terminology in, 57
 - views in, 73–75
- DATE, 62, 72, 388

- Date/time, 65, 72–73, 122, 233–258
 add duration in (Adddur), 252, 253
 arithmetic with, 242–243, 255
 built-in functions for, 243–252
 compile-time arrays and, 331–334
 conversion errors in, 503–505
 convert other data to (%Date), 246, 247–249, 256
 convert other data to time (%Time), 246, 249–250, 256
 convert other data to timestamp (%Timestamp), 246, 250, 256
 convert to character (%Char), 246, 251, 256
 convert to decimal (%Dec), 246, 251–252, 256
 data-conversion functions with, 246–252
 Data Definition Specifications (DDS) to define, 234–235
 data structures for, 235–236
 default values for, 234
 defining, 233
 difference between (%Diff) and, 243, 244–245, 256
 duration in, and duration functions, 242
 error management and, 502–503
 Eval and, 255, 256
 extract (Extret), 252
 extract portion of (%Subdt) in, 243, 245–246, 256
 figurative constants for, 240–241
 formats for, 236–238, 255
 ILE RPG and, defining, 235–236
 input specifications for, 236
 interactive applications and, 388
 ISO standard formats for, 236–238
 legacy code and, 252–255
 move/move left (Move, Movel), 254–255
 native vs. legacy dates in, 72
 relational comparisons in, 147–148
 reserved values for, 123
 separator characters in, 237
 simple assignment of, 239–241
 SQL to define, 234
 standalone variables for, 235–236
 subfields for, 235–236
 subtract duration in (Subdur), 252, 253–254
 timestamps in, 73, 233
 typed literals for, 239–240, 255
- Dcl-f, 51
 Dcl-s, 51
 Dcl-xx declaration instructions, 108, 142
- Debugging, 16–17, 18, 19. See also Error management**
 control options for, 28
 DECIMAL, 62, 70–71, 121
Decimal data. See also Radix
 addition, sizing results for, 186
 character data and, (%Decpos) in, 224, 229
 convert to (%Dec), 246, 251–252, 256
decimal points, 10. See also Radix characters
 division, sizing results for, 187
 error management and, 501–502
 maximum digits rule in, 190
 multiplication, sizing results for, 187
 overflow in, 185–186
 position, decimal position (%Decpos), 93–94, 141, 224, 229
 result decimal positions rule in, 190
 subtraction, sizing results for, 186–187
 truncation of, 185–186
- Decision logic, 4
Declarations, 23, 26, 27, 28–32, 50, 107–144, 438
 assigning initial values to data using, 122–124
 character literals in, 115
 data item name in, 118
 data structures and, 107, 124–132, 142
 data types in, 119–122
 Dcl-xx instructions in, 108, 142
 devices in, 109
 file keywords in, 110–113
 file usage in, 109–110
 filenames in, 108, 136
 files and, 107, 108–113, 141
 legacy code and, 136–141, 142
 named constants and, 107, 113–118, 142
 numeric literals in, 114
 override and Extxx keywords in, 112–113
 procedure interfaces and, 107
 procedures and, 440, 465, 464
 prototypes and, 107
 standalone variables and, 107, 118–122, 142
 typed literals in, 115
- Declare, 319
 DECLARE CURSOR, 58, 303
 Declare Parameters (Dcl-parm), 408
- Declare procedure (Dcl-proc), 439–440, 465, 466
 Declare procedure interface (Dcl-pi), 410–411, 435
 Declare Prototype (Dcl-pr), 407–408, 435
 Declared length, of character data, 207
 DEFAULT, 62, 63
 Default values, date/time data and, 234
 Definition specification, 39–40, 92, 139–141
 data item name in, 139
 data type and, 140–141
 decimal position in, 141
 definition type in, 139–140
 external description in, 139
 fixed-format programming and, 44–45
 keywords in, 141
 length in, 140
 Definition type, in fixed-format programming and, 45
 Delete/DELETE, 58, 274–275, 287, 288, 291, 293–294, 319
 cursors, SQL and, 308
Delete mode, in interactive applications and, 392–395, 392–394, 399
 Delete Override (DLTOVR), 489
 Deploying service programs, 471–475
- Design
 hierarchical decomposition in, 161–162
 modular, 162, 438–439
 spaghetti code and, 379
structured. See Structured design
 top-down, 161–162, 177
- Detail in report, 25–26, 26, 25**
 Development tools, SQL, 58–60
 Device files, 57, 109, 284
 device files for, 57
 device names, 29, 44
 fixed-format programming and, 44
- Dictionary, in documentation, 49
 Difference between dates/times (%Diff), 243, 244–245, 256
 Dimension (Dim) keyword, 324
 Display Attribute (DSPATR), 385
 Display attributes, 385
Display Command Line Window (QUSCMDLN), 546–547, 546
Display files, 373, 374–378, 374, 400–401
 combined, 378
 Create Display File (CRTDSPF) for, 374
 Data Description Specifications (DDS), 374, 382–390, 401

- declaring, 378–379
 - display attributes in, 385
 - documentation in, 378
 - field names in, 376
 - field-level entries in, 377
 - file-level entries in, 377
 - function keys in, 378
 - I/O and, 375
 - Indicator Area (INDARA) in, 378
 - keyboard shift attribute in, 377
 - record-level entries in, 377–378
 - reference files in, 376
 - Screen Design Aid (SDA) for, 374
 - Display Message (Dsply), 446, 466
 - Display Program (DSPPGM), 454, 466, 474, 474**
 - Display Service Program (DSPSRVPGM), 479
 - Display Spooled File (DSPFFD), 10, 12**
 - DIV, 194
 - Divider lines, in comments, 28
 - Division, 182–183
 - Integer division (%Div) in, 191, 198
 - Remainder (%Rem) in, 191–192, 198
 - sizing results for, 187
 - DLTOVR, 489
 - Do Until. See Dou*
 - Do while. See Dow*
 - Do/Enddo, 174
 - Documentation, 26, 27, 48–49**
 - // delimiter for, 49
 - blank lines in, 49
 - commenting/comments in, 16, 24, 49
 - dictionary as, 49
 - display files and, 378
 - internal vs. external, 16
 - maintenance programming and, 48
 - overview type, 48–49
 - revisions in, 49
 - Dou, 156–157, 157, 175, 177, 178**
 - file processing with, 160–161
 - Dow, 35–37, 51, 155–156, 157, 146, 175, 177, 178**
 - file processing with, 160–161, 160
 - DSPFFD, 10, 12**
 - DSPPGM, 454, 466, 474, 474**
 - DSPSRVPGM, 479
 - DTAARA, 7
 - Duration, in date/time data, 242
 - Dynamic program calls, 437
 - Dynamic SQL, 308–312, 319
 - Execute, 310–311, 318
 - Execute Immediate, 311–312
 - parameter values in, 308
 - Prepare, 309–310
- E**
- E Extender, 493–497
 - Early exits from looping, 159–160
 - EBCDIC, 66–67, 69
 - relational comparisons in, 147
 - Edit Code (EDTCDE), 384–385, 384
 - Edit codes, 85–86, 85, 100, 102**
 - edit with (%Editc), 219–221, 229
 - interactive applications and, 384–385
 - Edit with edit code (%Editc), 219–221, 229
 - Edit with edit word (%Editw), 221, 229
 - Edit Word (EDTWRD), 384–385
 - Edit words, 86–87, 102
 - edit with (%Editw), 221, 229
 - interactive applications and, 384–385
 - Editing/output editing, 85–87
 - edit codes in, 85–86, 85, 100, 102
 - edit words in, 86–87, 102
 - negative numbers/values, CR and, 85–86
 - zero suppression and, 85–86
 - Elements, in arrays, 324
 - Else/Elseif, 146, 150, 152–153, 153, 178**
 - Embedded SQL, 294
 - Embedded SQL Host Language
 - statements, 58
 - Emulator, 5250, 374
 - End Monitor Group (Endmon), 498–500
 - End-of-file (EoF) condition, 33–34, 51, 137, 260, 283, 288
 - End position, 100
 - End procedure (End-proc), 439–440, 465, 464
 - End Program Export List (ENDPGMEXP) in, 478–481
 - End subroutine (Endsr), 162, 178
 - End/Enddo, 35, 36–37, 51, 178
 - End/Endif, 35, 36–37, 51
 - ENDPGMEXP, 478–481
 - Entry Plist, 426–427
 - Eof. See End-of-file*
 - EQ, 80, 175
 - Equal, 288
 - Equal to operator (=), 146
 - Error logs, updating database files and, 279
 - Error management, 19, 493–518
 - capturing opcode errors in, 493
 - data-conversion errors and, 503–505
 - data errors and, 500–505
 - data structures for, 505–507
 - data validation in, 385–388
 - date errors in, 502–503
 - debugging and, 16–17, 18
 - decimal data errors and, 501–502
 - E Extender in, 493–497
 - End Monitor Group (Endmon) in, 498–500
 - error logs for, 279
 - Error Message (ERRMSG) in, 388
 - interactive applications and, 388
 - legacy code and, 513–515
 - logic errors and, 16–17, 19
 - Monitor for, 498–500
 - On-error operations for, 498–500
 - %Error in, 495–496
 - program status data structure for, 508–509, 513
 - PSSR subroutine for, 509–510
 - resulting indicators for, 513
 - SQL (Sqlcode and Sqlstate) and, 302–303, 511–513
 - Status of file or program (%Status) in, 496–497
 - subroutines for, 505, 507–508
 - syntax errors and, 16–17, 19
 - Whenever statement in, 514–515
- Error Message (ERRMSG), 388
- %Error, 495–496
- eServer, 5
- Eval, 35, 37–38, 50, 51, 181–185, 198, 201–205
 - character data and, 228
 - date/time data and, 255, 256
 - parameter passing with, 414–415
 - precision in, 189–190
- Evaluate expression, right adjust (Evalr), 203–205
- Except, 101, 102–103, 287, 288
- Except (Calculation time output), 284–286
- Exception lines/names, 96
- Exclusive locks, 423
- Exec SQL directive, 294–295, 319
- Execute, 310–311, 318
- Execute Command (QCMDEXC), 547–548
- Execute Format (Exfmt), interactive applications and, 379–380, 401
- Execute Immediate, 311–312, 319
- Execute subroutine (Exsr), 163, 178
- Exponentiation, 182–183
- EXPORT, 478–481
- Export a Program Symbol (EXPORT), 478–481
- Extdesc keyword, 112
- Extended Binary Coded Decimal Interchange Code. See EBCDIC*
- Extended factor 2 operands, in fixed-format programming, 47

External description, 139
 External documentation, 16
 Externally described data structures,
 129–131
 Externally described files, 56, 101
 arrays and, 329–332
 printer, 82–85
 subprocedures for, 459–460
 Extfile, 112
 Extmbr, 112
 Extpgm, 408, 442
 Extract date/time (Extret), 252, 254
 Extract portion of date/time (%Subdt),
 243, 245–246, 256

F

Factor 2 operands, in fixed-format
 programming, 46–47
 Factors, 35
 Fetch, 58, 306–307
 arrays and, 368
 Field description entries, 92–93
 Field-level keywords, 76, 384–388
 Fields, 7, 8–10, 19, 25, 55, 57, 288
 data dictionary and, 88
 data types in, 10, **11**
 display files and, 376, 377
 display of, in interactive program,
 399–400
 field description entries in, 92–93
 field reference files and, 87–88, **87**
 join fields in, 80–81
 key fields and, 56
 keywords for, in DDS, 76
 length of, 9–10, 10, **11**
 naming, 31, 94, 99
 position of, positional notation for,
 10, **11**
 result field definition in, 196
 selecting, 56
 size of, 185–186
 sub-, 124, 125–126
work. See Standalone variables
 Figurative constants, 117–118
 assigning values to, 182
 character data and, 205
 date/time data and, 240–241
 FILE, 7
 File addition, 283
 File declaration, 29
 File designation, 282
 fixed-format programming and, 43
 File format, 43–44
 File I/O
 resulting indicators and, 286–287
 subprocedures and, 459–460, 465
 File organization, 283

File specification, 39–40, 91–92,
 136–138, 282–284
 accessing database files with,
 282–284
 device files in, 284
 end-of-file in, 137, 283
 file addition in, 137, 283
 file designation in, 137, 282
 file format in, 137, 283
 file organization and, 138, 283
 file type in, 136–137
 filename in, 136, 282
 fixed-format programming and, 43
 key length in, 138, 283
 key type in, 138
 keywords in, 138, 284
 record address in, 283
 record length in, 138, 283
 File-level keywords, 76
 interactive applications and,
 382–383
 Filenames, 25, 29, 108, 136, 282
 fixed-format programming and, 43
*Files, 7, 8–10, 19, 55–105. See also
 Databases; Externally described
 files*
 access paths for, 56
 adding to, 137
 arrival sequence of records in, 56
 closing, 276–277
 combined I/O, in display files, 378
 Data Description Specifications
 (DDS) and, 81
 data dictionary and, 88
*database concepts and. See
 Databases*
 declaration for, 29, 107, 108–113,
 141
 designation of, 137
 device, 29, 44, 57, 109
 end-of-file in, 137
 externally described, 56, 101,
 329–332
 externally described printer, 82–85
 field reference, 87–88, **87**
 filenames in, 29, 136
 fixed-format programming and, 43
 format of, 137
 hierarchy/organization in, **9**
 input, 30
 key fields in, 56
 key sequence of records in, 56
 keywords for, 76, 110–113
 level checking and, 81
 locking, 275–276
 logical, 56, 101
 members in, 8
 open data paths and, 489–490

Open Database File (OPNDBF)
 and, 490
 opening, 276–277
 organization of, 138
 output, 30
 overflow indicator in, 30
 override and Extxx keywords in,
 112–113
 physical, 56, 101
 pointers to, 56
 program-described, 55–56, 88–91,
 101
 record format in, 25
 record length in, 138
 renaming, Rename keyword in,
 111–112
 selecting fields in, 56
 selecting records in, 56
*SQL database concepts and,
 57–65. See also Databases*
 type in, 43, 136–137
 Usage keyword, declaration of, 30,
 109–110, 141
 First-in first-out access, 259–265
 Fixed currency symbols, 13
 Fixed-format and legacy programming,
 4, 5, 6, 18, 39–48, 50, 88–101
 accessing database files in,
 282–287
 arithmetic and, 194–197
 arrays and, 355–368
 Calculation specifications in, 40,
 45–47, 173
 Call a Program (Call) in, 424
 calls, passing, and modular
 programming, 424–428
 character data and, 224–228
 compiler directives for SQL in,
 317–318
 conditioning indicators in, 176–177
 control options/control
 specifications in, 42
 data area data structures in,
 427–428
 data item names in, 44
 data type in, 45
 date/time data and, 72, 252–255
 declarations and, 136–141, 142
 Definition specification in, 39–40,
 44–45, 92, 139–141, 424
 definition type in, 45
 devices in, 44
 Do/Enddo in, 174
 Entry Plist in, 426–427
 error management and, 513–515
 Except (Calculation time output)
 in, 284–286
 extended factor 2 in, 47

- factor 2 operations in, 46–47
 - file designation in, 43
 - file format in, 43–44
 - file I/O resulting indicators, 286–287
 - File specification in, 39–40, 43, 91–92, 136–138, 282–284. See also File specification*
 - file type in, 43
 - filename in, 43
 - free form area in, 40
 - free-format programming vs., 47–48
 - Header specification in, 39–40
 - Input specification in, 39–40, 92–95
 - keywords in, 44, 45
 - lookup operations in, 355–358
 - numeric operators in, 194–196
 - operation code in, 46
 - order or sequence of specifications in, 40
 - output layouts/specifications in, 95–101
 - Output specification in, 40, 95–101
 - preruntime arrays in, 358–359
 - Procedure Boundary (P)
 - specification in, 40, 463–464
 - procedure interface in, 425
 - procedures and, 463–464
 - program status data structure in, 513
 - program workflow and, 173–177
 - program-described files and, 88–91, 101
 - prompts in, 41
 - prototypes in, 425
 - relational comparisons in, 174–176
 - result field definition in, 196
 - resulting indicators in, 196–197, 228, 513
 - ruler lines in, 41
 - semicolon delimitator in, 42
 - specifications in, 39–42
 - SQL and, 317–318
 - Whenever statement in, 514–515
 - Fixed-length character string, 68, 119–120, 207–208. See also Character data type; CHAR*
 - Fixed-logic cycles, 2
 - FLDA/FLDC, 389
 - Floating currency symbols, 13
 - Floats, 13
 - For, 146, 157–159, 178
 - Format of file, 283
 - Formatting, 12
 - Found, 288
 - Free-format programming/logic, 4, 5, 6, 18, 39, 50
 - columns in, 24
 - fixed-format vs., 47–48
 - semicolon as terminator character in, 24, 42
 - variable declaration in, 24
 - Fullselect, 74–75
 - Function keys, in display files, 378
 - Functions, 51. See also Built-in functions*
- G**
- GE, 80, 175
 - Get environment variable (QtmhGetEnv), 559
 - Get number of elements in (%Elem), 328, 369
 - Get/set length (%Len), 222–224, 229
 - Get/set substrings in (%Subst), 211–212, 229
 - Global variables, 407, 445
 - Greater than/Greater than or equal to (>, >=), 146
 - GT, 80, 175
- H**
- Header specification, 39–40
 - Headers/headings in report, 25–26, 26, 96**
 - Hexadecimal notation, 66
 - Hierarchical decomposition, 161–162
 - Hierarchy of library, file, member, 8, 9**
 - High Level Language (HLL), 1–2, 421
 - High-order bits, 66
 - History of RPG, 2
 - AS/400 computer in, 4–5
 - eServer and, 5
 - ILE RPG in, 4–5
 - Internet based applications and, 5
 - iSeries and, 5
 - OS/400 and, 4–5
 - Power Systems servers and, 5
 - procedural language character of, 2
 - punch card programming in, 2, 14
 - Report Program Generator (RPG)
 - early development in, 2
 - RPG II in, 2–3
 - RPG III in, 3–4
 - RPG/400 in, 4
 - System/3 computer and, 3
 - System/38 minicomputers and, 3
 - Systemi, 5
 - Host structures, 297–298
 - Host variables, 295–296, 318
 - host structures and, 297–298
 - indicator variables vs., 298
 - null values and, 298–300
 - Select Into, 296–297
 - HTTP, web page creation, 559–563
- I**
- I/O operations, 272
 - I/O resulting indicators, 286–287
 - i5/OS operating system, 5
 - IBM Data Studio client and SQL, 60**
 - IBM i operating system and databases, 5, 55–56
 - If, 35, 36–37, 51, 146, 148–150, 151, 175, 178**
 - nested, 151–152, 152
 - ILE RPG, 6
 - case-insensitive nature of, 27
 - development history of, 4–5
 - SQLRPGLE programs in, 313–314
 - In, 423–424, 435
 - Include, 460–461, 465–466
 - INDARA, 378, 382
 - Indexes, 75, 102
 - arrays and, 336
 - SQL and, 57
 - Indicator Area (INDARA), 378, 381
 - Indicator data structures, 380–381
 - Indicator structures, 299–300
 - Indicator variables, 2, 120
 - host variables vs., 298
 - Indicators
 - conditioning, in legacy code, 176–177
 - data type, 31
 - relational comparison using, 148
 - resulting, 196–197
 - Initialization
 - arrays and, 325
 - data structures, 126–127
 - initializing keyword, variable, 32
 - subfield, 126–127
 - variables, 122–124
 - Inlr, 434
 - Input files, 30
 - Input specification, 39–40, 55, 92–95
 - data types in, 94–95
 - date/time data and, 236
 - decimal positions in, 93–94
 - field description entries in, 92–93
 - field name, 94
 - record identification entries in, 92–93
 - INSERT, 58, 291, 292–293, 319
 - INTEGER, 62, 71–72
 - Integer division (%Div), 191, 198

- Integrated development environments (IDEs), 14
- Integrated Language Environment (ILE), 4
- calls, passing, and modular programming, 421
 - procedures and, 438
- Interactive applications, 3, 373–403
- Add mode in, 392–395, **392–394**, 399
 - Add Physical File Constraint (ADDPFCST) in, 391
 - Addmode subroutine in, 399
 - AND and OR in, 389
 - basic program example for, 381–382
 - batch processing vs., 373
 - Chain operation in, 399
 - Change mode in, 392–395, **392–394**, 399
 - CHECK in, 386, 387, 388
 - combined I/O files in, 378
 - Command Attention (CAnn) and Command Function (CFnn) keywords in, 383
 - Common User Access (CUA) in, 400, 401
 - COMP in, 386, 387
 - conditioning indicators in, 388–390, 401
 - constraints in, 391, 401
 - Create Display File (CRTDSPF) for, 374
 - Data Description Specifications (DDS) and, 374, 382–390, 401
 - data validation in, 385–388
 - DATE in, 388
 - Delete mode in, 392–395, **392–394**, 399
 - Display Attribute (DSPATR) in, 385
 - display attributes in, 385
 - display files in, 373, 374–378, **374**, 400–401
 - display files in, declaring, 378–379
 - Edit Code (EDTCDE) in, 384–385
 - Edit Word (EDTWRD) in, 384–385
 - emulator, 5250 emulator in, 374
 - Error Message (ERRMSG) in, 388
 - Exfmt (Execute Format) in, 379–380, 401
 - field-level keywords in, 384–388
 - fields displayed in, 399–400
 - file-level keywords in, 382–383
 - FLDA/FLDC in, 389
 - I/O and, 375
- Indicator Area (INDARA) in, 378, 382
- indicator data structures in, 380–381
 - interactive file maintenance in, 391–400
 - keywords in, 382–388, 401
 - locking records in, 399
 - looping in, 379
 - modules and organization in, 395
 - OVERLAY in, 383
 - pseudocode in, 395
 - RANGE in, 386
 - Read in, 379, 401
 - record-level keywords in, 383–384
 - REF in, 382
 - Reset operation in, 399, 401
 - sample code for, 395–398
 - Screen Design Aid (SDA) for, 374
 - screen design in, 400
 - screen I/O in, 379–380
 - Setmode subroutine in, 399
 - subroutine organization in, 395
 - TIME IN, 388
 - updating data in, 391
 - Valid Command Key (VLDCMDKEY) keyword in, 383
 - validating data in, 392, 401
 - validity checking codes in, 387
 - VALUES in, 386
 - Write in, 379, 401
- Interactive SQL facility, 59
- Internal documentation, 16
- Internet-based applications, 5
- Inz keyword, 123–124
- INZSR subroutine, 164
- iSeries, 5
- ISO standard date/time formats, 236–238
- Iter, 178
- iteration (repetition) operators, 146, 155–159
- J**
- Java, 5
- Join fields, 80–81
- K**
- Key data structure (%Kds) function, 269–270, 288
 - Key fields, 56
 - Key sequence, 56
 - Keyboard shift attribute i, 377
 - Keyed keyword, 111
 - Keyed array data structures, 354–355
- Keyed arrays, 368
- Keys
- accessing database files and, 259–265
 - arrays and, 354–355
 - composite, 267–270
 - Data Description Specifications (DDS) and, 77
 - key data structure (%Kds) function in, 269–270, 288
 - length of, 138, 283
 - partial composite, 268–269
 - read equal key (Reade) in, 262–263
 - set greater than (Setgt) in, 263–264, 287
 - set lower limit for (Setll), 261–262, 287
 - type of, 138
- Keywords, 138, 284
- conditioning indicators in, 388–390, 401
 - Data Description Specifications (DDS) and, 76
 - Definition specification, 141
 - field-level, 384–388
 - file declaration, 110–113
 - file-level, 382–383
 - fixed-format programming and, 44, 45
 - initialization using, 123–124
 - interactive applications and, 382–388, 401
 - record-level, 383–384
 - subfile programming and, 523–524
- L**
- Last Record indicator, 38
- LE, 80, 175
- Leading decision loops, 156
- Learning RPG, 6
- Leave, 178
- Leave subroutine (Leavesr), 163–164, 178
- Legacy code. See Fixed-format and legacy programming*
- Legacy dates, 72
- Length notation, 10, 11**
- Length, Definition specification, 140
- Less than/Less than or equal to (<, <=), 147
- Level checking, 81
- Libraries, 7, 19, 57, 64
- hierarchy within, 8, 9
 - library lists and, 7, 65
 - library lists in, 65
 - separator (//) for, 7

Like, Likeds, Likerec keywords, 132–135, 142

Linear main programs, 32, 416
procedures and, 446–449

Linux, 5

List Database File Members (QUSLMBR), 548

List panels, 519

List Spooled File (QUSLSPL), 545

Literals, 99–100
character data and, 115, 202
date/time data and, 239–240, 255
numeric, 114
as parameters, 414–415
typed, 115

Live Parsing Extensible Editor. See LPEX

Local data area (LDA), 422, 434

Local procedures, 443, 465, 464

Local variables, 407, 439, 441

Locking, 287
data areas, 423
exclusive, 423
In, Out and, 423–424, 435
Unlock, 423–424, 435

Locking records, 399

Logic, 2, 32
case, 152–153
control break, 164–173, **165**, **166**
read/write program sample of, 33

Logic errors, 16–17, 19

Logical files, 8, 19, 56, 57, 101
access paths in, 56
creating, with Create Logical File (CRTLF), 81, 102
Data Description Specification (DDS) and, 78–81
level checking and, 81
pointers in, 56
selecting fields with, 56
selecting records with, 56
simple, 78
updating database files and, 280

Logical operators, 182

Look up an array element (%Lookup), 340–343, 368, 369

Looping, 4, 34–36, 155–160. See also Dou, Dow, End/Enddo
condition-controlled, 157
count-controlled, 157
Do/Enddo in, 174
Dou in, 156–157, **157**
Dow in, 155–156, **157**
early exits from 159–160
For in, 157–159
interactive applications and, 379
leading decision loops in, 156

priming read, 34, 160–161
trailing decision loops in, 156

Low-order bits, 66

LPEX, 14, 15, 39

SQL and, 59

LR, 417, 434

LT, 80, 175

M

Machine language, 1, 19

Main procedure, 23, 26, 27, 32–38, 50, 438

arguments in, 36

built-in functions in, 36

cycle main program in, 32

end-of-file condition in, 33–34

factors in, 35

instructions in, 35

linear main program in, 32

logic in, 32

loops in, 33–34

operation codes (opcodes) in, 35–38

priming read in, 34, 160–161

pseudocode in, 32–35

RPG cycle in, 32

Maintenance, 16, 48, 56

database files, 272

interactive applications and, 391–400

modular programs, with Update

Program (UPDPGM), 466,

457, **457**

Master files, 8, 19

Maximum digits rule, 190

Member, source, 17

Member types, 17

Members, 8, 39

Modular programming, 4, 162, 395, 405–436. See also Calls, passing, and modular programming

MODULE, 473

Modules, 17

Monitor and monitoring errors,

498–500

Move/move left (Move, MoveL),

224–227, 254–255

MULT, 195

Multidimensional arrays, 353–354

Multiple entry-points programs, 470.

See also Service programs

Multiple-occurrence data structures

(MODS), 366–368

Multiple-row fetch, 350

Multiplication, 182–183

sizing results for, 187

MVR, 195

N

Named constants

character literals as, 115

declarations and, 107, 113–118, 142

Declare Constant for, 116–117

figurative constants and, 117–118

filenames for, 108

numeric literals as, 114

Naming conventions, 11, 64

data items, 44

databases and, 64

device names in, 29

field names in, 31, 94, 99

filenames and, 29, 108

qualified names in, 7, 64

record formats, 63–64

Rename keyword in, 111–112

simple object names in, 7

special characters in, 31

SQL, 64

system, 64

tables, 61–62

variables, 31

Native date, 72, 102

NE, 80, 175

Negative numbers/values, CR to

denote, 13, 85–86

Nested If, 151–152, 152

NG, 80

NL, 80

Nomain modules, 449–451, 465, 464

Not equal to operator, 147

NOT NULL, 62–63

Null values, 62–63, 117–118

host variables and, 298

SQL and, 298

NUMERIC, 62, 69–70, 121

Numeric data/data types, 65, 69–72

abend and, 185

Absolute value (%Abs) in, 191, 198

accumulation in, 184

addition in, 182–183, 186

arithmetic with, 181–200

built-in functions and, 182–183,

190–194

combined operators in, rules of precedence for, 183

compound operators in, 184–185

conversion errors in, 503–505

convert to character (%Char),

218–219, 229

convert to date (%Date), 246,

247–249, 256

Convert to packed decimal (%Dec)

in, 192–193, 198

Convert to packed decimal with

half-adjust (%Dech) in, 193,

198

- convert to time (%Time), 246, 249–250, 256
 - convert to timestamp (%Timestamp), 246, 250, 256
 - Convert to unsigned integer (%Uns/Unsh), 198
 - counting in, 184
 - data-conversion functions in, 192
 - decimal position in, 93–94, 141
 - DECIMAL, 70–71
 - division in, 182–183, 187
 - EBCDIC and, 69
 - Eval in, 181–185, 189–190, 198
 - exponentiation in, 182–183
 - field size and, 185–186
 - figurative constants and, 182
 - Integer division (%Div) in, 191, 198
 - INTEGER, SMALLINT, BIGINT in, 71–72
 - legacy code and, 194–197
 - literals, numeric literals, 114
 - logical operators and, 182–183
 - maximum digits rule in, 190
 - multiplication with, 182–183, 187
 - negative numbers/CR and, 13, 85–86
 - numeric operators for, 194–196
 - numeric overflow in, 185–186
 - NUMERIC, 62, 69–70, 121
 - packed decimal, 70–71, 120–121
 - precision in, 69, 189–190
 - relational comparison/operators in, 146–147, 182–183
 - Remainder (%Rem) in, 191–192, 198
 - result decimal positions rule in, 190
 - result field definition in, 196
 - resulting indicators in, 196–197
 - rounding in, 188–189
 - scale in, 69
 - simple assignment in, Eval and, 181–182
 - Square root (%Sqrt) in, 192, 198
 - subtraction in, 182–183, 186–187
 - truncation in, 185–186, 189–190
 - zeros in, 182
 - zoned representation in, 69–70, 120–121
- Numeric operators, 194–196
- Numeric overflow, 185–186
- O**
- Objects, 7, 19
- naming, 7
 - qualified names for, 7
- Older programs. See Fixed-format and legacy programs*
- On-error operations, 498–500
- Open, 277, 288, 319
- Open Cursor, 306
- Open data paths, 489–490
- Open Database File (OPNDBF), 490
- Operand values (factors), 35
- Operands, 50
- extended factor 2 in, 47
 - factor 2, 46–47
- Operation codes (opcodes), 35–38, 50–51
- fixed-format programming and, 46
- Operation extender, 276
- Operators
- AND/OR in, 149
 - combining, rules of precedence for, 183
 - compound, 184–185
 - concise, 38
 - numeric, 194–196
 - precedence of, 183
 - relational, 80
 - relational, 146–148
- OPNDBF, 490
- OR, 149, 389
- Original Program Model (OPM), 483
- OS/400, 4–5, 4
- Other, 146, 178
- Out, 423–424, 435
- Output editing, 12
- Output files, 30
- writing to (Write), 270–271
- Output specification, 40, 55
- constants or literals in, 99–100, 99
 - edit codes in, 100
 - end position in, 100
 - exception lines/names in, 96
 - field names in, 99
 - headings and, 25–26, 26, 96
 - record identification entries in, 96–99
 - skip entries in, 97–98
 - space entries in, 97–98
- Overflow indicator, 30
- Overflow, numeric, 185–186
- Overlapping subfields, 127–129
- OVERLAY, 383
- Overlay keyword, 127–129
- Override with Database File (OVRDBF), 488
- Overrides
- activation groups and, 488–489, 491
 - Extxx keywords in, 112–113
 - Delete Override (DLTOVR) and, 489
 - Override with Database File (OVRDBF) and, 488
- Overview documentation, 48–49
- OVRDBF, 488
- P**
- Packed decimal numeric data, 70–71, 120–121
- Convert to (%Dec) in, 192–193, 198
 - Convert to packed decimal with half-adjust (%Dech) in, 193, 198
- Parameters (arguments), 407–409
- application program interfaces (APIs) and, 545–546
 - by-reference passing, 414–415
 - calls, passing, and modular programming, 434
 - changing values of, 412–415
 - Const keyword for, 413, 434
 - data areas for, 421–424
 - Declare Parameters (Dcl-parm) in, 408
 - dynamic SQL and, 308
 - passing, 405, 412–415. *See also Calls, passing, and modular programming*
 - passing by value, 457–459, 465
 - read-only reference in passing, 413–415, 434
- Parse QUERY_STRING (QzhhCgiParse), 559
- Partial composite keys, 268–269
- Passing data. See data passing*
- Per record (Perrcd) keyword, 332
- PGM object, 7, 438
- Physical files, 8, 9, 14, 19, 56, 57, 101
- Add Physical File Constraint (ADDPFCST) in, 391
 - changing, with Change Physical File (CHGPF), 81
 - creating, with Create Physical File (CRTPF), 81, 102
 - Data Description Specifications (DDS) and, 76–78
 - data dictionary and, 88
 - deleting, 81
 - level checking and, 81
- Pointers, 56, 550
- Positional notation, 10, 11**
- Positioned updates, 308
- Power Systems servers, 5
- Precedence of operators, 183
- Precision, in numeric data, 69
- Prepare, 309–310, 319
- Preruntime arrays, 358–359
- PRIMARY KEY, 62, 63
- Priming read, 34, 160–161

- Printer files and reports, 25, 26, 50, 57, 102**
 constants or literals in, 99–100
 creating, with Create Printer File (CRTPRTF), 82
 date and time in, 84
 declarations and, 109
 designing, 12–13
 Detail lines in, 25–26, **26**, 96
 edit codes in, 100
 end position in, 100
 exception lines/names in, 96
 Exception Output, with Except operation, 101, 102–103
 externally described, 82–85
 field names in, 99
 headings in, 25–26, **26**, 96
 lines in, records and, 25, 95
 output editing in, 85–87
 output specifications for, 95–101
 overflow indicator in, 30
 records in, 25, 95
 Report Designer and, 82–85
 skip entries in, 97–98
 SKIPxx keywords in, 84
 spaces in, space entries, 97–98
 SPACExx keywords in, 84
 Total lines in, 25–26, **26**, 96
 zero suppression in, 85–86
- Problem definition, in program design, 14
- Procedural languages, 2
- Procedure Boundary (P) specification in, 40, 463–464
- Procedure interface, 434, 440–441, 465
 declarations and, 107
 fixed-format programming and, 425
- Procedures, 40, 437–467
 bind by copy in, 452, 465
 binding directories for, 455–457, 465
 binding of, 439, 451–457, 465
 Call a Prototyped Procedure or Program (Callp) operation in, 406, 409–410, 434, 435, 437, 465, 466, 545
 calls, passing, and modular programming, 410–412
 coding, 439–442
 compiler directives and, 460–461, 465–466
 compiling and, 439, 451–457
 conditional compiler directives and, 461–463
 Copy and Include, 460–461, 465–466
 copybooks for, 460–461, 465–466, 465
- Create Bound RPG Program (CRTBNDRPG) and, 451–457, 466
- Create Program (CRTPGM) and, 451–457, 465, 466
- Create RPG Module (CRTRPGMOD) and, 451–457, 465, 466
- cycle main programs and, 443–446, 465
- declarations in, 440, 465, 464
- Declare procedure (Dcl-proc) for, 439–440, 465, 466
- Declare procedure interface (Dcl-pi) in, 410–411, 435
- Display Message (Dsply) in, 446, 466
- Display Program (DSPPGM) in, 454, 466
- End procedure (End-proc) for, 439–440, 465
- executing, 442–443
- Extpgm keyword and, 442
- file I/O, subprocedures and, 459–460, 465
- global variables in, 445
- Integrated Language Environment (ILE) and, 438
- interfaces for, 410–412. See also Procedure interfaces*
- legacy code and, 463–464
- linear main programs and, 446–449
- local, 443, 465
- local variables in, 439, 441
- main procedure in, 438
- maintaining modular programs and, with Update Program (UPDPGM), 457, **457**, 466
- modular program creation with, 438–439, 451–457, **452**, **453**, **454**
- Nomain modules and, 449–451, 465
- passing parameters by value in, 457–459, 465
- PGM object in, 438
- Procedure Boundary (P) specification in, 463–464
- procedure interface and, 440–441, 465
- processing statements/calculations in, 440, 441
- prototype for, 4, 442, 465
- recursion in, 439
- resolution process in, 437
- return values in, 439, 440–441
- roles of, 439
- RPG cycle and, linear main programs and, 446–449
- static binding in, 438
- subprocedures and, 438, 441
- templates for, 461
- variable scope and, 439
- wrapper, for APIs, 554–559
- Program design, 14
- Program development cycle, 13–16
- Program Development Manager (PDM), 14
- Program entry phase, 16–17, 18**
- Program maintenance. See Maintenance*
- Program status data structure for error control, 508–509, 513
- Program testing, 16–17, 18**
- Program workflow, 145–180
 Calculation specification in, 173
 case logic in, 152–153
 conditioning indicators in, 176–177
 control break logic in, 164–173, **165**, **166**, 177
 control flow in, 146
 Do/Enddo in, 174
 Dou in, 156–157, **157**, 160–161, 175, 177, 178
 Dow in, 146, 155–156, **157**, 160–161, 175, 177, 178
 early exits from looping in, 159–160
 Else/Elseif in, 146, 150, 152–153, **153**, 178
 End/Endo in, 178
 file processing with Dow and Dou, 160–161
 For in, 146, 157–159, 178
 hierarchical decomposition in, 161–162
 If in, 146, 148–150, 151–152, **151**, **152**, 175, 178
 INZSR subroutine in, 164
 Iter in, 178
 iteration (repetition) operators in, 146, 155–159
 Leave in, 178
 legacy code and, 173–177
 logic in, 167
 looping in, 155–160
 modular programming in, 162
 nested If in, 151–152, **152**
 Other in, 146, 178
 pseudocode in, 166–167
 recursion and, 163
 relational comparisons in, 146–148, 174–176
 Select in, 146, 154–155, 177, 178
 selection (decision) operators in, 146, 148–155

sequential, 146
 structured design in, 145–146
 subroutines in, 162–164, 167, 177
 top-down design in, 161–162, 177
 When in, 146, 175, 177, 178
 Program-described files, 55–56, 88–91
 Except/Exception Output in, 101, 102–103
 processing of, 101
 Programming, 1–2
 program development cycle in, 13–16
 sections of the ILE RPG program.
 See also Sections of the ILE RPG program
 specifications for, 10
 Programming cycle, 19
 Programs, 7
 Prompts, in fixed-format programming, 41
 Prototypes and prototyping, 4
 Call a Prototyped Procedure or Program (Callp) operation in, 406, 409–410, 434–435, 437, 465, 466
 call interface, 407–409
 declarations and, 107
 Declare Prototype (Dcl-pr) in, 407–408, 435
 fixed-format programming and, 425
 naming prototypes in, Extpgm for, 408
 procedures and, 442, 465, 464
 Pseudocode, 32–35, 166–167
 interactive applications and, 395
 PSSR subroutine, 509–510
 Punch card programming, 2, 14
 Put environment variable (QtmhPutEnv), 559

Q

QCMDEXC, 547–548
 QtmhCvtDb, 559
 QtmhGetEnv, 559
 QtmhPutEnv, 559
 QtmhRdStin, 559
 QtmhWrStout, 559, 560–562
 Qualified data structures, 131–132
 Qualified names, 7, 64
QUSCMDLN, 546–547, 546
 QUSCRTUS, 548–550
 QUSLMBR, 548
 QUSLSPL, 545
 QUSPTRUS, 550–553
 QUSRSPLA, 545
 QzhhCgiParse, 559

R

Radix characters, 10
 Random access, 265–267
 Random retrieval (Chain), 265–267
 RANGE, 386
 Range tables, 364–365
 RDi development environment, SQL and, 60
 Read, 35, 51, 260–261, 278, 279, 281, 287
 interactive applications and, 379, 401
 operation extender for, 276
 priming read in, 34, 160–161
 Read equal key (Reade), 262–263
 Read from Stdin (QtmhRdStin), 559
 Read Only clause, 304, 305
 Read prior equal record in (Readpe), 264–265
 Read prior record in (Readp), 264–265
 Read/write program logic, 33
 Read-only reference parameter passing, 413–415, 434
 Reade, 262–263, 281, 287, 288
 operation extender for, 276
 Readp, 264–265, 287, 288
 operation extender for, 276
 Readpe, 264–265, 287, 288
 operation extender for, 276
 Record format, 25
 Record identification entries, 92–93, 96–99
 Record-level keywords, 76, 383–384
 Records, 8–10, 19, 25, 55, 57, 76
 adding, using Except (Calculation time output) in, 284–286
 address of, 283
 arrival sequence for, 56
 Delete operation for, 274–275
 display files and, 377–378
 format names for, with RCDFMT, 63–64
 key sequence for, 56
 keywords for, in DDS, 76
 length of, 138, 283
 locking, 275–276
 naming, 11
 read prior (Readp), 264–265
 read prior equal (Readpe), 264–265
 record identification entries in, 92–93, 96–99
 selecting, 56
 subfiles and. *See Subfiles*
 Update operation for, 272–274
 Recursive calls, recursion, 163, 406–407, 439
 Redundancy, 56
 REF, 382

Reference files, display files and, 376
 Relational comparisons, 146–148
 legacy code, 174–176
 Relational operators, 80, 146–148, 182
 Releases of RPG, 24
 Remainder (%Rem), 191–192, 198
Remote System Explorer for service programs, 474, 475
 Rename keyword, 111–112
 Replace character string in (%Replace), 212–213, 229
Report Designer, 12–13, 13, 82–85, 101
 Report lines, 25, 95
Report Program Generator. *See RPG Reports.* *See Printer files and reports*
 Reset operation, 399, 401
 Resolution process, 437
 Result data structure, accessing
 database files and, 261
 Result decimal positions rule, 190
 Result field definition, 196
 Resulting indicators, 196–197, 513
 character data and, 228
 file I/O, 286–287
 Retrieve Pointers to User Space (QUSPTRUS), 550–553
 Retrieve Spooled File Attributes (QUSRSPLA), 545
 Return, 35, 38, 51
 calls, passing, and modular programming, program execution and, 416–418, 435, 466
 Return codes, SQL, 300–303
 Return open file condition (%Open), 277
 Return values, 439, 440–441
 Reuse of code, 405
 Revisions, documentation and, 49
 Rounding, 188–189
 Rows, 57
 fullselect and, 74–75
 RPG, 18
 history of, 2
 SQLRPGLE programs in, 313–314
 RPG cycle, 32, 446–449
 RPG II, 2–3, 6
 RPG III, 3–4
 RPG/400, 4
 RPGLE, 50
 Ruler lines, fixed-format programming and, 41
Run SQL Statements (RUNSQLSTM) command in, 59, 60
RUNSQLSTM, 59, 60
 Runtime arrays, 325, 358–359

S

- Sample code for interactive program
 - 395–398
- Sample ILE RPG program, 25–38
- SBMJOB, 422
- Scale, in numeric data, 69
- Scan and replace characters in
 - (%Scanrpl), 213–214, 229
- Scan string (%Scan), 215–217, 229
- Schema (collections), 64–65
 - change value of, with Set Schema, 65
 - current, 65
 - SQL and, 57
- Screen design for interactive applications, 400, 401
- Screen Design Aid (SDA), 374
- Scrollable cursors, 305–306
- Sections of the ILE RPG program, 23–53
- Sections of the RPG program
 - arguments in, 36
 - binding, 39
 - building, 38
 - built-in functions in, 36
 - case insensitivity in, 27
 - comment, 26, 27
 - compiling, 39
 - control options in, 23, 26, 27, 28
 - Data Description Specifications (DDS), 26, 75–81
 - declarations in, 23, 26, 27, 28–32
 - documentation, 26, 27
 - editor for, 39
 - loops in, 33–34
 - main procedure in, 23, 26, 27, 32–38
 - operation codes (opcodes) in, 35–38
 - pseudocode in, 32–35
 - subprocedure, 23
- SELECT/Select, 58, 146, 154–155, 177, 178, 291, 292, 318, 319
 - cursors, SQL and, 304
 - Read Only clause and, 304, 305
 - Select Into, with host variables, 296–297
- SELECT INTO, 58
- Selection (decision) operators, program workflow and, 146, 148–155
- Semicolon (;)
 - as delimiter in SQL, 59
 - as terminator character, 24, 35
- Separator characters, date/time data and, 237
- Sequence, 40
 - arrival, 56
 - key, 56
 - program flow and, 146
- Sequential access, 259–265
- Serial cursors, 305
- Servers, 5
- Service programs, 469–492
 - activation groups in, 483–490, 941.
 - See also Activation groups
 - bind by copy and, 472
 - bind by reference and, 470, 472, 490
 - Bind Service Program (NDSRVPGM) in, 473
 - binder language in, 477, 478–482
 - binding directories in, 471
 - Binding Directory (BNDDIR) in, 473
 - binding in, 469–474, 491
 - calls made to, 471–475
 - clients for, 470, 491
 - compiling, 470–471
 - Create Service Program (CRTSRVPGM) for, 470–471, 471, 490
 - deploying, 471–475
 - Display Program (DSPPGM) in, 474, 474
 - Display Service Program (DPSRVPGM) in, 479
 - End Program Export List (ENDPGMEXP) in, 478–481
 - Export a Program Symbol (EXPORT) in, 478–481
 - exporting data items in, 482
 - maintaining, 475
 - MODULE command in, 473
 - multiple entry points in, 470
 - Remote System Explorer view of, 474, 475
 - signatures in, 476–481, 477, 491
 - Start Program Export List (STRPGMEXP) in, 478–481
 - superglobal definition in, 482
 - Update Service Program (UPDSRVPGM) in, 475, 476
- Set greater than (Setgt), 263–264, 287
- Set lower limit (Setll), 261–262, 287
- Set Option, 312–313, 319
- Set Schema, 65
- Set/get portion of array (%Subarra), 346–350, 369
- Setgt, 263–264, 287, 288
- Setll, 261–262, 287, 288
- Setmode subroutine, in interactive applications, 399
- Shared open, 490
- Signatures, in service programs, 476–481, 477, 491**
- Simple logical files, 78
- Simple object names, 7
- Single fetch, 350
- Skip entries, 97–98
- SKIPxx keywords, 84
- Slash-slash separator (//), 7, 24, 28, 42, 49, 50
- SMALLINT, 62, 71–72, 121
- Sort array elements (Sorta), 345–346, 369
- Sorta, 345–346, 369
- Source code, 2, 16, 17, 50
 - case-insensitive nature of, 27
- Source Entry Utility (SEU), 14, 15, 39**
 - SQL and, 59
- Source file member, 39
- Source member, 17
- Source physical file, 14
- Space entries, 97–98
- SPACExx keywords, 84
- Spaghetti code, 379
- Special characters in names, 31
- Specifications for programming, 10
 - Data Description Specifications (DDS) in, 57, 75
 - fixed-format programming and, 41–42
- Spooled files
 - List Spooled File (QUSLSPL), 545
 - Retrieve Spooled File Attributes (QUSRSPLA), 545
- SQL, 3, 57–65, 102, 291–321
 - Alwcpydta option in, 312–313
 - arrays and, 350–353, 368
 - Close, 307–308
 - Closqlcsr in, 313
 - column data type definitions in, 62, 65–73
 - columns in, 57
 - command line for, 59
 - Commit in, 313
 - compiler directives for, 317–318
 - constraints in, 62–63
 - Create SQL ILE RPG Object (CRTSQLRPGI) in, 313, 319
 - cursors in, 303–308, 318
 - Data Definition Language (DDL) statements in, 58
 - Data Description Specifications (DDS) and, 57, 75, 294.
 - See also Data Description Specifications (DDS)
 - Data Manipulation Language (DML) in, 58, 291–294, 318
 - DELETE in. See DELETE/Delete
 - development tools for, 58–60
 - dynamic, 308–312, 319
 - embedded, 294

- Embedded SQL Host Language
 - statements in, 58
 - error management and, 302–303, 511–513
 - example code using, 314–317
 - Exec SQL directive for, 294–295
 - Execute, 310–311, 318
 - Execute Immediate in, 311–312, 319
 - Fetch, 306–307, 368
 - Fetch, with arrays, 350–353
 - formatting scripts/code in, 61
 - host structures in, 297–298
 - host variables in, 295–296, 318
 - IBM Data Studio client for, 60
 - IBM i terminology vs., 57
 - indexes in, 57, 75
 - INSERT* in. *See INSERT*
 - Interactive SQL facility for, 59
 - legacy code and, 317–318
 - naming conventions and, 64
 - null values in, 62–63, 298–300
 - Open, 306
 - parameter values in, 308
 - positioned updates in, 308
 - precompiler in, 313–314
 - Prepare, 309–310
 - qualified names in, 64
 - RDi development environment and, 60
 - record format names in, with RCDFMT, 63–64
 - return codes in, 300–303
 - rows in, 57
 - Run SQL Statements
 - (RUNSQLSTM) command in, 59, 60
 - schema (collections) in, 57, 64–65
 - Select Into, with host variable, 296–297
 - SELECT* in. *See SELECT*
 - semicolon (;) delimiter in, 59
 - Set Option in, 312–313
 - SQL Communication Area (SQLCA) data structure in, 300, 318, 511–513
 - SQL naming convention in, 64
 - SQL Procedural Language (SPL) statements in, 58
 - Sqlcode in, 301–302, 511–513
 - Sqlerrd in, 511–513
 - SQLRPGLE programs in, 313–314, 319
 - Sqlstate in, 301–303, 511–513
 - Sqlwarn in, 511–513
 - Start SQL Interactive Session (STRSQL) in, 59, 59
 - system naming convention in, 64
 - table creation in, with CREATE TABLE, 61
 - table naming in, 61–62
 - tables in, 57
 - UPDATE* in. *See UPDATE*
 - Values clause in, 292
 - views in, 57, 73–75
 - Whenever statement in, 514–515
 - Where clause in, 293, 318–319
 - Xfields function in, 293
 - SQL Communication Area (SQLCA), 300, 318, 511–513
 - SQL Procedural Language (SPL) statements, 58
 - SQLCA, 300, 318, 511–513
 - Sqlcode, 301–302, 511–513
 - Sqlerrd, 511–513
 - SQLRPGLE programs, 313–314, 319
 - Sqlstate, 301–303, 511–513
 - Sqlwarn, 511–513
 - SQRT, 195
 - Square root (%Sqrt), 192, 198
 - Standalone variables, 30–32, 50
 - arrays as, 324, 368. *See also Arrays*
 - data item name in, 118
 - data types in, 119–122
 - date, time, timestamp in, 122, 235–236
 - declarations and, 107, 118–122, 142
 - declaring, 30–32, 118
 - indicators and, 120
 - Like, Likeds, Likerec keywords in, 132–135, 142
 - packed data types in, 120–121
 - unsigned integers in, 121
 - zoned data types in, 120–121
 - Start Program Export List (STRPGMEXP), 478–481
 - Statements, 24
 - semicolon terminator for, 24, 42
 - Static binding, 438
 - Status of file or program (%Status), 496–497
 - STRPGMEXP, 478–481
 - STRSQL**, 59, 59
 - Structured design, 3–4, 145–146
 - Structured Query Language*. *See SQL*
 - SUB, 195
 - Subfields, 124, 125–126
 - character data and, Eva-corr and, 203–205
 - date/time data and, 235–236
 - externally described, 129–131
 - initializing, 126–127
 - Like, Likeds, Likerec keywords in, 132–135, 142
 - overlapping, 127–129
 - qualified data structures and, 131–132
 - Subfile programming, 519–544
 - control record formats for, 523–524
 - entire subfile at once loading, 538–542
 - example of, 519–521, 520, 521, 522
 - keywords for, 523–524
 - list panels and, 519
 - loading, 524–542
 - page at a time loading of, 525–538
 - record formats for, 522–523
 - Submit Job (SBMJOB), 422
 - Subprocedures, 23, 438, 441
 - file I/O and, 459–460, 465
 - Subroutines, 162–164, 167, 177, 395
 - automatic execution of, with INZSR, 164
 - Begin subroutine (Begsr) for, 162, 178
 - End subroutine (Endsr) for, 162, 178
 - error management and, 505, 507–508
 - Execute subroutine (Exsr) for, 163, 178
 - INZSR, 164
 - Leave subroutine (Leavesr) for, 163–164, 178
 - PSSR, for error control, 509–510
 - recursion and, 163
 - Substrings, 211–212
 - Subtract duration (subdur), 252, 253–254
 - Subtraction, 182–183
 - sizing results for, 186–187
 - Sum elements of array (%Xfoot), 346, 369
 - Superglobal definition, 482
 - Syntax/syntax errors, 16–17, 19, 28
 - System naming convention, 64
 - System/3 computer, 3
 - System/38 minicomputers, 3
 - System-named activation groups, 484–485
 - System i, 5
- T**
- Tables, 57, 323–324
 - arrays and, 323–324, 359–365
 - changing values in, using arrays, 365
 - creating, with CREATE TABLE, 61

lookups in, using arrays, 361–362
 naming, 61–62
 range, 364–365
 related, using arrays, 363–364

Templates, 461

Terminator character (:), 24, 35

Testing, 19

TEXT keyword, 77

Thousands separators, 10

TIME, 62, 72–73, 388

Time data types, 72–73. *See also Date/time*

Timestamp, 73, 122, 233. *See also Date/time*
 convert other data to
 (%Timestamp), 246, 250, 256

TIMESTAMP, 62, 73

Title case, 27

Top-down design, 161–162, 177

Totals in report, 25–26, 26, 96
 crossfooting in, 346, 369
 summing elements of an array
 (%Xfoot), 346, 369

Trailing decision loops, 156

Transaction files, 8, 19

Translate (%Xlate), 214–215, 229

Trim leading/trailing blanks (%Trim),
 209–211, 229

Truncation, 185–186, 189–190

Typed literals, 115
 date/time data and, 239–240, 255

U

UNIQUE, 62, 63, 76–77

UNIX, 5

Unlock, 276, 287, 288, 423–424, 435

Unresolved imports, 456, 466

Unsigned integers, 121

UPDATE/Update, 58, 272–274, 278,
 287, 288, 291, 293, 319
 cursors, SQL and, 304, 308
 positioned updates in, 308

**Update Program (UPDPGM), 457,
 457, 466**

**Update Service Program
 (UPDSRVPGM) in, 475, 476**

Updating data in interactive programs,
 391
 Add mode in, 392–395, **392–394**,
 399
 Addmode subroutine in, 399
 Chain operation in, 399
 Change mode in, 392–395,
392–394, 399
 constraints and, 391, 401
 Delete mode in, 392–395,
392–394, 399

example of, 392–395, **392–394**
 fields displayed in, 399–400
 locking records in, 399
 modules and organization in, 395
 pseudocode in, 395
 Reset operation in, 399
 sample code for, 395–398
 Setmode subroutine in, 399
 subroutine organization in, 395
 validating data in, 392, 401

Updating database files, 272–290
 Delete records in, 274–275
 errors in, error log for, 279
 file I/O resulting indicators,
 286–287
 file maintenance and, 272
 locking in, 275–276
 logical files for, 280
 Read operation, 281
 Reade, 281
 Update records in, 272–274

UPDPGM, 457, 457, 466

UPDSRVPGM, 475, 476

URLs, 562

Usage keyword, 30, 109–110, 141

User open operation (Usroprn), 277

User profiles, 7, 65

User space for APIs, 548–553

User-named activation groups, 484

USRPRF, 7

V

Valid Command Key (VLDCMDKEY),
 383

Validating data
 interactive applications and,
 385–388, 392, 401

validity-checking codes, 387

VALUES, 386

Values clause, SQL, 292

VARCHAR, 62, 68, 119–120

Variable-length character string,
 68, 119–120, 207–208. *See also
 Character data type; VARCHAR*

Variables, 6–7, 18, 24, 27
 character data and, 207–208
 concise operators for, 38
 data structures for, 124
 data types for, 24, 31–32
 global vs. local, 407, 439, 441, 445
host. *See Host variables*
 indicator, 298
 indicators as, 2, 120
 initializing/assigning values to, 24,
 32, 122–124
 Last Record indicator in, 38

Like, Likeds, Likerec keywords in,
 132–135, 142

local, 439, 441

naming, 31

overlapping subfields for, 127–129

pointers as, 550

scope of, 407, 439

standalone. *See Standalone
 variables*

subfields and, 124, 125–126

value assigned to, using Eval, 37

variable-length, 207–208

Views, 57, 73–75, 102
 fullselect and, 74–75

W

Web page creation with HTTP APIs,
 559–563

CGI header, 562

Common Gateway Interface (CGI)
 and, 559

URL in, 562

When, 146, 175, 177, 178

Whenever statement, 514–515

Where clause, 293, 318–319

While loop, 33–34

Work fields. *See Standalone variables*

Work with Binding Directory Entries
 (WRKBNDDIRE), 456, 466

Workflow. *See Program workflow*

Workstation file, 57

Wrapper procedures, 554–559

Write, 35, 37, 51, 270–271, 287, 288
 interactive applications and, 379,
 401

Write to output files (Write), 270–271,
 270

Write to Stdout (QtmhWrStout), 559,
 560–562

WRKBNDDIRE, 456, 466

X

Xfields, 293

Z

Z-ADD, 195

Z-SUB, 195

Zero and one in programming, 1, 2,
 66–67

Zero suppression, 13, 85–86, 85

Zeros, 117–118, 182

Zoned decimal, 102

Zoned representation, numeric data,
 69–70, 120–121