

Index

Boldfaced text indicates tables or illustrations.

A

Access paths, 230, 231
Activation groups, database modernization/
restructuring and, 384–386
ADD, 167, 178
Add Binding Directory (ADDBNDDIRE) in,
24
ADD_MONTHS, 266–267
ADDBNDDIRE, 24
ADDUR, 123, 169, 178
ADDR function, 65, 66
Adsero Optima Foundation, database
modernization/restructuring and, 365
Advantages of using ILE RPG, 1–6
Aggregate functions, SQL, 245–248, 331
Aliasing and ALIAS, 236, 322, 368
ALL, 209, 246–248
debugging and, 188
ALLOC, 169, 177, 178
ALTER TABLE, 315–318
ALTER TRIGGER, 325–330
AND, 233, 244
ANDxx op codes, 168, 178

Architecture, multi-tier, modular programming
and, 395–399, **395**, **397**, **398**
Arithmetic operators, in SQL, 234
Array-related opcodes, 172–176
Arrays, 284
ASCII function, 249, 331
ASNA, 406
Assignment expressions, 179
Asterisk Operator (*), 179
Atomicity, 359
Attributes, in ERDs, 360–361, **361**
Authorization and privileges, SQL, 387
AVG, 245–247, 331

B

Batch jobs, debugging, 207–209, 210
BCD Software, 152, 407
BEGIN/END, 328
BETWEEN, 234
Bind by copy, 12, **12**, 15, 20
BINDEC, 218
Binding, 12, 15, 19–27
Add Binding Directory (ADDBNDDIRE)
in, 24

- bind by copy, 12, **12**, 15, 20
 - binding directory for, 22–23, 27, 154–156
 - compiling and, 20–21
 - Create Binding Directory (CRTBNDDIR)
 - in, 24
 - Create Bound RPG Program (CRTBNDRPG)
 - and, 188–189
 - Create Program (CRTPGM) in, 25
 - error management in, 25–26, 27
 - H–specs in, 23
 - service programs for, 15–16, **16**
 - shortcuts for, 26
 - user-defined options for, 26, 27
 - Work with Binding Directory (WRKBNDDIR) in, 24
 - Binding directories, 22–23, 27, 154–156
 - code organization and, 154–156
 - creating, 155
 - BIRT, 225, 304
 - BITAND/BITNOT/BITOR, 169, 178–180
 - BOTTOM, 193
 - BREAK, 196–198, 205
 - Breakpoints, 196–198, **197**, 210
 - conditional vs. unconditional, 201–203, **202**, **203**
 - Building procedures and functions, 98–118
 - Built-in functions (BIFs), 4, 41, 75–136, 177
 - array-related, 172–174
 - CAT, 175
 - CHAIN, 77–78
 - CHAR, 84–85
 - character conversion and, 84–85
 - data structures and, 87–88
 - DATE, 119, 123, 128
 - date operations with, 118–129
 - DAYS, 122, 123
 - DEC/DECH, 80–81, 84, 102
 - DIFF, 123–124, 125, 132
 - edit codes in, EDITC, 85–87
 - edit words in, EDITW, 85–87
 - EDITC/EDITW, 84
 - EDITFLT, 84
 - EOF, 76–77
 - EQUAL, 78–79
 - ERROR, 122
 - EVAL, 79, 80
 - file-related indicators vs., 75–76
 - find and replace text in strings with, 92–98
 - fixed-format coding in, 84
 - FOUND, 78
 - free-format coding in, 84
 - HOURS, 130
 - LOOKUPxx and TLOOKUPxx, 174–175
 - MINUTES, 130
 - MONTHS, 122, 123
 - MOVE and MOVEL removed with, 79–81
 - MSECONDS, 130
 - nested, 51
 - numeric conversion, 80–82, 84
 - operation codes replaced by, 169, 177
 - READxx, 75–76
 - REM, 125–126, 134
 - REPLACE, 94–96, 111
 - SCAN, 92, 93–96, 111, 116–118
 - SCANRPL, 96–98
 - SECONDS, 130
 - SETGT, 76
 - SETLL, 76
 - SETxx, 75–76
 - string operations with, 88–98
 - SUBARR, 172–174
 - SUBDT, 122, 123, 130
 - SUBST, 90–91
 - TIME, 129, 133
 - time operations with, 129–135
 - TRIM/TRIML/TRIMR in, 91–92
 - user-defined functions using, 98–118
 - XLATE, 111–115
 - YEARS, 123
 - Business rules migration, 372–386. *See also*
 - Database modernization/restructuring;
 - Modernization
- C**
- C prefix, 141
 - CABxx, 171–172, 177, 178
 - Calculate day of week (Clc_DayOfWeek) in, 124–126

- Calculate time until weekend
(Clc_HowLongUntilWE) in, 130–135
- CALL, 178
- CALLB, 178
- Calling functions, 43
- CALLP, 33, 40, 164, 178
- CAST, 247
- CASxx, 170–171, 177, 178
- CAT, 175, 178
 - string operations and, 89
- Catalogs, 230, 231
- CHAIN, 77–78, 165, 288
 - built-in functions (BIFs) and, 77–78
 - EOF with, 77
- Change case (Chg_Case), 111–118
- Change Command Defaults, 26
- CHAR, 84–85, 135, 218, 249–250
- Character data, 169, 177, 178. *See also* String data
 - ASCII function and, 249
 - CHAR, 249–250
 - numeric data converted to, using CHAR, 84–85
- CHECK, 169, 177, 178
- CHECKR, 169, 178
- Chg_Case function, 111–118
- CL commands
 - debugging and, 207
 - SQL and, 276–277
- CLASS, 219
- Clc_DayOfWeek, 124–126
- Clc_HowLongUntilWE, 130–135
- CLEAR, 198, 200
- CLOSE, 292, 294
- Codd, Edgar, 358
- Code organization, 152–156
 - binding directories and, 154–156
 - /COPY in, 154
 - /INCLUDE in, 154
 - program flow in, 153, **153**
 - service programs and, 153–154
- Collections, 230, 308. *See also* Schemas
- Column functions, SQL, 245–270
- Columns, 230, 231
- Comment first, code later concept, 150–151
- Commenting. *See also* Documentation
 - free-format coding and, 160, 165, 167, 177
 - functions and, 46
 - SQL and, 277
- Commitment Control in SQL, 273
- COMP, 170–171, 177, 178
- Comparison operators in SQL, 234
- Compilers and compiling, 20–21
 - binding directories and, 22–23, 154–156
 - embedded SQL and, 285–286
 - error management in, 25–26, 27
 - free-format coding and, 159
 - order of, 26
 - PGM objects, 24
 - procedures, line position and, 34–35
 - shortcuts for, 26
 - user-defined options for, 26, 27
- Create RPG Module (CRTRPGMOD) and, 182, 188–189, 285
 - debugging and, 182
 - order of, 20–21
 - service programs and, 21–24
- Components of code/runtime services, 6
- Composite attributes, 360–361
- Compound assignment (+=1), 166
- Concatenation, 259
- Conceptual models of databases, 358
- CONST, 57–59, 70
- Constants, 216, 284
 - defining, 220–221
 - free-format coding and, 227
 - naming convention for, 141, 220–221
- Constraints, 315–318
- Control statements, in free-format coding, 211–213, 227
- Controller layer, in MVC, 403–404
- Convert string to numeric value (Cvt_To_Numeric), 99–106
- Converting fixed-format to free-format code, 166–167, 177. *See also* Free-format code
- /COPY, 32, 37–39, 40
 - code organization and, 154

debugging and, 184–186, **185**, 207
 free-format coding and, 226
 Copy members, 104
 COUNT, 247–248, 331
 Create Binding Directory (CRTBNDDIR) in, 24
 Create Bound RPG Program (CRTBNDRPG), 188–189
 debugging and, 188–189
 CREATE FUNCTION, 304–307
 CREATE INDEX, 320–322
 CREATE PROCEDURE, 300–303
 Create Program (CRTPGM), 25
 Create RPG Module (CRTRPGMOD), 21, 182, 188–189, 285
 compiling and, 285
 debugging and, 182, 188–189
 CREATE SCHEMA, 308–310
 Create Service Program (CRTSRVPGM), 22–24
 Create SQL ILE RPG Object (CRTSQLRPGI) for, 285–286, 332
 CREATE TABLE, 310–318
 CREATE TRIGGER, 325–330
 CREATE VIEW, 318–320
 CRTBNDDIR, 24
 CRTBNDRPG, 188–189
 CRTPGM, 25
 CRTRPGMOD, 21, 182, 188–189, 285
 CRTSQLRPGI, 285–286, 332
 CRTSRVPGM, 22–24
 Crystal Reports, 225, 304
 CURDATE, 262–263
 Cursors in SQL, 289–296
 CLOSE, 292, 294
 DECLARE, 290, 292, 293, 294
 FETCH, 290–292, 293, 294
 input-defined, PREPARE and, 294–296
 OPEN, 292, 294
 Open Query File (OPNQRYF) replaced by, 292–294
 PREPARE and, 294–296
 SELECT and, 289
 SQLCOD variable and, 291–292

CURTIME, 262–263
 Cvt_To_Numeric function, 99–106

D
 D-spec, 216, 227
 Data Definition Language (DDL), 229, 307–330, 332–333
 database modernization/restructuring and, 363
 DDS files converted to, 366–372
 DROP and, 322–323
 indexes in, 320–322
 Override Database File (OVRDBF) and ALIAS instruction in, 322
 schemas in, 308–310
 tables in, 310
 triggers in, 323–330
 views in, 318–320
 Data definitions, in free-format coding, 216–226, 227
 Data dictionary, 368–369, 371
 Data File Utility (DFU), 240, 242
 Data Manipulation Language (DML), 232–245.
 See also SQL
 Data structures, 87–88, 216
 built-in functions (BIFs) and, 87–88
 defining, 222, 228
 free-format coding and, 227–228
 OVERLAY and, 222, 228
 Database modernization/restructuring, 340–341, 357–392
 activation groups in, 384–386
 Adsero Optima Foundation for, 365
 atomicity in, 359
 attributes defined for, 360–361, **361**
 authorization and privileges in, GRANT and REVOKE, 387
 business role migration to database for, 372–386
 conceptual, logical and physical models of, 358
 Data Definition Language (DDL) as tool for, 363
 data dictionary in, 368–369, 371

- data domains in, 371
- DB2 advanced functionality for, 386–391
- DDS files converted to DDL objects, 366–372
- DDS keywords ignored during SQL generation, **369**
- encryption, 387–388
- Entity Relationship Diagrams (ERDs) for, 359–363, **360**, **361**, **362**
- existing files, discovering and listing all files before beginning, 367
- first normal form (1NF) in, 358–359
- I/O server isolation in, with code, 374–384
- IBM Data Studio for, 363–364
- IBM InfoSphere Data Architect for, 364–365, 372
- keys and key generation for security in, 388
- logical model for, 371
- methodology for, step-by-step, 365–391
- naming conventions in, 368
- normalization in, 358–359, 371–372
- physical and logical file definition in, 368, 370
- recursive relationships in, 361–362, **362**
- relationships between files, mapping before beginning, 367–368
- ROWIDs in, 390–391
- second normal form (2NF) in, 359
- security issues in, 386–391
- sequence objects in, 389–390
- third normal form (3NF) in, 359
- tools for, 363–365
- validation routines for, 372–374
- DATE, 119, 123, 128, 136, 180, 218, 219, 260–261
- Dates, 118–129, 259–270, 283
 - ADD_MONTHS, 266–267
 - ADDDUR/SUBDUR in, 123
 - adding and subtracting, using DATE, 123
 - built-in functions (BIFs) and, 118–129
 - calculate day of week (Clc_DayOfWeek) in, 124–126
 - CURDATE, 262–263
 - CURTIME, 262–263
 - date conversion function, 47–50
 - date data type, 259–260
 - date formats for, DATFMT keyword for, 119
 - DATE, 119, 123, 128, 136, 180, 218, 219, 260–261
 - DAY, 264, 269
 - DAYNAME, 263, 286–289
 - DAYOFMONTH, 264, 269
 - DAYOFWEEK, 264–265, 331
 - DAYOFYEAR, 266
 - DAYS, 122, 123, 136, 269, 283
 - DIFF, 123–124, 125
 - E extender and, 121–122
 - EXTRACT, 263–264, 269
 - format conversion using CHAR, **250**
 - formatting, using DATFMT, 119, 219
 - HOUR, 264
 - last day of month calculation (LastDayOfMonth) in, 119–124
 - LAST_DAY, 268–269
 - MICROSECOND, 264
 - MIDNIGHT_SECONDS, 266
 - MINUTE, 264
 - MONTH, 264
 - MONTHNAME, 263
 - MONTHS, 122, 123
 - MONTHS_BETWEEN, 267–268
 - NEXT_DAY, 269
 - NOW, 262
 - QUARTER, 266
 - REM, 125–126
 - retrieve day of week (Rtv_DayOfWeek) in, 126–127
 - SECOND, 264
 - SQL date functions for, 259–270
 - SUBDT, 122, 123, 263–264
 - testing date-related functions for, 128–129
 - TIME, 261
 - TIMESTAMP, 261–262
 - WEEK, 265
 - WEEK_ISO, 265
 - YEAR, 264
 - YEARS, 123
- DAY, 264, 269

- DAYNAME, 263, 286–289
- DAYOFMONTH, 264, 269
- DAYOFWEEK, 264–265, 331
- DAYOFYEAR, 266
- DAYS, 122, 123, 136, 269, 283
- DB2, 230
 - in database modernization/restructuring, 386–391
- DBGENCKEY, 189–190, 209
- DBGVIEW, 21, 182, 190
- DCL-PROC/END-PROC delimiters for, 222, 223–224, 228
- DDS files converted to DDL, in database modernization, 366–372. *See also* Database modernization/restructuring:
- DDS keywords ignored during SQL generation, **369**
- Debug Encryption Key (DBGENCKEY), 189–190
- Debug View (DBGVIEW), 21
- Debugging, 6, 181–210
 - ALL in, 188, 209
 - batch job, 207–209, 210
 - BREAK command in, 196–198, 205
 - breakpoints in, 196–198, **197**, 201–203, **202**, **203**, 210
 - CL programs and, 207
 - CLEAR command in, 198, 200
 - commands for, **205**
 - compile time and, 182
 - /COPY and /INCLUDE statements in, 184–186, **185**, 207, 209
 - Create Bound RPG Program (CRTBNDRPG) and, 188–189
 - Create RPG Module (CRTRPGMOD) and, 182, 188–189
 - Debug View (DBGVIEW) in, 21, 182, 190
 - Decryption Key (DBGENCKEY) and, 189–190, 209
 - encrypting views for, with Debug Encryption Key (DBGENCKEY), 189–190
 - End Debug (ENDDBG), 205
 - End Service Job (ENDSRVJOB) in, 208–209
 - EVAL command in, 203, 205
 - FIND command in, 194
 - Interactive Source Debugger (ISDB) vs. new ILE debugger in, 6, 181–182
 - LIST view in, 186–188, **186**, **187**, 209
 - navigational commands for, 193–196, **193**
 - NONE in, 188, 209
 - OPM RPG programs and, 207
 - OPTIONS parameters for, 186, 188, 207
 - parameter settings for STRDBG in, 194–196, **194**, **195**
 - parameters and, 55
 - service programs and, 206–207, **206**
 - SET command in, 207
 - SOURCE view in, 182, 184–186, **184**, **185**, 209
 - Start Debugger (STRDBG) in, 21, 190–193, **191**, **192**, 194–196, **194**, **195**, 201
 - Start Service Job (STRSVRJOB) in, 208
 - STEP command in, 203, 205
 - step-by-step session in, 201–205
 - stepping into a function in, 203
 - STMT view in, 183–184, 183, 190, 209
 - Target Release (TGTRLS) parameter setting in, 189–190
 - views in, 182–189, 209
 - WATCH command in, 198–200, **200**, 203–205
 - watch conditions in, 198–200, **200**, 203–205, 210
 - WHEN in, 205
- DEC/DECH, 80–81, 84, 102, 135
- DECLARE, 290, 292, 293, 294
- Decryption Key (DBGENCKEY), 189–190, 209
- Default Activation Group (DAG), 384
- DEFINE, 178
- DELETE, 78, 230, 231, 232, 244–245, 331
 - DROP vs., 322–323
 - embedded SQL and, 295–298
- EXISTS, 245
- SELECT and, 245
- WHERE, 245
- Derived attributes, 361

- DESC, 234
 DETERMINISTIC/NOT DETERMINISTIC, 302, 306
 DIFF, 123–124, 125, 132, 136
 DIM, 284
 DISPLAY MODULE, 193
 DISTINCT, 242, 246–248
 DIV, 167, 169, 178
 DO, 178
 Docu-Mint tool, 152
 Documentation, 149–152, 156
 binding directories for, 22–23, 154–156
 comment first, code later concept and, 150–151
 comments, in free-format code, 160, 165, 167, 177
 Docu-Mint tool for, 152
 double slash (//) for comments in, 167
 for functions, 149–150
 ILEDPCS tool for, 152
 Javadoc for, 151
 for procedures, 149–150
 purpose and scope of, 152
 RPGLEDOC tool for, 151
 self-documenting features for, 151–152
 SQL and, 277
 standards and guidelines for, 152
 strategy for, 152
 Double slash (//) for comments in, 167
 DOUxx, 168, 172, 178
 DOWN, 193
 DOWxx, 168, 172, 178
 DROP, 322, 323, 331
 DROP TRIGGER, 325–330
 Duplicated field names, 147–149
- E**
 E extender, 121–122
 Eclipse, 151
 EDIFLT, 135
 Edit codes, EDITC, 84, 85–87, 135
 Edit words, EDITW, 84, 85–87, 135
 EDITC, 84, 85–87, 135
 EDITFLT, 84
 EDITW, 84, 85–87, 135
 ELEM, 284
 ELSEIF, 162–163
 Embedded SQL, 279–298, 332. *See also*
 Embedding RPG code in SQL
 compiling, 285–286
 Create SQL ILE RPG Object (CRTSQLRPGI) for, 285–286, 332
 cursors in. *See* Cursors in SQL
 DELETE and, 295–298
 editors for, 285
 embedding RPG code in SQL instead, 299–307. *See also* Embedding RPG code in SQL
 EXEC SQL/END SQL in, 280, 281, 283
 fixed-format code and, 280–281
 free-format code and, 281
 functions in, 286–289
 host variables in, 282–285
 INSERT and, 295–298
 INTO in, 282
 prepared statements in, 298
 RPG and SQL interaction in, 281–285
 RPGLE and editor recognition of, 285
 SQL cursors to replace record-level data access in, 289–296
 UPDATE and, 295–298
 WHERE, 282
 Embedding RPG code in SQL, 299–307. *See also* Embedded SQL
 CREATE FUNCTION in, 304–307
 CREATE PROCEDURE in, 300–303
 stored procedures for, 299, 300–303
 user-defined functions for, 299, 300, 303–307, **307**
 Encryption, 387–388
 End Debug (ENDDBG), 205
 End of instruction (:); semicolon indicator, 159, 167, 177
 End Service Job (ENDSRVJOB) in, 208–209
 END-FREE. *See* FREE/END-FREE
 END-MON. *See* MONITOR/END-MON
 ENDMON, 102
 ENDSRVJOB, 208–209

- Entity Relationship Diagrams (ERDs), 359–363, **360**, **361**, **362**
- Entry point modules, 14
- EOF, 76–77, 135
- EQUAL, 78–79, 135, 175
- ERROR, 122
- Error management
 - in binding, 25–26, 27
 - in compiling, 25–26, 27
 - ENDMON and, 102
 - ERROR in, 122
 - in functions, 46–47, 101–102
 - MONITOR in, 101–102
 - ON-ERROR in, 101–102, 175
 - RPGLEDOC tool for, 151
 - SQL and, SQLCOD and, 291–292
- EVAL, 53, 79, 80, 164, 169, 180, 205
 - debugging and, 203
 - operation codes replaced by, 167
- EVALR, 179
- Excel-like string operations, 106–115
- EXEC SQL/END SQL in, 280, 281, 283, 332
- EXISTS, 243–244, 245, 331
- EXPORT, 34
- Exporting a module, 14, 22, 154
- EXTRACT, 169, 178, 263–264, 269

- F**
- F-specs, 213
- FETCH, 290–292, 293, 294
- Field reference files, 230
- Fields, 216, 230, 231
 - duplicated names in, 147–149
 - free-format coding and, 227
 - naming conventions for, duplicated names and, 147–149
 - PREFIX keyword in, 147–149
 - standalone, 227
- File definitions, in free-format coding and, 213–216, 227
- Files
 - duplicated field names in, 147–149
 - naming conventions for, 146–147
- FIND, 193, 194
- First normal form (1NF), 358–359
- Fixed-format code, 29, 84, 161. *See also* Free-format code
 - converting to free-format, 166–167, 177
 - embedded SQL and, 280–281
 - EXEC SQL/END SQL in, 280
 - keywords in, with free-format counterparts, 218–219
- FLOAT, 82–84, 135, 218
- FOR, 172
- Foreign Key, 233, 315, 317
- Formatting Code
 - edit codes for, EDITC, 85–87
 - edit words for, EDITW, 85–87
 - INDENT (pipe character) for, 21
 - parameters and, 60–70
- Forward engineering approach to modernization, 350
- FOUND, 78, 135, 175
- Free-format code, 29, 84, 157–180
 - array-related opcodes and, 172–176
 - CABxx and, 171–172
 - CASxx in, 170–171
 - CAT in, 175
 - CHAIN in, 165
 - column restrictions in, 159, 176, 226, 228
 - comments in, 160, 165, 167, 177
 - COMP in, 170–171
 - compiling, 159
 - compound assignment (+-1) in, 166
 - constants in, 220–221, 227
 - control statements in, 211–213, 227
 - converting fixed-format to, 166–167, 177
 - COPY in, 226
 - D-spec in, 216, 227
 - data definitions in, 216–226, 227
 - data structures in, 222, 227–228
 - ELSEIF block in, 162–163
 - embedded SQL and, 281
 - end of instruction (;) semicolon indicator in, 159, 167, 177
 - EVAL in, 164
 - F-specs in, 213

- fields in, 227
 - file definitions in, 213–216, 227
 - fixed-format vs., code sample of, 161
 - FREE and END-FREE directives in, 159, 161, 167, 177, 211, 226, 283
 - function headers converted to, 224–226
 - functions and, 53
 - GOTO in, 171–172
 - H-specs in, 211–213, 227
 - I specs in, 225–226, 225
 - IFxx/ANDxx/ORxx op codes in, 168
 - indentation in, 161–162
 - ITER in, 172
 - KEYED in, 213–216, 227
 - keywords in, with fixed-format counterparts, 218–219
 - LOOKUPxx and TLOOKUPxx and, 174–175
 - loops in, 172
 - nested code in, 162
 - O specs in, 225–226
 - one statement per line in, 159, 177
 - operation codes in, 160–166, 177
 - operation codes not used in, 167–176, 178
 - OVERLAY in, 222, 228
 - positional indicators not used in, 160, 177
 - procedures in, 222–224, 228
 - programming languages and, 158–159
 - pros and cons of, 157–159, 176
 - RD*i* (Rational Developer for *i*) for automatic conversion to, 166–167
 - READE in, 164
 - rules for, 159–160, 176–177
 - SELECT block in, 164
 - spaghetti code and, 171
 - structured programming syntax and, 168
 - SUBARR in, 172–174
 - TAG in, 171–172
 - USAGE in, 213–216, 227
 - V7.1 Technology Refresh (TR) 11 in, 226
 - V7.2 Technology Refresh (TR3) and, 160, 226
 - variables in, 216–220
 - /FREE, /END-FREE, 159, 161, 167, 177, 211, 226, 283
 - Fresche Legacy, 407
 - FROM, 233, 244
 - Function headers, free-format conversion of, 224–226
 - Functions, 10, 14, 41–54, 331
 - building, using BIFs, 98–118
 - built-in (BIF). *See* Built-in functions (BIFs)
 - calling and, 43
 - column, in SQL. *See* Column functions, SQL
 - commenting code for, 46
 - convert string to numeric value (Cvt_To_Numeric), 99–106
 - CREATE FUNCTION in, 304–307
 - date conversion, 47–50
 - documenting, 149–150
 - embedded SQL and, 286–289
 - error management for, 46–47, 101–102
 - formatting code for, 53
 - free-format, 53
 - functions within, 47–50, 54
 - input parameter for, 98
 - line position of (P-line, etc.) for, 44
 - MONITOR and, 47
 - naming conventions for, 42–44, 45–47, 53, 99, 144–145
 - nested, 51–53, 54
 - P prefix for, 138
 - parameters in, 43–44, 46, 98
 - procedures converted into, 42, 53
 - readability of code for, 42–44
 - return value concept for, 45–47, 53
 - “shell,” 48, 52
 - SQL and, 235
 - user-defined, 51. *See also* User-defined functions, 299
- G**
- GOTO, 171–172, 177, 178, 292
 - GRANT, 387
 - GRAPH, 218
 - GROUP BY, 235–236, 242
 - aggregate functions in SQL and, 245

H

H-specs, 211–213, 227
 binding using, 23
 HIVAL, 105–106
 Host variables, 332
 embedded SQL and, 282–285
 HOUR, 264
 HOURS, 130, 136, 283

I

I Navigator Run, 274, 303
 I prefix, 141
 I specs, 225–226
 I/O server isolation in, with code, 374–384
 IBM Cryptographic Access Provider, 387
 IBM Data Studio, database modernization/
 restructuring and, 363–364
 IBM InfoSphere Data Architect, database
 modernization/restructuring and, 364–365,
 372
 Ifxx, 168, 179
 ILE, OPM programming vs., 1–6
 ILE debugger. *See* Debugging
 ILEDOCS tool for, 152
 Importing a module, imports, 14, 22, 155
 IN/INOUT, 241, 302
 /INCLUDE, 32, 37–39, 40
 code organization and, 154
 debugging and, 184–186, **185**, 207
 IND, 218
 INDENT, 21
 Indentation of lines, free-format coding and,
 161–162
 Indexes in SQL, 239, 231, 320–322
 Indicators, named, 141–143
 Initializing work variables, 100
 Inner joins, 236–240, **239**
 INSERT, 230, 231, 240–242, 254–255, 331
 Data File Utility (DFU) and, 240
 DISTINCT, 242
 embedded SQL and, 295–298
 GROUP BY, 242
 IN and, 241

SELECT and, 240, 241
 subselects and, 241
 triggers and, 323–330
 INT/INTH, 81–82, 84, 135, 218
 Integers, 284
 Interactive Source Debugger (ISDB), 6, 181.
See also Debugging
 INTO, 282, 284, 289
 ITER, 172, 177, 178

J

Java, 151
 Javadoc, 151
 JDBC, SQL and, 277–278, **278**
 Joins, 236–240, **239**

K

K prefix, 140
 KDS structure, 179
 Key expressions, 321
 Key fields, naming conventions for, 140
 KEYED, free-format coding and, 213–216, 227
 Keys
 for security, 388
 SQL and, 233
 Keywords. *See* Parameters and keywords
 KFLD, 179
 KLIST, 179

L

Languages. *See* Programming languages
 Last day of month function (LastDayofMonth),
 119–124
 LAST_DAY, 268–269
 LCASE, 251–252, 331
 LEAVE/LEAVESR, 172, 177, 178, 180
 LEFT, 106–115, 193, 258–259
 Left joins, 237–240, **239**
 LEN, 68–69
 Libraries, 230
 LIKE, 178, 235
 Linking. *See* Binding
 LIST debugging view, 186–188, **186**, **187**, 209

- Logical files, 230, 231
 database modernization/restructuring and,
 368, 370
 naming conventions for, 146–147
- Logical models of databases, 358
- Look-ahead fields, 284
- Looksoftware, 407
- LOOKUP:xx, 174–175, 179
 EQUAL and, 79
- Loops, 172
- LOWER, 111–115, 251–252
- LPAD/RPAD, 257
- M**
- Maintaining code, 344
- Maximum variable length, 143
- MERGE, 331
- MHHZO/MHLZO, 169, 179
- MICROSECOND, 264, 283
- Microsoft SQL Server, 230
- MID string handling function in Excel,
 106–115
- MIDNIGHT_SECONDS, 266
- MINUTE, 264
- MINUTES, 130, 136, 283
- Missing parameters and NOPASS, 60–67,
 68–69, 99
- Missing parameters and OMIT, 64–67, 68–69,
 99
- MLHZO/MLLZO, 169, 179
- Model layer, in MVC, 401–403, **402**
- Model-view-controller (MVC) concept, 3, 337,
 393, 399–405, **400, 402**
- Modernizing your applications, 337–356
 Adsero Optima Foundation for, 365
 approaches to and levels of, 340–341,
 347–348
 benefits of, 343–345
 communications in, importance of, 347
 cost benefits of, 338, 344
 database restructuring in, 340–341, 357–392.
See also Database modernization/
 restructuring
 evaluating existing applications value for,
 341–343
 forward engineering approach to, 350
 functionality loss and, 349
 goal setting in, 346, 347–354
 help with, 346
 IBM Data Studio for, 363–364
 IBM InfoSphere Data Architect for, 364–365
 impact of, 349
 incremental steps toward, 346
 maintenance benefits of, 344
 methodology for, 365–391. *See also*
 Database modernization/restructuring
 model-view-controller (MVC) concept, 3,
 337, 393, 393, 399–405, **400, 402**
 modular programming as, 343–345
 overview of, 337–356
 process of, 347
 proof of concept (PoC) in, 346
 reengineering approach to, 349, 404–405
 refacing approach to, 349
 refactoring approach to, 349
 replacement approach to, 349
 resistance to, dealing with 345–346
 reverse engineering approach to, 350
 RPG Open Access (RPG OA) and, 352,
 405–407
 support of organization for, 346
 tips for, 345–347
 tools for, 363–365
 training requirements and, 349
 Unified Modeling Language (UML) in, 350
 user experience/user interface (UX/UI),
 339
 user interface (UI) in, 338–339, 393–408
See also User interface (UI)
- Modular programming, 343–345. *See also*
 Modernizing your applications; Modules
 multi-tier architecture approach to, 395–399,
395, 397, 398
 user interface (UI) and, 394–405. *See also*
 User interface (UI)
- MODULE object, 16–17

- Modules, 4, 5, 9–15. *See also* Modular programming
 - bind by copy, 12, **12**, 15, 20
 - binding, 10, 12, **12**, 14, 15, 19–27. *See also* Binding
 - binding directory for, 154–156
 - compiling, 20–21
 - Create RPG Module (CRTRPGMOD) and, 21, 182, 188–189, 285
 - entry point, 14
 - exporting, 14, 154
 - functions in, 14
 - importing, 14, 22, 155
 - MODULE object and, 16–17
 - multi-tier architecture using, 395–399, **395**, **397**, **398**
 - naming conventions for, 145–146
 - online source for, 10
 - OPM coding vs. 10–11, **11**
 - PGM object and, 9, 17
 - procedures in, 14, 34
 - program flow through, 12–13, **12**, 153, **153**
 - programming languages and, 13
 - reusability concepts and, 11–15
 - service programs and, 15–16, 21–24, 153–154
 - shortcuts for, 26
 - SRVPGM object and, 15, 17
 - user end point in, 15
 - user interface (UI) example of use of, 394–405. *See also* User interface (UI)
 - user-defined options for, 26, 27
 - MONITOR/END-MON, 47, 101, 175, 180
 - EVAL and, 80
 - MONTH, 264
 - MONTHNAME, 263
 - MONTHS, 122, 123, 136, 283
 - MONTHS_BETWEEN, 267–268, 267
 - “Morphing” parameters and VARSIZE, 67–69
 - MOVE/MOVEL, 167, 179
 - numeric conversion using, 84, 87–88
 - removing from code, 79–81
 - string operations and, 89
 - MOVEA, 172, 179
 - MSECONDS, 130, 136
 - MULT, 167, 179
 - Multi-tier architecture approach, modular programming, 395–399, **395**, **397**, **398**
 - Multiple-dimension arrays, 284
 - Multivalued attributes, 361
 - MVR, 169, 179
- ## N
- Named Indicators, 141–143
 - Naming Conventions, 137–149, 156
 - aliasing and ALIAS in, 236, 322, 368
 - C prefix in, 141
 - for constants, 141, 220–221
 - for copy members, 104
 - database modernization/restructuring and, 368
 - for fields, duplicated names in, 147–149
 - for files, 146–147
 - for functions, 42–44, 53, 99, 144–145
 - I prefix in, 141
 - K prefix in, 140
 - for key fields, 140
 - length of, 143
 - for logical files, 146–147
 - for modules, 145–146
 - named indicators in, 141–143
 - P prefix in, 138
 - for physical files, 146–147
 - PREFIX keyword in, 147–149
 - for procedures, 33, 42, 99, 144–145
 - proper names in, 143–144
 - SQL and, 230, 273
 - for tables, 236
 - underscore character in, 143
 - for variables, 217
 - for variables and, prefixes for, 138–143
 - for variables and, proper names, 143–144
 - W prefix in, 139
 - Navigational commands in debug, 193–196, **193**
 - Nested code, free-format coding and, 162
 - Nested functions, 51–53, 54
 - Netbeans, 151

- NEXT, 193
 NEXT_DAY, 269
 NONE, 209
 debugging and, 188
 NOPASS, 60–67, 68–69, 71, 99
 Normalization of database, 358–359, 371–372
 NOT, 78, 234
 NOW, 331
 Null values in SQL, 311
 NULLIND, 70, 71
 Numeric data
 character conversion of, using CHAR, 84–85
 convert to FLOAT, 82–84
 decimal, DEC/DECH for, 80–81
 edit codes for, EDITC, 85–87
 edit words for, EDITW, 85–87
 integer conversion, INT and INTH for,
 81–82, 84
 MOVE/MOVE in conversion of, 84, 87–88
 TESTN in, 175
- O**
- O specs, 225–226
 OBJECT, 218, 219
 OCCUR, 169, 179, 284
 OFF, 100
 OMIT keyword, 64–67, 68–69, 71, 99
 ON, 100
 ON-ERROR, 101–102, 175, 180
 Online sources for modules, 10
 OPEN, 292, 294
 EOF with, 77
 Open Query File (OPNQRYF), 292–294
 SQL cursors instead of, 292–294
 OPNQRYF, 292–294
 Operation codes, 160
 array-related, 172–176
 built-in functions (BIFs) replacing, 169, 177
 CASxx, 170–171
 CAT in, 175
 COMP, 170–171
 free-format coding and, 160–166, 177
 no longer used, in fixed-format, 167–176,
 178
 TESTN in, 175
- Operators
 compound assignment (+=1) in, 166
 SQL and, 234
 OPM RPG programs, debugging, 207
 OPTIONS, 60, 71, 98, 225
 debugging and, 186, 188, 207
 OR, 233, 244
 Oracle, 230
 ORDER BY, 234, 235
 Organizing code. *See* Code organization
 Original Programming Model (OPM) vs. ILE,
 1–6
 ORxx op codes, 168, 179
 OTHER, 178
 Outer joins, 237–240, **239**
 OVERLAY, 222, 228, 284
 Override Database File (OVRDBF) and ALIAS
 instruction, 322
- P**
- P fields, 143
 P prefix, 138
 PACKED, 218
 Parameters and keywords, 30, 40, 55–71
 ADDR function and, 65, 66
 advantages of using, 55
 C-like strings in, 69
 CONST, 57–59, 70
 copy members in, 104
 DBGVIEW, 182
 DDS keywords ignored during SQL
 generation, **369**
 debugging and, 55
 formatting and, 69–70
 free-format vs. fixed-format, list of, 218–219
 functions and, 43–44, 46, 98
 initializing work variables for, 100
 input type, keyword selection for, 59
 LEN function and, 68–69
 missing, allowing with NOPASS, 60–67,
 68–69, 99
 missing, OMIT keyword for, 64–67, 68–69,
 99

- “morphing” type, using VARSIZE for, 67–69
- NOPASS keyword for, 60–67, 68–69, 71, 99
- NULLIND, 70, 71
- OFF, 100
- OMIT keyword for, 64–67, 68–69, 71, 99
- ON, 100
- OPTIONS, 60, 71
- P prefix for, 138
- PARMS function and, 65, 66
- passing, 55, 99
- procedure, 98
- QCMDEXC and, 67–69
- RIGHTADJ and, 69–70, 71
- SOURCE in, 182
- STRING, 69, 71
- System() function and, 69
- TRIM, 68–70, 71
- VALUE, 56–59, 70
- VARSIZE, 67–69, 71, 100, 101, 108
- PARM, 179
- PARMS, 65, 66
- Passing parameters, 99
- PGM object, 9, 17
 - compiling, 24
 - program flow in, 153, **153**
- Physical files, 230
 - database modernization/restructuring and, 368, 370
 - naming conventions for, 146–147
- Physical models of databases, 358
- Pipe character, 21
- PLIST, 179
- POINTER, 218, 219
- Pointers, 284
- POS, 284
- Positional indicators vs. free-format code, 160, 177
- PR/PI definitions, 179
- PREFIX, 147–149
- Prefixes in naming variables, 138–143
- PREPARE, 294–296, 332
- Prepared statements, embedded SQL and, 298
- Presto suite, 407
- PREVIOUS, 193
- PRIMARY KEY, 316
- Privileges, SQL, 387
- Procedures, 10, 14, 29–40
 - building, 30–37
 - building, using BIFs, 98–118
 - calling, CALL, CALLP and, 33, 40
 - code for, 33–34
 - compiling, 34–35
 - /COPY in, 32, 37–39, 40
 - CREATE PROCEDURE for, 300–303, 300
 - DCL-PROC/END-PROC delimiters for, 222, 223–224, 228
 - defining, 222–224
 - documenting, 149–150
 - EXPORT in, 34
 - fixed-format, 29
 - free-format coding and, 29, 228
 - functions created from, 42, 53
 - /INCLUDE in, 32, 37–39, 40
 - input parameters for, 98
 - line position (P line, D line, etc.) for, 34–35, 39
 - modules and, 34
 - naming conventions for, 32, 42, 99, 144–145
 - OPMexample of, 30, **30**, **31**
 - OPM program transformed into, 36–37, **36**
 - P prefix for, 138
 - parameters and keywords for, 30, 31–32, 40, 98
 - PLIST for, 31
 - program flow and, 37
 - prototype definition in, 32
 - source file for, QCPYLESRC and, 32
 - stored. *See* Stored procedures
 - subprocedures and, 29
 - when and why build, 35–37
- PROCPT, 219
- Profound Logic, 407
- Program flow through modules, 12–13, **12**, 153, **153**
- Programming languages, 5
 - free-format coding and, 158–159
 - module development in, 13

SQL and, 230
triggers and, 324
Proof of concept (PoC), in modernizing your applications, 346
PROPER case change function in Excel, 111–118
Proper names, 143–144
Prototype definition, 32, 216

Q

QCMDEXC API, 67–69
QCPYLESRC, 32, 154
QSQLRC, 303
QUARTER, 266

R

RDi (Rational Developer for i) for free-format conversion, 166–167
RDi Database Explorer, 279
READ, 288, 289
Readability of code, 42–44
READE, 164
READP, 289
READxx, 75–76
REALLOC, 169, 179
Record Level Access (RLA), 289
Records, 230, 231
Recursive relationships, in ERDs, 361–362, **362**
Reengineering approach to modernization, 349–354, 404–405
Refacing approach to modernization, 349–354
Refactoring approach to modernization, 349–354
References, module references, 14, 22
REM, 125–126, 134, 136, 169, 179
REPLACE, 94–96, 135, 253–254
Replacement approach to modernization, 349–354
Repositories for code, 3
Reserved words, in SQL, 233
Retrieve day of week (Rtv_DayOfWeek) in, 126–127
RETURN, 44

Return values, 216
functions and, 45–47, 53
Reusability of code, 3, 4, 5
modules and, 11–15
programming language and, 5
Reverse engineering approach to modernization, 350
REVOKE, 387
RIGHT, 106–115, 193, 224–226, 258–259
Right joins, 237–240, **239**
RIGHTADJ, 69–70, 71
ROWIDs, 390–391
Rows, 230, 231
ILE RPG, OPM programming vs., 1–6
Rpg Next Gen, 152
RPG Open Access (RPG OA), 352, 405–407
RPGLE, 285
RPGLEDOC tool, 151
Rtv_DayOfWeek, 126–127
RUN, 273
Run SQL Scripts tool, 274–279, **275–279**
RUNSQLSTM, 303
Runtime services, 4, 6

S

Scalar functions, 248–259, 331
SCAN, 92, 93–96, 111, 116–118, 135, 169, 177, 179
SCANRPL, 96–98, 136
Schemas, 230, 308–310
Screen scraping, 351–352, 405
SECOND, 264
Second normal form (2NF), 359
SECONDS, 130, 136, 283
Security
authorization and privileges in, GRANT and REVOKE, 387
database modernization/restructuring and, 386–391
encryption in, 387–388
keys, 388
ROWIDs, 390–391
sequence objects, 389–390

- SELECT, 164, 178, 230, 231, 232–240, 271, 284, 288, 289, 331
 - alias with, 236
 - AND/OR, 233
 - BETWEEN, 234
 - cursors and, 289
 - DELETE and, 245
 - DESC, 234
 - as embedded SQL, 281–285
 - foreign keys with, 233
 - FROM, 233
 - functions with, 235
 - GROUP BY, 235–236
 - inner joins and, 236–240, **239**
 - INSERT and, 240, 241
 - INTO, 282, 284
 - joins in, 236–240, **239**
 - left/right joins in, 237–240, **239**
 - LIKE, 235
 - NOT, 234
 - ORDER BY, 234, 235
 - outer joins in, 237–240, **239**
 - output options for, 273
 - scalar functions and, 248–259
 - subselects and, 241
 - UPDATE and, 242, 244
 - WHERE, 233, 284
- Self-documenting coding, 151–152
- Semicolon (;) end of instruction indicator, 159, 167, 177
 - SQL and, 278
- Sequence objects, 389–390
- Service jobs
 - End Service Job (ENDSRVJOB) in, 208–209
 - Start Service Job (STRSVRJOB) in, 208
- Service programs, 15–16, **16**, 21–24
 - Create Service Program (CRTSRVPGM) in, 22–24
 - debugging and, 206–207, **206**
 - module links through, 15–16, **16**
 - organization of, 153–154
 - program flow in, 153, **153**
- Services screen (F13) in SQL, 271–272, **272**
- Session Attributes screen in SQL, 272, **272**
- SET, 207, 244, 283–284
- SETGT, 76
 - EOF with, 77
- SETLL, 76
 - EOF with, 77
 - EQUAL and, 78–79
- SETOFF/SETON, 179
- SETxx, 75–76
- Shared code, 3, 6
- Sheehy, Patrick, 374
- “Shell” functions, 48, 52
- SHTDN, 169, 179
- SIZE, 284
- SOURCE debugging view, 182, 184–186, **184**, **185**, 209
- Spaghetti code, 171
- SQL, 229–333. *See also* DELETE; INSERT; SELECT; UPDATE
 - aggregate functions in, 245–248
 - alias in, 236
 - ALIAS in, 322, 368
 - authorization and privileges in, GRANT and REVOKE, 387
 - BETWEEN in, 234
 - case insensitivity of, 233
 - catalogs in, 231
 - CL commands and, 276–277
 - collections in, 230–231
 - column functions in, 245–270
 - columns in, 231
 - commands in. *See individual commands*
 - commenting in, 277
 - Commitment Control in, 273
 - cursors in, 289–296
 - Data Definition Language (DDL) in, 229, 307–330. *See also* Data Definition Language (DDL)
 - Data Manipulation Language (DML) in, 232–245
 - data types in, as host variables, 284
 - database management systems using, 230
 - DDS keywords ignored during SQL generation, **369**

- description and history of, 229–230
- double click to run statements in, 276–277, **277**
- DROP vs. DELETE in, 322–323
- editors for, 285
- embedding, in RPG code. *See* Embedded SQL
- error management, SQLCOD and, 291–292
- executing a command in, 273–274
- exit screen in, 274, **274**
- foreign keys in, 233
- freezing panes in, 274
- index in, 231, 320–322
- JDBC and, 277–278, **278**
- joins in, 236–240, **239**
- key expressions in, 321
- keys, 233
- language support for, 230
- LIKE operator in, 235
- logical files in, 231
- naming conventions in, 273
- naming conventions in, vs. RPG, 230
- navigating through, 273–274
- NOT in, 234
- null values in, 311
- opening separate windows for statements in, 278–279, **279**
- operators in, 234
- Override Database File (OVRDBF) and ALIAS instruction in, 322
- PREPARE and, 294–296
- prepared statements in, 298
- reserved words in, 233
- rows in, 231
- RPG code embedded in SQL, 299–307. *See also* Embedding RPG code in SQL
- RPGLE and editor recognition of, 285
- Run SQL Scripts tool and, 274–279, **275–279**
- saving scripts in, 279
- scalar functions in, 248–259
- schema in, 230–231, 308–310
- SELECT output options in, 273
- semicolon (;) end of statement delimiter, 278
- Services screen (F13) in, 271–272, **272**
- Session Attributes screen in, 272, **272**
- SQL Communications Area (SQLCA) in, 291
- SQLCOD variable in, 291–292
- SQLRPGLE and, 285
- Start Interactive SQL Session (STRSQL) and, 270
- statement processing in, RUN and, 273
- stored procedures in, 234, 299, 300–303
- subselects in, 241
- tables in, 231, 310
- triggers in, 323–330
- user-defined functions in, 299, 300, 303–307, **307**
- using, 270–279
- views in, 231, 318–320
- SQL Communications Area (SQLCA), 291
- SQLCOD, 291–292
- SQLRPGLE, 285
- SQRT, 169, 179
- SRVPGM object, 15, 17
 - compiling, 22–23
 - Create Service Program (CRTSRVPGM) in, 22–24
- Standalone fields, 227
- Start Debug (STRDBG), 21, 190–193, **191**, **192**, 194–196, **194**, **195**, 201
- Start Interactive SQL Session (STRSQL), 270
- Start Service Job (STRSVRJOB), 208
- STEP, 203, 205
- Step-by-step debugging session, 201–205
- STMT debugging view, 183–184, **183**, 190, 209
- Stored procedures, 234, 299, 300–303
 - CREATE PROCEDURE for, 300–303
- STRDBG, 21, 190–193, **191**, **192**, 194–196, **194**, **195**, 201
- STRING, 69, 71
- String data, 88–89
 - built-in functions (BIFs) and, 88–98
 - CAT in, 89, 175
 - change case in, Chg_Case function for, 111–118
 - change case in, XLATE, 111–115
 - concatenation in, 259

- convert string to numeric value (Cvt_To_Numeric) for, 99–106
 - Excel-like, 106–115
 - find and replace text within, 92–98
 - INSERT, 254–255
 - LCASE, 251–252
 - LEFT, 258–259
 - LEFT/RIGHT, in Excel, using SUBST instead, 106–115
 - LOWER, 251–252
 - LPAD/RPAD, 257
 - MID, in Excel, using SUBST in, 106–115
 - MOVE in, 89
 - REPLACE in, 94–96, 111, 253–254
 - RIGHT, 258–259
 - SCAN in, 92, 93–96, 111, 116–118
 - SCANRPL in, 96–98
 - SUBST, 90–91, 106–115
 - TRANSLATE, 252–253
 - TRIM/LTRIM/RTRIM, 255–257
 - TRIM/TRIML/TRIMR, 91–92
 - VARCHAR, 250–251
 - XLATE, 111–115
 - STRSQL, 270
 - STRSVRJOB, 208
 - Structured programming syntax, 168
 - SUB, 167, 179
 - SUBARR, 172, 179
 - SUBDT, 122, 123, 130, 136, 178, 263–264
 - SUBDUR, 123, 169, 179
 - Subprocedures, 29. *See also* Procedures
 - Subselects, 241
 - SUBST, 90–91, 106–115, 135, 169, 179
 - SUBSTR, 258–259, 258
 - SUM, 235, 245, 331
 - System(), 69
- T**
- Tables, 230, 231, 284, 310
 - aliasing in, 236
 - ALTER TABLE and, 315–318
 - constraints on, 315–318
 - CREATE TABLE and, 310–318
 - foreign keys and, 315, 317
 - joins in, 236–240, **239**
 - naming, 236
 - TAG, 171–172, 177, 180, 292
 - Target Release (TGTRLS) parameter setting in, 189–190
 - TEMBO, 374
 - Test program (TST_NBROPS), 103
 - TESTB, 169, 180
 - TESTN, 175, 180
 - TESTZ, 169, 180
 - TGTRLS, 189–190
 - Third normal form (3NF), 359
 - TIME, 129, 133, 136, 169, 180, 218, 219, 261
 - Time data type, 259–260
 - Time formats, 129–130
 - Time, 129–135, 259–260
 - calculate time until weekend (Clc_HowLongUntilWE) in, 130–135
 - DIFF, 132
 - formats for, 129–130
 - HOURS in, 130
 - MINUTES in, 130
 - MSECONDS in, 130
 - REM, 134
 - SECONDS in, 130
 - SUBDT with, 130
 - TIME, 133
 - TIME in, 129
 - Time until weekend (Clc_HowLongUntilWE) in, 130–135
 - TIMESTAMP, 180, 218, 331
 - Timestamp data type, 259–260
 - TIMESTAMP, 261–262
 - TIMFMT, 219
 - TLOOKUPxx, 174–175, 179
 - Toad, 279
 - TOP, 193
 - TRANSLATE, 252–253
 - Triggers, 323–330
 - BEGIN/END in, 328
 - TRIM/LTRIM/RTRIM, 68–79, 71, 135, 255–257, 331
 - TRIM/TRIML/TRIMR, 91–92
 - TST_NBROPS, 103

U

UCS2, 218
 UPDATE, 284
 UDAY, 284
 UDPATE, 232
 UMONTH, 284
 Underscore character in names, 143
 Unified Modeling Language (UML), 350
 UNIQUE, 315, 316, 321
 UNS, 218
 Unsupported operation codes for fixed-format, 167–176
 UPDATE, 231, 242–244, 283, 284, 331
 AND/OR, 244
 Data File Utility (DFU) and, 242
 embedded SQL and, 295–298
 EXISTS, 243–244
 FROM, 244
 keyed tables and, 243
 SELECT and, 242, 244
 SET, 244
 triggers and, 323–330
 WHERE, 243, 244
 UPPER case change function in Excel, 111–115
 USAGE, in free-format coding, 213–216, 227
 User end point, 15
 User interface (UI), 338–339, 351–353, 393–408
 model-view-controller (MVC) concept in, 393, 399–405, **400**, **402**
 modular programming techniques for, 394–405
 multi-tier architecture approach to, 395–399, **395**, **397**, **398**
 reengineering, using MVC, 404–405
 refacing approach to modernizing, 351–353
 refactoring, 353–354
 RPG Open Access (RPG OA) and, 352, 405–407
 screen scraping in, 351–352, 405
 separating code for, important reasons for, 393–394

User-defined functions (UDFs), 51, 299, 300, 303–307
 built-in functions (BIFs) and in, 98–118
 CREATE FUNCTION in, 304–307
 User-defined options, 26, 27
 UX/UI. *See* User interface (UI)
 UYEAR, 284

V

V7.1 Technology Refresh (TR) 11, 226
 V7.2 Technology Refresh (TR3), 226
 free-format coding and, 160
 Validation
 database modernization/restructuring and, 372–374
 I/O server isolation in, with code, 374–384
 VALUE, 56–59, 70
 VARCHAR, 218, 220, 225, 250–251, 314, 331
 VARGRAPH, 219, 220
 Variables
 compound assignment (+=1) in, 166
 defining, 216–220
 initializing work variables and, 100
 length of, maximum, 143
 naming conventions for, 217
 naming conventions for, prefixes for, 138–143
 naming conventions for, proper names, 143
 P prefix for, 138
 W prefix for, 139
 VARSIZE, 67–69, 71, 100, 101, 108, 225
 VARUCS2, 219, 220
 VARYING, 98, 225
 View layer, in MVC, 404
 Views in debugging, 182–188, 209
 Views in SQL, 230, 231, 318–320

W
 W prefix, 139
 Warehouse shelf master table definition, 333
 Watch conditions, 198–200, **200**, 203–205, 210
 WATCH, 198–200, **200**, 203–205
 Web Studio Development Client, 166
 WEEK, 265

WEEK_ISO, 265
WHEN, 178, 180, 205
WHENxx, 168
WHERE, 233, 243, 244, 245, 282, 284, 298,
 331
 aggregate functions in SQL and, 245
Wings, 406
Work variables, naming conventions for,
 139
Work with Binding Directory (WRKBNDDIR)
 in, 24
WRKBNDDIR, 24

X
XFOOT, 169, 180
XLATE, 111–115, 136, 169, 177, 180, 252–254

Y
YEAR, 264
YEARS, 123, 136, 283

Z
Z-ADD, 167, 180
Z-SUB, 167, 180
ZONED, 219