Index

ABS numeric function, 355
Active State Perl, SQL*Plus with, 61
ADD_MONTHS, 360
AFTER DELETE ROW trigger, 202
AFTER DELETE STATEMENT trigger, 202
AFTER-INSERT-ROW (AIR) trigger, 172–174, 177, 179–180, 186–187
AFTER STATEMENT trigger, 198–201
Alerts, email notification, 398–402
ALL_OBJECTS.LST, 218–219
ALL_SOURCE, 290
ALTER-INSERT-STATEMENT trigger, 193–194
ALTER TABLE statement, 74–75, 93
American National Standards Institute (ANSI), 2, 5–6
  SQL standard, 5–6
Application partitioning, 266–268
Application Programming Interface (API), 218–221
  DBMS_ALERT package, 395–398
  DBMS_PIPE package, 387–392
  learning how to use, 220
  viewing packages, 218–219
Application tablespace, 22–25
Arguments, command line, 52–57
Arithmetic expressions, 301
ASCII string function, 354
Assignments and statements, 291–292; *see also* PL/SQL language features
Association of Computing Machinery (ACM), 2
AUTOTRACE command, 64–67
  syntax, 65–66
  types of output, 65–67

Batch command files, 60–61
BEFORE-INSERT-ROW (BIR) trigger, 172–174, 179–180, 186–187
BEFORE STATEMENT trigger, 198–201
BFILE, 20
BINARY_INTEGER, 307
BLOB, 19
Blocks with exception handlers, 367
Booch (Grady Booch) diagram, 261, 267*f*
Boolean expression, 292–293, 304–305
Business rules
  enforcement. *see* Declarative constraints
  table with Oracle constraints and, 183–191

Cardinality, check constraints and, 132
CASCADE option, 93
CASE statements, 328–333
  DECODE within the SELECT, using, 332–333
  searched CASE statement, 329–330*f*
  with SELECTOR, 330–331
  using within the SELECT statement, 331–332
CATPROC.SQL, 386
CEIL numeric function, 356
CHANGE commands, SQL*Plus, 36–46
CHAR, 17, 306
Character datatypes, 16–17
CHECK constraints, 126–133
  cardinality, 132
  designing for, 132–133
  lookup tables *versus,* 131–132
  multicolumn, 128–130
  students table example, 131
  supplementing unique constraints, 130–131
  with UNIQUE constraints, combining, 101
CHECK_CONSTRAINT_VIOLATION, 376
CHECK_PLSQL_OBJECTS, 226, 229
CHILD-INSERT, 107–109
CHILD-UPDATE, 107–109
CHR string function, 354
CLOB, 19, 322
COLUMN command, 48–49
Column constraint clause, 72–73
Command files, batch, 60–61
Command line, SQL*Plus, 35–46, 52–57
  arguments, 52–57
  CHANGE commands, 36–46
Comments, 289–290
  C-style, 289
Compilation dependency, PL/SQL environment, 241–244
Complex rule enforcement, 205–209
CONCAT string function, 354

Connecting
  to an infrastructure using SQL*Plus, 32–34
  to SQL*Plus, 30–34
Constraints
  CHECK, 126–133
    designing, 132–133
  data model with. *see* Students data model
  declarative, 69–141
  foreign key, 104–126
  lookup tables *versus* check constraints, 131–132
  modifying, 135–136
  multicolumn check, 128–130
  NOT NULL, 133–134, 297
  primary key, 70–99
  UNIQUE, 99–104
Constraint scripts, samples, 161–170
  CHECK constraint rule, 163–165
  constraints of a table, 161–162
  constraint status, 167
  querying a constraint name, 162–163
  querying child tables, 166
  querying parent tables, 165–166
Constraint views, 158–161
  USER_CONS_COLUMNS, 158–159
  USER_CONSTRAINTS, 159–161
COSH (X) numeric function, 357
COS (X) numeric function, 357
COURSES, sample data, 151
CREATE_PIPE, 391
CREATE TABLESPACE, 22–24
CURRVAL sequence sttribute, 82–87
Cursors, explicit, 381–383

Data abstraction, 268–271
Database
  access with SQL, 377–383
    cursor FOR LOOP, 377–378
    explicit cursors, 381–383
    inserts and updates, 378–381
    select when expecting a single row, 378
  relational tables, introduction to, 1–26
  SQL, 6–7
  student data model, example tables, 11–14
    create table script, 12
    data model view, 12
    describing, 12
    table data, 13–14
Data definition language (DDL), 12, 14, 145–150
Data dictionary, 25–26, 153–170
  constraint scripts, samples, 161–170
    CHECK constraint rule, 163–165
    constraints of a table, 161–162
    constraint status, 167
    querying a constraint name, 162–163
    querying child tables, 166
    querying parent tables, 165–166
    validated, 168–170
  constraint views, 158–161
    USER_CONS_COLUMNS, 158–159
    USER_CONSTRAINTS, 159–161
  and system tablespace, 25*f*
  views, 154–157
  what you can see, 154–155
Data manipulation language (DML), 13–15
Data model with constraints. *see* Students data model
DATE datatypes, 18
Date functions, 359–364
  ADD_MONTHS, 360
  LAST_DAY, 360–361
  MONTHS_BETWEEN, 362
  NEW_TIME, 362–363
  NEXT_DAY, 363–364
  ROUND, TRUNC, 364
  SYSDATE, 359
  TO_CHAR, TO_DATE, 359–360
DBMS_ALERT package, 249, 384, 393–398
  interface description, 395–398
  signaling events with, 393–398
DBMS_LOB package, 22
DBMS_OUTPUT package, 213, 218–221, 235–238, 284, 349
  documentation, 235–238
DBMS_PIPE, 383–393
  interface description, 387–392
DBMS_PIPS, 249
DBMS_RANDOM, 357–358
DBMS_ROWID package, 22
DBMS_STANDARD package, 374
DBMS_STATS package, 43
DBMS_UTILITY.COMPILE_SCHEMA, 226, 228
Declarative constraints, 69–141
  CHECK, 126–133
  data loads, 138–141
  default values, 134–135

exception handling, 137–138
foreign key, 104–126
modifying constraints, 135–136
NOT NULL, 133–134
primary key, 70–99
UNIQUE, 99–104
Declarative part
of a function, 253–254
of a procedure, 250
Declaring variables, 302–304
DECODE statement, 6, 328
Default parameters, 278–281
DEFERRABLE option
foreign key constraints, 122–126
primary key constraints, 94–98
UNIQUE constraint, 103
Delete cascade, foreign key constraints, 109–110
DESCRIBE (DESC) command, 36–37
Designing check constraints, 132–133
Direct SQL, 7–8
Disconnecting from SQL*Plus, 34–35
Document Object Model (DOM), 21
DO UNTIL LOOP, 333*f*–335
DO-WHILE-DO LOOP, 338*f*–339
DUAL table, 63–64

Email notification with triggers and alerts, 398–402
Entity relationship diagram, 143
Error codes and messages, managing, 190–191
Error handling in PL/SQL, 103–104
with parent/child tables, foreign key constraint, 122
primary key constraints, 99
UNIQUE constraints, 103–104
Errors
ORA-00001, 76
ORA-00372, 374
ORA-00923, 38
ORA-02272, 136
ORA-02290, 127, 129
ORA-02291, 125
ORA-06510, 372, 376
ORA-06556, 392
ORA-20000, 375
Exception clause, 368–370
Exception handlers, 373
blocks with, 254
of a function, 254
of a procedure, 251
Exception handling, 137–138, 392–393
EXCEPTION_INIT, 375–376
Exceptions, 364–376
blocks with exception handlers, 367
exception clause, 368–370
EXCEPTION_INIT, 375–376
RAISE_APPLICATION_ERROR, 374–375
RAISE statement, 371–372
SQLCODE and SQLERRM, 370–371
unhandled exceptions and exception propagation, 372–373
user-defined, 365–366
Explicit cursors, 381–383
EXP (X) numeric function, 357

FEEDBACK command, 52
FLOOR numeric function, 356–357
Foreign key constraints, 104–126
DEFERRABLE option, 122–126
delete cascade, 109–110
four error types, 107–109
mandatory columns, 111–112
many-to-many relationships, 117–119
multiple parents and DDL migration, 114–117
PL/SQL error handling with parent/child tables, 122
referencing the parent syntax, 111–112
referential integrity across schemas and databases, 112–114
self-referential integrity, 120–121
FOR LOOP, 336–338, 337*f*
Formal parameter name, 276
%FOUND, 383
Functions
date, 359–364
and modes, 274–275
numeric, 355–357
PL/SQL, 252–255, 253*f*
key components of, 253–254
string, 354–355
string manipulation, 342–354

GEN_COUNTS.SQL, 51
GET_NEXT_RECORD, 341
GREATEST string function, 355

HEADING command, 52
Hello World program, 212–217
  building the HELLO procedure, 215

IF statements, 323–328
  DECODE and CASE in SQL, use of, 327–328
  IF-THEN-ELSE, 323*f*
  IF-THEN-ELSIF No ELSE, 324–325*f*
  IF-THEN-ELSIF with ELSE, 324*f*
  simple IF, 323
  statement expressions, 325–327
Implementation, SQL, 7–9
Index-by-tables, 311–313
INITCAP string function, 355
IN mode (default) parameter, 271–273
IN OUT parameters, 272–273, 352
Insert and update statements, 378–381
INSERT command, 37
INSERT_TEMP, 251–252
INSTR function, 344–348
Interface description, 387–392, 395–398
  DBMS_ALERT, 395–398
  DBMS_PIPE, 387–392
INVALID error, 224
INVALID_NUMBER, 284
%ISOPEN, 383

JDeveloper, 289

Korn Shell, SQL*Plus with, 57–60

Large-character objects datatypes, 19–20
Large objects (LOBs), 320–322
  creating a table to store, 320–321
LAST_DAY, 360–361
LAST_DDL_TIME, 226
LEAST string function, 355
LENGTH string function, 355
LN (X) numeric function, 357
LOG (A, X) numeric function, 357
Logical operators, 298–300
Login, SQL*Plus, 31*f*
LONG datatype, 22
Lookup tables *versus* check constraints, 131–132
Loops, 333–342
  DO UNTIL LOOP, 333*f*–335
  DO-WHILE-DO LOOP, 338*f*–339
  encapsulating the logic of, 339–342
  FOR LOOP, 336–338, 337*f*
  WHILE LOOP, 335*f*–336
LOWER string function, 355
LPAD function, 348–350
LTRIM function, 350–351

MAJOR_LOOKUP, sample data, 150
Many-to-many relationships, 117–119
Mathematical operations. *see* Arithmetic expressions
Modifying constraints, 135–136
MOD numeric function, 355
MONTHS_BETWEEN, 362
Multicolumn check constraint, 128–130
Multiple parents and DDL migration, 114–117
Multi-Threaded Server (MTS), 34

Named notation, 275–278
NCHAR, 17
NCLOB, 19
NEW_TIME, 362–363
NEXT_DAY, 363–364
NEXT_ITEM_TYPE, 391
NEXT_TOKEN, 345
NEXTVAL sequence attribute, 82–87
%NOTFOUND, 382
NOT NULL constraints, 133–134, 297
  combining with UNIQUE constraints, 101
  common syntax, 133
NOVALIDATE option
  primary key constraints, 98–99
  UNIQUE constraint, 103
NULL statement, expressions with, 293–298
NUMBER, 306–307
Number datatypes, 17
Number generation, random, 357–358
Numeric functions, 355–357
NVARCHAR2, 17
NVL operator, 296–297

Objects, 317–320
Optimal Flexible Architecture (OFA), 24
ORA-00001, 76
ORA-00372, 374
ORA-00923, 38
ORA-02272, 136
ORA-02290, 127, 129

ORA-02291, 125
ORA-06510, 372, 376
ORA-06556, 392
ORA-20000, 375
Oracle
  Application Programming Interface (API), 218–221
  constraints and business rules, table with, 183–191
  infrastructure, connecting to, 32–34
  interacting with. *see* SQL*Plus, interacting with Oracle
  PL/SQL packages, referencing, 217–221, 219*f*
  SQL constraints, 69–141
Oracle Advanced Queuing, 395
Oracle Enterprise Manager (OEM), 8
OUT parameters, 274
Overloading, 284–286

Package body, 264–266
Package specification, 257–264
  developing a specification, 261–264
  syntax and style, 258–261
PACK_MESSAGE, 384, 388
PAD function, 348–350
PAGESIZE command, 52
Parameters
  default, 278–281
  and modes, 271–284
    functions and modes, 274–275
    IN mode (default) as constant, 272–273
    OUT mode, 274
    IN OUT mode, 273
    scaling code with defaults, 281–282
    %TYPE, 282–284
  named *versus* positional notation, 275–278
PARENT-DELETE, 107–108
PARENT-UPDATE, 107–108
PARKING_TICKETS, sample data, 151
Pipe messages (DBMS_PIPE), sending, 383–393
  exception handling, 392–393
  interface description, 387–392
  send-receive example, 385–387
PLS_INTEGER, 307
PL/SQL environment, 211–245
  block, 215–217
  compilation dependency, 241–244
    package body referencing another package, 242*f*
    two interdependent packages, 243*f*
  Hello World program, 212–217
  procedures, dependencies among, 223–229
  referencing Oracle packages, 217–221, 219*f*
    Application Programming Interface (API), 218–221
    creation of the environment, 217–218
  sharing code, 238–241
  USER_DEPENDENCIES, 229–232
  USER_ERRORS, 244–245
  USER_OBJECTS, 221–223
  USER_SOURCE, 232–238
PL/SQL language features, 287–402
  arithmetic expressions, 301
  assignments and statements, 291–292
  Boolean expression, 292–293
  CASE statements, 328–333
    CASE within the SELECT, using, 331–332
    CASE with Selector, 330–331
    DECODE within the SELECT, using, 332–333
    searched CASE statement, 329–330*f*
  comments, 289–290
  database access with SQL, 377–383
    cursor FOR LOOP, 377–378
    explicit cursors, 381–383
    inserts and updates, 378–381
    select when expecting a single row, 378
  date functions, 359–364
    ADD_MONTHS, 360
    LAST_DAY, 360–361
    MONTHS_BETWEEN, 362
    NEW_TIME, 362–363
    NEXT_DAY, 363–364
    ROUND, TRUNC, 364
    SYSDATE, 359
    TO_CHAR, TO_DATE, 359–360
  email notification with triggers and alerts, 398–402
  exceptions, 364–376
    blocks with exception handlers, 367
    exception clause, 368–370
    EXCEPTION_INIT, 375–376
    RAISE_APPLICATION_ERROR, 374–375
    RAISE statement, 371–372

PL/SQL language features (*cont.*)
  SQLCODE and SQLERRM, 370–371
  unhandled exceptions and exception propagation, 372–373
  user-defined, 365–366
 expressions with NULL, 293–298
 IF statements, 323–328
  DECODE and CASE in SQL, use of, 327–328
  IF-THEN-ELSE, 323*f*
  IF-THEN-ELSIF No ELSE, 324–325*f*
  IF-THEN-ELSIF with ELSE, 324*f*
  simple IF, 323
  statement expressions, 325–327
 logical operators, 298–300
 loops, 333–342
  DO UNTIL LOOP, 333*f*–335
  DO-WHILE-DO LOOP, 338*f*–339
  encapsulating the logic of, 339–342
  FOR LOOP, 336–338, 337*f*
  WHILE LOOP, 335*f*-336
 numeric functions, 355–357
 random number generation, 357–358
 sending pipe messages (DBMS_PIPE), 383–393
  exception handling, 392–393
  interface description, 387–392
  send-receive example, 385–387
 signaling events with alerts (DBMS_ALERT), 393–398
  interface description, 395–398
 string concatenation, 300–301
 string functions, miscellaneous, 354–355
 string manipulation functions, 342–354
  INSTR function, 344–348
  LPAD, RPAD function, 348–350
  LTRIM, RTRIM function, 350–351
  REPLACE, 351–352
  SUBSTR function, 342–344
  TRANSLATE function, 353–354
 types, 304–322
  Boolean, 304–305
  index-by tables, 311–313
  large objects (LOBs), 320–322
  objects, 317–320
  records, 307–311
  %ROWTYPE, 311
  scalar, 305–307
  VARRAY and nested tables, 314–317
 variable declarations, 302–304
PL/SQL program units, 247–286
 application partitioning, 266–268
 data abstraction, 268–271
 functions, 252–255, 253*f*
  key components of, 253–254
 overloading, 284–286
 package body, 264–266
 package specification, 257–264
  developing a specification, 261–264
  syntax and style, 258–261
 parameters and modes, 271–284
  default parameters, 278–281
  functions and modes, 274–275
  IN mode (default) as constant, 272–273
  named *versus* positional notation, 275–278
  OUT mode, 273
  IN OUT mode, 273
  scaling code with defaults, 281–282
  %TYPE, 282–284
 procedures, 250*f*–252
  exception handler, 251
  key components of, 250–251
  subprogram body, 251
 subprogram encapsulation: introduction to packages, 256
POINT_TYPE, 314–317
Positional notation, 275–278
POWER numeric function, 357
Primary key constraints, 70–99
 concatenated, 87–89
 creating the constraint, 72–75
  ALTER TABLE statement, 74–75
  column constraint clause, 72–73
  table constraint clause, 73–74
 DEFERRABLE option, 94–98
 enable, disable, and drop, 92–94
 error handling in PL/SQL, 99
 extra indexes with pseudo keys, 89–92
 index, 77–82, 80*f*
 naming the constraint, 75–77
 NOVALIDATE option, 98–99
 sequences, 82–85
  in code, 86–87
Privileges, SQL*Plus, 61–62
Procedures, PL/SQL, 250*f*–252
 key components, 250–251

PROCESS_ALERTS, 399, 401
PROFESSORS, sample data, 151
Pseudo keys, extra indexes with, 89–92
PURGE procedure, 390

Querying
  child tables, sample script, 166
  a constraint name, sample script, 162–163
  parent tables, sample script, 165–166
  tablespace quotas, 79
QUERY_V1.SQL, 54
QUERY_V3.SQL, 54–55

RAISE_APPLICATION_ERROR, 374–375
RAISE statement, 371–372
Random number generation, 357–358
RAW datatype, 22
RECEIVE_MESSAGE, 384, 389
Records, 307–311
  declaring, 308
Relational database tables, 1–26
  example tables, 11–14
  history prior to, 2–4
  logical and physical look at tables, 22–26
    application tablespaces, 22–25
    data dictionary, 25–26
  SQL, 4–11
  table column datatypes, 16–22
REMOVE_PIPE, 391
REPLACE, 351–352
RESET_BUFFER, 390
ROUND numeric function, 356, 364
%ROWCOUNT, 383
ROWID datatype, 22
Row level trigger, 201–203
Row trigger mechanics, 171–191
  before-insert *versus* after-insert, 172–174
  example row triggers, 181–183
  overview, 171–172
  syntax, 174–178
  table with Oracle constraints and business rules, 183–191
    before-insert *versus* after-insert, 186–187*f*
    environment, 183–185, 184*f*
    managing error codes and messages, 190–191
    procedural constraints to enforce, 185–186
    trigger architecture, 191*f*
    using packages for procedural constraints, 187–190
  trigger body, 178–180
%ROWTYPE, 311, 379
RPAD function, 348–350
RTRIM function, 350–351

Sample data, 150–152; *see also* Students data model
  COURSES, 151
  MAJOR_LOOKUP, 150
  PARKING_TICKETS, 151
  PROFESSORS, 151
  STATE_LOOKUP, 150
  STUDENT_COURSES, 152
  STUDENTS, 150
  STUDENT_VEHICLES, 151
SAVE command, 48
Scalar types, 305–307
Scaling code with defaults, 281–282
Scripts, 46–49, 161–170
  data dictionary constraints, 161–170
  script output, 49–52
Searched CASE statement, 329–330
SELECT statement
  using CASE statements within, 331–332
Self-referential integrity, 120–121
SEND_ALERT, 400
SEND_MESSAGE, 388
Sequences, primary key constraint, 82–85
  in code, 86–87
Session control statements, 15
SET FEEDBACK OFF, 57
SET SERVEROUTPUT ON, 56
Sharing code, 238–241, 239*f*
Signaling events with alerts (DBMS_ALERT), 393–398
  interface description, 395–398
SIGN numeric function, 356
SINH (X) numeric function, 357
SIN (X) numeric function, 357
SPOOL command, 49–50
SQL, 4–11
  ANSI standard, 5–6
  database, 6–7
    access with, 377–383

SQL (*cont.*)
  Direct, 7–8
  implementation, 7–9
  statements, 14–16
    data definition language (DDL), 14
    data manipulation language (DML), 14–15
    session control, 15
    system control, 15
    transaction control, 15
SQLCODE, 137–138, 370–371
SQLERRM, 370–371
SQL*Loader, 139
SQL*Plus, 9–11
  advantages to learning, 10–11
  in client/server mode, 9*f*
  Hello World program, 212–217
SQL*Plus, interacting with Oracle
  with Active State Perl, 61
  AUTOTRACE command, 64–67
  batch command files, 60–61
  command line, 35–46, 52–57
    arguments, 52–57
    CHANGE commands, 36–46
  connecting, 30–34
    to an infrastructure, 32–34
    using UNIX, 31
    using Windows, 31
  desktop shortcut, creating, 28–30, 29*f*
  disconnecting, 34–35
  DUAL table, 63–64
  editing in, 36–46
  with Korn Shell, 57–60
  login, 31*f*
  privileges, 61–62
  script output, 49–52
  scripts, 46–49
SQLPLUS.EXE, 29
SQL%ROWCOUNT function, 378
SQRT numeric function, 356
Stack package, 268–269
Stand-alone procedure, 256
STATE_LOOKUP, sample data, 150
Statement level triggers, 193–203
  aggregation, 198–201
  processing row captured data, 201–203
  and row triggers, types of, 196–198
  sequence of events, 194–195*f*
  syntax, 194–198
Statements
  assignments and, 291–292; *see also* PL/SQL language features
  CASE, 328–333
  IF, 323–328
  insert and update, 378–381
  SELECT, 331–332
  session control, 15
  SQL, 14–16
  system control, 15
  transaction control, 15
String concatenation, 300–301
  operator, 300
String functions, common, 354–355
String manipulation functions, 342–354
  INSTR function, 344–348
  LPAD, RPAD function, 348–350
  LTRIM, RTRIM function, 350–351
  REPLACE, 351–352
  SUBSTR function, 342–344
  TRANSLATE function, 353–354
Structured English Query Language (SEQUEL), 2
STUDENT_COURSES, sample data, 152
Student Registration Application (SRA), 267*f*–268
STUDENTS, sample data, 150
Students data model, 11–14, 131, 143–152, 144*f*
  with constraints, 143–152
    DDL, 145–150
    entity relationship diagram, 143
    sample data, 150–152
    table descriptions, 143–145
  example tables, 11–14
  table example
    check constraint, 131
    check-unique constraint, 131
    UNIQUE constraints, 101–103
STUDENT_VEHICLES, sample data, 151
Subprogram body
  of a function, 254
  of a procedure, 251
Subprogram encapsulation: introduction to packages, 256
SUBSTR function, 342–344
SYSDATE, 359
System control statements, 15
SYSTEM tablespace, data dictionary and, 25*f*–26

Table column datatypes, 16–22
  character, 16–17
  DATE types, 18
  large objects, 19–20
  LONG/RAW, 22
  number, 17
  ROWID, 22
  XMLType, 21–22
Table constraint clause, 73–74
TANH (X) numeric function, 357
TAN (X) numeric function, 357
TERM command, 52
TIMESTAMP, 18
TOAD, 8
TO_CHAR, 359–360
TO_DATE, 359–360
Transaction control statements, 15
TRANSLATE function, 351, 353–354
Triggers
  email notification, 398–402
  row trigger mechanics, 171–191
  statement level, 193–203
TRIM function, 350–351
TRUNC numeric function, 356, 364
%TYPE, 305
  syntax, 282–284

UNIQUE constraints, 99–104
  combining NOT NULL/CHECK constraints with, 101
  DEFERRABLE/NOVALIDATE options, 103
  error handling in PL/SQL, 103–104
  students table example, 101–103
UNIQUE_SESSION_NAME, 385, 391
UNPACK_MESSAGE, 384, 390
UPPER string function, 355
UROWID datatype, 22
USER_CONS_COLUMNS, 153, 158–159
USER_CONSTRAINTS, 156–157, 159–161
User-defined exceptions, 365–366
USER_DEPENDENCIES, 229–232
  invalid procedures, 230*f*
USER_ERRORS, 244–245
  columns, descriptions of, 245
USER_OBJECTS, 156–157, 221–223
  columns, descriptions of, 222
USER_SOURCE, 232–238
  columns, descriptions of, 233
USER_TAB_COLUMNS, 26
UTL_SMTP, 218

VALUE_ERROR, 283–284, 368
VARCHAR2, 16, 306
Variable declarations, 302–304
VARRAY and nested tables, 314–317
  declaring a varray of numbers, 314

WAITANY, 397
WAITONE, 397–398
WHILE LOOP, 335*f*–336

XMLType, 21

ZERO-AMOUNT, 372
ZERO_DIVIDE, 365