E

PL/SQL Predefined Data Types

This appendix groups by data type family the data types and subtypes that the package `STANDARD` predefines.

Constants

This constant defines the maximum name length possible.

```
ORA_MAX_NAME_LEN CONSTANT PLS_INTEGER := 128;
```

BFILE Data Type Family

```
type BFILE is BFILE_BASE;
```

BLOB Data Type Family

```
type BLOB is BLOB_BASE;

subtype "BINARY LARGE OBJECT" is BLOB;
```

BOOLEAN Data Type Family

```
type BOOLEAN is (FALSE, TRUE);
```

CHAR Data Type Family

```
type VARCHAR2 is new CHAR_BASE;
type MLSLABEL is new CHAR_BASE;
type UROWID   is new CHAR_BASE;

DBMS_ID and DBMS_QUOTED_ID define the length of identifiers in objects for SQL, PL/SQL
and users.
subtype DBMS_ID              is VARCHAR2(ORA_MAX_NAME_LEN);
subtype DBMS_QUOTED_ID       is VARCHAR2(ORA_MAX_NAME_LEN+2);

DBMS_ID_30 and DBMS_QUOTED_ID_30 define the length of SQL objects whose limits is 30
bytes.
subtype DBMS_ID_30           is VARCHAR2(30);
subtype DBMS_QUOTED_ID_30    is VARCHAR2(32);

subtype VARCHAR              is VARCHAR2;
subtype STRING               is VARCHAR2;
subtype LONG                 is VARCHAR2(32760);
subtype RAW                  is VARCHAR2;
subtype "LONG RAW"           is RAW(32760);
subtype ROWID                is VARCHAR2(256);
subtype CHAR                 is VARCHAR2;
subtype CHARACTER            is CHAR;
subtype "CHARACTER VARYING"  is VARCHAR;
subtype "CHAR VARYING"       is VARCHAR;
subtype "NATIONAL CHARACTER" is CHAR CHARACTER SET NCHAR_CS;
subtype "NATIONAL CHAR"      is CHAR CHARACTER SET NCHAR_CS;
subtype "NCHAR"              is CHAR CHARACTER SET NCHAR_CS;
subtype "NVARCHAR2"          is VARCHAR2 CHARACTER SET NCHAR_CS;
```

CLOB Data Type Family

```
type CLOB is CLOB_BASE;

subtype "CHARACTER LARGE OBJECT"          is CLOB;
subtype "CHAR LARGE OBJECT"               is CLOB;
subtype "NATIONAL CHARACTER LARGE OBJECT" is CLOB CHARACTER SET NCHAR_CS;
subtype "NCHAR LARGE OBJECT"              is CLOB CHARACTER SET NCHAR_CS;
subtype "NCLOB"                           is CLOB CHARACTER SET NCHAR_CS;
```

DATE Data Type Family

```
type DATE                            is     DATE_BASE;

type TIMESTAMP                       is new DATE_BASE;

type "TIMESTAMP WITH TIME ZONE"      is new DATE_BASE;
type "INTERVAL YEAR TO MONTH"        is new DATE_BASE;
type "INTERVAL DAY TO SECOND"        is new DATE_BASE;
type "TIMESTAMP WITH LOCAL TIME ZONE" is new DATE_BASE;

subtype TIME_UNCONSTRAINED          is TIME(9);
subtype TIME_TZ_UNCONSTRAINED       is TIME(9) WITH TIME ZONE;
subtype TIMESTAMP_UNCONSTRAINED     is TIMESTAMP(9);
subtype TIMESTAMP_TZ_UNCONSTRAINED  is TIMESTAMP(9) WITH TIME ZONE;
subtype YMINTERVAL_UNCONSTRAINED    is INTERVAL YEAR(9) TO MONTH;
subtype DSINTERVAL_UNCONSTRAINED    is INTERVAL DAY(9) TO SECOND (9);
subtype TIMESTAMP_LTZ_UNCONSTRAINED is TIMESTAMP(9) WITH LOCAL TIME ZONE;
```

JSON Data Type Family

```
type JSON is BLOB_BASE;
```

NUMBER Data Type Family

```
type NUMBER is NUMBER_BASE;

subtype FLOAT              is NUMBER; -- NUMBER(126)
subtype REAL               is FLOAT;  -- FLOAT(63)
subtype "DOUBLE PRECISION" is FLOAT;

subtype INTEGER  is NUMBER(38,0);
subtype INT      is INTEGER;
subtype SMALLINT is NUMBER(38,0);

subtype DECIMAL is NUMBER(38,0);
subtype NUMERIC is DECIMAL;
subtype DEC     is DECIMAL;

subtype BINARY_INTEGER is INTEGER range '-2147483647'..2147483647;
subtype NATURAL        is BINARY_INTEGER range 0..2147483647;
subtype NATURALN       is NATURAL not null;
subtype POSITIVE       is BINARY_INTEGER range 1..2147483647;
subtype POSITIVEN      is POSITIVE not null;
subtype SIGNTYPE       is BINARY_INTEGER range '-1'..1;  -- for SIGN functions
subtype PLS_INTEGER    is BINARY_INTEGER;

type BINARY_FLOAT  is NUMBER;
type BINARY_DOUBLE is NUMBER;
```

```
subtype SIMPLE_INTEGER is BINARY_INTEGER NOT NULL;
subtype SIMPLE_FLOAT   is BINARY_FLOAT   NOT NULL;
subtype SIMPLE_DOUBLE  is BINARY_DOUBLE  NOT NULL;
```

See Also:

- PL/SQL Data Types for more information about PL/SQL data types
- "User-Defined PL/SQL Subtypes" for information that also applies to predefined subtypes

Index

Symbols

_ wildcard character, *3-39*
% wildcard character, *3-39*
%BULK_EXCEPTIONS cursor attribute, *13-20*
%BULK_ROWCOUNT cursor attribute, *13-23*
%FOUND cursor attribute
    for implicit cursor, *7-7*
    for named cursor, *7-21*
%ISOPEN cursor attribute
    for implicit cursor, *7-7*
    for named cursor, *7-20*
%NOTFOUND cursor attribute
    for implicit cursor, *7-8*
    for named cursor, *7-22*
%ROWCOUNT cursor attribute
    for implicit cursor, *7-8*
    for named cursor, *7-23*
%ROWTYPE attribute, *6-56*
    column alias and, *7-15*
    explicit cursor and, *7-15*
    invisible columns and, *6-60*
    syntax diagram, *14-159*
    virtual columns and, *6-59*
%TYPE attribute, *3-18*
    initial value and, *3-18*
    NOT NULL constraint and, *3-18*
    syntax diagram, *14-186*
$$PLSQL_LINE inquiry directive, *3-57*
$$PLSQL_UNIT inquiry directive, *3-57*
$$PLSQL_UNIT_OWNER inquiry directive, *3-57*
$$PLSQL_UNIT_TYPE inquiry directive, *3-57*

A

Abstract Data Type (ADT), *2-8*, *15-13*
    creating, *15-67*
accent-insensitive comparison, *3-38*
ACCESS_INTO_NULL exception, *12-11*
ACCESSIBLE BY clause, *11-1*, *14-3*
    in CREATE FUNCTION statement, *15-24*
    in CREATE PACKAGE statement, *15-35*
    in CREATE PROCEDURE statement, *15-43*
    in CREATE TYPE BODY statement, *15-80*
    in CREATE TYPE statement, *15-67*
    in package specification, *11-1*
ACCESSIBLE BY clause *(continued)*
    in subprogram, *9-2*
accessor, *14-3*
accessor list, *14-3*
ADT
    See Abstract Data Type (ADT)
aggregate, *6-18*
AGGREGATE clause, *14-8*
    in CREATE FUNCTION statement, *15-24*
aggregate function, *15-24*
    in PL/SQL expression, *3-47*
    pipelined table function as, *13-48*
    SQL%NOTFOUND attribute and, *7-8*
alias
    column
        in cursor FOR LOOP, *7-25*
        in explicit cursor, *7-15*
    table
        for avoiding inner capture, *B-8*
        for row expression, *B-10*
        for table element attribute or method, *B-9*
aliasing (problem)
    SELECT BULK COLLECT INTO statement
            and, *13-28*
    subprogram parameter, *9-20*
ALTER FUNCTION statement, *15-2*
ALTER LIBRARY statement, *15-4*
ALTER PACKAGE statement, *15-6*
ALTER PROCEDURE statement, *15-8*
ALTER TRIGGER statement, *15-10*
ALTER TYPE statement, *15-13*
AND operator, *3-31*
anonymous block, *2-4*
    AUTHID property and, *9-52*
ANYDATA data type, *13-43*
ANYDATASET data type, *13-43*
ANYTYPE data type, *13-43*
application common object
    metadata-linked, *14-167*
architecture of PL/SQL, *2-10*
array
    associative
        See associative array, *6-2*
    non-PL/SQL, *6-2*

assignment of value
  to composite variable
    collection, *6-23*
    record, *6-62*
  to scalar variable, *3-24*
assignment statement, *3-25*
  syntax diagram, *14-9*
associative array, *6-4*
  characteristics of, *6-2*
  comparisons, *6-29*
  declaring constant, *6-7*
  FIRST and LAST methods for, *6-40*
  in FORALL statement, *14-103*
  NLS parameters and, *6-8*
    See also collection
atomic (lexical) unit, *3-4*
atomically null collection
  See null collection
attribute
  %ROWTYPE
    See %ROWTYPE attribute, *6-56*
  %TYPE
    See %TYPE attribute, *3-18*
  cursor
    See cursor attribute, *7-5*
AUTHID property, *9-52*
  syntax diagram, *14-118*
autonomous routine, *7-54*
  declaring, *7-56*
autonomous transaction, *7-54*
  controlling, *7-57*
  pipelined table function in, *14-137*
autonomous trigger, *7-58*
AUTONOMOUS_TRANSACTION pragma, *7-56*
  for pipelined table function, *13-43*
  syntax diagram, *14-12*

B

bag data structure, *6-2*
base type, *4-1*
basic LOOP statement, *5-10*
  syntax diagram, *14-13*
BETWEEN operator, *3-41*
BINARY_DOUBLE data type
  predefined constants for, *4-2*
  subtype of, *4-3*
  tuning code and, *13-9*
BINARY_FLOAT data type
  predefined constants for, *4-2*
  subtype of, *4-3*
  tuning code and, *13-9*
BINARY_INTEGER data type
  See PLS_INTEGER data type
bind variable
  avoiding SQL injection with, *8-24*
bind variable *(continued)*
  placeholder for
    See placeholder for bind variable, *8-2*
blank-padding
  in assignment, *4-25*
  in comparison, *4-25*
block, *2-4*
  syntax diagram, *14-15*
BOOLEAN data type, *4-3*
BOOLEAN expression, *3-42*
BOOLEAN static expression, *3-51*
BOOLEAN variable, *3-27*
built-in function
  See SQL function
bulk binding, *13-11*
BULK COLLECT clause, *13-25*, *13-38*
  aliasing and, *13-28*
  of FETCH statement, *13-33*
  of RETURNING INTO clause
    FORALL statement and, *13-39*
  of SELECT INTO statement, *13-26*
  query result set processing and, *7-25*
  that returns no rows, *13-25*
bulk SQL, *13-11*
  in compound DML trigger, *10-11*

C

C declaration, *14-24*
C procedure, invoking, *9-55*
cache, function result, *9-37*
calculated column
  See virtual column
call specification, *9-55*
  in CREATE FUNCTION statement, *15-24*
  in CREATE PROCEDURE statement, *15-43*
  in CREATE TYPE statement, *15-67*
  in package, *11-1*
  syntax diagram, *14-24*
call stack, AUTHID property and, *9-52*
capture, *B-7*
cascading triggers, *10-46*
CASE expression
  searched, *3-46*
  simple, *3-43*
case sensitivity
  character comparison and, *3-38*
  character literal and, *3-10*
  identifier and, *3-5*
    quoted user-defined identifier, *3-8*
  keyword and, *D-1*
  LIKE operator and, *3-39*
  reserved word and, *D-1*
CASE statement, *5-1*
  searched, *5-8*
    syntax diagram, *14-27*

CASE statement *(continued)*
  simple, *5-6*
    IF THEN ELSIF statement and, *5-4*
    syntax diagram, *14-27*
CASE_NOT_FOUND exception, *12-11*
case-insensitive comparison, *3-38*
CHAR data type, *4-23*
CHAR data type family, *E-1*
character code, *3-1*
character literal, *3-10*
    See also string
character set, *3-1*
CLOB data type and comparison operator, *3-36*
CLOB data type family, *E-1*
CLOSE statement, *14-30*
collating sequence, *3-38*
collation, *14-55*
collection, *6-1*, *6-4*, *6-10*, *6-13*
  as public package item, *6-49*
  assigning one to another, *6-23*
  comparing one to another, *6-29*
  cursor variable and, *14-51*
  declaration syntax, *14-34*
  empty, *6-2*
    creating with constructor, *6-16*
  index
    See index collection, *13-15*
  internal size of
    DELETE method and, *6-33*
    EXTEND method and, *6-38*
    TRIM method and, *6-36*
  multidimensional, *6-27*
  null, *6-2*
    assigning to collection variable, *6-24*
  pipelined table function and, *13-42*
  querying
    with dynamic SQL, *8-8*
    with static SQL, *7-37*
  retrieving query results into, *13-25*
  types of, *6-2*
collection constructor, *6-16*
collection method, *6-32*
  as subprogram parameter, *6-32*
  invocation syntax, *14-31*
  null collection and, *6-32*
COLLECTION_IS_NULL exception, *12-11*
column alias
  in cursor FOR LOOP, *7-25*
  in explicit cursor, *7-15*
COLUMNS pseudo-operator, *13-57*
comment, *3-12*
  nested, *3-13*
  PL/SQL Wrapper utility and, *A-2*
  syntax diagram, *14-39*
COMMIT statement, *7-44*
  FOR UPDATE cursor and, *7-52*
COMMIT statement *(continued)*
  in autonomous transaction, *7-58*
comparison operator, *3-36*
  cursor variable and, *14-51*
compatible data type
  for collection variables, *6-24*
  for scalar variables, *3-24*
compilation
  conditional, *3-54*
  for native execution, *13-70*
  interpreted, *13-72*
compilation parameter, *2-11*
  displaying value of, *3-57*
  predefined inquiry directive for, *3-57*
compile clause, *14-40*
compile_clause
  in ALTER FUNCTION statement, *15-2*
compile_clause syntax diagram, *14-40*
compile-time warning, *12-2*
compiler directive
  See pragma
composite data type, *6-1*
composite variable, *6-1*
compound trigger, *10-10*
computation-intensive code, *13-9*
concatenation operator (||), *3-28*
concurrent transactions, *7-58*
condition, SQL multiset, *6-31*
conditional compilation, *3-54*
conditional compilation directive, *3-55*
  error, *3-56*
  inquiry, *3-57*
  restrictions on, *3-63*
  selection, *3-56*
conditional predicate, *10-5*
conditional selection statement, *5-1*
conditional trigger, *10-1*
constant
  declaring, *3-16*
    associative array, *6-7*
    record, *6-51*
    syntax diagram, *14-43*
  initial value of, *3-17*
  predefined, *4-2*
  static, *3-53*
    in DBMS_DB_VERSION package, *3-60*
constrained subtype, *4-32*
  in performance-critical code, *13-10*
  subprogram parameter and, *9-11*
constraint
  cursor parameter and, *14-84*
  NOT NULL
    See NOT NULL constraint, *3-15*
  trigger compared to, *10-2*
constructor
  See collection constructor

context of transaction, *7-55*
CONTINUE statement
  syntax diagram, *14-44*
CONTINUE WHEN statement
  syntax diagram, *14-44*
control statement, *5-1*
control token, *3-55*
correlated subquery, *7-28*
correlation name, *10-28*
  with LONG or LONG RAW column, *10-42*
    See also pseudorecord
COUNT collection method, *6-44*
COVERAGE pragma, *14-46*
CREATE FUNCTION statement, *15-24*
CREATE LIBRARY statement, *15-31*
CREATE PACKAGE statement, *15-35*
CREATE TRIGGER statement, *15-48*
CREATE TYPE BODY statement, *15-80*
CREATE TYPE statement, *15-67*
CREATE_WRAPPED procedure, *A-8*
crossedition trigger, *10-1*
CURRENT OF clause, *7-51*
  FOR UPDATE cursor and, *7-51*
  ROWID pseudocolumn instead of, *7-52*
CURRVAL pseudocolumn, *7-3*
cursor, *7-5*
  explicit
    See explicit cursor, *7-9*
  FOR UPDATE, *7-51*
    after COMMIT or ROLLBACK, *7-52*
  implicit
    See implicit cursor, *7-6*
  in SERIALLY_REUSABLE package, *11-9*
  named, *7-9*
    pipelined table function and, *13-47*
    See also explicit cursor *and* cursor variable, *7-9*
  nested, *7-42*
cursor attribute
  for cursor variable, *7-38*
  for explicit cursor, *7-20*
    %FOUND, *7-21*
    %ISOPEN, *7-20*
    %NOTFOUND, *7-22*
    %ROWCOUNT, *7-23*
  for implicit cursor, *7-6*
    DBMS_SQL package and, *8-11*
    native dynamic SQL and, *8-2*
    SQL%BULK_EXCEPTIONS, *13-20*
    SQL%BULK_ROWCOUNT, *13-23*
    SQL%FOUND, *7-7*
    SQL%ISOPEN, *7-7*
    SQL%NOTFOUND, *7-8*
    SQL%ROWCOUNT, *7-8*
  where you can use, *7-5*
CURSOR expression, *7-42*
  passing to pipelined table function, *13-48*
cursor FOR LOOP statement, *5-10*
  query result set processing with, *7-25*
  recursive invocation in, *9-35*
  syntax diagram, *14-49*
cursor number
  converting cursor variable to, *8-17*
  converting to cursor variable, *8-15*
  DBMS_SQL.GET_NEXT_RESULT procedure and, *8-14*
  DBMS_SQL.RETURN_RESULT procedure and, *8-12*
cursor parameter, *7-16*
cursor specification, *14-84*
cursor variable, *7-30*, *7-31*
  converting cursor number to, *8-15*
  converting to cursor number, *8-17*
  DBMS_SQL.GET_NEXT_RESULT procedure and, *8-14*
  DBMS_SQL.RETURN_RESULT procedure and, *8-12*
  declaration syntax diagram, *14-51*
CURSOR_ALREADY_OPEN exception, *12-11*
cursor_control, *5-21*

D

data abstraction, *2-7*
data definition language statement
  See DDL statement
Data Pump Import and triggers, *10-48*
data type, *4-1*
  collection
    See collection, *6-1*
  compatible
    for collection variables, *6-24*
    for scalar variables, *3-24*
  composite, *6-1*
  JSON, *4-5*
  object
    See Abstract Data Type (ADT), *2-8*
  of expression, *3-27*
  predefined, *E-1*
  RECORD
    See record, *6-1*
  scalar, *4-1*
  SQL, *4-1*
  user-defined
    See Abstract Data Type (ADT), *2-8*
  what it determines, *4-1*
    See also subtype
data type conversion, *4-1*
  implicit
    See implicit data type conversion, *13-10*
  SQL injection and, *8-22*

data type family, 4-1
  overloaded subprogram and, 9-29
  predefined data types grouped by, E-1
  subtypes with base types in same, 4-34
Data-bound collation, 14-55
database character set, 3-1
database links
  DR units, 9-54
DATABASE trigger, 10-35
datatype
  in CREATE TYPE statement, 15-67
Datatype, 14-54
DATE data type family, E-1
DBMS_ASSERT package, 8-25
DBMS_DB_VERSION package, 3-60
DBMS_DDL package, A-8
DBMS_PARALLEL_EXECUTE package, 13-67
DBMS_PREPROCESSOR package, 3-62
DBMS_SQL package, 8-11
  switching to native dynamic SQL from, 8-11
DBMS_SQL.GET_NEXT_RESULT procedure, 8-14
DBMS_SQL.RETURN_RESULT procedure, 8-12
DBMS_SQL.TO_NUMBER function, 8-17
DBMS_SQL.TO_REFCURSOR function, 8-15
DBMS_STANDARD package, 3-19
DBMS_WARNING package, 12-3
dbmsupgin.sql script, 13-72
dbmsupgnv.sql script, 13-72
DDL statement, 7-58
  dynamic SQL for, 8-1
  in trigger, 7-58
  subprogram side effects and, 9-51
deadlock
  autonomous transaction and, 7-58
  implicit rollback and, 7-49
declaration, 3-15
  exception raised in, 12-22
DEFAULT COLLATION clause, 14-55
  in CREATE FUNCTION statement, 15-24
  in CREATE PACKAGE statement, 15-35
  in CREATE PROCEDURE statement, 15-43
  in CREATE TRIGGER statement, 15-48
  in CREATE TYPE statement, 15-67
default value, 9-23
  of cursor parameter, 7-17
  of subprogram parameter, 9-23
    See also initial value
DEFINE
  binding category, 13-11
definer’s rights clause, 14-118
definer’s rights unit
  See DR unit
DELETE collection method, 6-33
  COUNT method and, 6-44
  EXISTS method and, 6-39
DELETE collection method *(continued)*
  EXTEND method and, 6-38
  FIRST method and, 6-39
  LAST method and, 6-39
  NEXT method and, 6-46
  PRIOR method and, 6-46
  TRIM method and, 6-36
DELETE statement, 14-57
  BEFORE statement trigger and, 10-39
  PL/SQL extension to, 14-57
    See also DML statement
DELETING conditional predicate, 10-5
delimiter, 3-4
dense collection, 6-2
DEPRECATE pragma, 14-57
Descriptive Intermediate Attributed Notation for Ada (DIANA), C-1
DETERMINISTIC
  in CREATE FUNCTION statement, 15-24
DETERMINISTIC clause, 14-67
DETERMINISTIC option, 14-108
  for pipelined table function, 13-43
DIAGNOSTIC_DEST initialization parameter, 12-1
direct-key partitioning, 15-24, 15-80
directive, 3-55
  compiler
    See pragma, 3-12
  error, 3-56
  inquiry, 3-57
  selection, 3-56
    See also conditional compilation directive
DML statement, 14-57, 14-119, 14-188
  avoiding inner capture in, B-8
  in FORALL statement, 13-12
  inside pipelined table function, 13-43
  on pipelined table function result, 13-51
  PL/SQL syntax of, 7-1
  repeating efficiently, 13-12
DML trigger, 10-4
dot notation, 3-19
  for collection method, 6-32
  for identifier in named PL/SQL unit, 3-19
  for pseudocolumn, 7-3
  for record field, 6-1
  name resolution and, B-1
double quotation mark ("), 3-1
DR unit, 9-52
  call stack and, 9-52
  database links, 9-54
  dynamic SQL and, 9-52
  INHERIT REMOTE PRIVILEGES privilege, 9-54
  name resolution and, 9-52
  privilege checking and, 9-52
  SCHEMA trigger and, 10-35

DR unit *(continued)*
static SQL and, *9-52*
See also AUTHID property
DROP FUNCTION statement, *15-84*
DROP LIBRARY statement, *15-86*
DROP PACKAGE statement, *15-87*
DROP PROCEDURE statement, *15-89*
DROP TRIGGER statement, *15-90*
DROP TYPE BODY statement, *15-93*
DUP_VAL_ON_INDEX exception, *12-11*
dynamic SQL, *8-1*
AUTHID property and, *9-52*
native, *8-2*
switching to DBMS_SQL package from, *8-11*
placeholder for bind variable in
EXECUTE IMMEDIATE statement and, *8-2*
repeated, *8-10*
tuning, *13-4*

E

editioned, *15-13*
editioned Abstract Data Type (ADT), *15-13*
element of collection, *6-1*
element specification
in CREATE TYPE statement, *15-67*
syntax diagram, *14-68*
embedded SQL
See static SQL
empty collection, *6-2*
creating with constructor, *6-16*
error directive, *3-56*
error handling, *12-1*
error-reporting function
SQLCODE, *14-177*
SQLERRM, *14-178*
SQL%BULK_EXCEPTIONS and, *13-20*
escape character, *3-39*
escaped identifier, *B-6*
evaluation order, *3-29*
events publication, *10-49*
evolution of type, *15-13*
exception, *12-1*
handling, *12-5*
in FORALL statement, *13-18*
in trigger, *10-38*
See also exception handler, *12-5*
internally defined
See internally defined exception, *12-9*
predefined
See predefined exception, *12-11*
raised in cursor FOR LOOP statement, *7-25*
raised in declaration, *12-22*
raised in exception handler, *12-23*

exception *(continued)*
raising explicitly, *12-15*
reraising, *12-17*
unhandled, *12-27*
in FORALL statement, *13-18*
user-defined
See user-defined exception, *12-13*
exception handler, *12-5*
continuing execution after, *12-28*
exception raised in, *12-23*
for NO_DATA_NEEDED, *13-51*
GOTO statement and, *14-110*
locator variables for, *12-7*
retrieving error code and message in, *12-27*
retrying transaction after, *12-30*
syntax diagram, *14-77*
EXCEPTION_INIT pragma, *14-74*
for giving error code to user-defined exception, *12-18*
for giving name to internally defined exception, *12-9*
EXECUTE IMMEDIATE statement, *8-2*
syntax diagram, *14-79*
EXISTS collection method, *6-39*
EXIT statement
syntax diagram, *14-82*
EXIT WHEN statement
syntax diagram, *14-82*
exiting a loop, *5-9*
explicit cursor, *7-9*
declaration syntax diagram, *14-84*
in package
declaring, *11-12*
opening and closing, *11-10*
query result processing with
in FOR LOOP statement, *7-25*
with OPEN, FETCH, and CLOSE statements, *7-28*
explicit format model, *8-27*
expression, *3-27*
CURSOR, *7-42*
passing to pipelined table function, *13-48*
data type of, *3-27*
in explicit cursor, *7-15*
SQL function in PL/SQL, *3-47*
static, *3-48*
syntax diagram, *14-88*
EXTEND collection method, *6-38*
external subprogram, *9-55*
call specification, *14-24*

F

FETCH FIRST clause, *13-32*
FETCH statement
across COMMIT, *7-52*

FETCH statement *(continued)*
  record variable and, *6-66*
  syntax diagram, *14-98*
  that returns no row, *7-11*
  with BULK COLLECT clause, *13-33*
  with cursor variable, *7-33*
  with explicit cursor, *7-11*
field of record, *6-1*
FIRST collection method, *6-39*
FOR LOOP index, *5-12*
FOR LOOP iterand, *5-12*
FOR LOOP statement, *5-10*
  FOR LOOP statement
    bounds of, *14-100*
  FORALL statement and, *13-12*
  STEP clause and, *5-10*
  syntax diagram, *14-100*
    See also cursor FOR LOOP statement
FOR UPDATE cursor, *7-51*
  after COMMIT or ROLLBACK, *7-52*
FORALL statement, *13-12*
  associative array in, *14-103*
  bulk binding and, *13-11*
  BULK COLLECT clause and, *13-39*
  for sparse collection, *13-15*
    SQL%BULK_EXCEPTIONS and, *13-23*
  handling exception raised in
    after FORALL completes, *13-20*
    immediately, *13-19*
  number of rows affected by, *13-23*
  syntax diagram, *14-103*
  unhandled exception in, *13-18*
format model, *8-27*
forward declaration of subprogram, *9-9*
function, *9-1*
  aggregate
    See aggregate function, *13-48*
  built-in
    See SQL function, *13-11*
  declaration syntax diagram, *14-108*
  error-reporting
    SQLCODE, *14-177*
    SQLERRM, *13-20*
  invoking, *9-2*
    in SQL statement, *9-51*
  options for, *9-5*
  SQL
    See SQL function, *13-11*
  structure of, *9-5*
  table
    See table function, *13-42*
    See also subprogram
function result cache, *9-37*
function specification, *14-108*

G

generated column
  See virtual column
GET_NEXT_RESULT procedure, *8-14*
global identifier, *3-20*
GOTO statement, *5-24*
  restrictions on, *14-110*
  syntax diagram, *14-110*
granting roles to PL/SQL units, *9-54*

H

hardware arithmetic, *13-9*
hash table, *6-2*
hiding PL/SQL source text
  See wrapping PL/SQL source text
host variable
  bulk-binding, *13-41*
  cursor variable as, *7-41*
  packages and, *11-2*

I

identifier, *3-5*, *3-19*
  ambiguous reference to, *B-1*
  escaped, *B-6*
  global, *3-20*
  in static SQL, *7-1*
  local, *3-20*
  reference to, *3-19*
  scope of, *3-20*
  user-defined, *3-6*
    collecting data about, *13-68*
  visibility of, *3-20*
    See also name
IDL, *C-1*
IF statement, *5-1*
  IF THEN ELSE form, *5-3*
  IF THEN ELSIF form, *5-4*
    nested IF THEN ELSE statement and, *5-4*
    simple CASE statement and, *5-4*
  IF THEN form, *5-1*
  nested, *5-3*
  syntax diagram, *14-113*
immutable iterand, *5-14*
imp and triggers, *10-48*
implicit cursor, *7-6*
  CURSOR expression with, *7-42*
  declaration syntax, *14-115*
  dynamic SQL and, *8-11*
  query result processing with
    with cursor FOR LOOP statement, *7-25*
    with SELECT INTO statement, *7-25*
implicit data type conversion
  minimizing, *13-10*

implicit data type conversion *(continued)*
- of subprogram parameter, *9-13*
  - causing overload error, *9-32*
- of subtypes
  - constrained, *4-32*
  - unconstrained, *4-31*
  - with base types in same family, *4-34*

implicit ROLLBACK statement, *7-49*
implicitly returning query results, *8-12*
Import and triggers, *10-48*
IN operator, *3-41*
IN OUT parameter mode, *9-14*
IN parameter mode, *9-14*
in-bind, *13-11*
independent transaction
- *See* autonomous transaction

index collection, *13-15*
- representing subset with, *13-15*

index iterator choice, *6-18*
index of collection, *6-1*
index-by table
- *See* associative array

indices_of_control, *5-18*
infinite loop, *5-10*
INFORMATIONAL compile-time warning, *12-2*
INHERIT ANY PRIVILEGES privilege, *9-52*
INHERIT PRIVILEGES privilege, *9-52*
INHERIT REMOTE PRIVILEGES privilege, *9-54*
initial value, *9-23*
- %TYPE attribute and, *3-18*
- NOT NULL constraint and, *3-15*
- of constant, *3-17*
- of variable
  - nested table, *6-13*
  - record, *6-51*
  - scalar, *3-17*
  - varray, *6-10*
  - *See also* default value

initialization parameter, *2-11*
INLINE pragma, *13-2*
- syntax diagram, *14-117*

inner capture, *B-7*
- avoiding, *B-8*

input, *2-6*
inquiry directive, *3-57*
INSERT statement, *14-119*
- inserting record with, *6-69*
  - restrictions on, *6-71*
- PL/SQL extension to, *14-119*
  - *See also* DML statement

INSERTING conditional predicate, *10-5*
INSTEAD OF trigger, *10-1*
- for CREATE statement, *10-36*
- on DML statement, *10-5*
  - compound, *10-10*
  - for pipelined table function result, *13-51*

INSTEAD OF trigger *(continued)*
- on DML statement *(continued)*
  - on nested table column, *10-5*

Interface Definition Language (IDL), *C-1*
internally defined exception, *12-9*
- giving name to, *12-9*
- raising explicitly, *12-16*

interpreted compilation, *13-72*
INVALID_CURSOR exception, *12-11*
INVALID_NUMBER exception, *12-11*
invisible column, *6-60*
invoker's rights clause, *14-118*
invoker's rights unit
- *See* IR unit

IR unit, *9-52*
- call stack and, *9-52*
- dynamic SQL and, *9-52*
- granting roles to, *9-54*
- name resolution and, *9-52*
- privilege checking and, *9-52*
- static SQL and, *9-52*, *9-54*
- template objects for, *9-54*
  - *See also* AUTHID property

IS [NOT] NULL operator, *3-37*
isolation level of transaction, *7-55*
iterand, *14-121*
iterand declaration, *14-121*
iterand mutability property, *5-14*
iteration control, *14-121*
iterator, *14-121*
- FOR LOOP statement
  - bounds of, *14-121*
  - PAIRS OF, *14-121*
- INDICES OF, *14-121*
- syntax diagram, *14-121*
- VALUES OF, *14-121*

J

Java class method invocation, *9-55*
Java method declaration, *14-24*
JavaScript declaration, *14-24*
JavaScript function, invoking, *9-55*
JSON data type family, *E-1*

K

key-value pair
- *See* associative array

keywords, *3-6*
- list of, *D-1*

L

LAST collection method, *6-39*
LEVEL pseudocolumn, *7-3*

lexical unit, *3-4*
library
  creating, *15-31*
  dropping, *15-86*
  explicitly recompiling, *15-4*
library arithmetic, *13-9*
LIKE operator, *3-39*
LIMIT clause, *13-36*
LIMIT collection method, *6-45*
line-continuation character, *3-10*
literal, *3-10*
local identifier, *3-20*
locator variable, *12-7*
lock mode, *7-51*
LOCK TABLE statement, *7-51*
locking
  overriding default, *7-50*
  result set row, *7-51*
  table, *7-51*
logical operator, *3-31*
logical value, *4-3*
LOGIN_DENIED exception, *12-11*
LONG data type, *4-26*
  in trigger, *10-42*
LONG RAW data type, *4-26*
  in trigger, *10-42*
LOOP statement
  exiting, *5-9*
  kinds of, *5-9*
  labeled, *5-9*
  optimizing, *13-8*
LOOP UNTIL structure, *5-23*

M

MALFORMED_WRAP_INPUT exception, *A-8*
manageability, *2-3*
MapReduce workloads, *15-24*, *15-80*
materialized view, trigger and, *15-48*
membership test, *3-41*
Method 4, *8-11*
method, collection
  See collection method
mixed parameter notation, *9-25*
mode
  lock, *7-51*
  subprogram parameter, *9-14*
multibyte character set
  as database character set, *3-1*
  variables for values from, *4-24*
multidimensional collection, *6-27*
multiline comment, *3-13*
multiple data transformations, *13-42*
multiset condition, *6-31*
mutable iterand, *5-14*
mutating table, *10-42*
mutating-table error
  for function, *9-51*
  for trigger, *10-42*

N

name, *3-19*
  qualified
    See dot notation, *3-19*
  qualified remote, *3-19*
  remote, *3-19*
  simple, *3-19*
    See also identifier
name resolution, *B-1*
  AUTHID property and, *9-52*
  in static SQL, *B-6*
  PL/SQL and SQL differences, *B-5*
named choice list, *6-18*
named cursor, *7-9*
  pipelined table function and, *13-47*
    See also explicit cursor and cursor variable
named parameter notation, *9-25*
national character set, *3-3*
native dynamic SQL, *8-2*
  switching to DBMS_SQL package from, *8-11*
native execution, compilation for, *13-70*
NATURAL subtype, *4-28*
NATURALN subtype, *4-28*
nested comment, *3-13*
nested cursor, *7-42*
nested IF statement, *5-3*
  IF THEN ELSIF form and, *5-4*
nested record
  assignment example, *6-63*
  declaration example, *6-52*
nested subprogram, *9-2*
  declaration and definition of, *9-2*
  forward declaration for, *9-9*
nested table, *6-13*
  assigning null value to, *6-24*
  assigning set operation result to, *6-25*
  characteristics of, *6-2*
  column in view, trigger on, *10-5*
  comparing to NULL, *6-29*
  comparing two, *6-30*
  correlation names and, *10-28*
  COUNT method for, *6-44*
  FIRST and LAST methods for, *6-42*
  returned by function, *13-42*
  SQL multiset conditions and, *6-31*
    See also collection
nested transaction, *7-54*
NEW correlation name, *10-28*
  with LONG or LONG RAW column, *10-42*
NEXT collection method, *6-46*
NEXTVAL pseudocolumn, *7-3*

NLS parameters
    associative array and, 6-8
    character comparison and, 3-38
    SQL injection and, 8-22
NO_DATA_FOUND exception, 12-11
NO_DATA_NEEDED exception, 13-51
    SQLCODE for, 12-11
no-op (no operation) statement, 5-24
NOCOPY hint, 14-105
    subprogram parameter aliasing and, 9-20
    tuning subprogram invocation with, 13-7
nonpadded comparison semantics, 4-25
IS [NOT] NULL operator
    collections and, 6-29
NOT NULL constraint, 3-15
    %TYPE attribute and, 3-18
    EXTEND method and, 6-38
NOT operator, 3-31
NOT_LOGGED_ON exception, 12-11
null collection, 6-2
    assigning to collection variable, 6-24
    collection method and, 6-32
NULL statement
    syntax diagram, 14-130
    uses for, 5-24
null string, 3-10
NULL value
    assigning to record variable, 6-68
    comparing to collection
        associative array, 6-29
        nested table, 6-29
        varray, 6-29
    comparison operator and, 3-36
    concatenation operator and, 3-28
    for $$PLSQL_UNIT inquiry directive, 3-57
    for $$PLSQL_UNIT_OWNER inquiry
            directive, 3-57
    for collection variable, 6-24
    for subprogram parameter, 9-23
    for unresolvable inquiry directive, 3-60
    in control statement, 3-31
    IN operator and, 3-41
    in set, 3-41
    in USING clause, 8-2
    simple CASE expression and, 3-43
    simple CASE statement and, 5-6
NUMBER data type family
    inefficiency of, 13-9
    members of, E-1

O

obfuscating PL/SQL source text
    See wrapping PL/SQL source text
object type
    See Abstract Data Type (ADT)
OBJECT_VALUE pseudocolumn, 10-33
OCI
    associative array and, 6-10
    cursor variable and, 7-41
of RETURNING INTO clause, 13-38
OLD correlation name, 10-28
OPEN FOR statement, 14-132
    recursive invocation and, 9-35
OPEN statement, 14-130
    recursive invocation and, 9-35
operation, 3-29
operator
    comparison, 3-36
        cursor variable and, 14-51
    logical, 3-31
    relational, 3-37
        collection and, 6-29
operator precedence, 3-29
optimizer
    PL/SQL, 13-1
    SQL, 13-47
OR operator, 3-31
ORA-*n* error
    See internally defined exception
Oracle Call Interface (OCI)
    associative array and, 6-10
    cursor variable and, 7-41
Oracle RAC environment, result caches in, 9-48
ordinary user-defined identifier, 3-7
Original Import and triggers, 10-48
others choice, 6-18
OUT parameter mode, 9-14
out-bind, 13-11
outer capture, B-7
output, 2-6
overloaded subprogram, 9-29
    INLINE pragma and, 13-2

P

package, 11-1
    body of
        See package body, 11-5
    DBMS_STANDARD, 3-19
    explicitly recompiling, 15-6
    features of, 11-2
    granting roles to, 9-54
    guidelines for writing, 11-12
    initialization of, 11-7
    of static constants, 3-53
    private items in, 11-5
    product-specific, 11-1
    public items in
        See public package item, 11-3
    reasons to use, 11-2
    SERIALLY_REUSABLE, 11-8

package *(continued)*
specification of
See package specification, *11-3*
STANDARD
See STANDARD package, *11-18*
state of, *11-7*
supplied by Oracle, *11-1*
wrapping, *A-1*
guideline for, *A-2*
package body, *11-1*
creating, *15-39*
dropping, *15-87*
initialization part of, *11-5*
assigning initial values in, *11-12*
replacing, *15-39*
package specification, *11-1*, *11-3*
creating, *15-35*
cursor variable in, *14-51*
dropping, *15-87*
replacing, *15-35*
See also public package item
package subprogram, *9-2*
pairs_of_control, *5-18*
parallel DML
bulk binding and, *13-11*
for large table, *13-67*
PARALLEL_ENABLE clause, *14-134*
in CREATE FUNCTION statement, *15-24*
PARALLEL_ENABLE option, *14-108*
for pipelined table function, *13-43*
for table function, *13-42*
parameter
compilation
See compilation parameter, *2-11*
explicit cursor, *7-16*
initialization, *2-11*
subprogram
See subprogram parameter, *9-9*
parameter mode, *9-14*
PARENT correlation name, *10-28*
with LONG or LONG RAW column, *10-42*
parentheses
nested, *3-29*
to control evaluation order, *3-29*
to improve readability, *3-29*
pattern matching, *3-39*
percent sign (%) wildcard character, *3-39*
PERFORMANCE compile-time warning, *12-2*
PIPE ROW statement, *13-43*
in autonomous routine, *7-58*
PIPELINED Function
syntax diagram, *14-138*
PIPELINED option, *13-42*
where to specify, *13-43*
pipelined table function, *13-42*
as aggregate function, *13-48*
pipelined table function *(continued)*
in autonomous transaction, *14-137*
See also table function
PL/Scope tool, *13-68*
PL/SQL architecture, *2-10*
PL/SQL block
See block
PL/SQL engine, *2-10*
PL/SQL function result cache, *9-37*
PL/SQL language
advantages of, *2-1*
high performance of, *2-2*
high productivity with, *2-2*
lexical units of, *3-4*
limits of, *C-1*
main features of, *2-3*
manageability and, *2-3*
portability of, *2-2*
program limits of, *C-1*
scalability of, *2-3*
SQL integration in, *2-1*
syntax and semantics, *14-1*
PL/SQL optimizer, *13-1*
PL/SQL table
See associative array
PL/SQL unit, *2-11*
stored
See stored PL/SQL unit, *2-11*
PL/SQL Wrapper utility, *A-2*
placeholder for bind variable
in conditional compilation directive, *3-63*
in dynamic SQL
EXECUTE IMMEDIATE statement and, *8-2*
repeated, *8-10*
in static SQL, *7-1*
OPEN FOR statement and, *7-32*
in trigger body, *10-28*
PLS_INTEGER data type, *4-27*
tuning code and, *13-9*
PLS_INTEGER static expression, *3-51*
PLSQL_CCFLAGS compilation parameter, *3-59*
PLSQL_IMPLICIT_CONVERSION_BOOL initialization parameter, *4-3*
PLSQL_OPTIMIZE_LEVEL compilation parameter, *13-1*
PLSQL_WARNINGS compilation parameter
displaying value of
with ALL_PLSQL_OBJECT_SETTINGS view, *12-2*
with DBMS_WARNING subprogram, *12-3*
setting value of
with ALTER statements, *12-2*
with PLSQL_WARNINGS subprogram, *12-3*

polymorphic table function, *13-53*
  compilation and execution, *13-57*
  definition, *13-54*
  implementation, *13-54*
  invocation, *13-55*
    COLUMNS pseudo-operator, *13-57*
  optimization, *13-57*
portability, *2-2*
positional choice list, *6-18*
positional parameter notation, *9-25*
POSITIVE subtype, *4-28*
POSITIVEN subtype, *4-28*
post-processed source text, *3-62*
pragma, *3-12*
  AUTONOMOUS_TRANSACTION, *14-12*
    for pipelined table function, *13-43*
  COVERAGE, *14-46*
  DEPRECATE, *14-57*
  EXCEPTION_INIT, *14-74*
  INLINE, *13-2*
    syntax diagram, *14-117*
  RESTRICT_REFERENCES, *14-151*
  SERIALLY_REUSABLE, *14-164*
  SUPPRESSES_WARNING_6009, *14-180*
  UDF, *14-188*
precedence, operator, *3-29*
predefined constant, *4-2*
predefined data type, *E-1*
predefined exception, *12-11*
  raising explicitly, *12-16*
  redeclared, *12-13*
predefined inquiry directive, *3-57*
predefined subtype, *E-1*
preprocessor control token, *3-55*
PRIOR collection method, *6-46*
privilege checking and AUTHID property, *9-52*
procedure, *9-1*
  declaration syntax, *14-141*
  invoking, *9-2*
  structure of, *9-5*
    See also subprogram
procedure specification, *14-141*
product-specific package, *11-1*
profiling and tracing programs, *13-68*
program limits, *C-1*
PROGRAM_ERROR exception, *12-11*
pseudocolumn, *7-3*
  OBJECT_VALUE, *10-33*
pseudoinstruction
  See pragma
pseudorecord, *10-28*
    See also correlation name
public package item, *11-3*
  appropriate, *11-3*
  collection type as, *6-49*
  cursor variable as, *14-51*
public package item *(continued)*
  declaring, *11-3*
  RECORD type as, *6-52*
  referencing, *11-3*
  remote variable, *11-3*
  scope of, *11-3*
  visibility of, *11-3*
publishing events, *10-49*
purity rules for subprograms, *9-51*

Q

qualified expression, *6-18*
  syntax diagram, *14-143*
qualified name
  See dot notation
qualified remote name, *3-19*
query, *7-1*
  implicitly returning results of, *8-12*
  invoking function in, *13-5*
  processing result set of, *7-24*
    multiple-row dynamic query, *8-8*
    See also SELECT INTO statement
quotation mark, single or double, *3-1*
quoted user-defined identifier, *3-8*

R

RAISE statement, *12-15*
  syntax diagram, *14-148*
RAISE_APPLICATION_ERROR procedure, *12-18*
raising exception explicitly, *12-15*
range test, *3-41*
read-only transaction, *7-49*
read-write transaction, *7-49*
recompiling stored PL/SQL unit, *15-1*
record, *6-1*
  as public package item, *6-52*
  assigning value to, *6-62*
  comparing one to another, *6-69*
  creating, *6-50*
    syntax diagram, *14-149*
  declaring constant, *6-51*
  nested
    See nested record, *6-52*
  representing row, *6-56*
  types of, *6-52*
record type variables, *6-62*
recursive subprogram, *9-35*
  result-cached, *9-43*
recursive trigger, *10-39*
REF CURSOR
  See cursor variable
REF CURSOR type, *7-31*
relational operator, *3-37*
  collection and, *6-29*

RELEASE constant, *3-60*
remote exception handling
  subprograms and, *12-19*
  triggers and, *10-38*
remote name, *3-19*
remote public package variable, *11-3*
remote subprogram
  exceptions in, *12-19*
  invoked by trigger, *10-36*
  with composite parameter, *6-1*
REPEAT UNTIL structure, *5-23*
replacing stored PL/SQL unit, *15-1*
reraising exception, *12-17*
reserved preprocessor control token, *3-55*
reserved words
  information about, *3-6*
  list of, *D-1*
RESTRICT_REFERENCES pragma, *14-151*
result cache, *9-37*
RESULT_CACHE clause, *9-38*, *14-157*
RESULT_CACHE option for function, *14-108*, *15-24*
RESULT_CACHE_EXECUTION_THRESHOLD, *9-37*
RESULT_CACHE_MAX_RESULT, *9-37*
RESULT_CACHE_MAX_SIZE, *9-37*
RESULT_CACHE_MAX_TEMP_SIZE, *9-37*
RETURN clause of function, *9-5*
RETURN INTO clause
  See RETURNING INTO clause
RETURN statement, *9-6*
RETURN_RESULT procedure, *8-12*
RETURNING INTO clause, *14-154*
  BULK COLLECT clause of, *13-38*
    FORALL statement and, *13-39*
returning query results implicitly, *8-12*
REUSE SETTINGS clause, *2-11*
ROLLBACK statement, *7-46*
  FOR UPDATE cursor and, *7-52*
  implicit, *7-49*
  in autonomous transaction, *7-58*
  transparent, *10-39*
row-level trigger, *10-4*
rowid, *4-26*
ROWID data type, *4-26*
ROWID pseudocolumn, *7-3*
  instead of CURRENT OF clause, *7-52*
ROWNUM pseudocolumn
  bulk SELECT operation and, *13-32*
  single-row result set and, *7-25*
ROWTYPE_MISMATCH exception
  error code for, *12-11*
  example of, *12-11*
runtime error
  See exception

S

same-scope capture, *B-7*
SAMPLE clause, *13-32*
SAVEPOINT statement, *7-47*
  in autonomous transaction, *7-58*
scalability
  SERIALLY_REUSABLE packages and, *11-8*
  subprograms and, *2-3*
scalar data type, *4-1*
scalar variable
  assigning value to, *3-24*
  declaration, *3-16*
    syntax diagram, *14-161*
  initial value of, *3-17*
schema object
  See stored PL/SQL unit
SCHEMA trigger, *10-35*
scope of identifier, *3-20*
searched CASE expression, *3-46*
searched CASE statement, *5-8*
  syntax diagram, *14-27*
security mechanism
  against SQL injection, *8-18*
  PL/SQL source text wrapping
    benefit of, *A-1*
    limitations of, *A-2*
  trigger as, *10-2*
SELECT FOR UPDATE statement, *7-51*
SELECT INTO statement, *7-1*
  assigning values with
    to record variable, *6-66*
    to scalar variables, *3-26*
  avoiding inner capture in, *B-8*
  query result set processing with, *7-25*
  SQL%NOTFOUND attribute and, *7-8*
  SQL%ROWCOUNT attribute and, *7-8*
  syntax diagram, *14-162*
  with BULK COLLECT clause, *13-26*
    See also query
selection directive, *3-56*
selector
  in simple CASE expression, *3-43*
  in simple CASE statement, *5-6*
SELF_IS_NULL exception, *12-11*
sequence, *7-3*
sequence iterator choice, *6-18*
sequential control statement, *5-24*
SERIALLY_REUSABLE package, *11-8*
SERIALLY_REUSABLE pragma, *14-164*
session cursor, *7-5*
SESSION_EXIT_ON_PACKAGE_STATE_ERROR initialization parameter, *11-7*
set data structure, *6-2*
set membership test, *3-41*
SET TRANSACTION statement, *7-49*

SEVERE compile-time warning, 12-2
SHARD_ENABLE clause, 14-165
  in CREATE FUNCTION statement, 15-24
SHARING clause
  in CREATE FUNCTION statement, 15-24
  in CREATE LIBRARY statement, 15-31
  in CREATE PACKAGE statement, 15-35
  in CREATE PROCEDURE statement, 15-43
  in CREATE TRIGGER statement, 15-48
  in CREATE TYPE statement, 15-67
  sharing_clause syntax diagram, 14-167
short-circuit evaluation
  how it works, 3-36
  tuning code and, 13-11
side effects of subprogram, 9-37
SIGNTYPE subtype, 4-28
simple CASE expression, 3-43
simple CASE statement, 5-6
  IF THEN ELSIF statement and, 5-4
  syntax diagram, 14-27
simple DML trigger, 10-4
simple name, 3-19
SIMPLE_DOUBLE subtype, 4-3
  tuning code and, 13-9
SIMPLE_FLOAT subtype, 4-3
  tuning code and, 13-9
SIMPLE_INTEGER subtype, 4-29
  tuning code and, 13-9
single quotation mark ('), 3-1
single-line comment, 3-13
sort ordering, 14-55
sparse collection, 6-2
  FORALL statement for, 13-15
  SQL%BULK_EXCEPTIONS and, 13-23
  traversing, 6-46
specification
  cursor, 14-84
  function, 14-108
  package
    See package specification, 11-1
  procedure, 14-141
SQL
  bulk, 13-11
    in compound DML trigger, 10-11
  dynamic
    See dynamic SQL, 8-1
  static
    See static SQL, 7-1
SQL cursor
  See implicit cursor
SQL data type, 4-1
SQL function, 13-11
  in PL/SQL expression, 3-47
  tuning and, 13-11
SQL injection, 8-18
SQL integration in PL/SQL, 2-1
SQL multiset condition, 6-31
SQL MULTISET operator, 6-25
SQL optimizer, 13-47
SQL statement, 2-4
  for stored PL/SQL unit, 15-1
  in trigger, 10-1
  invoking collection method in, 6-32
  invoking PL/SQL function in, 9-51
  tuning, 13-5
    See also anonymous block
SQL_MACRO clause, 14-169
  in CREATE FUNCTION statement, 15-24
SQL*Loader and triggers, 10-48
SQL%BULK_EXCEPTIONS cursor attribute,
    13-20
SQL%BULK_ROWCOUNT cursor attribute, 13-23
SQL%FOUND cursor attribute, 7-7
SQL%NOTFOUND cursor attribute, 7-8
SQL%ROWCOUNT cursor attribute, 7-8
SQLCODE function, 14-177
SQLERRM function, 14-178
  SQL%BULK_EXCEPTIONS and, 13-20
standalone subprogram, 9-2
  function
    creating, 15-24
    dropping, 15-84
    explicitly recompiling, 15-2
    replacing, 15-24
  procedure
    creating, 15-43
    dropping, 15-89
    explicitly recompiling, 15-8
    replacing, 15-43
STANDARD package
  data type defined in
    See predefined data type, E-1
  exception defined in
    See predefined exception, 12-11
  how it defines PL/SQL environment, 11-18
  listing identifiers defined in, 3-6
  referencing item defined in, 3-19
statement injection, 8-20
statement modification, 8-19
statement-level trigger, 10-4
static constant, 3-53
  in DBMS_DB_VERSION package, 3-60
static expression, 3-48
static SQL, 7-1
  AUTHID property and, 9-52
  name resolution in, B-6
  PL/SQL identifier in, 7-1
  placeholder for bind variable in, 7-1
    OPEN FOR statement and, 7-32
STORAGE_ERROR exception, 12-11
  recursive invocation and, 9-35
store table, 6-16

stored PL/SQL unit, 2-11, 15-1
    creating, 15-1
    recompiling, 15-1
    replacing, 15-1
    wrapping, A-1
stored subprogram, 9-2
    unhandled exception in, 12-27
    wrapping, A-1
string, 3-10
    null, 3-10
    zero-length, 3-10
        See also character literal
STRING subtype, 4-24
strong REF CURSOR type
    creating, 7-31
    FETCH statement and, 7-33
subprogram, 9-1
    inlining, 13-2
    invoked by trigger, 10-36
    remote
        See remote subprogram, 6-1
    unhandled exception in, 12-27
subprogram invocation
    optimization of, 13-2
    resolution of, 9-27
    syntax of, 9-2
    tuning, 13-7
subprogram parameter, 9-9
    collection as, 6-32
    composite variable as, 6-1
    CURSOR expression as actual, 7-42
    cursor variable as, 7-38
    optional, 9-23
    query result as, 7-38
    required, 9-23
subprogram property, 9-3
subquery
    correlated, 7-28
    result set processing with, 7-28
SUBSCRIPT_BEYOND_COUNT exception, 12-11
SUBSCRIPT_OUTSIDE_LIMIT exception, 12-11
subtype, 4-1
    constrained, 4-32
        subprogram parameter and, 9-11
    of BINARY_DOUBLE data type, 4-3
    of BINARY_FLOAT data type, 4-3
    of PLS_INTEGER data type, 4-28
    predefined, E-1
    unconstrained, 4-31
    user-defined, 4-31
        See also data type
SUPPRESSES_WARNING_6009 pragma, 14-180
synonym, 3-19
SYS_INVALID_ROWID exception, 12-11
SYS_REFCURSOR type, 7-31
system trigger, 10-34

T

table
    hash, 6-2
    index-by
        See associative array, 6-4
    mutating, 10-42
    nested, 6-13
        characteristics of, 6-2
    parallel DML for large, 13-67
    PL/SQL
        See associative array, 6-4
    store, 6-16
    unordered, 6-2
    updating large in parallel, 13-67
table alias
    for avoiding inner capture, B-8
    for row expression, B-10
    for table element attribute or method, B-9
table function, 13-42
    pipelined
        See pipelined table function, 13-42
    weak cursor variable argument to, 7-31
TABLE operator, 7-37
TCL statement, 7-1
    in subprogram invoked by trigger, 10-36
    in trigger, 7-58
template object, 9-54
TIMEOUT_ON_RESOURCE exception, 12-11
timing point
    of DML trigger
        compound, 10-10
        simple, 10-4
    of system trigger, 10-34
    trigger firing order and, 10-46
TO_NUMBER function, 8-17
TO_REFCURSOR function, 8-15
TOO_MANY_ROWS exception, 12-11
trace file, 12-1
tracing and profiling programs, 13-68
transaction
    autonomous, 7-54
        pipelined table function in, 14-137
    context of, 7-55
    ending
        with COMMIT statement, 7-44
        with ROLLBACK statement, 7-46
    isolation level of, 7-55
    nested, 7-54
    read-only, 7-49
    read-write, 7-49
    retrying after handling exception, 12-30
    SQL%ROWCOUNT cursor attribute and, 7-8
    visibility of, 7-55
Transaction Control Language
    See TCL statement

TRANSACTIONS initialization parameter, *7-58*
tri-state logic, *3-31*
trigger, *10-1*
  as security mechanism, *10-2*
  AUTHID property and, *9-52*
  autonomous, *7-58*
  cascading, *10-46*
  DDL statement in, *7-58*
  hiding implementation details of, *A-2*
  materialized view and, *15-48*
  recursive, *10-39*
  TCL statement in, *7-58*
TRIM collection method, *6-36*
tuning PL/SQL code, *13-1*
type
  See data type
type-compatible data type
  for collection variables, *6-24*
  for scalar variables, *3-24*
typemark, *6-18*

U

UDF pragma, *14-188*
unconstrained subtype, *4-31*
underscore (_) wildcard character, *3-39*
unhandled exception, *12-27*
  in FORALL statement, *13-18*
unordered table, *6-2*
UPDATE statement, *14-188*
  BEFORE statement trigger and, *10-39*
  PL/SQL extensions to, *14-188*
  with values in record, *6-70*
    restrictions on, *6-71*
    See also DML statement
UPDATING conditional predicate, *10-5*
UROWID data type, *4-26*
user-defined exception, *12-13*
  giving error code to, *12-18*
  raising
    with RAISE statement, *12-15*
    with RAISE_APPLICATION_ERROR
      procedure, *12-18*
user-defined identifier, *3-6*
  collecting data about, *13-68*
user-defined subtype, *4-31*
user-defined type
  See Abstract Data Type (ADT)
USING_NLS_COMP, *14-55*
utlrp.sql script, *13-72*

V

V$RESERVED_WORDS view, *D-1*
V$RESULT_CACHE_OBJECTS, *9-37*
validation check for avoiding SQL injection, *8-25*
VALUE_ERROR exception, *12-11*
values_of_control, *5-18*
VARCHAR subtype, *4-24*
VARCHAR2 data type, *4-23*
VARCHAR2 static expression, *3-52*
variable
  binding of, *13-11*
  BOOLEAN, *3-27*
  collection
    See collection, *6-1*
  composite, *6-1*
  cursor
    See cursor variable, *7-30*
  host
    cursor variable as, *7-41*
    packages and, *11-2*
  in cursor variable query, *7-35*
  in explicit cursor query, *7-13*
  locator, *12-7*
  record
    See record, *6-1*
  remote public package, *11-3*
  scalar
    See scalar variable, *3-16*
  with undefined value, *7-1*
variable-size array
  See varray
variadic pseudo-operator, *13-56*
varray, *6-10*
  assigning null value to, *6-24*
  characteristics of, *6-2*
  comparing to NULL, *6-29*
  COUNT method for, *6-44*
  FIRST and LAST methods for, *6-41*
  returned by function, *13-42*
    See also collection
VECTOR data type, *4-14*
VERSION constant, *3-60*
view
  AUTHID property and, *9-52*
  INSTEAD OF trigger and, *15-48*
  materialized, trigger and, *15-48*
virtual column, *7-15*
  %ROWTYPE attribute and, *6-59*
  explicit cursors and, *7-15*
visibility
  of identifier, *3-20*
  of transaction, *7-55*

W

warning, compile-time, *12-2*
weak REF CURSOR type
  creating, *7-31*
  FETCH statement and, *7-33*

WHILE LOOP statement, *5-23*
    syntax diagram, *14-189*
white list
    See ACCESSIBLE BY clause
whitespace character
    between lexical units, *3-14*
    in character literal, *3-10*
    in database character set, *3-1*
wildcard character, *3-39*
WRAP function, *A-8*
wrap utility
    See PL/SQL Wrapper utility
wrapping PL/SQL source text, *A-1*
    inquiry directives and, *3-60*

Z

ZERO_DIVIDE exception, *12-11*
zero-length string, *3-10*