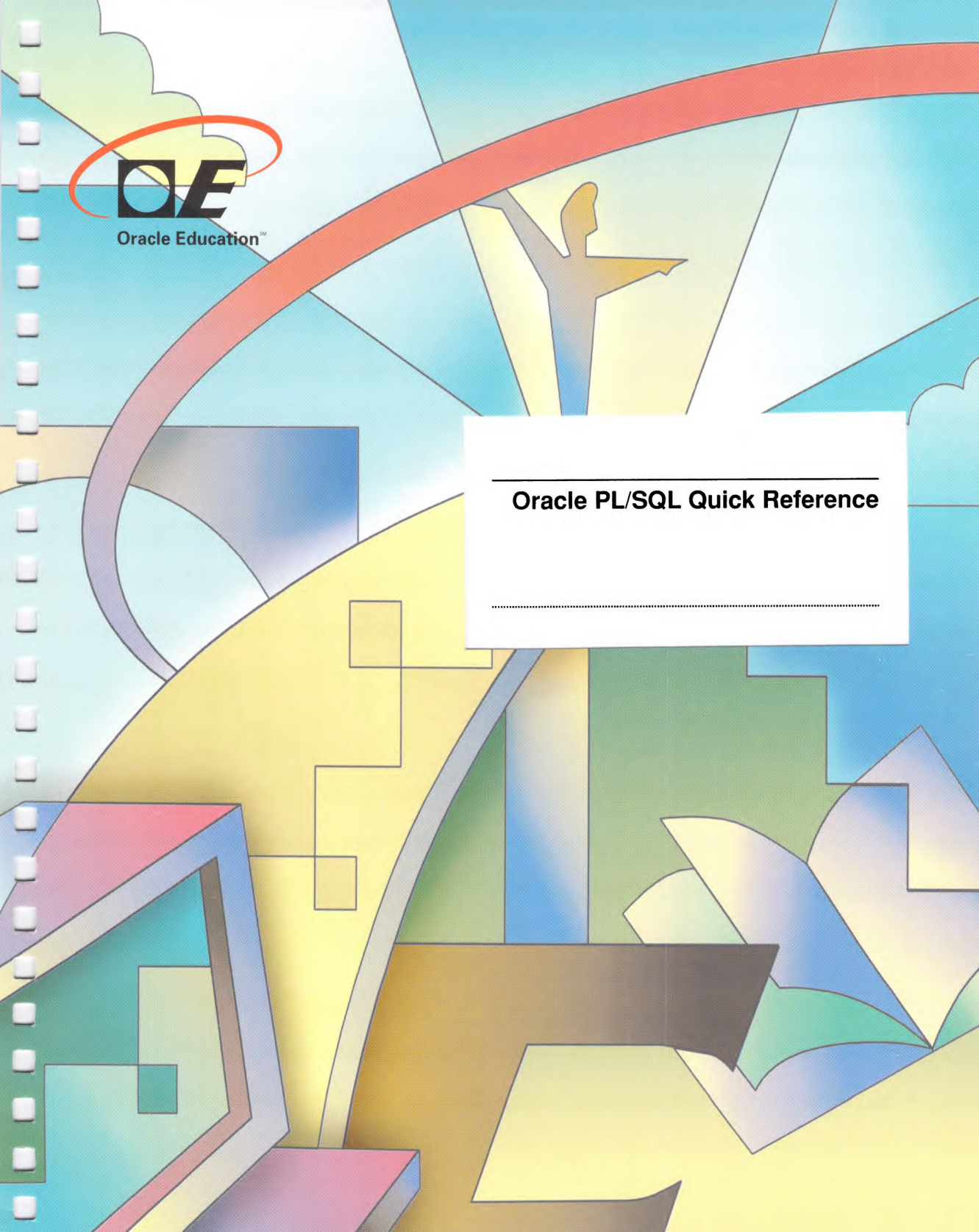

Oracle Education℠
Oracle PL/SQL Quick Reference

Oracle PL/SQL Quick Reference

40021GC10
Production 1.0
April 1999
M08685

ORACLE®

Oracle PL/SQL Quick Reference

Edition 1.0

Published by Oracle Corporation

Author: Kuljit Jassar

Table of Contents

Conventions Used in This Reference

Text

The text in this reference adheres to the following conventions:

UPPERCASE	Uppercase text calls attention to SQL keywords, filenames, andinitialization parameters.
italics	Italicized text calls attention to parameters of SQL statements.
boldface	Boldface text calls attention to definitions of terms.

Syntax Diagrams and Notation

This reference uses syntax diagrams to show PL/SQL statement and other elements of the PL/SQL language. These syntax diagrams use lines and arrows to show syntactic structure.
If you are not familiar with syntax diagrams, refer to*Oracle8 SQL Reference*.

Keywords appear in UPPERCASE. You must use keywords in your SQL statement exactly as they appear in the syntax diagram, although keywords are not case-sensitive.

Parameters appear in lowercase and act as placeholders. Parameters are usually names of database objects, Oracle datatype names, or expressions.

Reading Syntax Diagrams

Syntax diagrams are drawings that illustrate valid SQL syntax. To read a diagram, trace it from left to right, in the direction shown by the arrows.
Commands and other keywords appear in UPPERCASE inside rectangles. Type them exactly as shown in the rectangles. Parameters appear in lowercase inside ovals. Variables are used for the parameters. Punctuation, operators, delimiters, and terminators appear inside circles.
If the syntax diagram has more than one path, you can choose any path to travel.
If you have the choice of more than one keyword, operator, or parameter, your options appear in a vertical list.

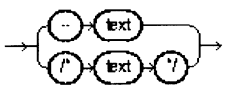
Required Keywords and Parameters

Required keywords and parameters can appear singly or in a vertical list of alternatives. Single required keywords and parameters appear on the main path, that is, on the horizontal line you are currently traveling. In the following example,library_name is a required parameter:

If there is a library named HQ_LIB, then, according to the diagram, the following statement is valid:

```
SAVEPOINT s1;
```

If multiple keywords or parameters appear in a vertical list that intersects the main path, one of them is required. That is, you must choose one of the keywords or parameters, but not necessarily the one that appears on the main path. In the following example, you must choose one of the four settings:



Optional Keywords and Parameters

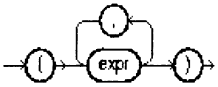
If keywords and parameters appear in a vertical list above the main path, they are optional. In the following example, instead oftraveling down a vertical line, you can continue along the main path:

According to the diagram, all of the following statements are valid:

```
EXIT;
EXIT this_loop;
EXIT this_loop WHEN x >= 1000;
```

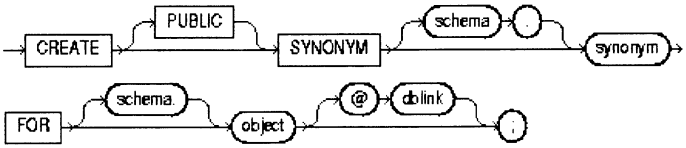
Syntax Loops

Loops let you repeat the syntax within them as many times as you like. In the following example, after choosing one expression, you can go back repeatedly to choose another, separated by commas.



Multipart Diagrams

Read a multipart diagram as if all the main paths were joined end to end. The following example is a two-part diagram:



According to the diagram, the following statement is valid:

```
CREATE SYNONYM emp FOR scott.employees;
```

Database Objects

The names of Oracle identifiers, such as tables and columns, must not exceed 30 characters in length. The first character must be a letter, but the rest can be any combination of letters, numerals, dollar signs ($), pound signs (#), and underscores (_).

However, if an Oracle identifier is enclosed by double quotation marks ("), it can contain any combination of legal characters, including spaces but excluding quotation marks.

Oracle identifiers are not case-sensitive except when enclosed by double quotation marks.

Note: In this quick reference guide, not all syntax statements have been included, and of those included, not all clauses have been included. For further information, see*Oracle8 SQL Reference*.

Datatypes

Scalar Types

BINARY_INTEGER
DEC
DECIMAL
DOUBLE PRECISION
FLOAT
INT
INTEGER
NATURAL
NATURALN
NUMBER
NUMERIC
PLS_INTEGER
POSITIVE
POSITIVEN
REAL
SIGNTYPE

SMALLINT
CHAR
CHARACTER
LONG
LONG RAW
NCHAR
NVARCHAR2
RAW
ROWID
STRING
VARCHAR
VARCHAR2

BOOLEAN

DATE

Composite Types

RECORD
TABLE
VARRAY

Reference Types

REF CURSOR
REF object_type

LOB Types

BFILE
BLOB
CLOB
NCLOB

Object Naming Rules

The following rules apply when naming objects:

- Names must be from 1 to 30 characters long.
- Names cannot contain quotation marks.
- Names are not case-sensitive.
- Unless surrounded by double quotation marks, a name must begin with an alphabetic character from your database character set.
- Names can contain only alphanumeric characters from your database character set and the characters underscore(_), dollar sign ($), and pound sign (#). Oracle Corporation strongly discourages you from using $ and #. Names of database links can also contain periods (.) and at-signs (@).
  If your database character set contains multibyte characters, Oracle recommends that each name for a user or a role contain at least one single-byte character.
- A name cannot be an Oracle8 reserved word.
  Depending on the Oracle product that you plan to use to access a database object, names might be further restricted by other product-specific reserved words.
- Do not use the word DUAL as a name for an object. DUAL is the name of a dummy table.
- The Oracle SQL language contains other words that have special meanings. These words include datatypes, function names, and keywords. These words are not reserved. However, Oracle server uses them internally, and this may make your statements difficult to read and may lead to unpredictable results

Oracle Reserved Words

Oracle reserved words have special meaning to the Oracle server. You cannot use these words as names for database objects.
Keywords also have special meaning to Oracle server, but are not reserved words. Because some may become reserved words, and for maximum portability to other implementations of SQL, do not use keywords as object names.

PL/SQL Reserved Words

The words listed here are reserved by PL/SQL; that is, they have a special syntactic meaning to PL/SQL. Thus, you should not use them to name program objects such as constants, variables, and cursors. Also, some of these words (marked by an asterisk) are reserved by SQL. You should not use them to name database objects such as columns, tables, and indexes.

ALL*	BINARY_INTEGER	COMMIT	DEFAULT*
ALTER*	BODY*	COMPRESS*	DELETE*
AND*	BOOLEAN	CONNECT*	DESC*
ANY*	BULK	CONSTANT	DISTINCT*
ARRAY	BY*	CREATE*	DO
AS*	CHAR*	CURRENT*	DROP*
ASC*	CHAR_BASE	CURRVAL	ELSE*
AUTHID	CHECK*	CURSOR	ELSIF
ASSIGN	CLOSE	DATE*	END
AVG	CLUSTER*	DAY	EXCEPTION
BEGIN	COLLECT	DECLARE	EXCLUSIVE*
BETWEEN*	COMMENT*	DECIMAL*	EXECUTE

EXISTS*
EXIT
EXTENDS
FALSE
FETCH
FLOAT*
FOR*
FORALL
FROM*
FUNCTION
GOTO
GROUP*
HAVING*
HEAP
HOUR
IF
IMMEDIATE*
IN*
INDEX*
INDICATOR
INSERT*
INTEGER*
INTERFACE
INTERSECT*
INTERVAL
INTO*
IS*
ISOLATION
JAVA
LEVEL*
LIKE*
LIMITED
LOCK*
LONG*
LOOP
MAX
MIN
MINUS*
MINUTE
MLSLABEL*
MOD
MODE*
MONTH
NATURAL
NATURALN
NEW
NEXTVAL
NOCOPY
NOT*
NOWAIT*
NULL*
NUMBER*
NUMBER_BASE
OCIROWID
OF*
ON*
OPAQUE
OPEN
OPERATOR
OPTION*
OR*
ORDER*
ORGANIZATION
OTHERS
OUT
PACKAGE
PARTITION
PCTFREE*
PLS_INTEGER
POSITIVE
POSITIVEN
PRAGMA
PRIOR*
PRIVATE
PROCEDURE
PUBLIC*
RAISE
RANGE
RAW*
REAL
RECORD
REF
RELEASE
RETURN
REVERSE
ROLLBACK
ROW*
ROWID*
ROWLABEL*
ROWNUM*
ROWTYPE
SAVEPOINT
SECOND
SELECT*
SEPARATE
SET*
SHARE*
SMALLINT*
SPACE
SQL
SQLCODE
SQLERRM
START*
STDDEV
SUBTYPE
SUCCESSFUL*
SUM
SYNONYM*
SYSDATE*
TABLE*
THEN*
TIME
TIMESTAMP
TO*
TRIGGER*
TRUE
TYPE
UID*
UNION*
UNIQUE*
UPDATE*
USE
USER*
VALIDATE*
VALUES*
VARCHAR*
VARCHAR2*
VARIANCE
VIEW*
WHEN
WHENEVER*
WHERE*
WHILE
WITH*
WORK
WRITE
YEAR
ZONE

Words followed by an asterisk (*) are also ANSI reserved words.

SQL Reserved Words

ACCESS	EXCLUSIVE	MODE	SELECT*
ADD	EXISTS*	MODIFY	SESSION
ALL*	FILE	NOAUDIT	SET*
ALTER	FLOAT*	NOCOMPRESS	SHARE
AND*	FOR*	NOT*	SIZE
ANY*	FROM*	NOWAIT	SMALLINT*
AS*	GRANT*	NULL*	START
ASC*	GROUP*	NUMBER	SUCCESSFUL
AUDIT	HAVING*	OF*	SYNONYM
BETWEEN*	IDENTIFIED	OFFLINE	SYSDATE
BY*	IMMEDIATE	ON*	TABLE*
CHAR*	IN*	ONLINE	THEN
CHECK*	INCREMENT	OPTION*	TO*
CLUSTER	INDEX	OR*	TRIGGER
COLUMN	INITIAL	ORDER*	UID
COMMENT	INSERT*	PCTFREE	UNION*
COMPRESS	INTEGER*	PRIOR	UNIQUE*
CONNECT	INTERSECT	PRIVILEGES*	UPDATE*
CREATE*	INTO*	PUBLIC*	USER*
CURRENT*	IS*	RAW	VALIDATE
DECIMAL	LEVEL	RENAME	VALUES*
DEFAULT*	LIKE*	RESOURCE	VARCHAR
DELETE*	LOCK	REVOKE	VARCHAR2
DESC*	LONG	ROW	VIEW*
DISTINCT*	MAXEXTENTS	ROWID	WHENEVER
DROP	MINUS	ROWNUM	WHERE*
ELSE	MLSLABEL	ROWS	WITH*

Words followed by an asterisk (*) are also ANSI reserved words.

SQL Keywords

ADMIN
AFTER
ALLOCATE
ANALYZE
ARCHIVE
ARCHIVELOG
AUTHORIZATION*
AVG*
BACKUP
BECOME
BEFORE
BEGIN*
BLOCK
BODY
CACHE
CANCEL
CASCADE
CHANGE
CHARACTER*
CHECKPOINT
CLOSE*
COBOL*
COMMIT*
COMPILE
CONSTRAINT
CONSTRAINTS
CONTENTS
CONTINUE*
CONTROLFILE
COUNT*
CURSOR*
CYCLE
DATABASE
DATAFILE
DBA
DEC*
DECLARE*
DISABLE
DISMOUNT
DOUBLE*
DUMP
EACH
ENABLE
END*
ESCAPE*
EVENTS
EXCEPT
EXCEPTIONS
EXEC*
EXECUTE
EXPLAIN
EXTENT
EXTERNALLY
FETCH*
FLUSH
FORCE
FOREIGN*
FORTRAN*
FOUND*
FREELIST
FREELISTS
FUNCTION
GO*
GOTO*
GROUPS
INCLUDING
INDICATOR*
INITRANS
INSTANCE
INT*
KEY*
LANGUAGE*
LAYER
LINK
LISTS
LOGFILE
MANAGE
MANUAL
MAX*
MAXDATAFILES
MAXINSTANCES
MAXLOGFILES
MAXLOGHISTORY
MAXLOGMEMBERS
MAXTRANS
MAXVALUE
MIN*
MINEXTENTS
MINVALUE
MODULE*
MOUNT
NEXT
NEW
NOARCHIVELOG
NOCACHE
NOCYCLE
NOMAXVALUE
NOMINVALUE
NONE
NOORDER
NORESETLOGS
NORMAL
NOSORT
NUMERIC*
OFF
OLD
ONLY
OPEN*
OPTIMAL
OWN
PACKAGE
PARALLEL
PASCAL*
PCTINCREASE
PCTUSED
PLAN
PLI*
PRECISION*
PRIMARY*
PRIVATE
PROCEDURE*
PROFILE
QUOTA
READ
REAL*
RECOVER
REFERENCES*
REFERENCING
RESETLOGS
RESTRICTED
REUSE ROLE
ROLES
ROLLBACK*
SAVEPOINT
SCHEMA*
SCN
SECTION*
SEGMENT
SEQUENCE
SHARED
SNAPSHOT
SOME*
SORT
SQLCODE*
SQLERROR*
STATEMENT_ID
STATISTICS
STOP
STORAGE
SUM*
SWITCH
SYSTEM
TABLES
TABLESPACE
TEMPORARY
THREAD
TIME
TRACING
TRANSACTION
TRIGGERS
TRUNCATE
UNDER
UNLIMITED
UNTIL
USE
USING
WHEN
WRITE
WORK*

SYSTEM PRIVILEGES

PROCEDURES

System Privilege	Allows Grantee To ...
CREATE PROCEDURE	Create stored procedures, functions, and packages in grantee's schema
CREATE ANY PROCEDURE	Create stored procedures, functions, and packages in any schema except SYS
ALTER ANY PROCEDURE	Alter stored procedures, functions, or packages in any schema except SYS
DROP ANY PROCEDURE	Drop stored procedures, functions, or packages in any schema except SYS
EXECUTE ANY PROCEDURE	Execute procedures or functions (standalone or packaged) Reference public package variables in any schema except SYS

TRIGGER

System Privilege	Allows Grantee To ...
CREATE TRIGGER	Create a database trigger in grantee's schema
CREATE ANY TRIGGER	Create database triggers in any schema except SYS
ALTER ANY TRIGGER	Enable, disable, or compile database triggers in any schema except SYS
DROP ANY TRIGGER	Drop any trigger in any schema except SYS
ADMINISTER DATABASE TRIGGER	Create a trigger on DATABASE (You must also have the CREATE TRIGGER or CREATE ANY TRIGGER privilege.)

TYPES

System Privilege	Allows Grantee To ...
CREATE TYPE	Create object types and object type bodies in grantee's schema
CREATE TYPE	Create object types and object type bodies in any schema except SYS
ALTER ANY TYPE	Alter object types in any schema except SYS
DROP ANY TYPE	Drop object types and object type bodies in any schema except SYS
EXECUTE ANY TYPE	Use and reference object types and collection types in any schema except SYS, and invoke methods of an object type in any schema*if you make the grant to a specific user* (If you grant EXECUTE ANY TYPE to a role, users holding the enabled role will not be able to invoke methods of any object type in any schema.)

OBJECT PRIVILEGES

These privileges apply to specific objects.

Object Privilege	Allows Grantee To ...
ALL [PRIVILEGES]	All of the object privileges that can be applied
TABLE PRIVILEGES	
ALTER	Change the table definition with the ALTER TABLE statement
DELETE	Remove rows from the table with the DELETE statement **Note:** You must grant the SELECT privilege on the table along with the DELETE privilege
INDEX	Create an index on the table with the CREATE INDEX statement
INSERT	Add new rows to the table with the INSERT statement
REFERENCES	Create a constraint that refers to the table. (You cannot grant this privilege to a role.)
SELECT	Query the table with the SELECT statement
UPDATE	Change data in the table with the UPDATE statement **Note:** You must grant the SELECT privilege on the table along with the UPDATE privilege.
VIEW PRIVILEGES	
DELETE	Remove rows from the view with the DELETE statement
INSERT	Add new rows to the view with the INSERT statement
SELECT	Query the view with the SELECT statement.
UPDATE	Change data in the view with the UPDATE statement
SEQUENCE PRIVILEGES	
ALTER	Change the sequence definition with the ALTER SEQUENCE statement
SELECT	Examine and increment values of the sequence with the CURRVAL and NEXTVAL pseudocolumns
PROCEDURE, FUNCTION, and PACKAGE PRIVILEGE	
EXECUTE	Compile the procedure or function or execute it directly, or access any program object declared in the specification of a package
SNAPSHOT PRIVILEGE	
SELECT	Query the snapshot with the SELECT statement

Object Privilege	Table	View	Sequence	Procedures, Functions, Packages	User-Defined Type
ALTER	X		X		
DELETE	X	X			
INDEX	X				
INSERT	X	X			
READ					
REFERENCES	X				
SELECT	X	X	X		
UPDATE	X	X			
EXECUTE				X	X

Built-in Functions

PL/SQL provides many powerful functions to help you manipulate data. These built-in functions fall into the following categories:

- Error-reporting
- Number
- Character
- Conversion
- Date
- Miscellaneous

You can use all the functions in SQL statements except the error-reporting functions SQLCODE and SQLERRM. Also, you can use all the functions in procedural statements except the miscellaneous functions DECODE, DUMP, and VSIZE.

The SQL group functions AVG, MIN, MAX, COUNT, SUM, STDDEV, and VARIANCE are not built into PL/SQL. Nevertheless, you can use them in SQL statements but not in procedural statements.

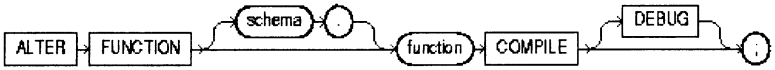
Error	Number	Character	Conversion
SQLCODE	ABS	ASCII	CHARTOROWID
SQLERRM	ACOS	CHR	CONVERT
	ASIN	CONCAT	HEXTORAW
	ATAN	INITCAP	RAWTOHEX
	ATAN2	INSTR	ROWIDTOCHAR
	CEIL	INSTRB	TO_CHAR
	COS	LENGTH	TO_DATE
	COSH	LENGTHB	TO_MULTI_BYTE
	EXP	LOWER	TO_NUMBER
	FLOOR	LPAD	TO_SINGLE_BYT
			E
	LN	LTRIM	
	LOG	NLS_INITCAP	
	MOD	NLS_LOWER	
	POWER	NLS_UPPER	
	ROUND	NLSSORT	
	SIGN	REPLACE	
	SIN	RPAD	
	SINH	RTRIM	
	SQRT	SOUNDEX	
	TAN	SUBSTR	
	TANH	SUBSTRB	
	TRUNC	TRANSLATE	
		TRIM	
		UPPER	

Date	Object Reference	Miscellaneous
ADD_MONTHS	DEREF	BFILENAME
LAST_DAY	REF	DECODE
MONTHS_BETWEE	VALUE	DUMP
N		
NEW_TIME		EMPTY_BLOB
NEXT_DAY		EMPTY_CLOB
ROUND		GREATEST
SYSDATE		LEAST
TRUNC		NLS_CHARSET_DECL_LEN
		NLS_CHARSET_ID
		NLS_CHARSET_NAME
		NVL
		SYS_CONTEXT
		SYS_GUID
		UID
		USER
		USERENV
		VSIZE

SQL Commands Syntax

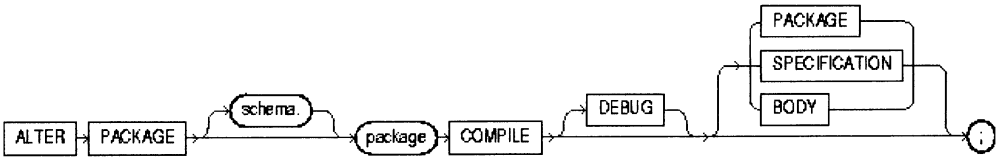
ALTER FUNCTION

Use this command to recompile an invalid stand-alone stored function.



ALTER PACKAGE

Use this command to explicitly recompile either a package specification, body, or both.

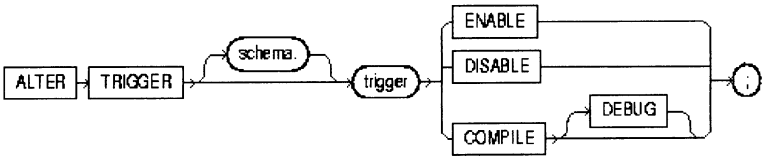


ALTER PROCEDURE

Use this command to explicitly recompile a stand-alone stored procedure.

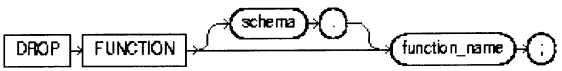
ALTER TRIGGER

Use this command to enable, disable, or compile a database trigger.



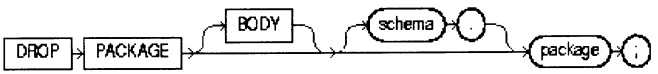
DROP FUNCTION

Use this command to remove a stored function from the database.



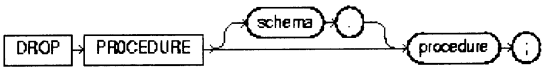
DROP PACKAGE

Use this command to remove a stored package from the database. This statement drops the body and specification of a package.



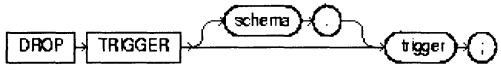
DROP PROCEDURE

Use this command to remove a stored procedure from the database.



DROP TRIGGER

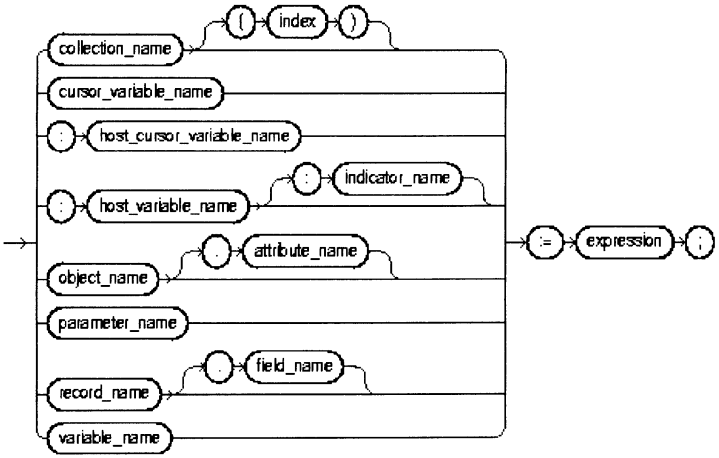
Use this command to remove a database trigger from the database.



Assignment Statement

An assignment statement sets the current value of a variable, field, parameter, or element.

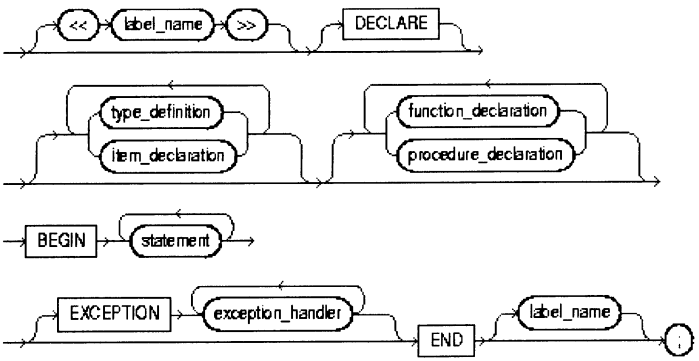
assignment_statement



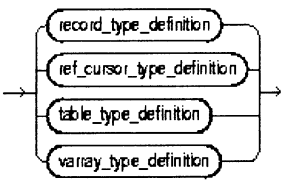
Block

The basic program unit in PL/SQL is the block.

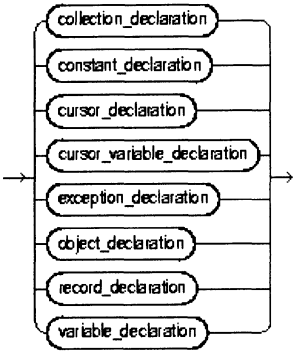
plsql_block



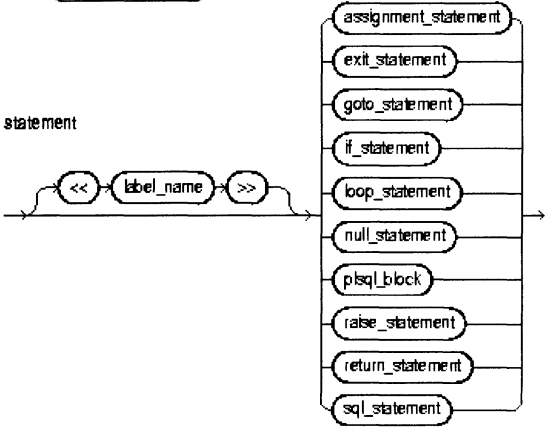
type_definition



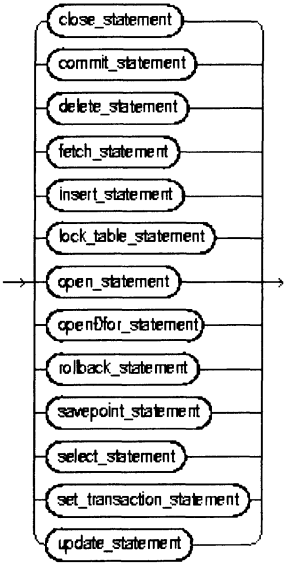
item_declaration



statement

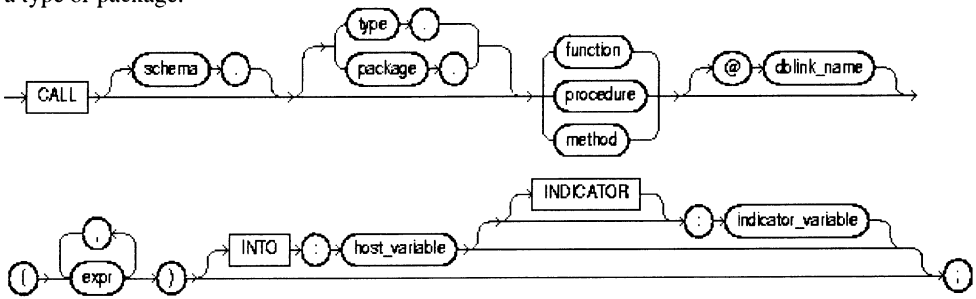


sql_statement



CALL

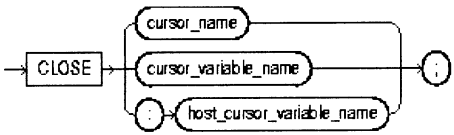
Allows you to execute a stand-alone procedure or function, or a procedure or function defined within a type or package.



CLOSE Statement

The CLOSE statement enables resources held by an open cursor or cursor variable to be reused. No more rows can be fetched from a closed cursor or cursor variable.

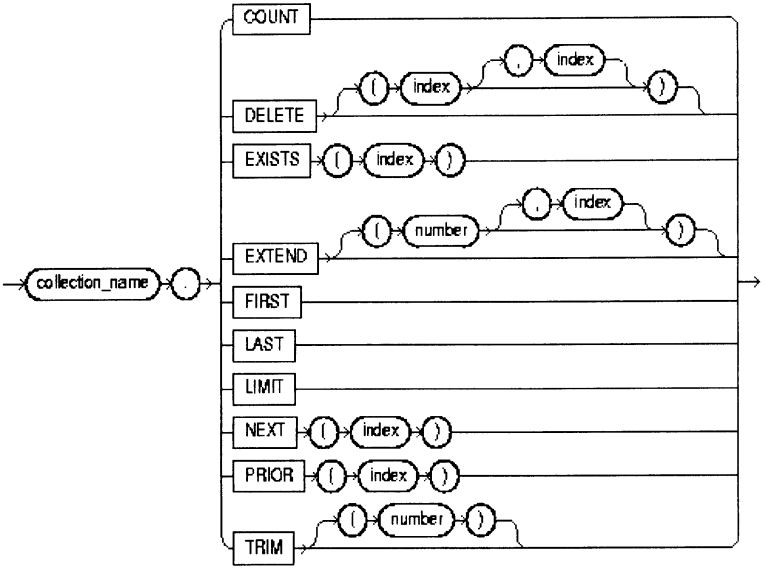
close_statement



Collection Method

A collection method is a built-in function or procedure that operates on collections and is called using dot notation. The methods EXISTS, COUNT, LIMIT, FIRST, LAST, PRIOR, NEXT, EXTEND, TRIM, and DELETE help generalize code, make collections easier to use, and make your applications easier to maintain.

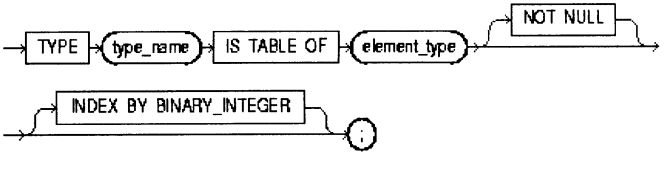
collection_method_call



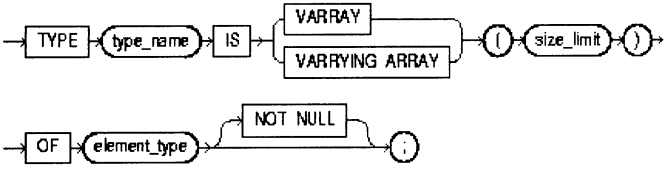
Collection

A collection is an ordered group of elements, all of the same type (for example, the grades for a class of students). Each element has a unique subscript that determines its position in the collection. PL/SQL offers three kinds of collections: index-by tables, nested tables, andvarrays.

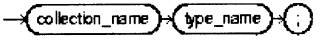
table_type_definition



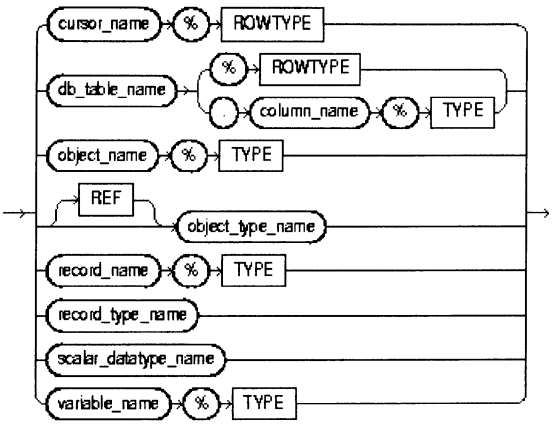
varray_type_definition



collection_declaration



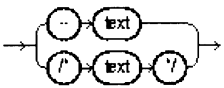
element_type



Comments

Comments can appear within a statement at the end of a line. You cannot nest comments.

comment



COMMIT Statement

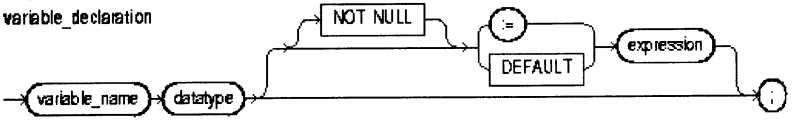
The COMMIT statement explicitly makes permanent any changes made to the database during the current transaction.

commit_statement

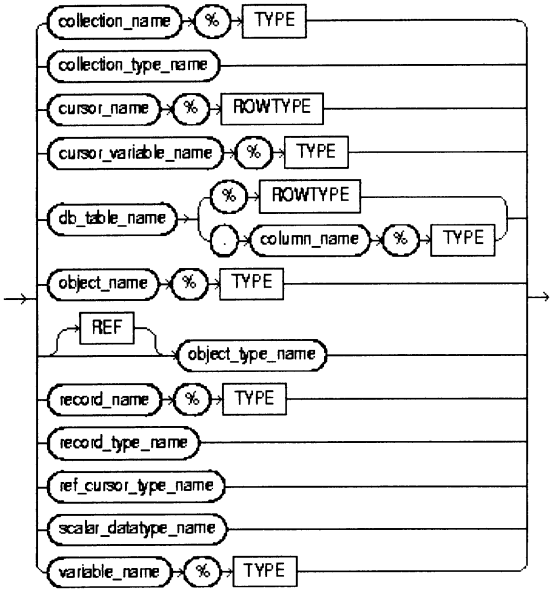
Constant and Variable

You can declare constants and variables in the declarative part of any PL/SQL block, subprogram, or package. Declarations allocate storage space for a value, specify its datatype, and name the storage location so that you can reference it. Declarations can also assign an initial value and impose the NOT NULL constraint.

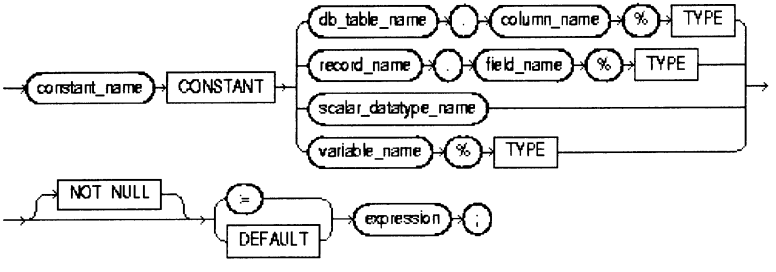
variable_declaration



datatype



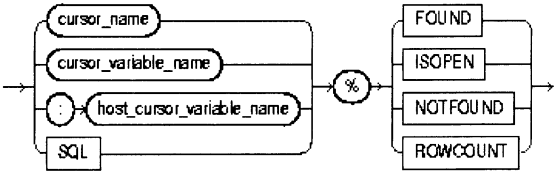
constant_declaration



Cursor Attribute

Cursors and cursor variables have four attributes that give you useful information about the execution of a data manipulation statement.

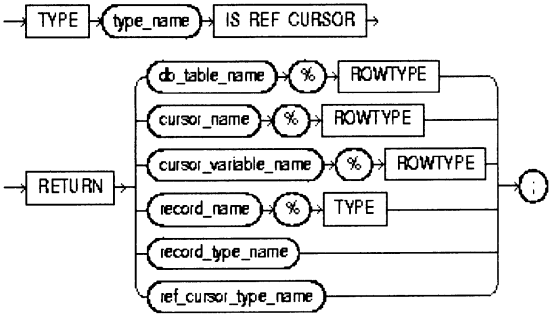
cursor_attribute



Cursor Variable

To execute a multirow query, Oracle server opens an unnamed work area that stores processing information. To access the information, you can use an explicit cursor, which names the work area. Or you can use a cursor variable, which points to the work area.

ref_cursor_type_definition

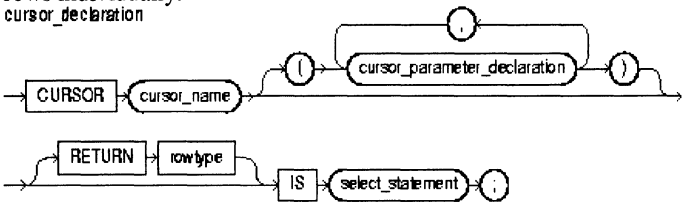


cursor_variable_declaration

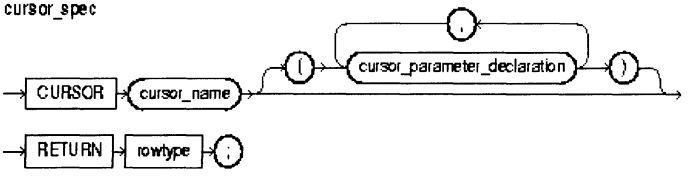
Cursor

To execute a multirow query, the Oracle server opens an unnamed work area that stores processing information. A cursor enables you to name the work area, access the information, and process the rows individually.

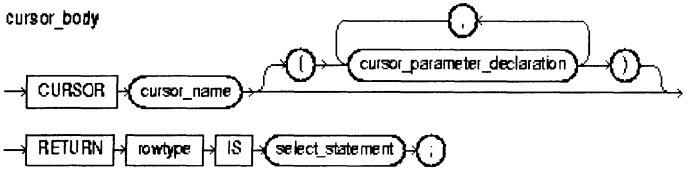
cursor_declaration



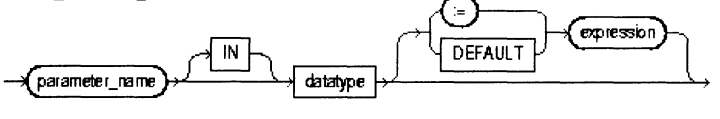
cursor_spec



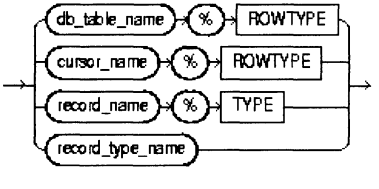
cursor_body



cursor_parameter_declaration



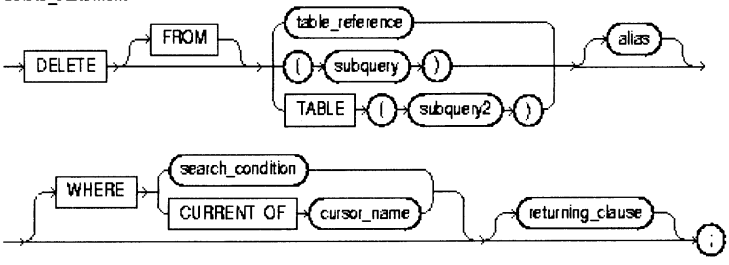
rowtype



DELETE Statement

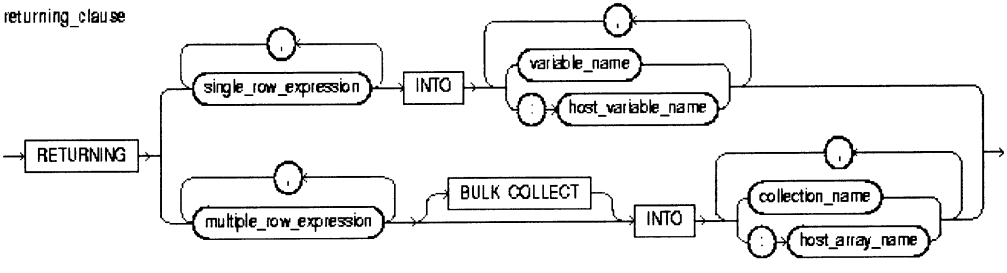
The DELETE statement removes entire rows of data from a specified table or view.

delete_statement



table_reference

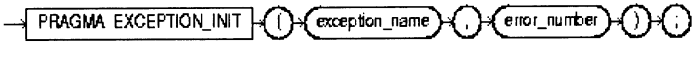
returning_clause



EXCEPTION_INIT PRAGMA

The PRAGMA EXCEPTION_INIT associates an exception name with an Oracle error number. This enables you to refer to any internal exception by name and to write a specific handler for it instead of using the OTHERS handler.

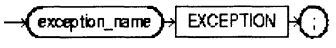
exception_init_pragma



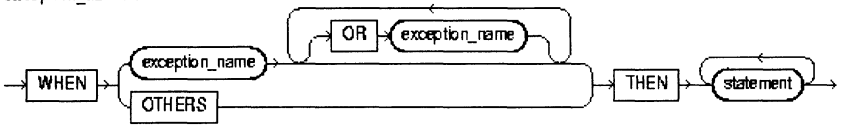
Exception

An exception is a run-time error or warning condition, which can be predefined or user-defined. Predefined exceptions are raised implicitly (automatically) by the run-time system. User-defined exceptions must be raised explicitly by RAISE statements. To handle raised exceptions, you write separate routines called exception handlers.

exception_declaration



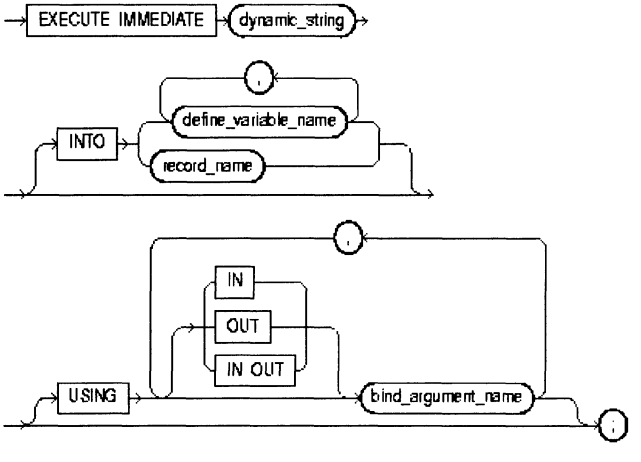
exception_handler



EXECUTE IMMEDIATE Statement

The EXECUTE IMMEDIATE statement prepares (parses) and immediately executes a dynamic SQL statement or an anonymous PL/SQL block.

execute_immediate_statement



EXIT Statement

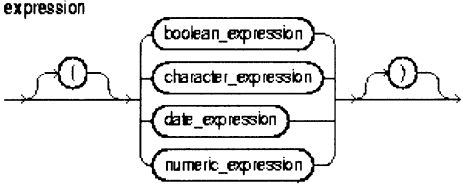
You use the EXIT statement to exit a loop.

exit_statement

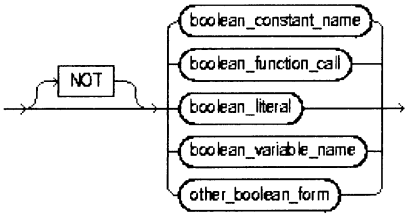
Expression

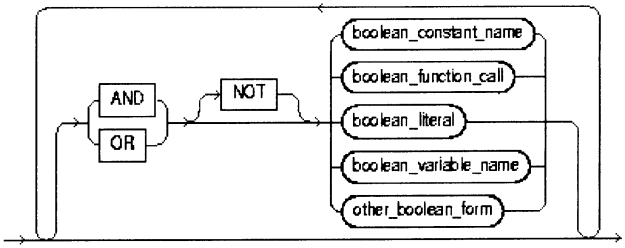
An expression is an arbitrarily complex combination of variables, constants, literals, operators, and function calls.

expression

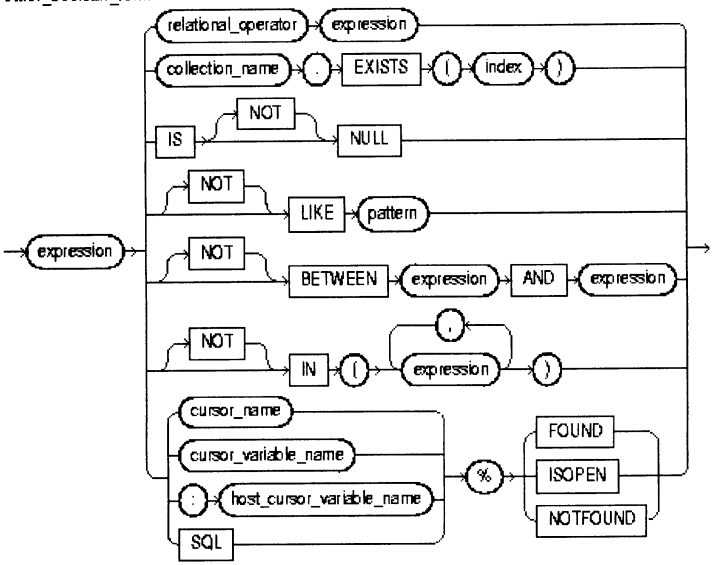


boolean_expression

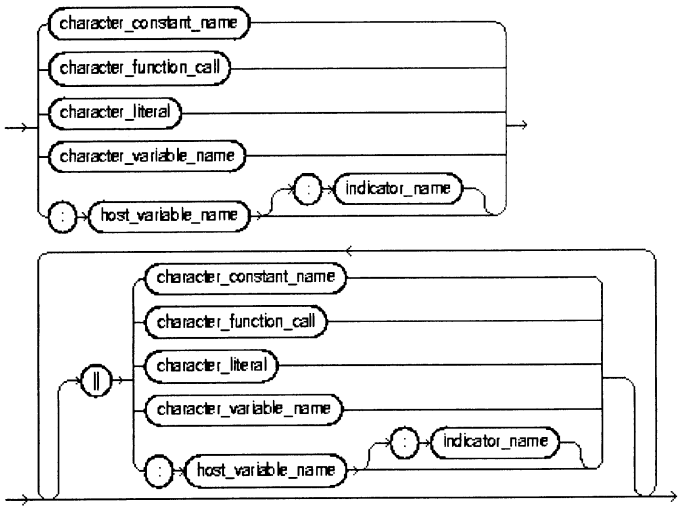




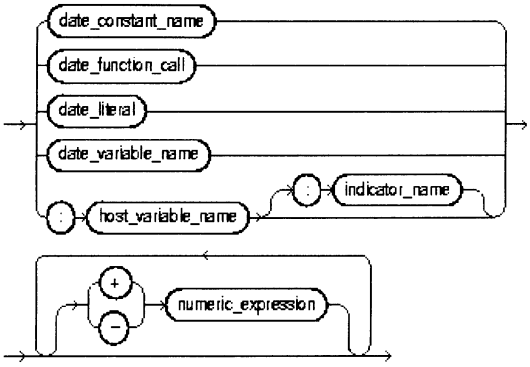
other_boolean_form



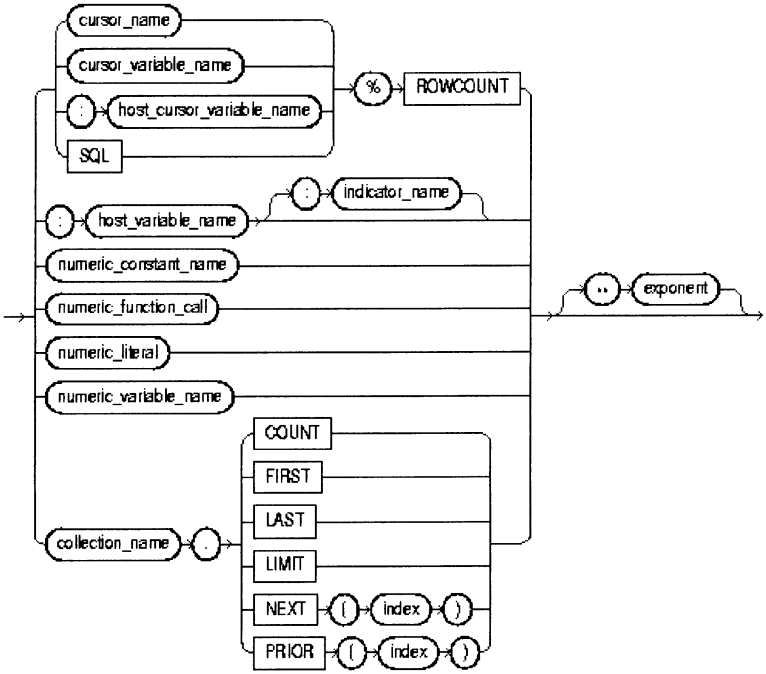
character_expression

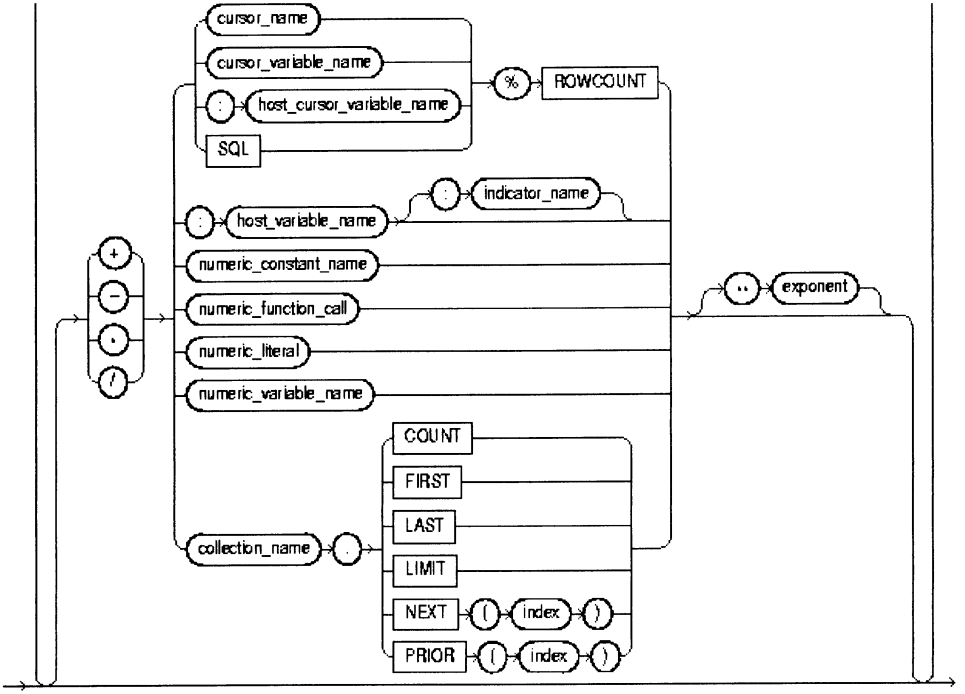


date_expression



numeric_expression

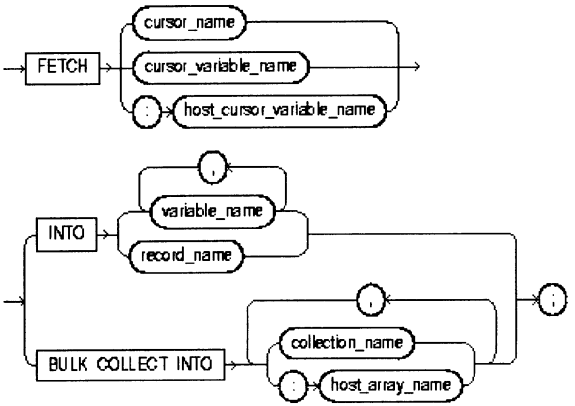




FETCH Statement

The FETCH statement retrieves rows of data one at a time from the result set of a multirow query. The data is stored in variables or fields that correspond to the columns selected by the query.

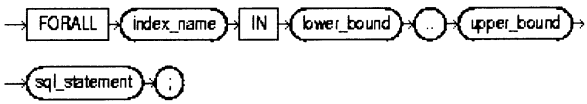
fetch_statement



FORALL Statement

The FORALL statement instructs the PL/SQL engine to bulk-bind input collections before sending them to the SQL engine. Although the FORALL statement contains an iteration scheme, it is not a FOR loop.

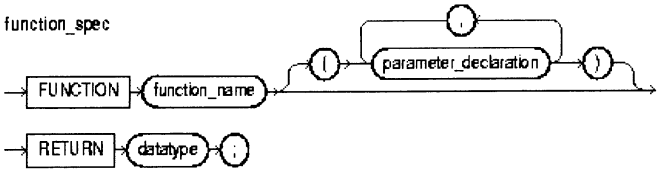
forall_statement



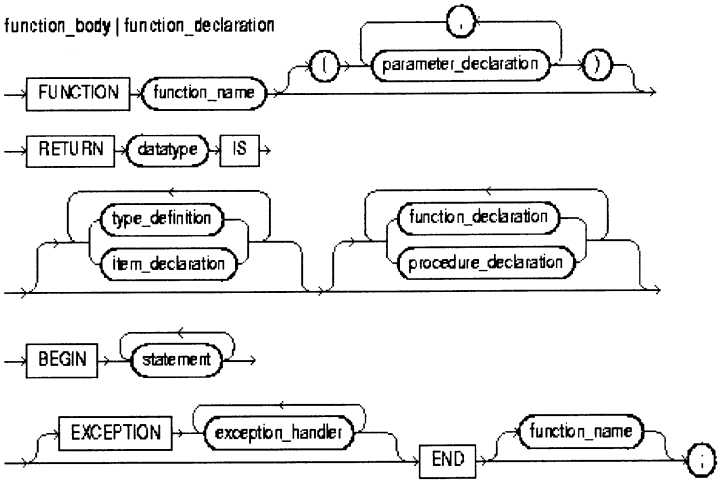
Functions

A function is a subprogram that can take parameters and be invoked. Generally, you use a function to compute a value.

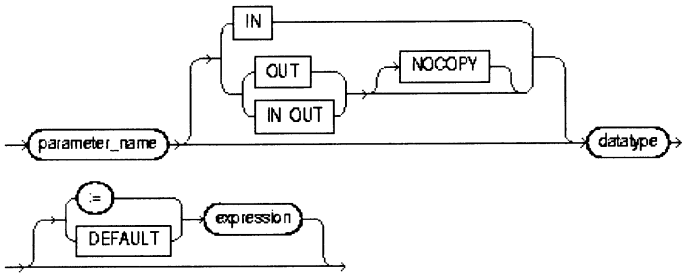
function_spec



function_body | function_declaration



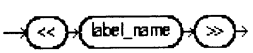
parameter_declaration



GOTO Statement

The GOTO statement branches unconditionally to a statement label or block label.

label_declaration

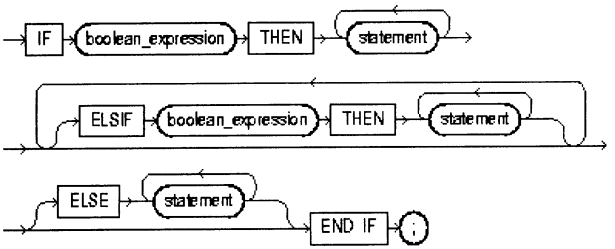


goto_statement

IF Statement

The IF statement lets you execute a sequence of statements conditionally. Whether the sequence is executed or not depends on the value of a Boolean expression.

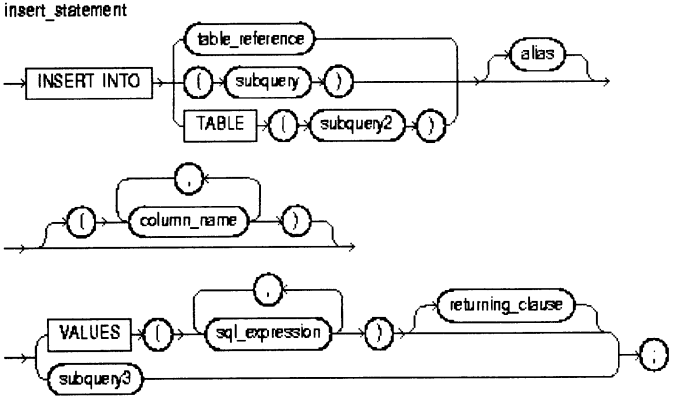
if_statement



INSERT Statement

The INSERT statement adds new rows of data to a specified database table or view.

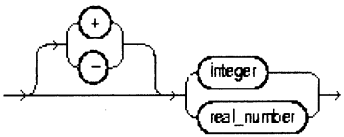
insert_statement



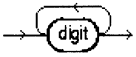
Literal

A literal is an explicit numeric, character, string, or Boolean value not represented by an identifier. The numeric literal `135` and the string literal `'hello world'` are examples.

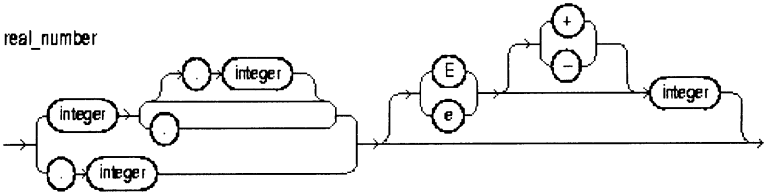
numeric_literal



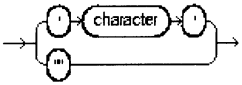
integer



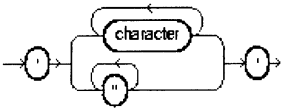
real_number



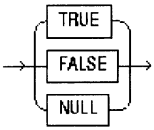
character_literal



string_literal



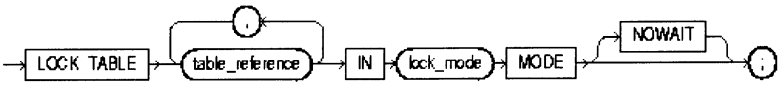
boolean_literal



LOCK TABLE Statement

The LOCK TABLE statement enables you to lock entire database tables in a specified lock mode. In this way you can share or deny access to the tables while maintaining their integrity.

lock_table_statement

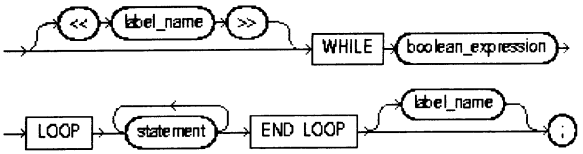


LOOP Statement

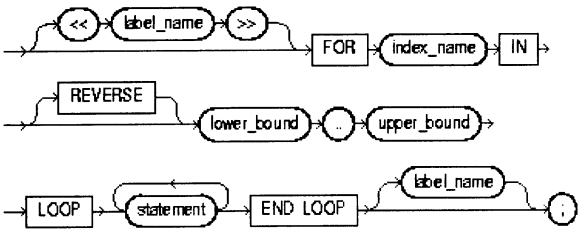
The LOOP statement executes a sequence of statements multiple times. The loop encloses the sequence of statements that is to be repeated. PL/SQL provides basic loops, WHILE loops, FOR loops, and cursor FOR loops.

basic_loop_statement

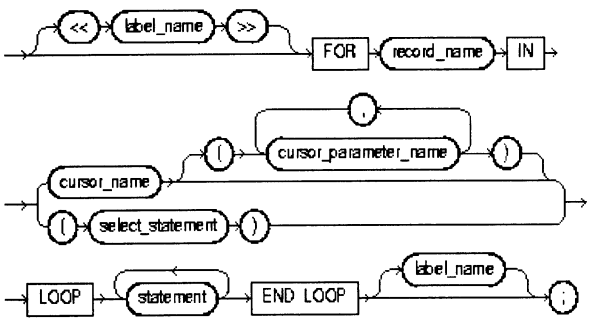
while_loop_statement



for_loop_statement



cursor_for_loop_statement



NULL Statement

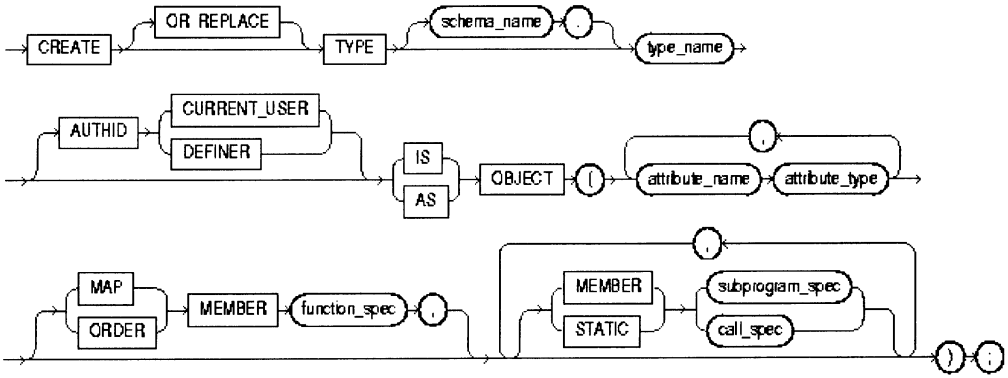
The NULL statement explicitly specifies inaction; it does nothing other than pass control to the next statement. In a construct allowing alternative actions, the NULL statement serves as a placeholder.

null_statement

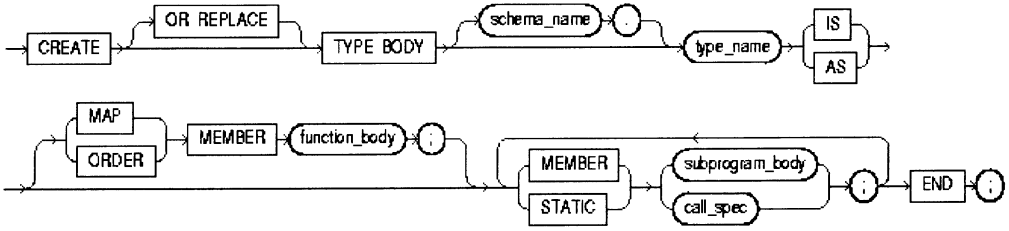
Object Type

An object type is a user-defined composite datatype that encapsulates a data structure along with the functions and procedures needed to manipulate the data. The variables that form the data structure are called attributes. The functions and procedures that characterize the behavior of the object type are called methods.

object_type_declaration | object_type_spec



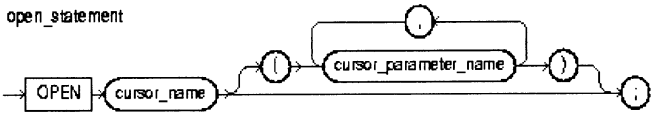
object_type_body



OPEN Statement

The OPEN statement executes the multirow query associated with an explicit cursor. It also allocates resources that Oracle uses to process the query, and it identifies the result set, which consists of all rows that meet the query search criteria.

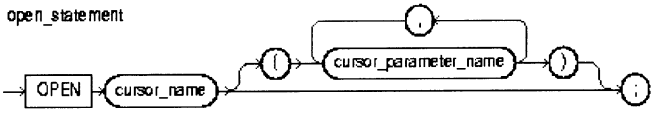
open_statement



OPEN-FOR Statement

The OPEN-FOR statement executes the multirow query associated with a cursor variable. It also allocates resources that Oracle server uses to process the query, and it identifies the result set, which consists of all rows that meet the query search criteria.

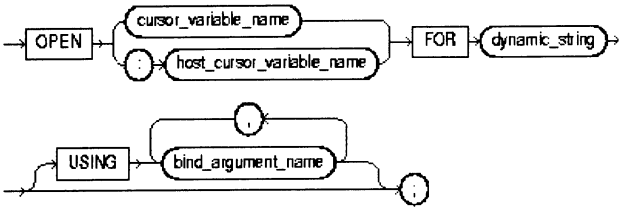
open_statement



OPEN-FOR-USING Statement

The OPEN-FOR-USING statement associates a cursor variable with a multirow query, executes the query, identifies the result set, positions the cursor on the first row in the result set, then zeroes the rows-processed count kept by %ROWCOUNT.

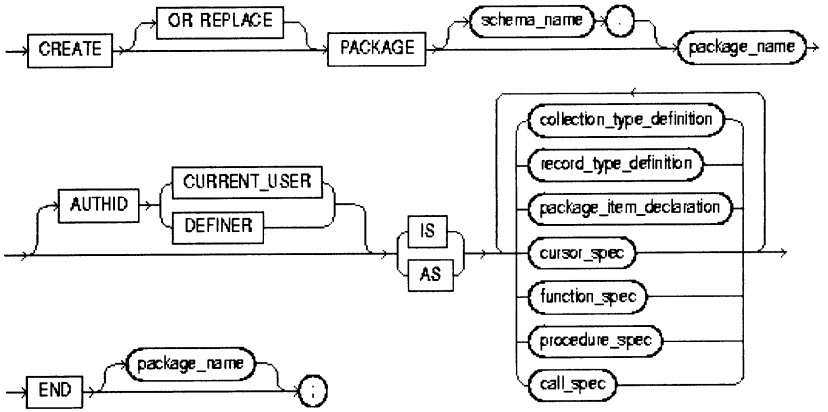
open_for_using_statement



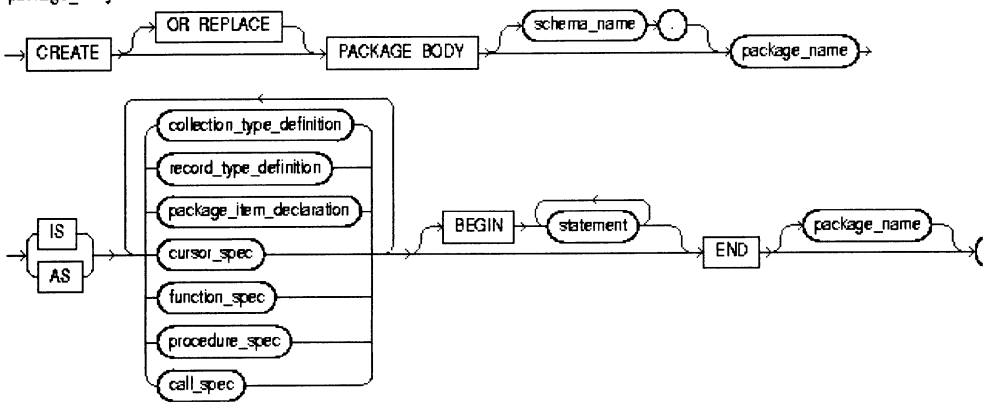
Package

A package is a schema object that groups logically related PL/SQL types, items, and subprograms. Packages have two parts: a specification (spec for short) and a body.

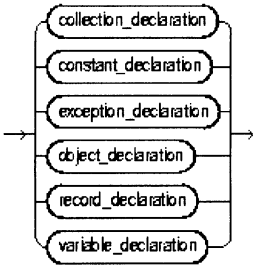
package_declaration | package_spec



package_body



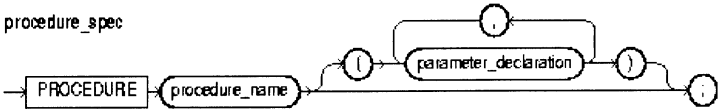
package_item_declaration



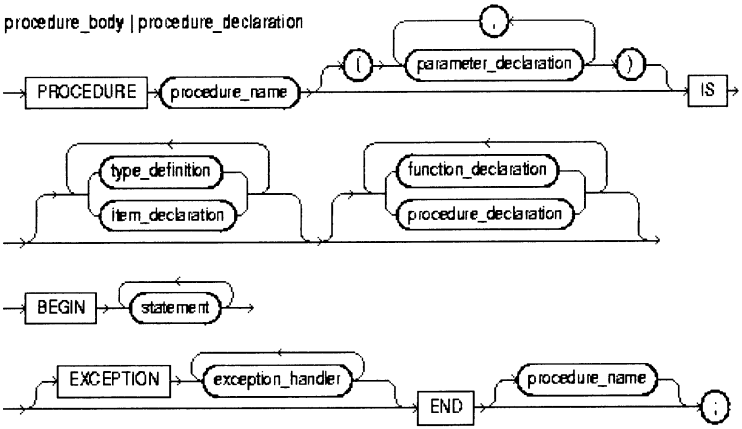
Procedure

A procedure is a subprogram that can take parameters and be invoked. Generally, you use a procedure to perform an action.

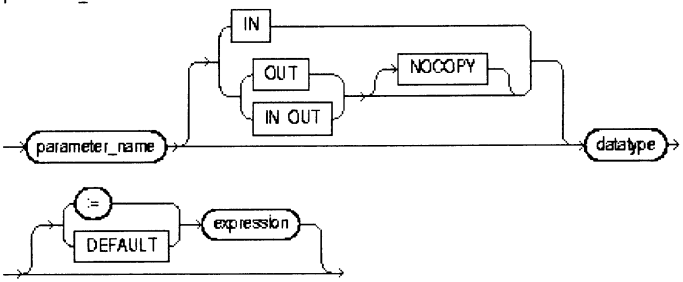
procedure_spec



procedure_body | procedure_declaration



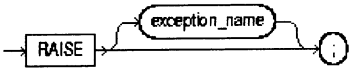
parameter_declaration



RAISE Statement

The RAISE statement stops normal execution of a PL/SQL block or subprogram and transfers control to the appropriate exception handler.

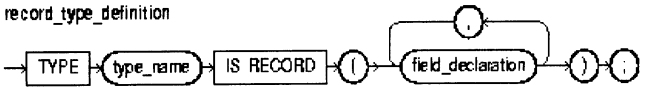
raise_statement



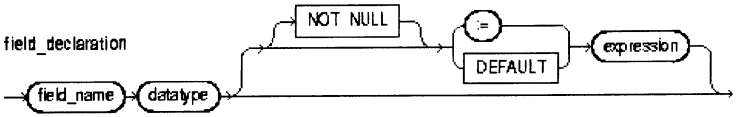
Records

Records are items of type RECORD. Records have uniquely named fields that can store data values of different types.

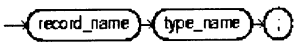
record_type_definition



field_declaration



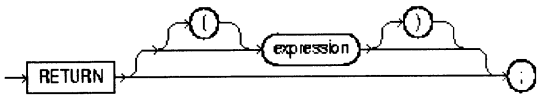
record_declaration



RETURN Statement

The RETURN statement immediately completes the execution of a subprogram and returns control to the caller. Execution then resumes with the statement following the subprogram call. In a function, the RETURN statement also sets the function identifier to the result value.

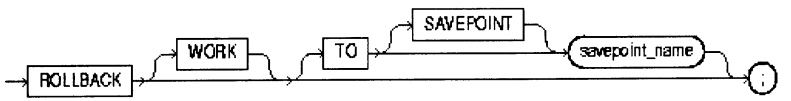
return_statement



ROLLBACK Statement

The ROLLBACK statement is the inverse of the COMMIT statement. It undoes some or all database changes made during the current transaction.

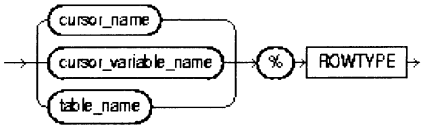
rollback_statement



%ROWTYPE Attribute

The %ROWTYPE attribute provides a record type that represents a row in a database table. The record can store an entire row of data selected from the table or fetched from a cursor or cursor variable. Fields in a record and corresponding columns in a row have the same names and datatypes.

rowtype_attribute



SAVEPOINT Statement

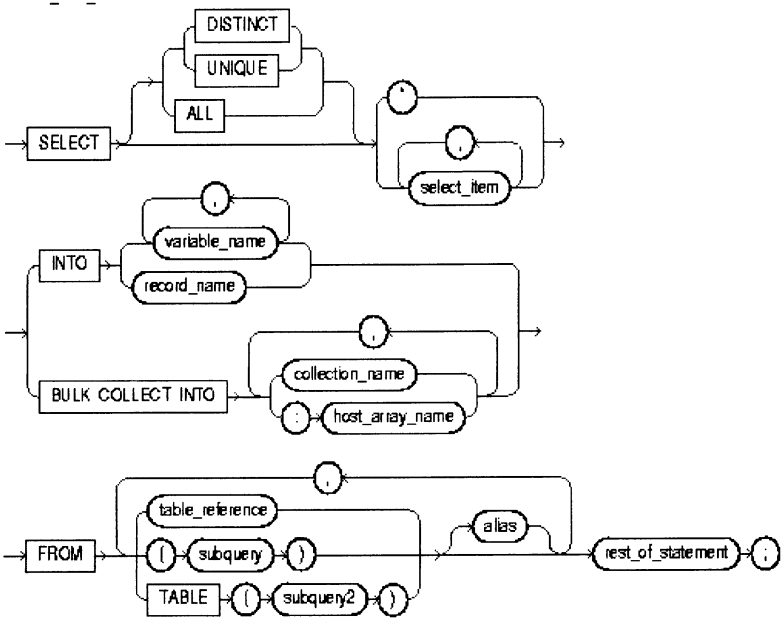
The SAVEPOINT statement names and marks the current point in the processing of a transaction. With the ROLLBACK TO statement, savepoints enable you to undo parts of a transaction instead of the whole transaction.

savepoint_statement

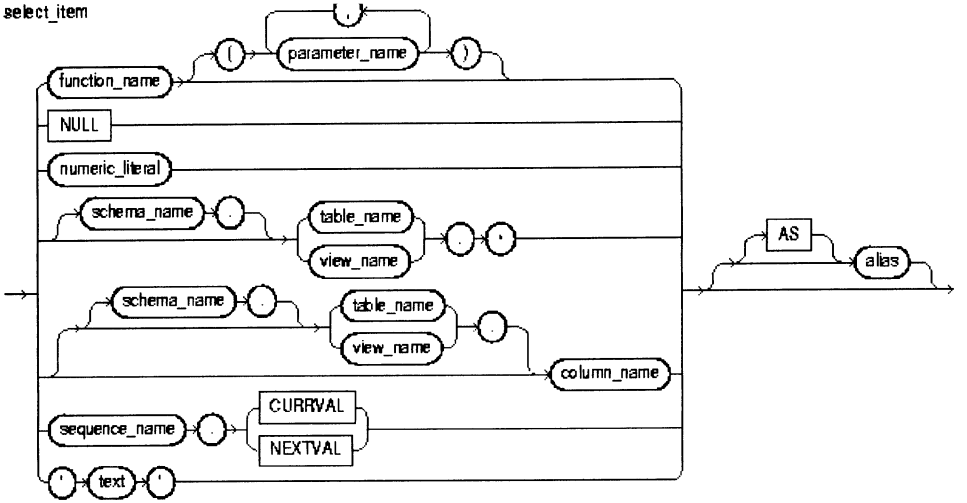
SELECT INTO Statement

The SELECT INTO statement retrieves data from one or more database tables, then assigns the selected values to variables or fields.

select_into_statement



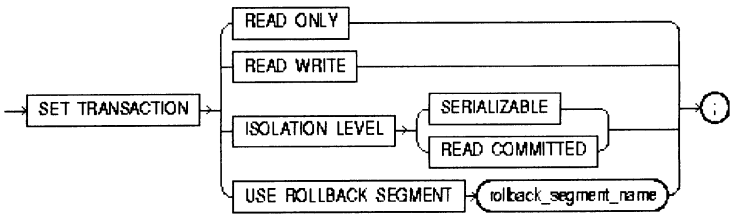
select_item



SET TRANSACTION Statement

The SET TRANSACTION statement begins a read-only or read-write transaction, establishes an isolation level, or assigns the current transaction to a specified rollback segment.

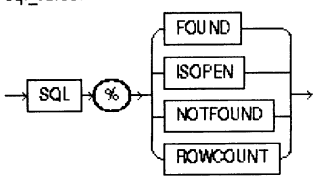
set_transaction_statement



SQL Cursor

The Oracle server implicitly opens a cursor to process each SQL statement that is not associated with an explicit cursor. PL/SQL lets you refer to the most recent implicit cursor as the SQL cursor, which has four attributes: %FOUND, %ISOPEN, %NOTFOUND, and %ROWCOUNT. These attributes give you useful information about the execution of data manipulation statements.

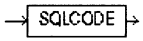
sql_cursor



SQLCODE Function

The function SQLCODE returns the number code associated with the most recently raised exception. SQLCODE is meaningful only in an exception handler. Outside a handler, SQLCODE always returns 0.

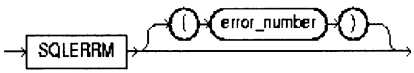
sqlcode_function



SQLERRM Function

The function SQLERRM returns the error message associated with its error-number argument or, if the argument is omitted, with the current value of SQLCODE. SQLERRM with no argument is meaningful only in an exception handler. Outside a handler, SQLERRM with no argument always returns the message of a normal, successful completion.

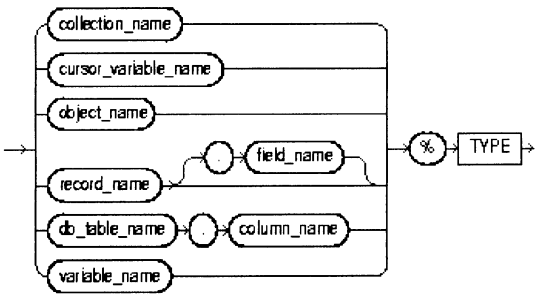
sqlerrm_function



%TYPE Attribute

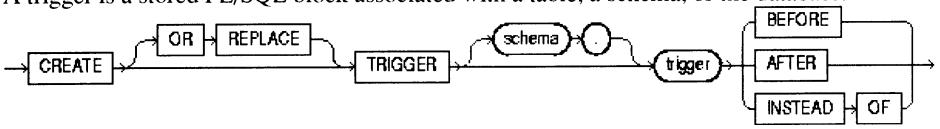
The %TYPE attribute provides the datatype of a field, record, nested table, database column, or variable. You can use the %TYPE attribute as a datatype specifier when declaring constants, variables, fields, and parameters.

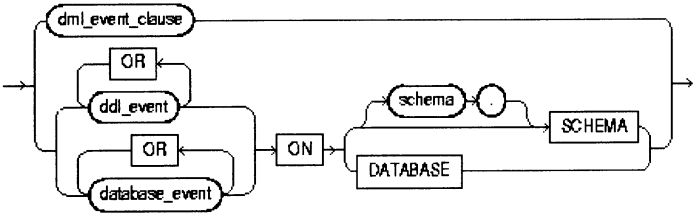
type_attribute

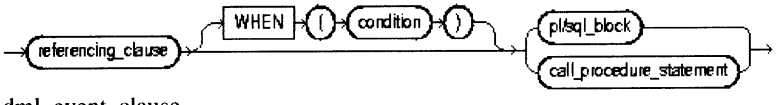


Trigger

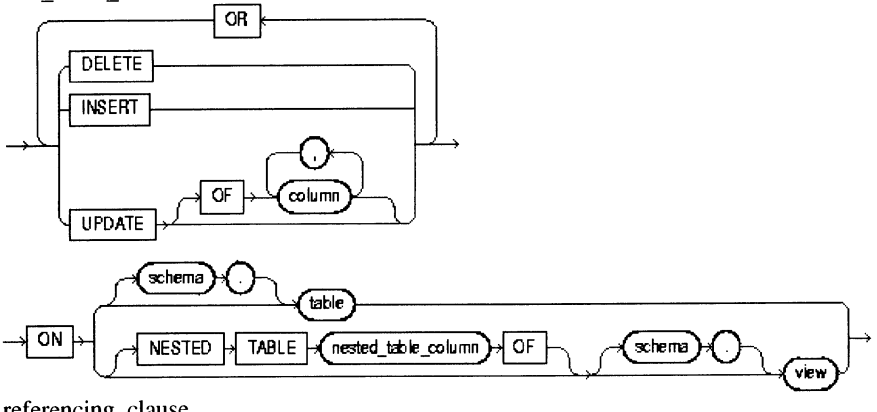
A trigger is a stored PL/SQL block associated with a table, a schema, or the database.



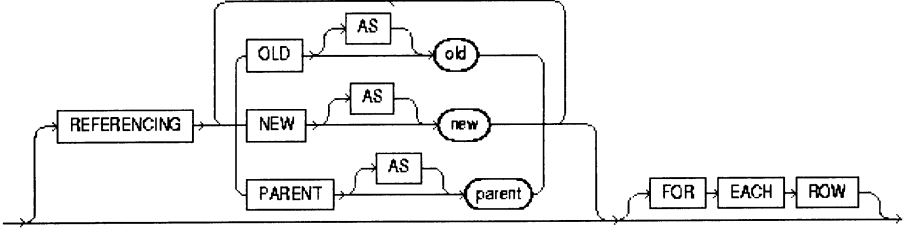




dml_event_clause



referencing_clause



UPDATE Statement

The UPDATE statement changes the values of specified columns in one or more rows in a table or view.

update_statement