Oracle SQL Language Quick Reference

40020GC10
Production 1.0
April 1999
M08684

Oracle SQL Quick Reference

Edition 1.0

Published by Oracle Corporation

Authors: Charlie Gray
Kuljit Jassar
Jeremy Tenison-Smith

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them in writing to Education Products, Oracle Corporation, 500 Oracle Parkway, Box SB-6, Redwood Shores, CA 94065. Oracle Corporation does not warrant that this document is error-free.

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, and initialization 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 used syntax diagrams to show SQL statements, and to show other elements of the SQL language. These syntax diagrams use lines and arrows to show syntactic structure.
If you are not familiar with this type of syntax diagram, refer to*Oracle8 SQL Reference*.

Keywords appear in UPPERCASE. You must use keywords in your SQL statements 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.

The following list shows parameters that appear in the syntax diagrams.

Parameter	**Description**	**Examples**
c	A single character from your database character set	T S
char	An expression of datatype CHAR or VARCHAR2 or a character literal in single quotes	ename ′Smith′
condition	An expression that evaluates to TRUE or FALSE	ename > ′A′
date	A date constant or an expression of datatype DATE	TO_DATE(′01-Jan-1994′, ′DD-Mon-YYYY′)
db_name	The name of a nondefault database in an embedded SQL program	sales_db
db_string	The database identification string for a Net8 database connection	
expr	An expression of any datatype as defined in the syntax description of *expr*	Sal + 100
integer	A number constant or expression of datatype NUMBER	72
number	A number constant or expression of datatype NUMBER	AVG(sal)
raw	An expression of datatype RAW	

subquery	A SELECT statement that will be used in another SQL statement	SELECT ename
		FROM emp

table	The name of an object of the type specified by the parameter	emp
′text′	A text string in single quotes	′Employee records′

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.

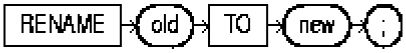
Commands and other keywords appear in uppercase inside rectangles. Type them exactly as shown in the rectangles. Parameters appear in lowercase inside ovals. Variables are used for the parameters. Punctuation, operators, delimiters, and terminators appear inside circles.

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

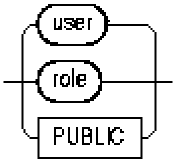
Required keywords and parameters can appear singly or in a vertical list of alternatives. Single required keywords and parameters appear on the main path, that is, on the horizontal line you are currently traveling. In the following example, library_name is a required parameter:



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

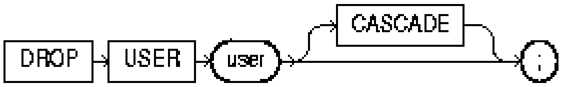
```
RENAME emp TO employees;
```

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 of traveling down a vertical line, you can continue along the main path:

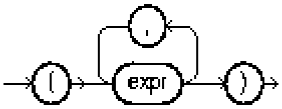


According to the diagram, all of the following statements are valid:

```
DROP USER scott;
DROP USER scott CASCADE;
```

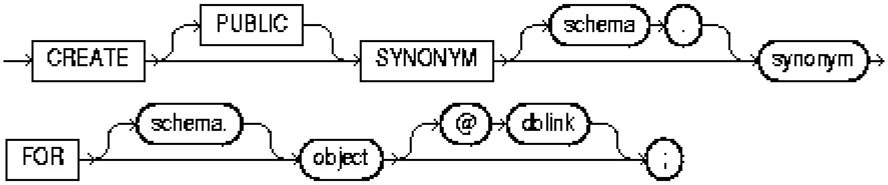
Syntax Loops

Loops enable you to 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*.

Functions

Single-Row Number Functions

Single-row number functions accept numeric input and return numeric values.

Function	**Value Returned**
ABS(*n*)	Absolute value of *n*
ACOS(*n*)	Arc cosine of n
ATAN(*n*)	Arc tangent of n
CEIL(*n*)	Smallest integer greater than or equal to *n*
COS(*n*)	Cosine of *n* in radians
COSH(*n*)	Hyperbolic cosine of *n*
EXP(*n*)	*e* raised to the *n*th power
FLOOR(*n*)	Largest integer equal to less than *n*
LN(*n*)	Natural logarithm of *n*, where *n* > 0
LOG(*m*,*n*)	Logarithm, base *m*, of *n*
MOD(*m*,*n*)	Remainder of *m* divided by *n*
POWER(*m*,*n*)	*m* raised to the *n*th power
ROUND(*n*[,*m*])	*n* rounded to *m* decimal places; *m* defaults to 0
SIGN(*n*)	if *n*<0, -1; if *n*=0, 0; if *n*>0, 1
SIN(*n*)	Sine of *n* in radians
SINH(*n*)	Hyperbolic sine of *n*
SQRT(*n*)	Square root of *n*; if *n*<0, NULL
TAN(*n*)	Tangent of *n* in radians
TANH(*n*)	Hyperbolic tangent of *n*
TRUNC(*n*[,*m*])	*n* truncated to *m* decimal places; *m* defaults to 0

Single-Row Character Functions

Single-row character functions that return character values

Function	**Value Returned**
CHR(*n*)	Character with numeric value *n*
CONCAT(*char1*,*char2*)	Concatenates *char1* and *char2*
INITCAP(*char*)	*char*, with first letter of each word capitalized
LOWER(*char*)	*char*, with all letters lowercase
LPAD(*char1*,*n*[,*char2*])	*char1*, left-padded to display length *n* with the sequence of character in *char2*; *char2* defaults to ”. (A multibyte character that does not fit is replaced with a blank.)
LTRIM(*char*[,*set*])	char, with initial characters removed up to the first character not in set (Set defaults to ”.)
NLS_INITCAP(*n* [,*nls_sort*])	*char*, with first letter in each word capitalized. *nls_sort* defines the handling of language-specific capitalization.
NLS_LOWER(*n* [,*nls_sort*])	*char*, with all characters lowercase. *nls_sort* defines the handling of language-specific capitalization.
NLS_UPPER(*n* [,*nls_sort*])	*char*, with all characters uppercase. *nls_sort* defines the handling of language-specific capitalization.

REPLACE(*char*, *search_string*[, *replacement_string*])	*char*, with every occurrence of *search_string* replaced by *replacement string.* If you do not specify *replacement_string,* Oracle server removes all occurrences of *search_string.*
RPAD(*char1*,*n*[,*char2*])	*char1*, right-padded to display length *n* with the sequence of character in *char2*; *char2* defaults to a blank. (A multibyte character that does not fit is replaced with a blank.)
RTRIM(*char*[,*set*])	*char*, with final characters removed after the last character not in *set.* (Set defaults to ”.)
SOUNDEX(*char*)	A *char* value representing the sound of the characters in *char*, in English
SUBSTR(*char*,*m*[,*n*])	A substring of *char*, beginning at character *m*, *n* characters long (if *n* is omitted, to end of *char*)
SUBSTRB(*char*,*m*[,*n*])	A substring of *char,* beginning at byte *m*, *n* bytes long (if *n* is omitted, to end of *char*)
TRANSLATE(*char*,*from*,*to*)	*char*, translated from the character set *from* to the character set *to*
TRIM([LEADING| TRAILING| BOTH] [*char* FROM] *source*)	Trims heading or trailing characters (or both) from a character string
UPPER(*char*)	*char*, with all characters uppercase.

Single-row character functions that return numeric values

Function	**Value Returned**
ASCII(*char*)	Returns a decimal number equivalent to the value of the first character of *char* in the database character set. (an ASCII value on ASCII systems). Returns value of first byte in a multibyte character.
INSTR(*char1*,*char2*[,*n*[,*m*]])	Position of the *m*th occurrence of *char2* in *char1*, beginning search at character position *n*. *m* and *n* default to 1. Position is relative to the first character of *char1*, even if *n*>1.
INSTRB(*char1*,*char2* [,*n*[,*m*]])	Position of the *m*th occurrence of *char2* in *char1*, beginning search at byte position *n*. *m* and *n* default to 1. Position is relative to the first byte of *char1*, even if *n*>1.
LENGTH(*char*)	Length of *char* in characters
LENGTHB(*char*)	Length of *char* in bytes
NLSSORT(*char* [,*nls_sort*])	The NLS-equivalent value used when sorting *char*

Date Functions

Single-row date functions that return values of datatype DATE, except MONTHS_BETWEEN, which returns a number.

Function	Value Returned
ADD_MONTHS(*d*,*n*)	Date *d* plus *n* months
LAST_DAY(*d*)	Date of last day of month containing *d*
MONTHS_BETWEEN(*d*,*e*)	Number of months by which *e* precedes *d*
NEW_TIME(*d*,*a*,*b*)	Date and time in time zone *b* when date and time in zone *a* are *d*. *a* and *b* are CHAR values identifying time zones.
NEXT_DAY(*d*,*char*)	Date of the first day of the week named by *char* that is equal to or later than the date *d*
SYSDATE	Current date and time

Date Truncation and Rounding

Function	Value Returned
ROUND(*d*,*fmt*)	*d* rounded as specified by rounding unit *fmt*
TRUNC(*d*,*fmt*)	*d* truncated as specified by *fmt*

Format Models for ROUND and TRUNC

This table lists the date format elements used to control truncation and rounding. If no format element is specified, *fmt* defaults to ‘DD’.

Format Element	Value Returned
CC or SCC	First day of century
YYYY or SYYYY YYY or YY or Y Y,YYY or YEAR or SYEAR	First day of year (rounds up on July 1)
Q	First day of the quarter (rounds up on 16th day of 2nd month of the quarter)
MONTH or MON or MM or RM	First day of the month (rounds up on 16th day)
WW or IW	Same day of the week as Jan 1 of that year
W	Same day of the week as the first of the month
DD or DD or J	Day
DAY or DY or D	First day of the week
HH or HH12 or HH24	Hour
MI	Minute

Conversion Functions

Convert a value from one datatype to another.

Function	**Value Returned**
CHARTOROWID(*char*)	*char*, converted from a CHAR value to a ROWID value
CONVERT(*char*, *dest_char_set* [,*source_char_set*])	*char*, converted from source_char_set representation to dest_char_set representation
HEXTORAW(*char*)	*char*, converted from a hexadecimal number to a binary RAW value
RAWTHOHEX(*raw*)	*raw*, converted from a binary value of datatype RAW to a hexadecimal number of datatype CHAR
ROWIDTOCHAR (*rowid*)	*rowid*, converted from a ROWID value to a CHAR value.
TO_CHAR (*expr*[,*fmt* [,’*nls_num_fmt*’]])	*expr*, converted from a NUMBER or DATE value to a CHAR value in the format specified by*fmt*. If you omit *fmt*, ORACLE converts DATE values to default date format and NUMBER values to CHAR values exactly wide enough to hold all significant digits. nls_num_fmt sets the international number-format specifications. In Trusted Oracle, converts values of type MLS or MLS_LABEL to type VARCHAR2.
TO_DATE(*char*[,*fmt* [,’*nls_lang*’]])	*char*, converted from a CHAR value in the format fmt to a DATE value. If you omit *fmt*, char must be in default date format. nls_lang specifies the language used for day and month
TO_MULTI_BYTE (*char*)	*char*, with all single-byte characters that have a multibyte equivalent converted to their multibyte form
TO_NUMBER(*char*[, *fmt* [,’*nls_lang*’]])	*char*, which is a character value containing a number in the format specified by the optional*fmt*, converted to a NUMBER value. nls_lang specifies the language used for numeric characters and currency symbols.
TO_SINGLE_BYTE (*char*)	*char*, with all multibyte characters that have a single-byte equivalent converted to their single-byte form

Format Models for TO_CHAR and TO_DATE

This table lists the date format elements. You can use any combination of these elements as the *fmt* argument of the TO_CHAR or TO_DATE functions. If no format is specified, *fmt* defaults to the default DATE format, ’DD-MON-YY’.

Format Element	**Value Returned**	**Available in TO_DATE**
SCC or CC	Century; ‘S’ prefixes BC date with ‘-’	No
YYYY or SYYYY	Year; ‘S’ prefixes BC date with ‘-’	Yes
YYY or YY or Y	Last 3, 2, or 1 digit(s) of year. Century defaults to current	Yes
IYYY	4-digit ISO standard year	No
IYY or IY or I	Last 3, 2, or 1 digit(s) of ISO year	No
Y,YYY	Year with comma in this position	Yes
SYEAR or YEAR	Year spelled out; ‘S’ prefixes BC date with ‘-’	No
RR	Given a year with 2 digits and knowing the current year and century, calculates the century of the year you specify	Yes
RRRR	Round year. Accepts either 4-digit or 2-digit input. If 2-digit, provides the same return as RR. If you don't want this functionality, enter the 4-digit year.	Yes
BC or AD or B.C. or A.D.	BC/AD indicator with or without periods	Yes
Q	Quarter of year (1,2,3,4; JAN-MAR=1)	No
MM	Month of year (01-12; JAN=01)	Yes
RM	Roman numeral month (I..XII;JAN=I)	Yes
MONTH	Month name, padded with blanks to 9 characters	Yes
MON	Name of month, abbreviated (JAN, FEB)	Yes
WW or W	Week of year (1052) or month (1-5)	No
IW	ISO standard week of the year (1..21 or 1..53)	No
DDD or DD or D	Day of year (1-366) or month (1-31) or week (1-7)	Yes
DAY	Name of day, blank-padded to 9 characters	Yes
DY	Name of day, 3-lettter abbreviation	Yes
J	Julian day (days since December 31, 4713 BC)	Yes
AM or A.M.	Meridian indicator with or without periods	Yes
PM or P.M.	Meridian indicator with or without periods	No
HH or HH12	Hour of the day (1-12)	Yes
HH24	Hour of day (0-23)	Yes
MI	Minute (0-59)	Yes
SS or SSSSS	Second (0-59) or seconds past midnight (0-86399)	Yes
-/,.;:	Punctuation is reproduced in the result.	Yes
"...text..."	Quoted string is reproduced in the result.	Yes

Date Format Prefixes and Suffixes

You can add these prefixes to date format elements:

FM	“Fill mode.” Suppresses blank padding when prefixed to MONTH or DAY. FM is a toggle. Use FM twice to reenable blank padding.
FX	“Format exact.” Specifies exact matching for the character argument and date format model of a TO_DATE function

You can add these suffixes to date format elements:

TH	Ordinal number (“DDTH” for “4TH”)
SP	Spelled-out number (“DDSP” for “FOUR”)
SPTH and THSP	Spelled-out ordinal number (“DDSPTH” for “FOURTH”)

Date Format Case Control

The following strings output in uppercase, initial caps, or lowercase:

Uppercase	**Initial Caps**	**Lowercase**
DAY	Day	day
DY	Dy	dy
MONTH	Month	month
MON	Mon	mon
YEAR	Year	year
AM	Am	am
PM	Pm	pm
A.M. A.m.		am
P.M., P.m.		pm

When prefixes or suffixes are added to a date format, the case (uppercase, initial caps, or lowercase) is determined by the format element, not by the prefix or suffix. For example, 'ddTH' produces "04th", not "04TH"

Number Format Models for TO_CHAR

This table lists the number format elements. You can use a combination of these elements as the *fmt* argument of the TO_CHAR function.

Format Element	**Example**	**Function**
9	9999	Number of *9*s determines length of returned character
0	0999	Prefixes value with leading zeroes
$	$9999	Prefixes value with dollar sign
B	B9999	Returns zero value as blank, instead of “0”
MI	9999MI	Returns “-” after negative values
S	S9999	Returns “+” for positive values and “-” for negative values
PR	9999PR	Returns negative values in <angle brackets>
D	99D99	Returns the decimal character
G	9G999	Returns the group separator
C	C999	Returns the international currency symbol
L	L999	Returns the local currency symbol
,	9,999	Returns a comma in this position
.	99.99	Returns a period in this position
V	999V99	Multiplies value by 10*n*, where *n* is the number of *9*s after the *V*

EEEE	9.999EEEE	Returns value in scientific notation.*fmt* must contain exactly four *E*s
RN or rn	RN	upper- or lower-case Roman numerals (numbers in range 1..3999)
DATE	‘DATE’	Returns value converted from Julian date to ‘MM/DD/YY’ date format

Miscellaneous Single-Row Functions

Functions that do not fall into any of the other single-row function categories

Function	**Value Returned**
DECODE(*expr*,*search1*, *return1*, [*search2*,*return2*,]... [*default*])	If *expr* equals any search, returns the following return; if not, returns default
DUMP(*expr* [,*display_format* [,*start_position* [,*length*]]])	*expr* in Oracle internal format
GREATEST(*expr* [,*expr*]...)	*expr* with the greatest value
LEAST(*expr*[,*expr*]...)	*expr* with the least value
NVL(*expr1*,*expr2*)	*expr2*, if *expr1* is null; otherwise returns *expr1*
UID	Number that uniquely identifies the current user
USER	Name of the current user
USERENV(*‘option’*)	Information about the current session. Specify option in single quotes. Options: ENTRYID, SESSIONID, TERMINAL, and LANGUAGE.
VSIZE(*expr*)	The number of bytes in Oracle’s internal representation of *expr*.

Object Reference Functions

Manipulate REFs, which are references to objects of specified object types

Function	**Value Returned**
DEREF(*e*)	The object reference of argument *e*
MAKE_REF(*table*|*view*, *key*[,*key*]...)	Creates a REF to a row of an object view or a row in an object table whose identifier is based on a primary key
REF(*correlation_variable*)	REF takes as its argument a correlation variable (table alias) associated with a row of an object table or an object view.
REFTOHEX(*r*)	Converts argument *r* to a character value containing its hexadecimal equivalent
VALUE(*correlation_variable*)	VALUE takes as its argument a correlation variable (table alias) associated with a row of an object table and returns object instances stored in the object table.

Group Functions

Return a single row based on groups of rows, rather than on single rows

Function	Value Returned
AVG([DISTINCT|ALL] *n*)	Average value of *n*, ignoring nulls
COUNT([ALL] *)	Number of rows returned by a query or subquery
COUNT([DISTINCT| ALL] *expr*)	Number of rows where *expr* is not null
MAX([DISTINCT|ALL] *expr*)	Maximum value of *expr*
MIN([DISTINCT|ALL] *expr*)	Minimum value of *expr*
STTDEV([DISTINCT|ALL] *n*)	Standard deviation of *n*, ignoring null values
SUM([DISTINCT|ALL] *n*)	Sum of values of *n*
VARIANCE ([DISTINCT| ALL] *n*)	Variance of n, ignoring null values

Pseudocolumns

Pseudocolumns are similar to table columns. You can query any of these pseudocolumns, but you cannot change their values with DML statements

Column Name	Value
sequence.CURRVAL	Current value of sequence for the current session (sequence.NEXTVAL must be referenced first.)
sequence.NEXTVAL	Next value of sequence for the current session
table.LEVEL	1 for a root node, 2 for a child of a root, and so on. Used in the SELECT statement during tree-structured queries.
[*table*.]ROWID	Value that uniquely identifies a single row among other rows in the table. ROWID values are of datatype ROWID, not NUMBER or CHAR.
ROWNUM	Position of a single row among other rows selected by a query. Oracle server selects rows in an arbitrary order and evaluates ROWNUM before sorting rows for an ORDER BY clause.

Datatypes

ORACLE accepts the following datatypes:

Datatype	**Explanation**
CHAR(*size*)	Fixed-length character data of length *size* bytes. Maximum *size* is 2000. Default and minimum *size* is 1 byte.
VARCHAR2(*size*)	Variable-length character string having maximum length *size* bytes. Maximum *size* is 4000, and minimum is 1. You must specify a *size*.
NCHAR(*size*)	Fixed-length character data of length *size* characters or bytes, depending on the choice of national character set. Maximum *size* is determined by the number of bytes required to store each character, with an upper limit of 2000 bytes. Default and minimum *size* is 1 character or 1 byte, depending on the character set.
NVARCHAR2(*size*)	Variable-length character string having maximum length *size* characters or bytes, depending on the choice of national character set. Maximum *size* is determined by the number of bytes required to store each character, with an upper limit of 4000 bytes. You must specify a *size*.
NUMBER[(*precision* [,*scale*])]	Number having *precision* and *scale*. The precision can range from 1 to 38. The scale can range from -84 to 127.
LONG	Character data of variable length up to 2 gigabytes.
LONG RAW	Raw binary data of variable length up to 2 gigabytes.
RAW(*size*)	Raw binary data of length *size* bytes. Maximum *size* is 2000 bytes. You must specify a *size*.
DATE	Valid date range from January 1, 4712 BC to December 31, 9999 AD.
BLOB	A binary large object. Maximum size 4 gigabytes.
CLOB	A character large object containing single-byte characters. Both fixed-width and variable-width character sets are supported, both using the CHAR database character set. Maximum size is 4 gigabytes.
NCLOB	A character large object containing single-byte characters. Both fixed-width and variable-width character sets are supported, both using the NCHAR database character set. Maximum size is 4 gigabytes. Stores national character set data.
BFILE	Contains a locator to a large binary file stored outside the database.
ROWID	Hexadecimal string representing the unique address of a row in its table. This datatype is primarily for values returned by the ROWID pseudocolumn.
UROWID(*size*)	Hexadecimal string representing the logical address of a row of an index-organized table. The optional *size* is the size of a column of type UROWID. The maximum size and default is 4000 bytes.

The Oracle server accepts the following ANSI-supported datatypes:
CHARACTER(*size*)
CHARACTER VARYING(*size*)
CHAR VARYING(*size*)
VARCHAR(*size*)
NATIONAL CHARACTER(*size*)
NATIONAL CHAR(*size*)
NATIONAL CHARACTER VARYING (*size*)
NATIONAL CHAR VARYING(*size*)
NCHAR VARYING(*size*)
NUMERIC[(*precision* [,*scale*])]
DECIMAL[(*precision* [,*scale*])]
DEC[(*precision* [,*scale*])]
INTEGER
INT
SMALLINT
FLOAT[(*size*)]
DOUBLE PRECISION
REAL

Object Naming Rules

The following rules apply when naming objects:

- Names must be from 1 to 30 characters long with these exceptions:
  - Names of databases are limited to 8 characters.
  - Names of database links can be as long as 128 characters.
- 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 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 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, the Oracle server uses them internally, and this may cause your statements to be difficult to read and may lead to unpredictable results.
- Within a namespace, no two objects can have the same name. Tables and views are in the same namespace; therefore, a table and a view in the same schema cannot have the same name.

Namespaces for Schema Objects

TABLES VIEWS SEQUENCES PRIVATE SYNONYMS STAND-ALONE PROCEDURES STAND-ALONE STORED FUNCTIONS PACKAGES MATERIALIZED VIEWS/SNAPSHOTS

INDEXES

CONSTRAINTS

CLUSTERS

DATABASE TRIGGERS

PRIVATE DATABASE LINKS

DIMENSIONS

Note: You cannot use special characters from European or Asian character sets in a database name, global database name, or database link names. For example, you cannot use the umlaut.

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 the 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.

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

SQL Reserved Words

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

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*

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

SYSTEM PRIVILEGES

INDEX

System Privilege	Allows Grantee To ...
CREATE INDEX	Create in the grantee’s schema an index on any table in the grantee’s schema or a domain index
CREATE ANY INDEX	Create in any schema except SYS a domain index or an index on any table in any schema except SYS
ALTER ANY INDEX	Alter indexes in any schema except SYS
DROP ANY INDEX	Drop indexes in any schema except SYS
QUERY REWRITE	Enable rewrite using a materialized view, or create a function-based index, when that materialized view or index references tables and views that are in the grantee’s own schema
GLOBAL QUERY REWRITE	Enable rewrite using a materialized view, or create a function-based index, when that materialized view or index references tables or views in any schema except SYS

ROLES

System Privileges	
CREATE ROLE	Create roles
ALTER ANY ROLE	Alter any role in the database
DROP ANY ROLE	Drop roles
GRANT ANY ROLE	Grant any role in the database

SEQUENCES

System Privileges	
CREATE SEQUENCE	Create a sequence in grantee’s schema
CREATE ANY SEQUENCE	Create sequences in any schema except SYS
ALTER ANY SEQUENCE	Alter any sequence in the database
DROP ANY SEQUENCE	Drop sequences in any schema except SYS
SELECT ANY SEQUENCE	Reference sequences in any schema except SYS

SESSIONS

System Privileges

CREATE SESSION	Connect to the database
ALTER RESOURCE COST	Set costs for session resources
ALTER SESSION	Issue ALTER SESSION statements
RESTRICTED SESSION	Log on after the database is started using the SQL*Plus STARTUP RESTRICT statement

SYNONYMS

System Privileges

CREATE SYNONYM	Create synonyms in grantee’s schema
CREATE ANY SYNONYM	Create private synonyms in any schema except SYS
CREATE PUBLIC SYNONYM	Create public synonyms
DROP ANY SYNONYM	Drop private synonyms in any schema, except SYS
DROP PUBLIC SYNONYM	Drop public synonyms

TABLES

System Privileges

CREATE TABLE	Create tables in grantee’s schema. To create a table, the grantee must also have space quota on the tablespace to contain the table.
CREATE ANY TABLE	Create tables in any schema. The owner of the schema containing the table must have space quota on the tablespace to contain the table.
ALTER ANY TABLE	Alter any table or view in the schema
BACKUP ANY TABLE	Use the Export utility to incrementally export objects from the schema of other users
DELETE ANY TABLE	Delete rows from table, table partitions, or views in any schema except SYS
DROP ANY TABLE	Drop or truncate tables in any schema except SYS
INSERT ANY TABLE	Insert rows into tables and views in any schema except SYS
LOCK ANY TABLE	Lock tables and views in any schema except SYS
UPDATE ANY TABLE	Update rows in tables and views in any schema except SYS
SELECT ANY TABLE	Query tables, views, or snapshots in any schema except SYS
COMMENT ANY TABLE	Comment on any table, view, or column in any schema except SYS

USERS

System Privileges

CREATE USER	Create users. This privilege also allows the creator to: • Assign quotas on any tablespace • Set default and temporary tablespaces • Assign a profile as part of a CREATE USER statement
ALTER USER	Alter any user. This privilege authorizes the grantee to: • Change another user’s password or authentication method • Assign quotas on any tablespace • Set default and temporary tablespaces • Assign a profile and default roles
BECOME USER	Become another user (required by any user performing a full database import)
DROP USER	Drop users

VIEWS

System Privileges

CREATE VIEW	Create views in grantee’s schema
CREATE ANY VIEW	Create view in any schema except SYS
DROP ANY VIEW	Drop views in any schema except SYS

MISCELLANEOUS

System Privileges

GRANT ANY PRIVILEGE	Grant any system privilege

OBJECT PRIVILEGES

These privileges apply to specific objects.

Object Privilege	Allows Grantee To ...
ALL [PRIVILEGES]	Apply 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

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

Operators

Arithmetic Operators

Operator	Purpose
+ -	Denotes a positive or negative expression
* /	Multiplies, divides
+ -	Adds, subtracts

Logical Operators

Operator	Function
NOT	Returns TRUE if the following condition is FALSE Returns FALSE if it is TRUE; if it is UNKNOWN, it remains UNKNOWN
AND	Returns TRUE if both component conditions are TRUE Returns FALSE if either is FALSE; otherwise returns UNKNOWN
OR	Returns TRUE if either component condition is TRUE Returns FALSE if both are FALSE; otherwise returns UNKNOWN

Comparison Operators

Operator	Purpose
=	Equality test
!= ^= ¬= <>	Inequality test. Some forms of the inequality operator may be unavailable on some platforms
> >= < <=	‘Greater than’, ‘greater than or equal to’, ‘less than’, and ‘less than or equal to’ tests
IN	‘Equal to any member of’ test. Equivalent to ‘= ANY’.
NOT IN	Equivalent to ‘!=ALL’. Evaluates to FALSE if any member of the set is NULL.
ANY SOME	Compares a value to each value in a list or returned by a query. Must be preceded by =, !=. >, <, >=, <=. Evaluates to FALSE if the query returns no rows
ALL	Compares a value to every value in a list or returned by a query. Must be preceded by =, !=. >, <, >=, <=. Evaluates to TRUE if the query returns no rows
[NOT] BETWEEN x AND y	[Not] greater than or equal to *x* and less than or equal to *y*
[NOT] EXISTS	TRUE if a subquery returns [does not return] at least one row
x [NOT] LIKE y [ESCAPE 'z']	TRUE if *x* does [not] match the pattern *y*. Within *y*, the character ‘%’ matches any string of zero of more characters except null. The character ‘_’ matches any single character. Any character, except percent (%) and underbar (_), may follow ESCAPE. A wildcard character

	is treated as a literal if preceded by the character designated as the escape character.
IS [NOT] NULL	Tests for nulls. This is the only operator that you should use to test for nulls.

Set Operators

Operator	**Returns**
UNION	All rows selected by either query
UNION ALL	All rows selected by either query, including all duplicates
INTERSECT	All distinct rows selected by both queries
MINUS	All distinct rows selected by the first query but not the second

<table>
<tr><th>Operator</th><th>Purpose</th></tr>
<tr><td>|| Concatenation operator</td><td>Concatenates character strings</td></tr>
<tr><td>(+) Outer Join operator</td><td>Indicates that the preceding column is an outer join column in a join</td></tr>
<tr><td>PRIOR</td><td>Evaluates the following expression for the parent row of the current row in a hierarchical, or tree-structured, query. You must use this operator in the CONNECT BY clause to define the relationship between parent and child rows. You can also use this operator in other parts of a SELECT statement that performs a hierarchical query.</td></tr>
</table>

Operator Precedence

When evaluating an expression, the Oracle server evaluates operators with greater precedence first. Operators on the same line have the same precedence.
Note: Parentheses() override normal precedence. The Oracle server evaluates operators inside parentheses before those outside. Without parentheses, operators at the same precedence are evaluated from left to right.

SQL Operator Precedence

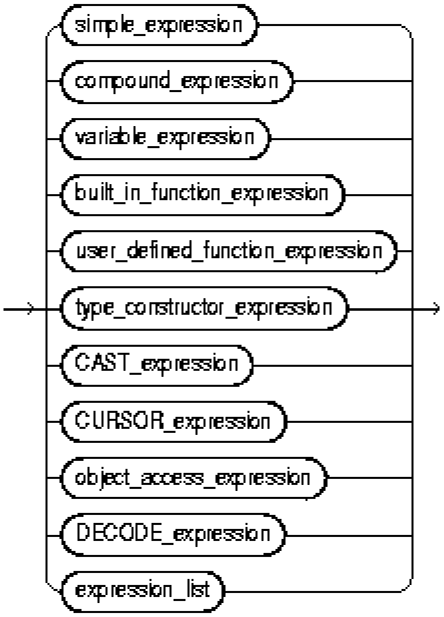
<table>
<tr><th>Operator</th><th>Operation</th></tr>
<tr><td>+, -</td><td>identity, negation</td></tr>
<tr><td>*, /</td><td>multiplication, division</td></tr>
<tr><td>+, -, ||</td><td>addition, subtraction, concatenation</td></tr>
<tr><td>=, !=, <, >, <=, >=, IS NULL, LIKE, BETWEEN, IN</td><td>comparison</td></tr>
<tr><td>NOT</td><td>exponentiation, logical negation</td></tr>
<tr><td>AND</td><td>conjunction</td></tr>
<tr><td>OR</td><td>disjunction</td></tr>
</table>

Expressions and Conditions

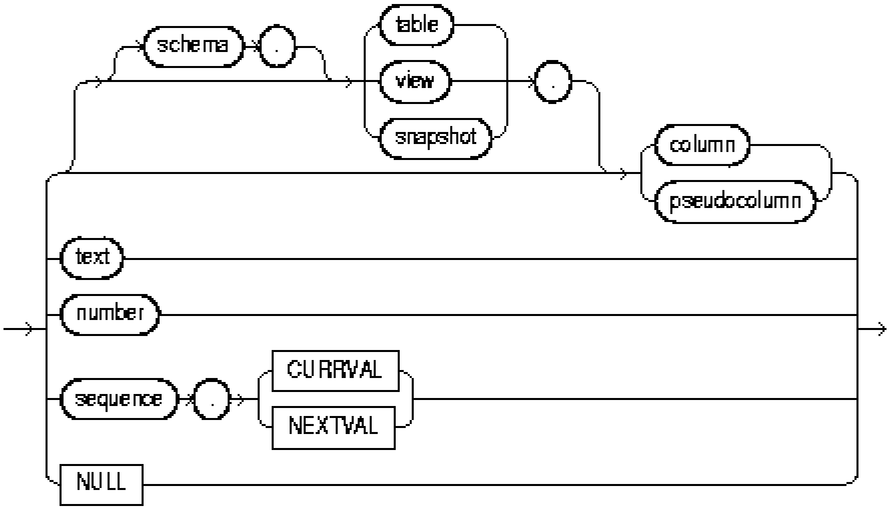
The following diagrams show the possible expressions and conditions denoted by “expr” and “condition” in the SQL command syntax diagrams.

Expressions

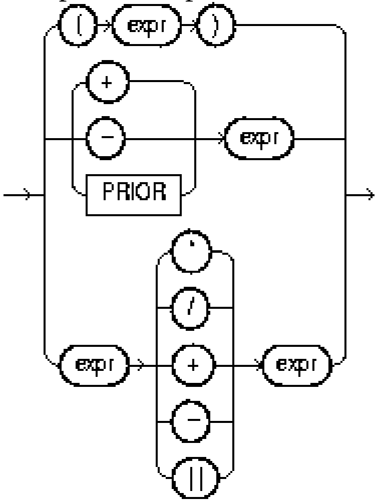
expr ::=



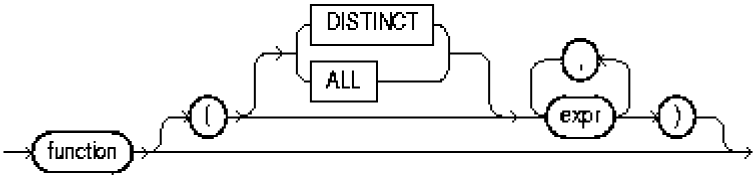
simple_expression ::=



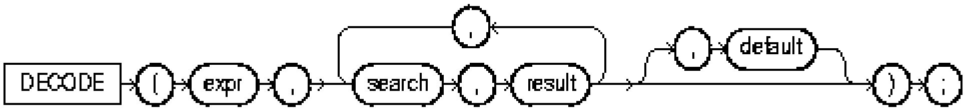
compound_expression ::=



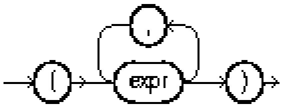
built_in_function_expression ::=



DECODE_expression ::=

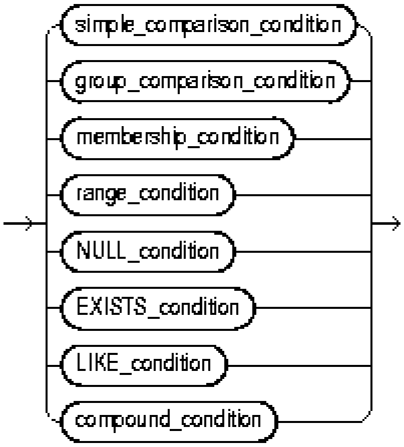


expression_list ::=

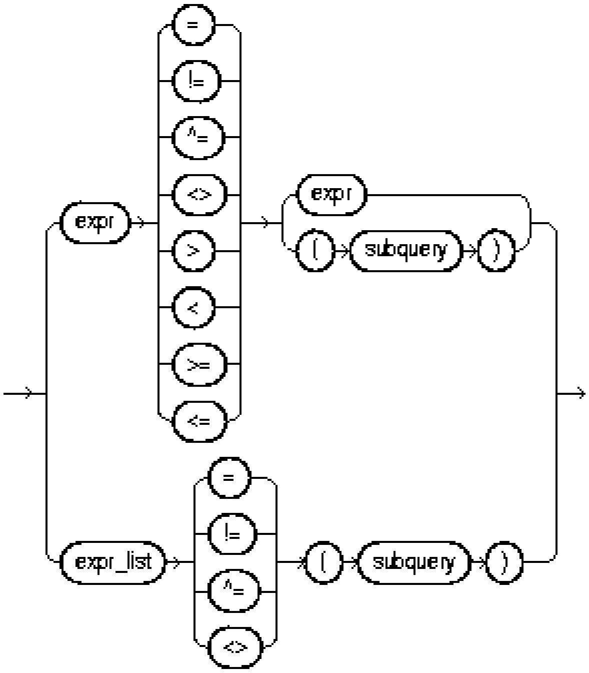


Conditions

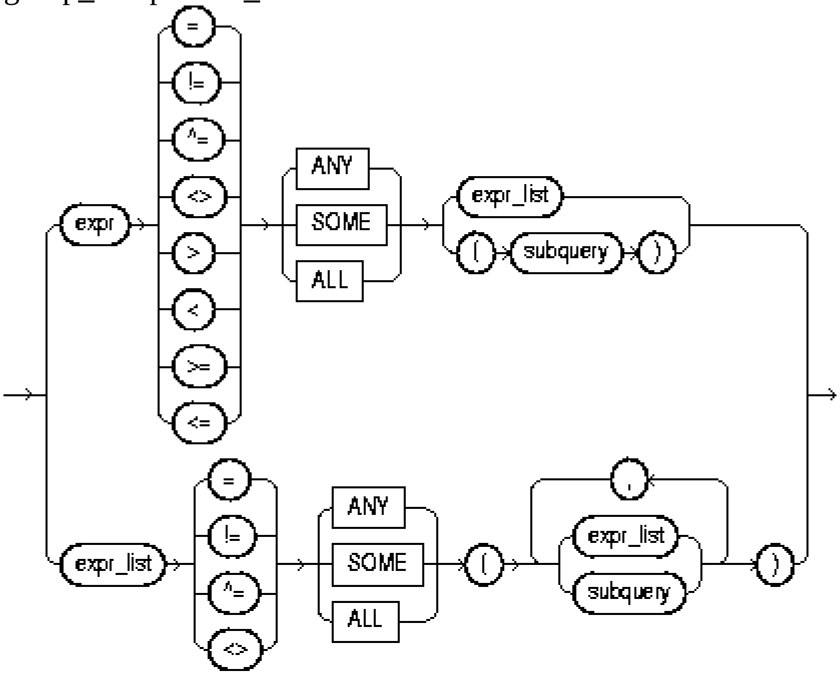
condition ::=



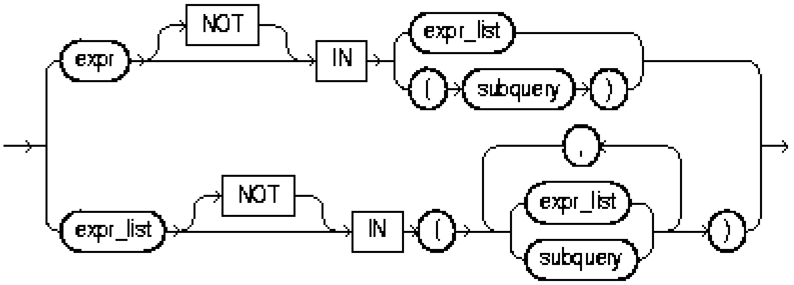
simple_comparison_condition ::=



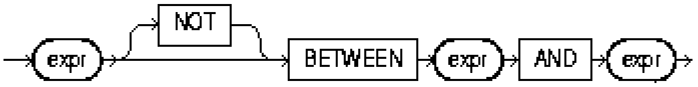
group_comparison_condition ::=



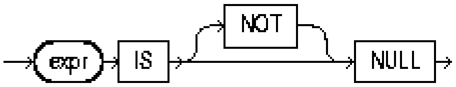
membership_condition ::=



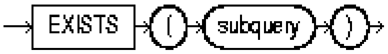
range_conditions ::=



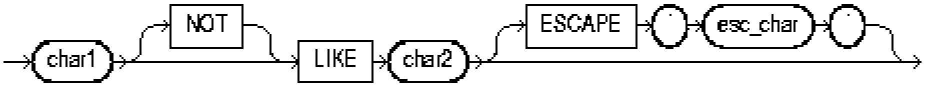
NULL_condition ::=



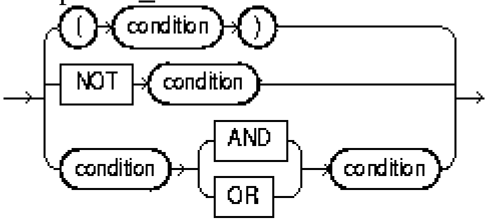
EXISTS_condition ::=



LIKE_condition ::=



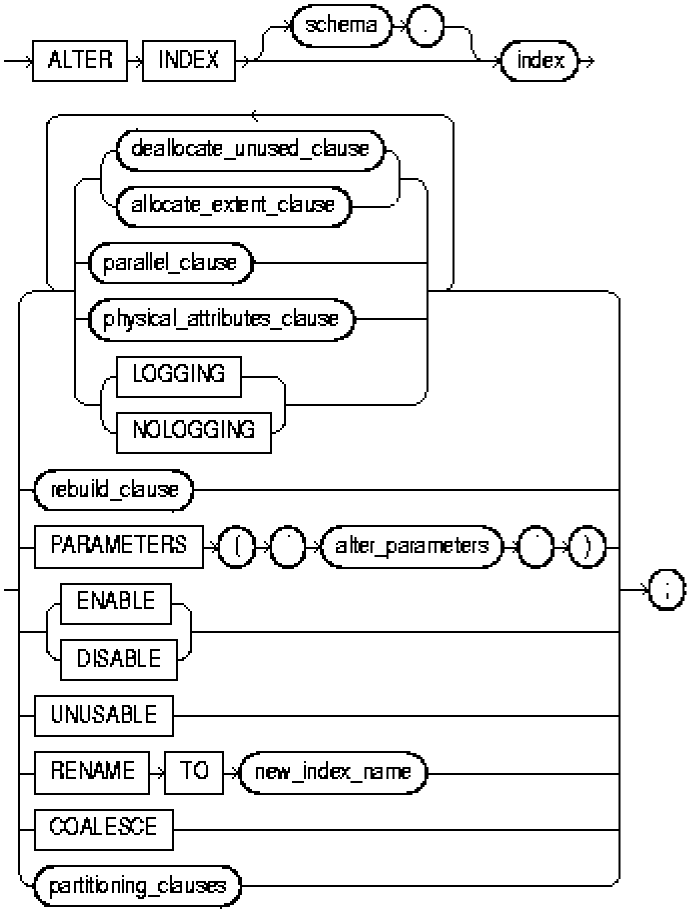
compound_condition ::=



SQL Commands Syntax

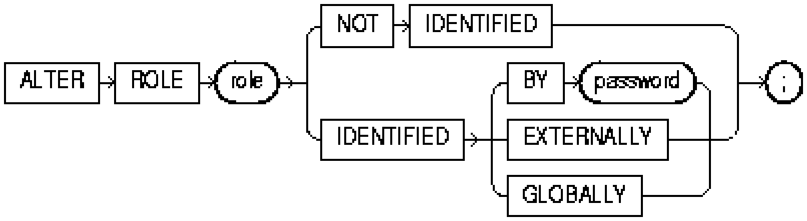
ALTER INDEX

Use this command to change or rebuild an existing index.



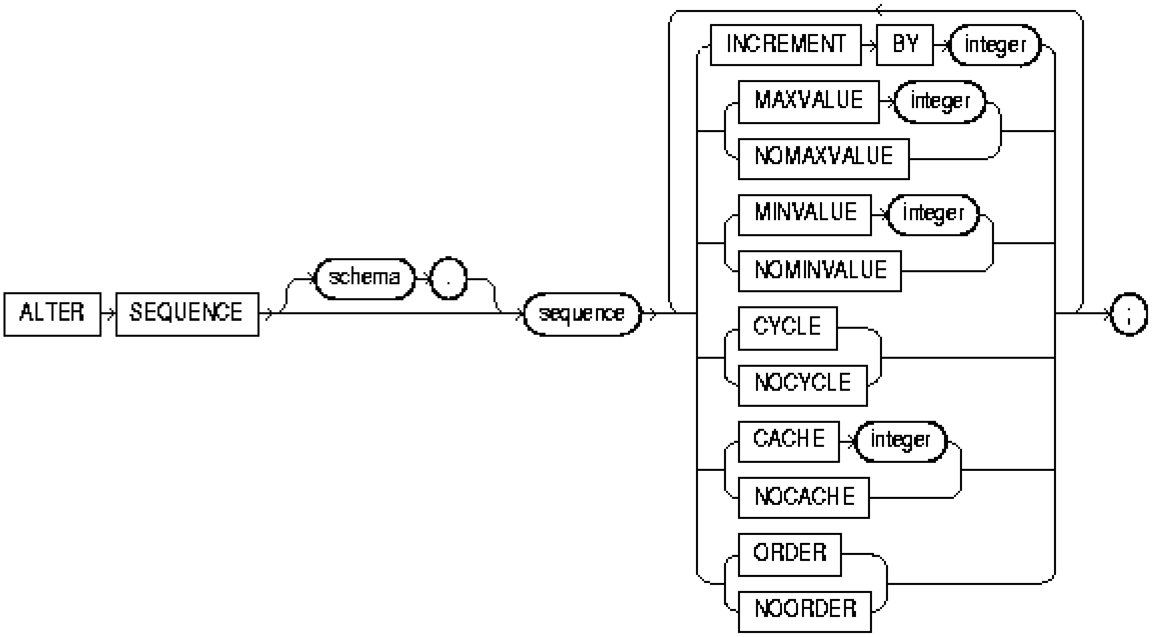
ALTER ROLE

Use this command to change the authorization needed to enable a role.



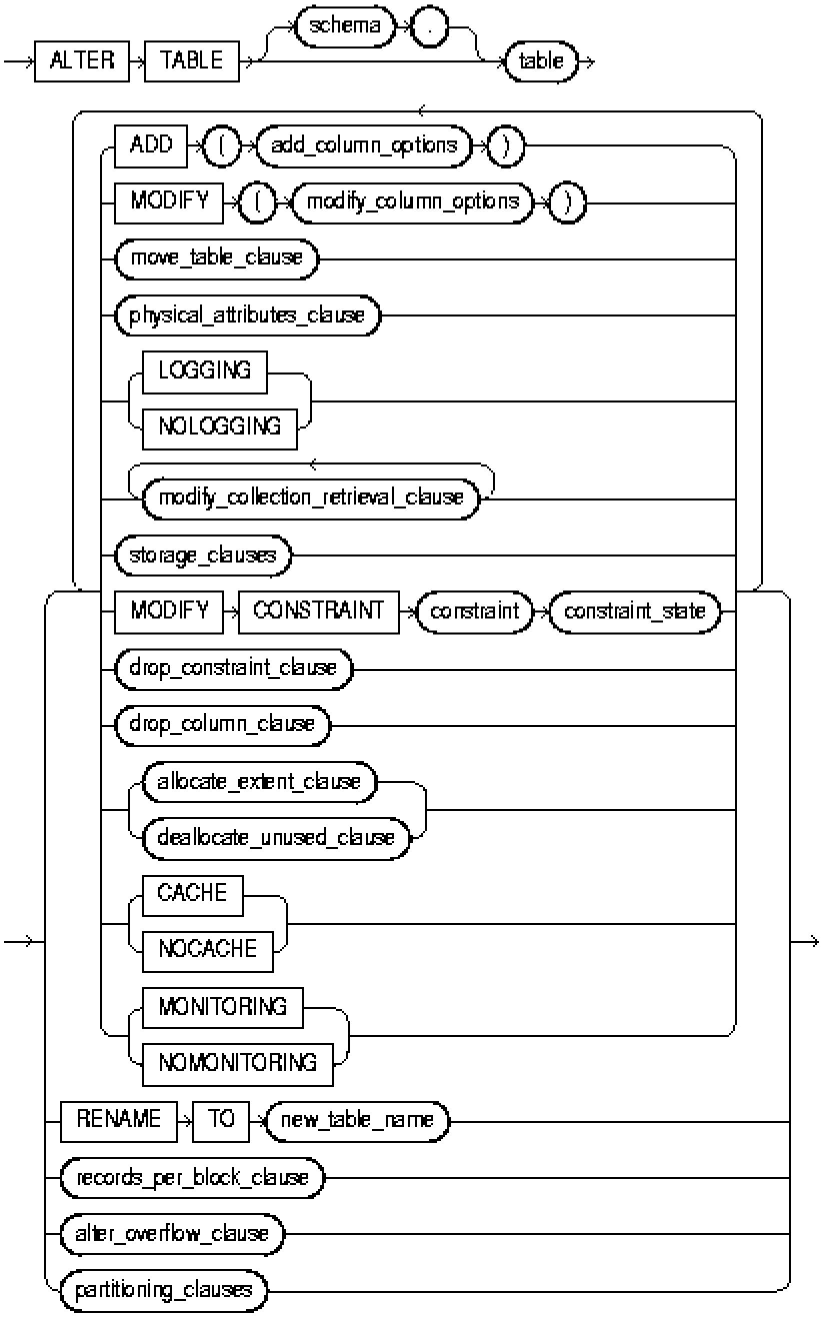
ALTER SEQUENCE

Use this command to change the increment, minimum and maximum values, cached numbers, and behavior of an existing sequence. This statement affects only future sequence numbers.

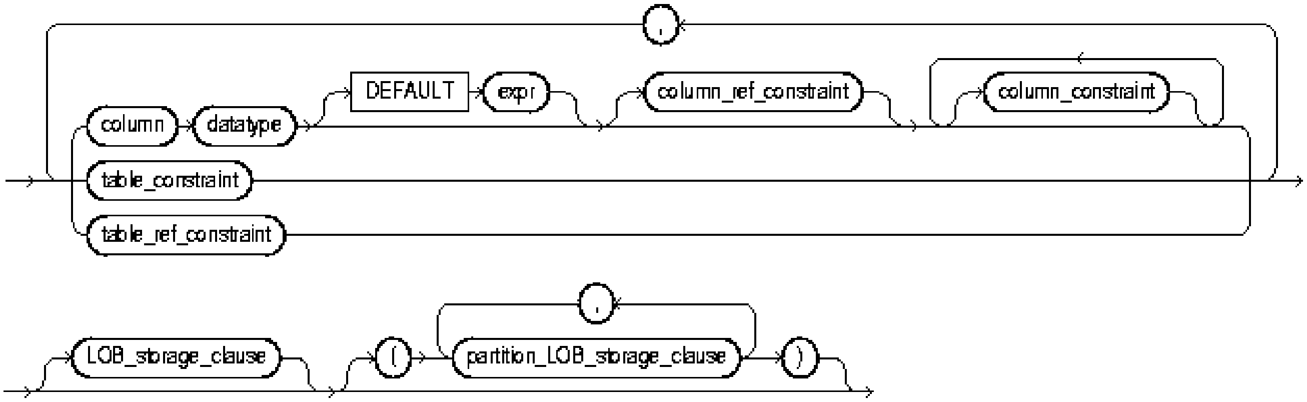


ALTER TABLE

Use this command to alter the definition of a nonpartitioned table, a partitioned table, a table partition, or a table subpartition.

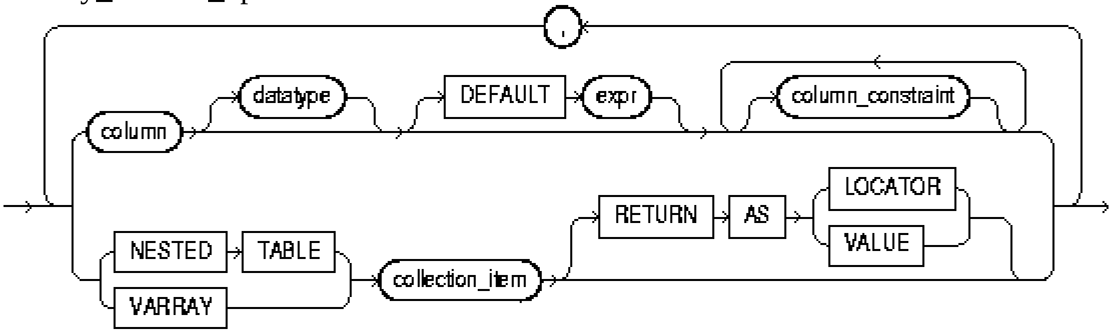


add_column_options ::=

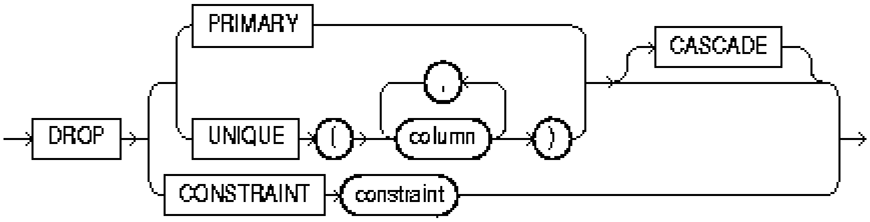


column_constraint, table_constraint, column_ref_constraint, table_ref_constraint, constraint_state ::=
See *constraint_clause*

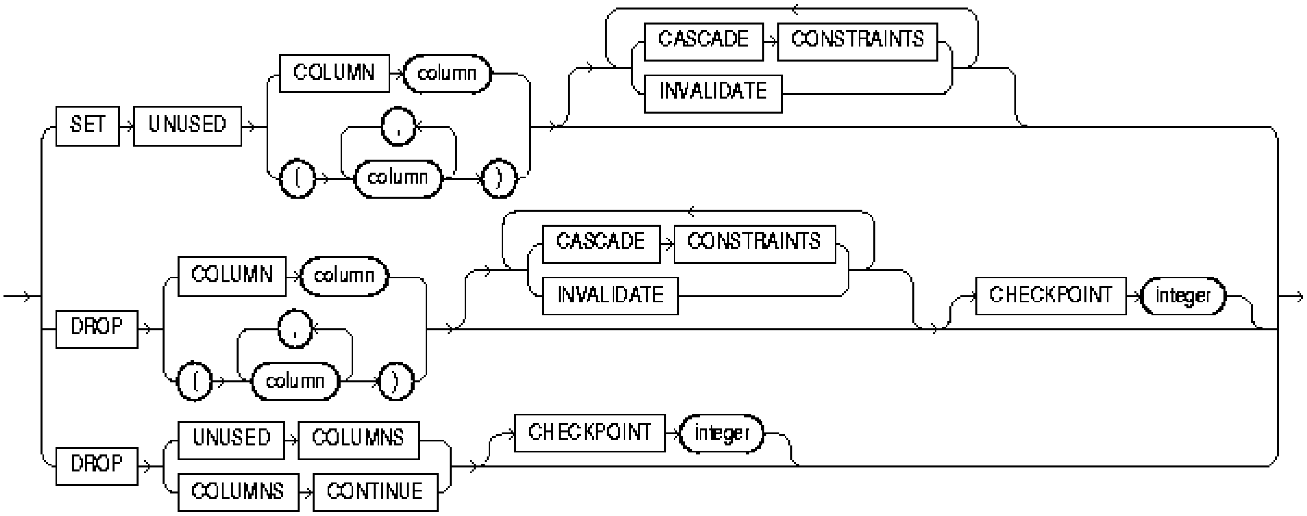
modify_column_options ::=



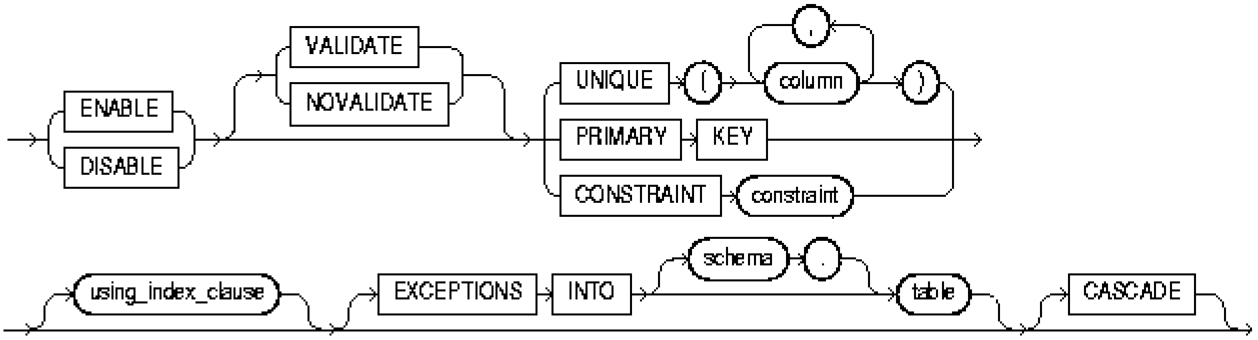
drop_constraint_clause ::=



drop_column_clause ::=

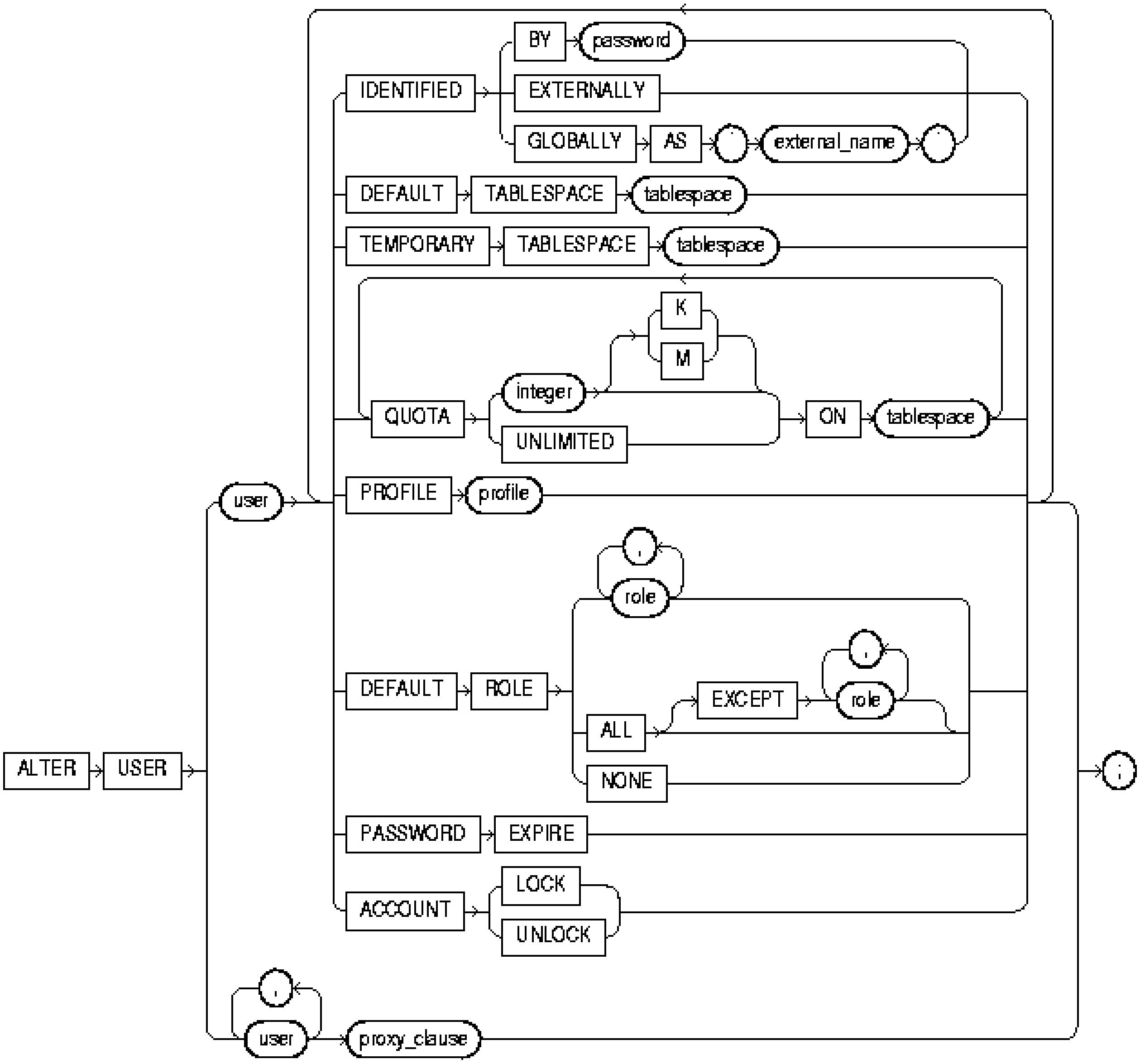


enable_disable_clause ::=



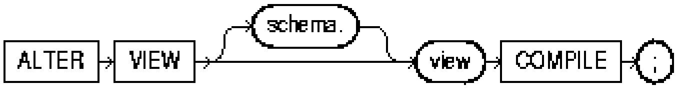
ALTER USER

Use this command to change the authentication or database resource characteristics of a database user.



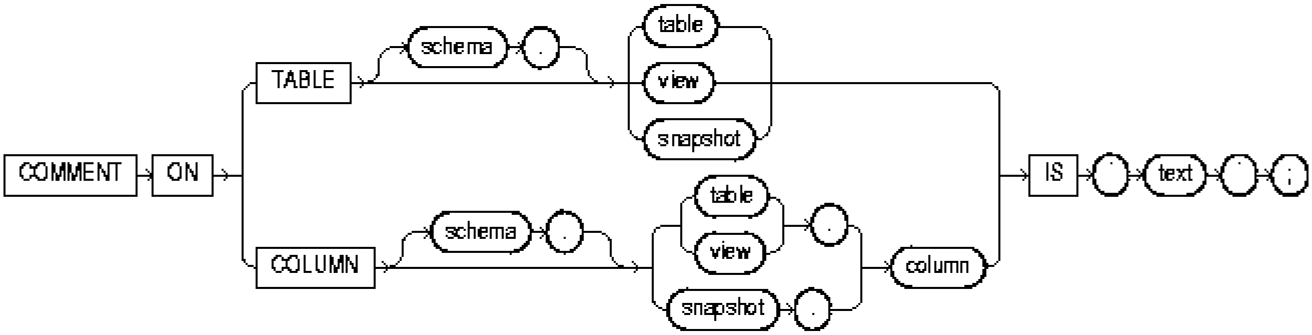
ALTER VIEW

Use this command to explicitly recompile a view that is invalid.



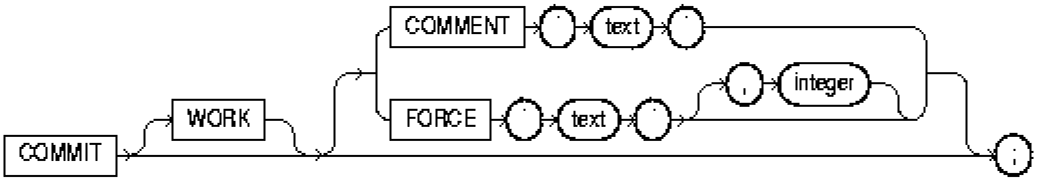
COMMENT

Use this command to add a comment about a table, view, materialized view, or column into the data dictionary.



COMMIT

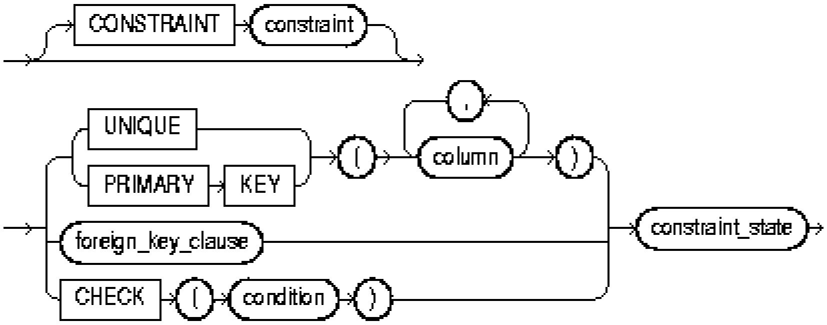
Use this command to end your current transaction and make permanent all changes performed in the transaction.



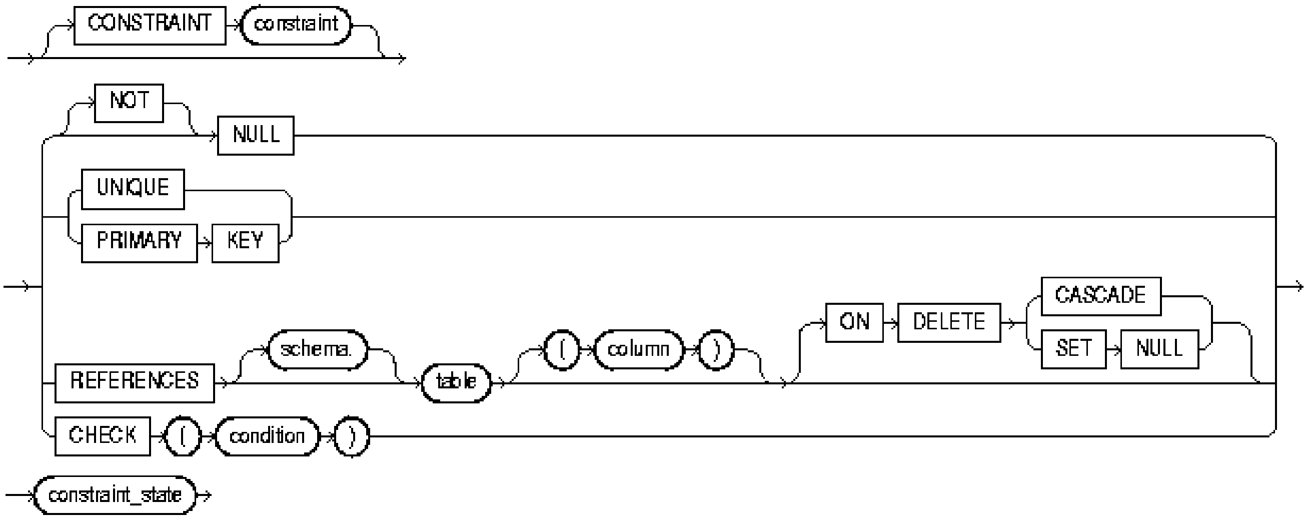
constraint_clause

To define an integrity constraint. An integrity constraint is a rule that restricts the values for one or more columns in a table or an index-organized table.

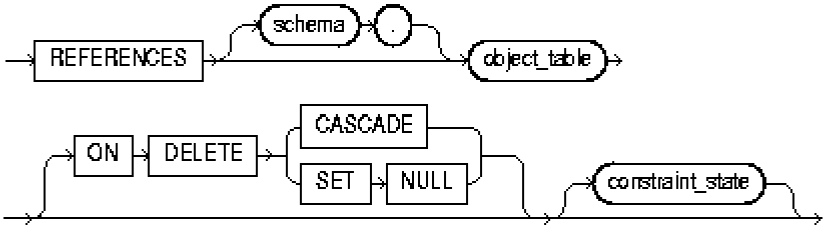
table_constraint::=



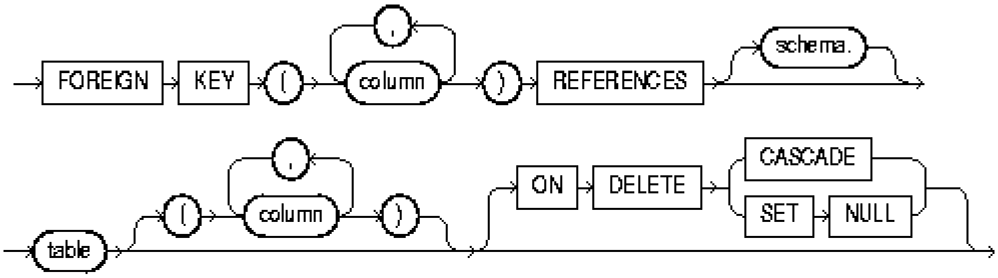
column_constraint ::=



references_clause ::=

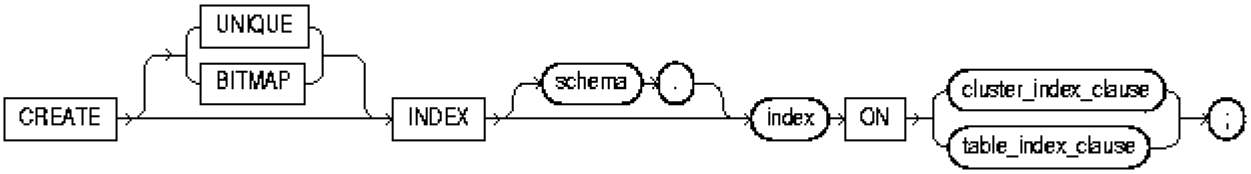


foreign_key_clause ::=

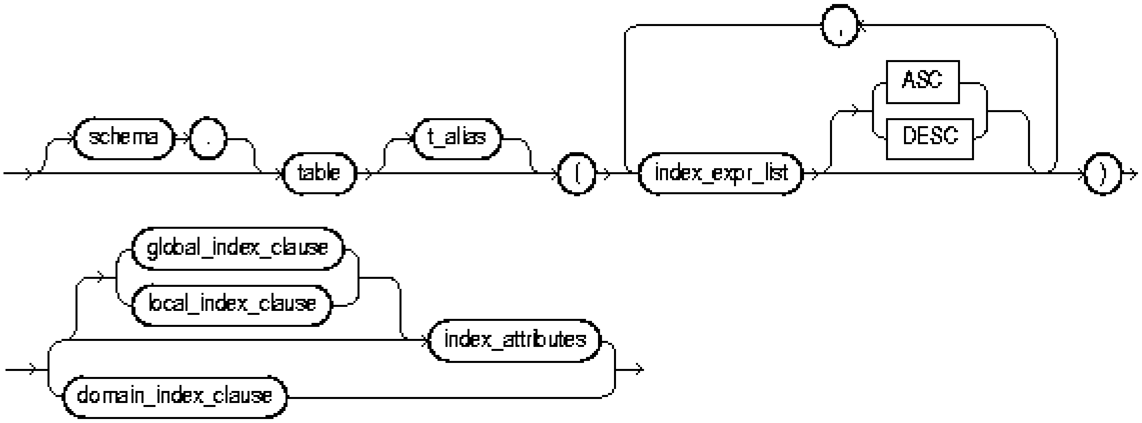


CREATE INDEX

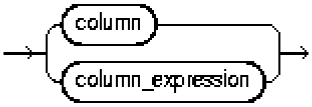
Use this command to create an index on one or more columns of a table, a partitioned table, an index-organized table, a cluster, one or more scalar typed object attributes of a table or a cluster, or a nested table storage table for indexing a nested table column.



table_index_clause ::=

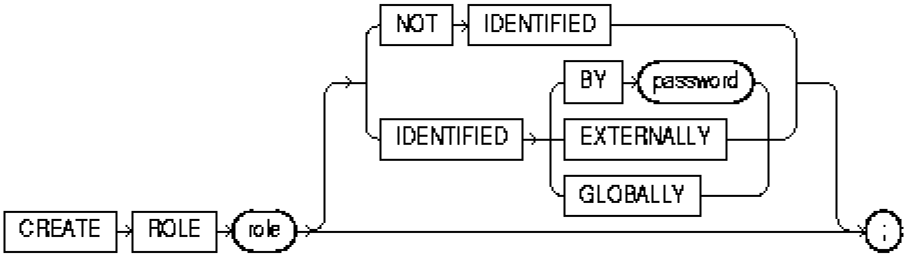


index_expr_list ::=



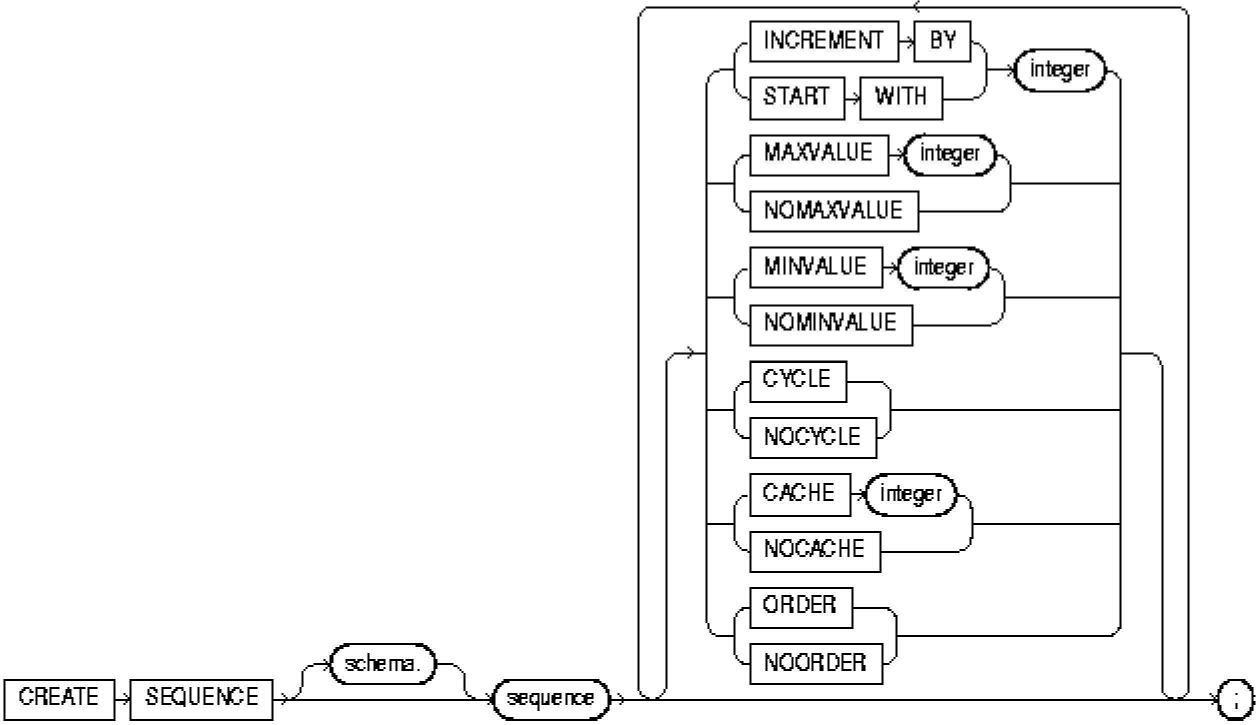
CREATE ROLE

Use this command to create a role, which is a set of privileges that can be granted to users or to other roles.



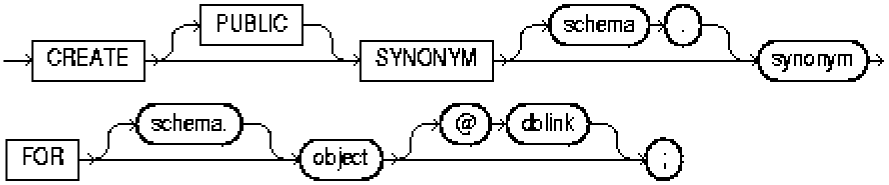
CREATE SEQUENCE

Use this command to create a sequence for generating sequential numbers.



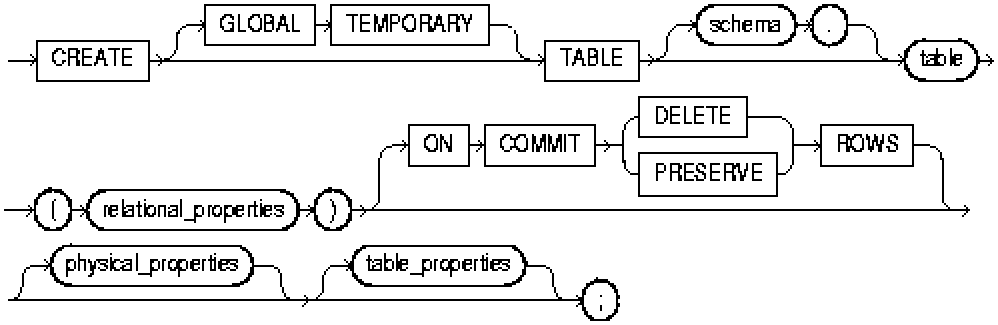
CREATE SYNONYM

Use this command to create a synonym for a schema object.

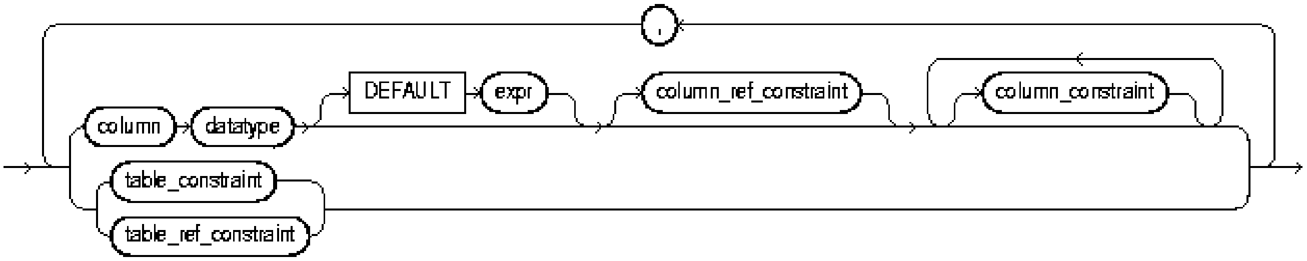


CREATE TABLE

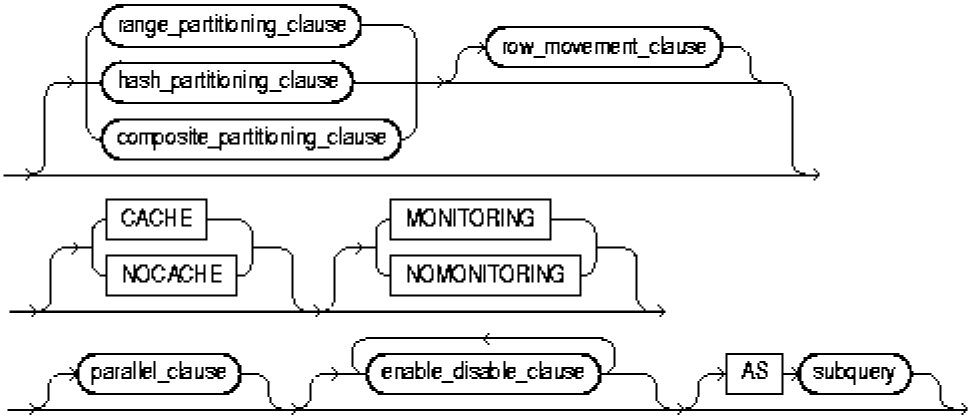
Use this command to create a relational table, the basic structure to hold user data, and to create an object table or a table that uses an object type for a column definition.



relational_properties ::=



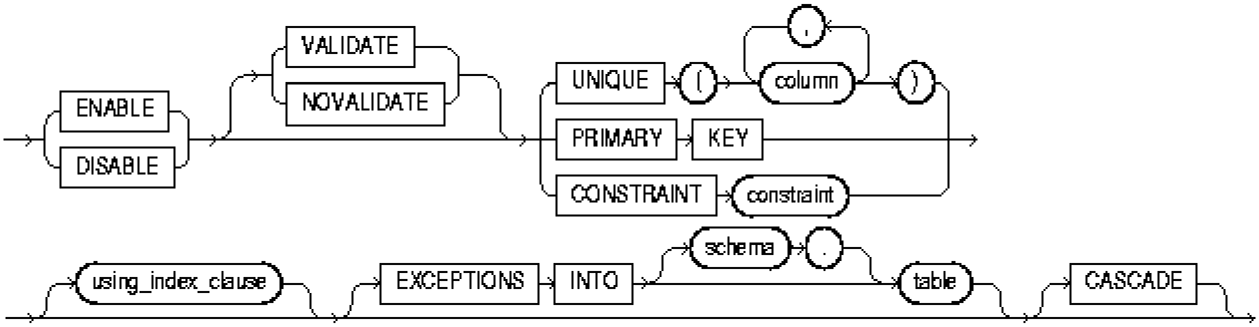
table_properties ::=



subquery
See *SELECT and subqueries*

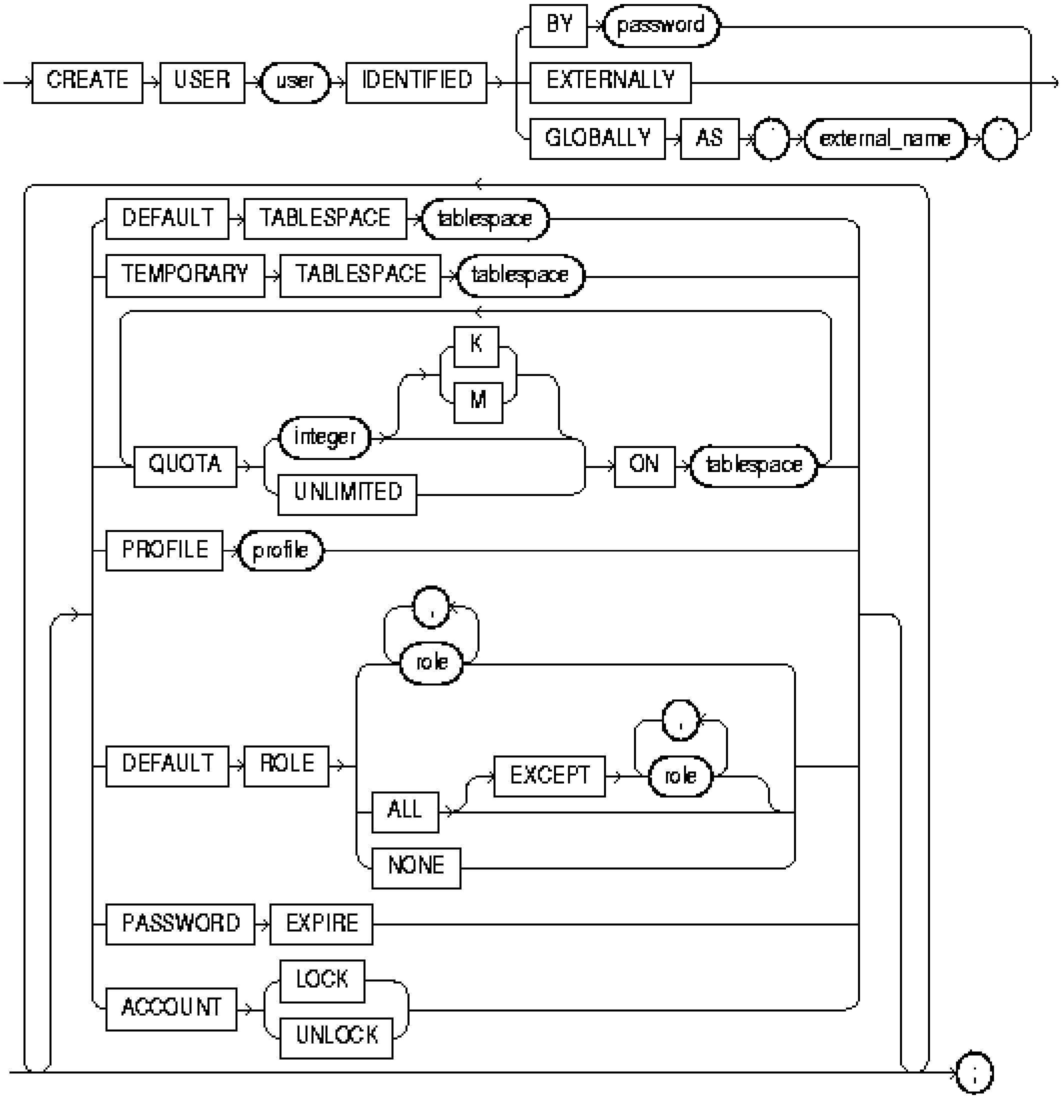
table_constraint, column_constraint, table_ref_constraint, column_ref_constraint, constraint_state
See *constraint_clause*

enable_disable_clause ::=



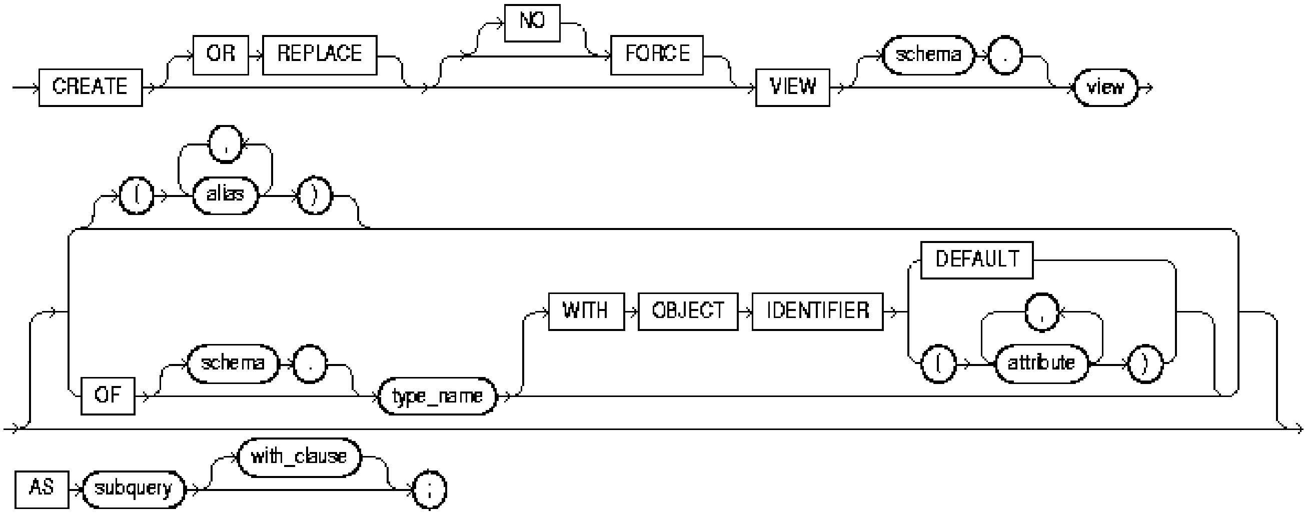
CREATE USER

Use this command to create and configure a database user, or an account through which you can log in to the database and establish the means by which Oracle server permits access by the user.



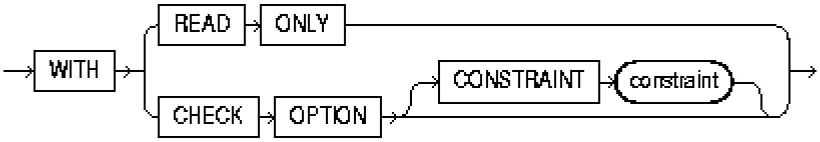
CREATE VIEW

Use this command to define a view, a logical table based on one or more tables or views.



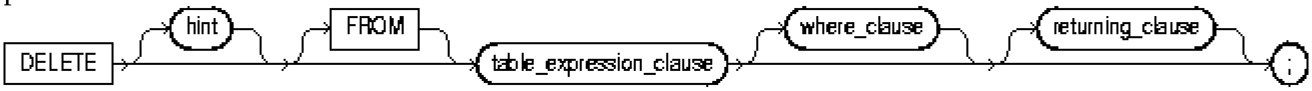
subquery
See *SELECT and subqueries*

with_clause ::=

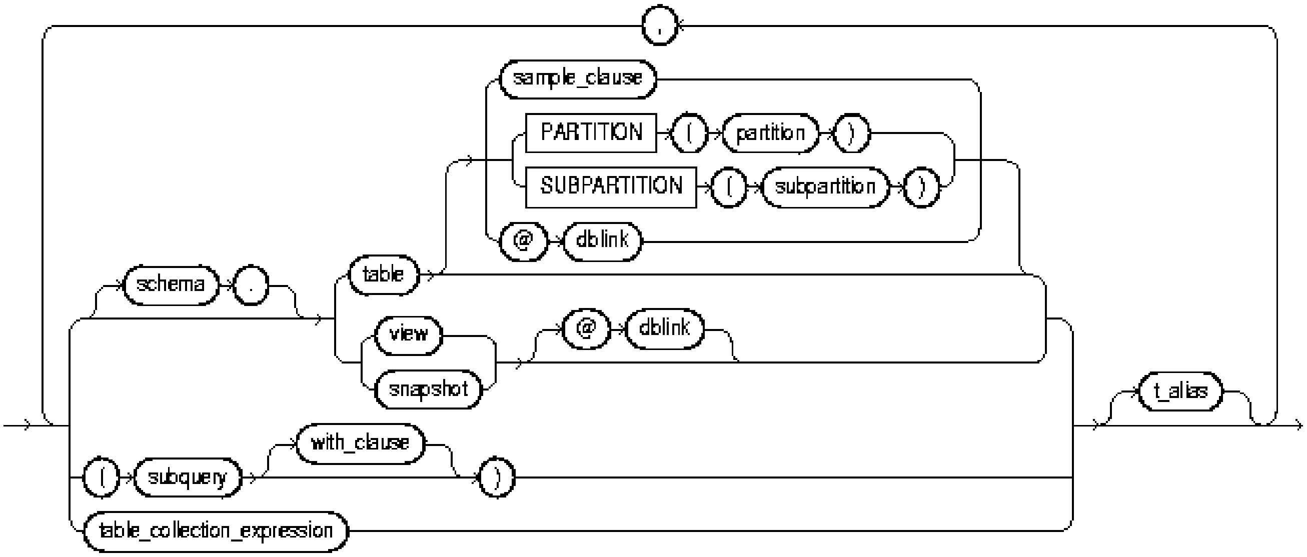


DELETE

Use this command to remove rows from a table, a partitioned table, a view’s base table, or a view’s partitioned base table.

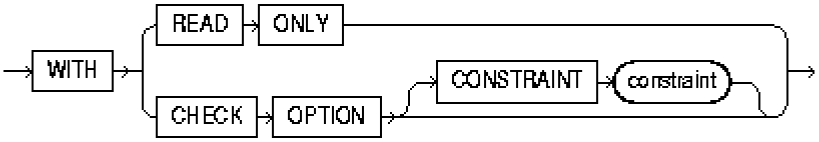


table_expression_clause ::=

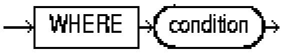


subquery
See *SELECT and subqueries*

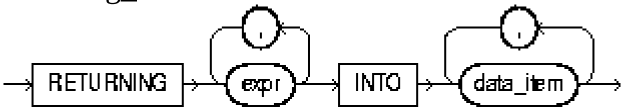
with_clause ::=



where_clause ::=

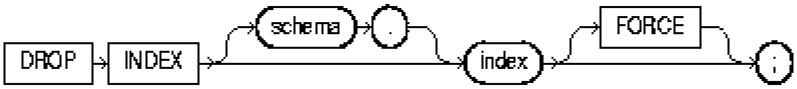


returning_clause ::=



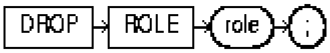
DROP INDEX

Use this command to remove an index or domain index from the database.



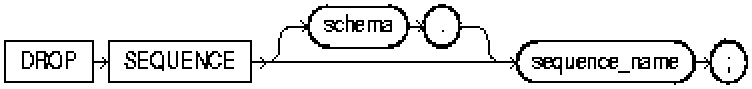
DROP ROLE

Use this command to remove a role from the database.



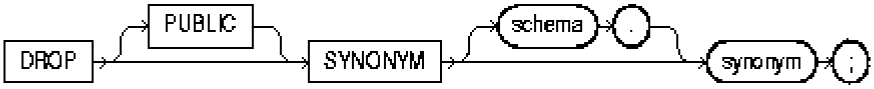
DROP SEQUENCE

Use this command to remove a sequence from the database.



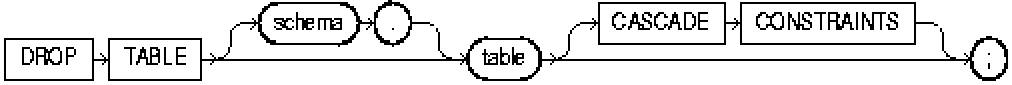
DROP SYNONYM

Use this command to remove a synonym from the database, or to change the definition of a synonym by dropping and re-creating it.



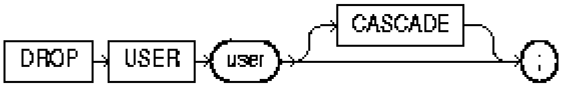
DROP TABLE

Use this command to remove a table or an object table and all its data from the database.



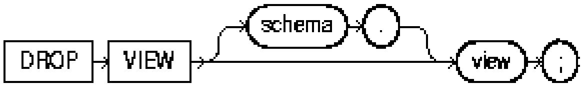
DROP USER

Use this command to o remove a user and the user’s schema from the database.



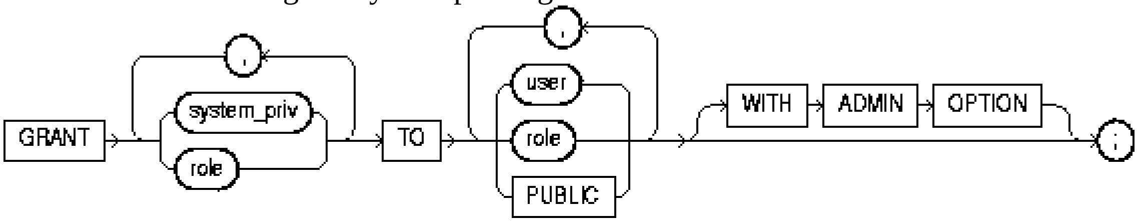
DROP VIEW

Use this command to remove a view or an object view from the database.



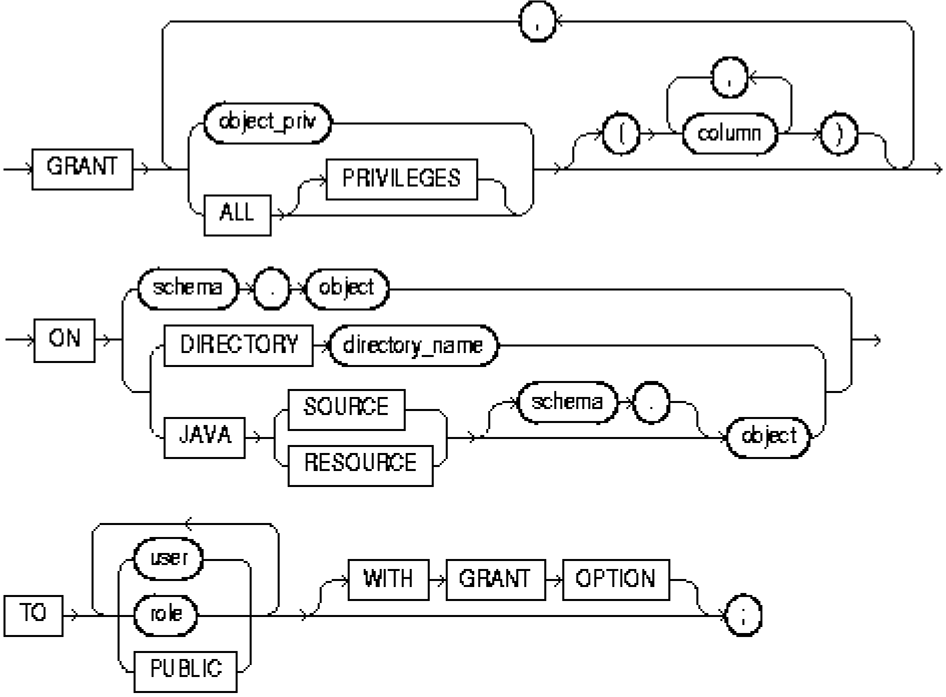
GRANT (*system_privileges_and_roles*)

Use this command to grant system privileges and roles to users and roles.



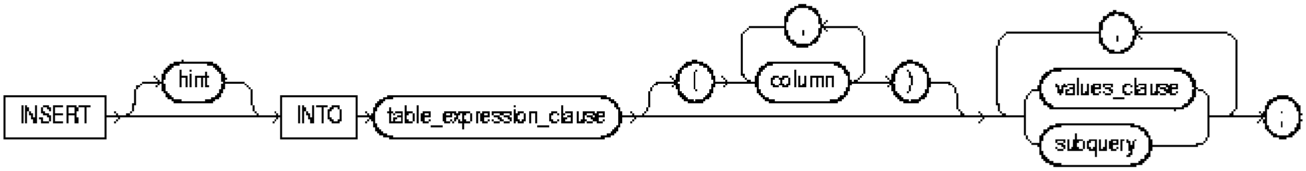
GRANT (*object_privileges*)

Use this command to grant privileges for a particular object to users, roles, and PUBLIC.

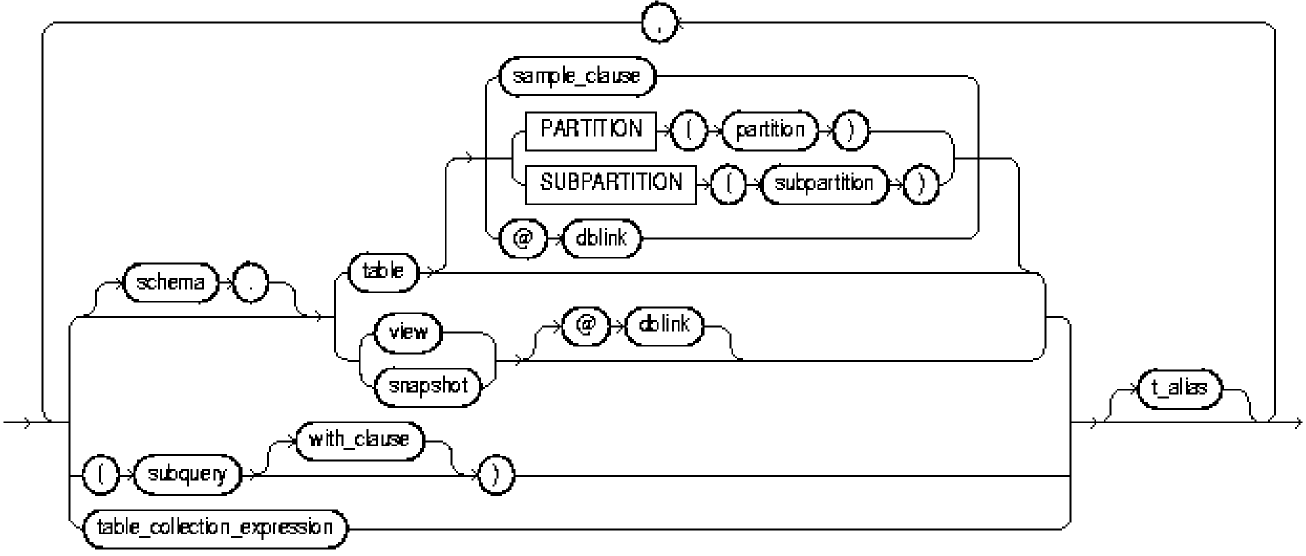


INSERT

Use this command to add rows to a table or a view's base table.

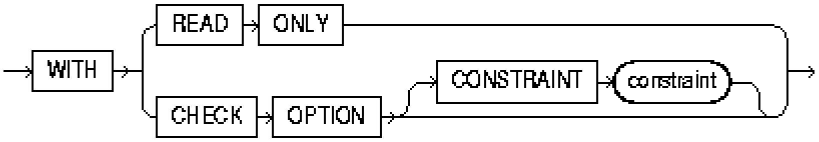


table_expression_clause ::=

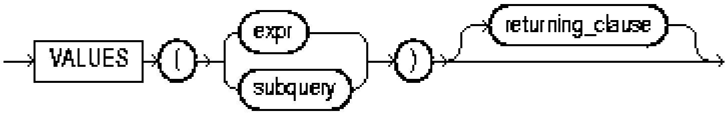


subquery
See *SELECT and subqueries*

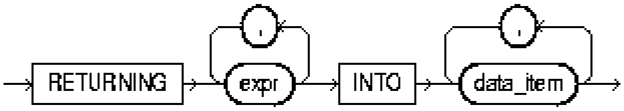
with_clause ::=



values_clause ::=

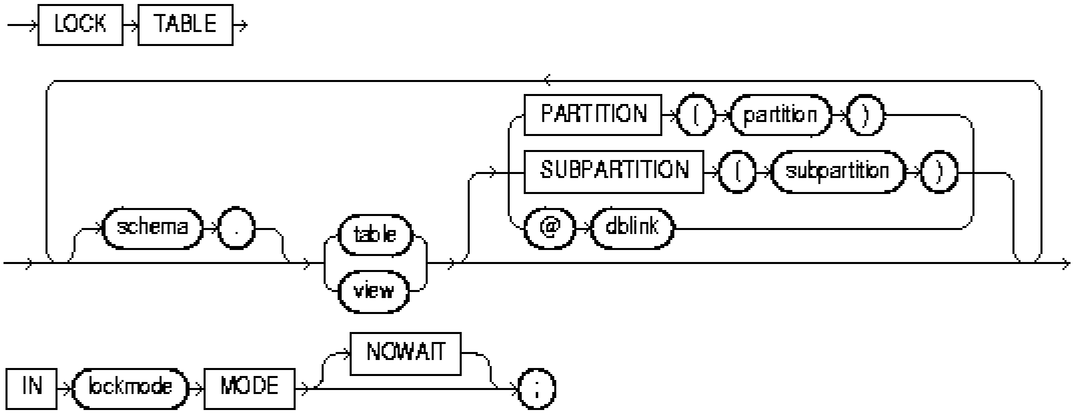


returning_clause ::=



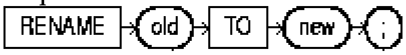
LOCK TABLE

Use this command to lock one or more tables (or table partitions or subpartitions) in a specified mode. This lock manually overrides automatic locking and permits or denies access to a table or view by other users for the duration of your operation.



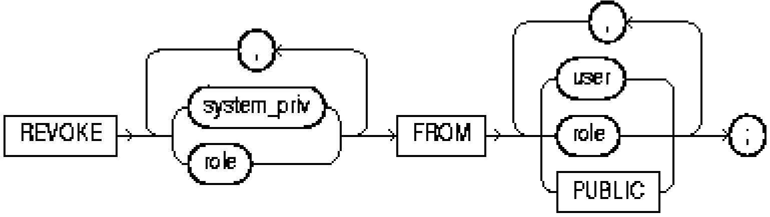
RENAME

Use this command to rename a table, view, sequence, or private synonym for a table, view, or sequence.



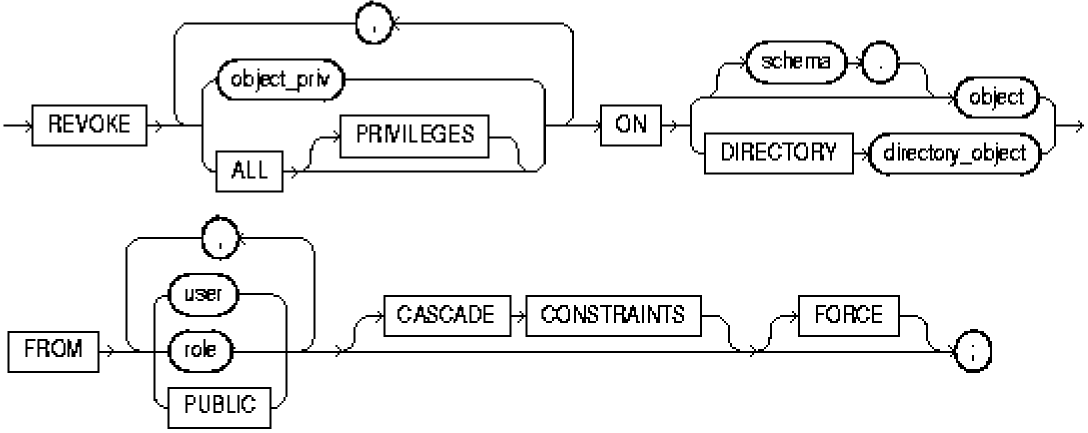
REVOKE (*system_privileges_and_roles*)

Use this command to revoke system privileges and roles from users and roles.



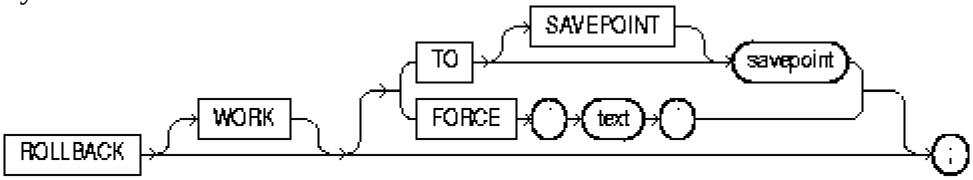
REVOKE (*object_privileges*)

Use this command to revoke object privileges for a particular object from users and roles.



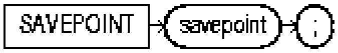
ROLLBACK

Use this command to undo work done in the current transaction, or to manually undo the work done by an in-doubt distributed transaction.



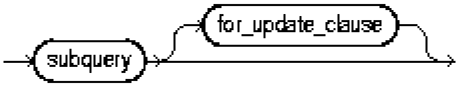
SAVEPOINT

Use this command to identify a point in the transaction to which you can later roll back.

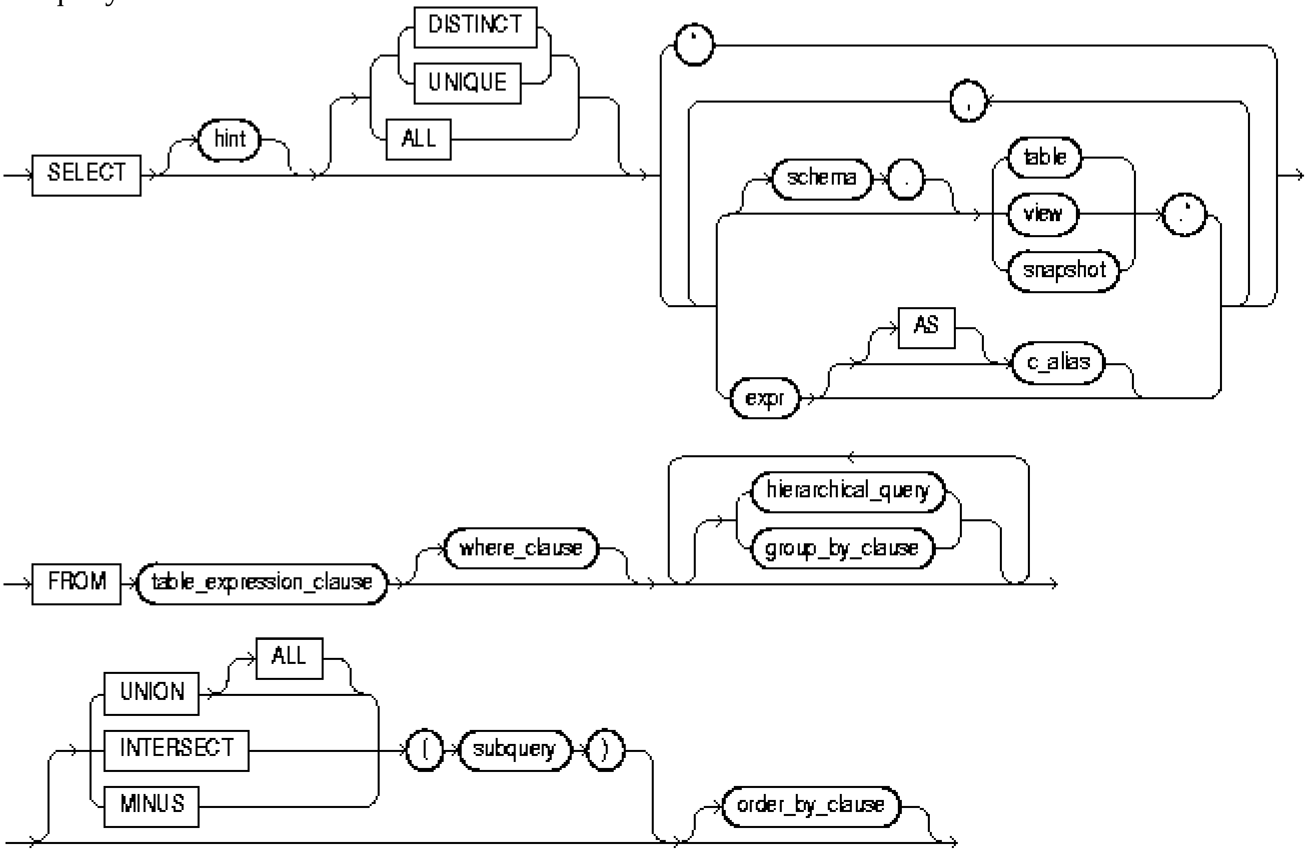


SELECT

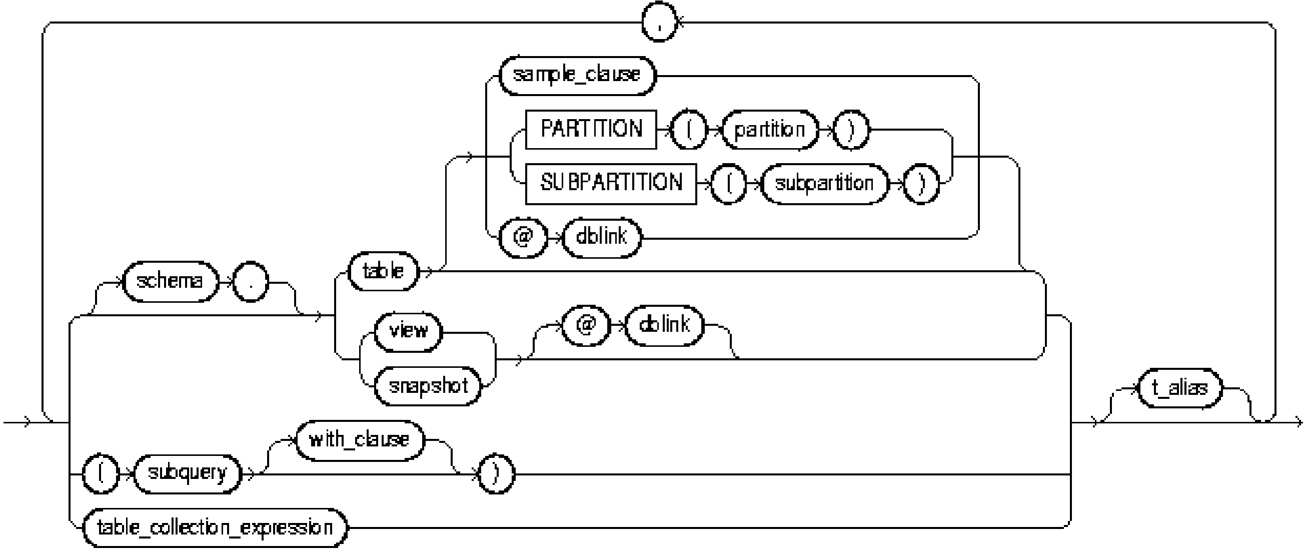
Use this command to retrieve data from one or more tables, object tables, views, object views, or materialized views.



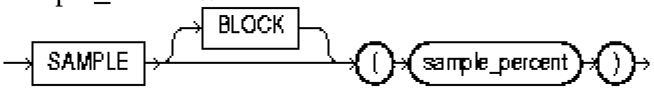
subquery ::=



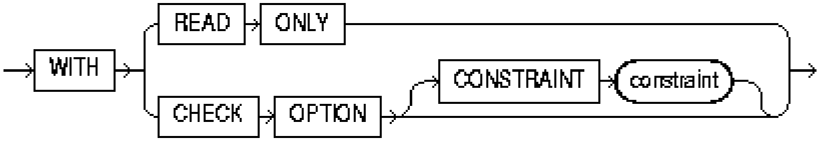
table_expression_clause ::=



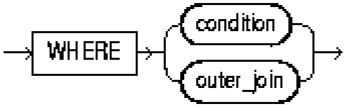
sample_clause ::=



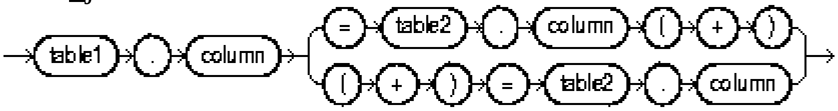
with_clause ::=



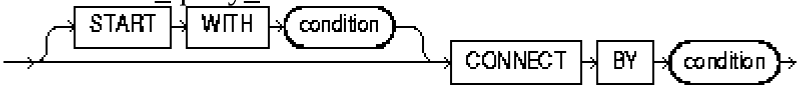
where_clause ::=



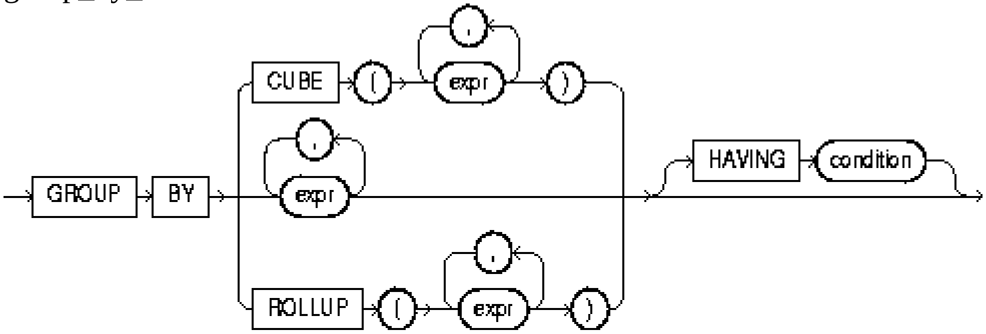
outer_join ::=



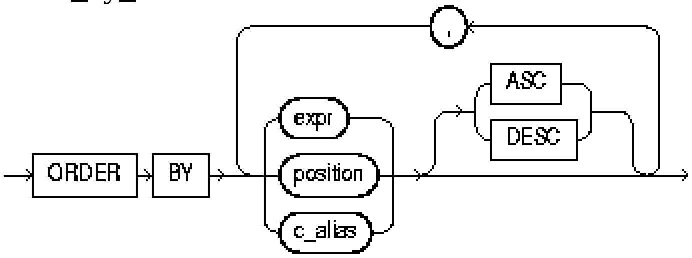
hierarchical_query_clause ::=



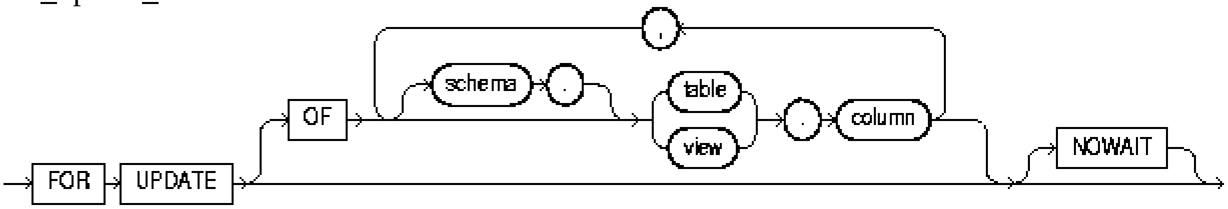
group_by_clause ::=



order_by_clause ::=

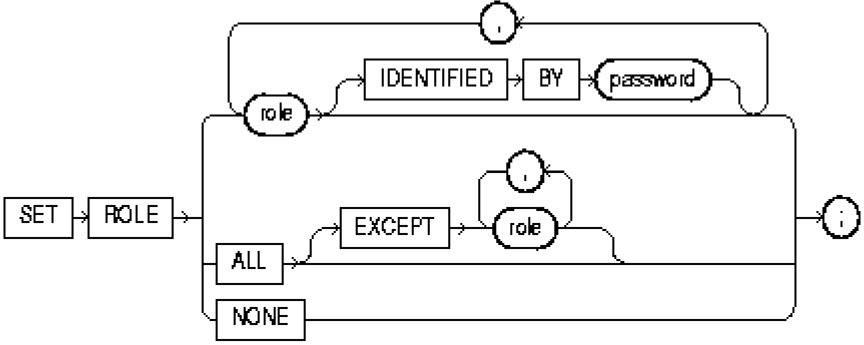


for_update_clause ::=



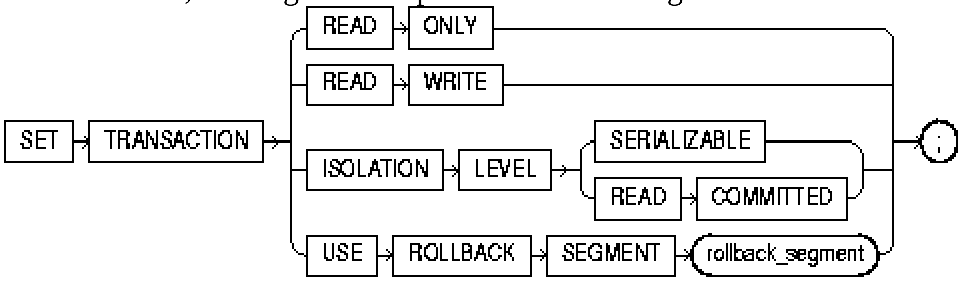
SET ROLE

Use this command to enable and disable roles for your current session.



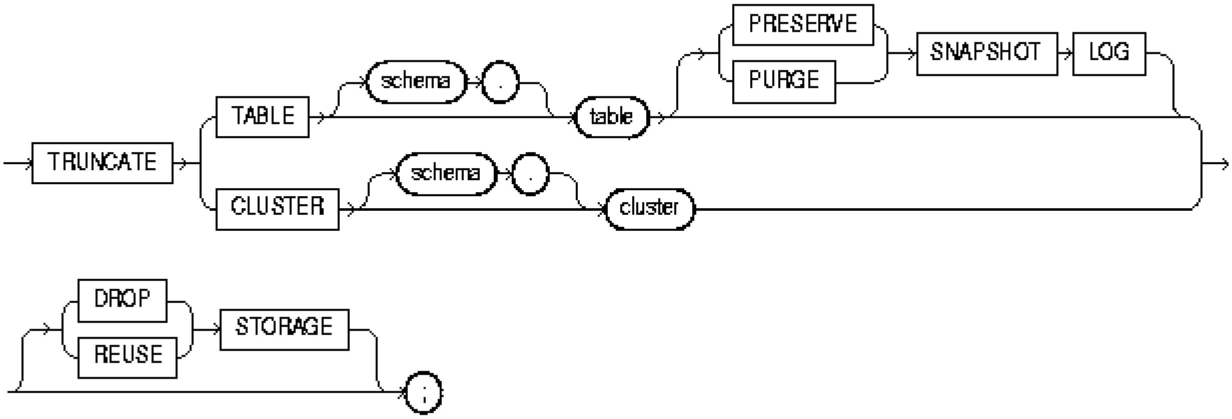
SET TRANSACTION

Use this command to establish the current transaction as a read-only or read-write, establish its isolation level, or assign it to a specified rollback segment.



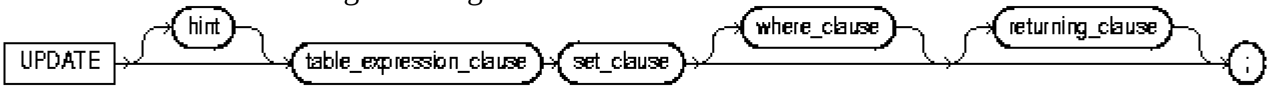
TRUNCATE

Use this command to remove all rows from a table or cluster and reset the storage parameters to the values when the table or cluster was created.

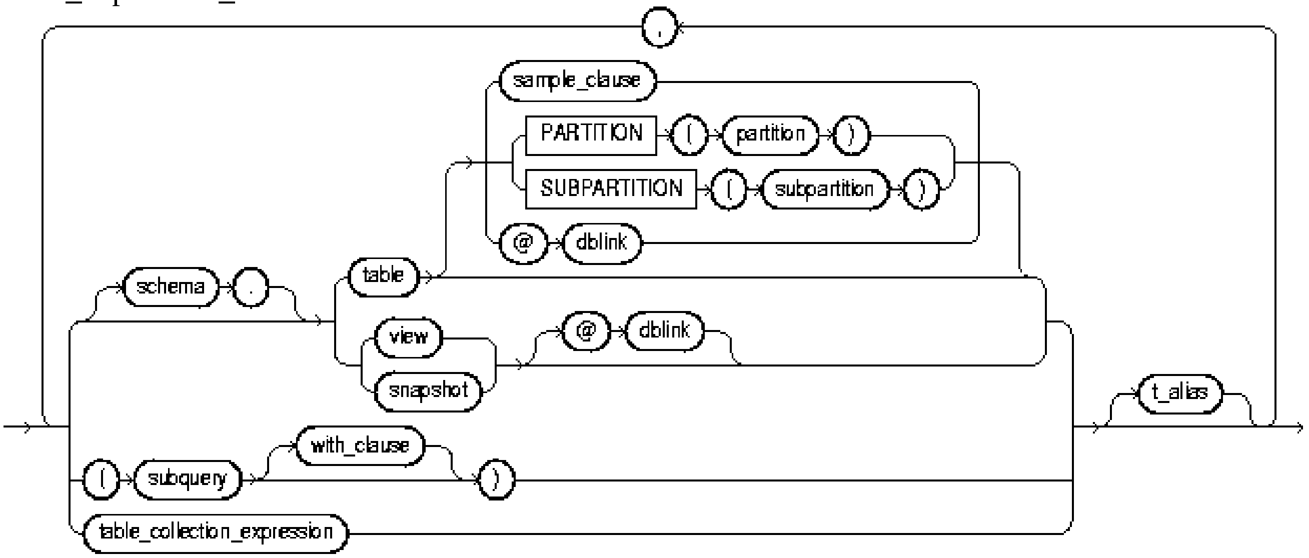


UPDATE

Use this command to change existing values in a table or view’s base table.

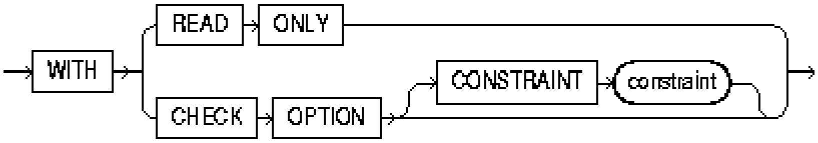


table_expression_clause ::=

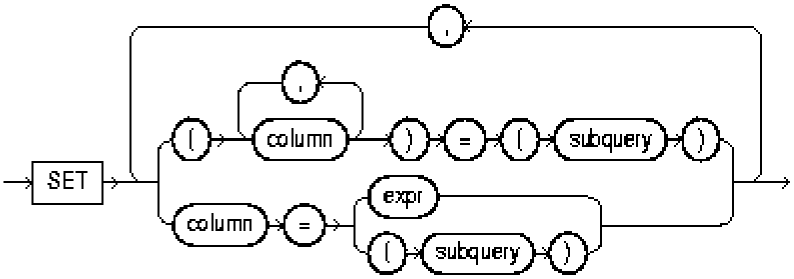


subquery
See *SELECT and subqueries*

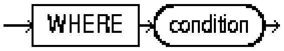
with_clause ::=



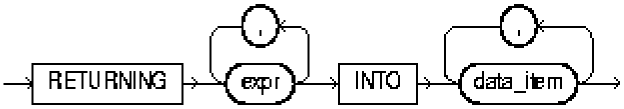
set_clause ::=



where_clause ::=



returning_clause ::=



Data Dictionary Views

Data dictionary view available to all users

View	Description
ALL_ALL_TABLES USER_ALL_TABLES	Description of all object and relational tables accessible to the user Description of all object and relational tables owned by the user
ALL_CATALOG USER_CATALOG	All tables, views, synonyms and sequences accessible to the user All tables, views, synonyms, and sequences owned by the user
ALL_COL_COMMENTS USER_COL_COMMENTS	Comments on columns of accessible tables and views Comments on columns of user tables and views
ALL_COL_PRIVS USER_COL_PRIVS	Grants on columns for which the user is the grantor, grantee, owner, or an enabled role or PUBLIC is the grantee Grants on columns for which the user is the owner, grantor, or grantee
ALL_COL_PRIVS_MADE USER_COL_PRIVS_MADE	Grants on columns for which the user is owner or grantor All grants on columns of objects owned by the user
ALL_COL_PRIVS_RECD USER_COL_PRIVS_RECD	Grants on columns for which the user, PUBLIC, or enabled role is the grantee Grants on columns for which the user is the grantee
ALL_CONSTRAINTS USER_CONSTRAINTS	Constraint definitions on accessible tables Constraint definitions on user's own tables
ALL_CONS_COLUMNS USER_CONS_COLUMNS	Information about accessible columns in constraint definitions Information about accessible columns in constraint definitions
ALL_INDEXES USER_INDEXES	Descriptions of indexes on tables accessible to the user Description of the user's own indexes
ALL_IND_COLUMNS USER_IND_COLUMNS	Columns comprising indexes on accessible tables Columns comprising user's indexes and indexes on user's tables
ALL_OBJECTS USER_OBJECTS	Objects accessible to the user Objects owned by the user
ALL_SEQUENCES USER_SEQUENCES	Description of sequences accessible to the user Description of the user's own sequences
ALL_TABLES USER_TABLES	Description of relational tables accessible to the user Description of the user's own relational tables
ALL_TAB_COLUMNS USER_TAB_COLUMNS	Columns of user's tables, views, and clusters Columns of user's tables, views, and clusters
ALL_TAB_PRIVS USER_TAB_PRIVS	Grants on objects for which the user is the grantor, grantee, or owner, or an enabled role or PUBLIC is the grantee Grants on objects for which the user is the owner, grantor, or grantee
ALL_TAB_PRIVS_MADE USER_TAB_PRIVS_MADE	User's grants and grants on user's objects All grants on objects owned by the user
ALL_TAB_PRIVS_RECD USER_TAB_PRIVS_RECD	Grants on objects for which the user, PUBLIC, or enabled role is the grantee Grants on objects for which the user is the grantee
ALL_TAB_COMMENTS USER_TAB_COMMENTS	Comments on tables and views accessible to the user Comments on the tables and views owned by the user
ALL_USERS USER_USERS	Information about all users of the database Information about the current user
ALL_VIEWS USER_VIEWS	Description of views accessible to the user Description of the user's own views
COLUMN_PRIVILEGES	Grants on columns for which the user is the grantor, grantee, owner, or an enabled role or PUBLIC is the grantee
DICTIONARY	Description of data dictionary tables and views
DICT_COLUMNS	Description of columns in data dictionary tables and views
DUAL	Dummy table owned by SYS

ROLE_ROLE_PRIVS	Roles that are granted to roles
ROLE_SYS_PRIVS	System privileges granted to roles
ROLE_TAB_PRIVS	Table privileges granted to roles
SESSION_PRIVS	Privileges which the user currently has set
SESSION_ROLES	Roles that the user currently has enabled
TABLE_PRIVILEGES	Grants on objects for which the user is the grantor, grantee, or owner, or an enabled role or PUBLIC is the grantee
CAT	Synonym for USER_CATALOG
CLU	Synonym for USER_CLUSTERS
COLS	Synonym for USER_TAB_COLUMNS
DICT	Synonym for Dictionary
IND	Synonym for USER_INDEXES
OBJ	Synonym for USER_OBJECTS
SEQ	Synonym for USER_SEQUENCES
SYN	Synonym for USER_SYNONYMS
TAB	Synonym for USER_TABLES