

The background of the cover is a vibrant, abstract collage of geometric shapes and colors, including teal, yellow, purple, and brown. A prominent red arc curves across the top. In the center, a stylized human figure with arms raised is set against a bright yellow sunburst. The Oracle logo is in the top left, and the title is in a white box on the right.

**ORACLE**<sup>®</sup>  
Education

---

**Oracle SQL Language Quick  
Reference**

---



---

# Oracle SQL Language Quick Reference

---

40020GC10  
Production 1.0  
April 1999  
M08684

**ORACLE®**

# Oracle SQL Quick Reference

Edition 1.0

Published by Oracle Corporation

Copyright© Oracle Corporation, 1999. All rights reserved.

Authors: Charlie Gray  
Kuljit Jassar  
Jeremy Tenison-Smith

This documentation contains proprietary information of Oracle Corporation. It is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. If this documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

#### Restricted Rights Legend

Use, duplication or disclosure by the Government is subject to restrictions for commercial computer software and shall be deemed to be Restricted Rights software under Federal law, as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

This material or any portion of it may not be copied in any form or by any means without the express prior written permission of Oracle Corporation. Any other copying is a violation of copyright law and may result in civil and/or criminal penalties.

If this documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Rights," as defined in FAR 52.227-14, Rights in Data-General, including Alternate III (June 1987).

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.

Oracle is a registered trademark and all references to Oracle products are trademarks or registered trademarks of Oracle Corporation.

All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

## Table of Contents

Conventions Used in This Reference	5
Syntax Diagrams and Notation	5
Reading Syntax Diagrams	6
Required Keywords and Parameters	6
Optional Keywords and Parameters	6
Syntax Loops	7
Multipart Diagrams	7
Database Objects	7
Functions	8
Single-Row Number Functions	8
Single-Row Character Functions	8
Date Functions	10
Date Truncation and Rounding	10
Format Models for ROUND and TRUNC	10
Conversion Functions	11
Format Models for TO_CHAR and TO_DATE	12
Date Format Prefixes and Suffixes	13
Date Format Case Control	13
Number Format Models for TO_CHAR	13
Miscellaneous Single-Row Functions	14
Object Reference Functions	14
Group Functions	15
Pseudocolumns	15
Datatypes	16
Object Naming Rules	18
Oracle Reserved Words	19
SQL Reserved Words	19
SQL Keywords	20
PL/SQL Reserved Words	21
SYSTEM PRIVILEGES	22
OBJECT PRIVILEGES	25
Operators	26
Arithmetic Operators	26
Logical Operators	26
Comparison Operators	26
Set Operators	27
Operator Precedence	27
Expressions and Conditions	28
Expressions	28
Conditions	29
SQL Commands Syntax	32
ALTER INDEX	32
ALTER ROLE	32
ALTER SEQUENCE	33

ALTER TABLE	34
ALTER USER	37
ALTER VIEW	37
COMMENT	38
COMMIT	38
<i>constraint_clause</i>	38
CREATE INDEX	39
CREATE ROLE	40
CREATE SEQUENCE	41
CREATE SYNONYM	41
CREATE TABLE	41
CREATE USER	43
CREATE VIEW	44
DELETE	44
DROP INDEX	45
DROP ROLE	45
DROP SEQUENCE	46
DROP SYNONYM	46
DROP TABLE	46
DROP USER	46
DROP VIEW	46
GRANT ( <i>system_privileges_and_roles</i> )	46
GRANT ( <i>object_privileges</i> )	47
INSERT	47
LOCK TABLE	48
RENAME	48
REVOKE ( <i>system_privileges_and_roles</i> )	49
REVOKE ( <i>object_privileges</i> )	49
ROLLBACK	49
SAVEPOINT	49
SELECT	49
SET ROLE	52
SET TRANSACTION	52
TRUNCATE	52
UPDATE	52
Data Dictionary Views	54

## Conventions Used in This Reference

### Text

The text in this reference adheres to the following conventions:

<b>UPPERCASE</b>	Uppercase text calls attention to SQL keywords, filenames, and initialization parameters.
<i>italics</i>	Italicized text calls attention to parameters of SQL statements.
<b>boldface</b>	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
<i>c</i>	A single character from your database character set	T S
<i>char</i>	An expression of datatype CHAR or VARCHAR2 or a character literal in single quotes	ename 'Smith'
<i>condition</i>	An expression that evaluates to TRUE or FALSE	ename > 'A'
<i>date</i>	A date constant or an expression of datatype DATE	TO_DATE( '01-Jan-1994', 'DD-Mon-YYYY')
<i>db_name</i>	The name of a nondefault database in an embedded SQL program	sales_db
<i>db_string</i>	The database identification string for a Net8 database connection	
<i>expr</i>	An expression of any datatype as defined in the syntax description of <i>expr</i>	Sal + 100
<i>integer</i>	A number constant or expression of datatype NUMBER	72
<i>number</i>	A number constant or expression of datatype NUMBER	AVG(sal)
<i>raw</i>	An expression of datatype RAW	
<i>subquery</i>	A SELECT statement that will be used in another SQL statement	SELECT ename FROM emp
<i>table</i>	The name of an object of the type specified by the parameter	emp
<i>'text'</i>	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.

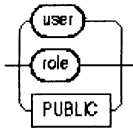
### Required Keywords and Parameters

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:



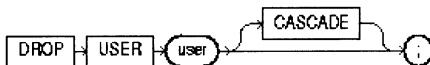
If there is a library named `HQ_LIB`, then, according to the diagram, the following statement is valid:  
`RENAME emp TO employees;`

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:

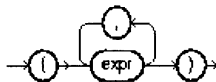


According to the diagram, all of the following statements are valid:

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

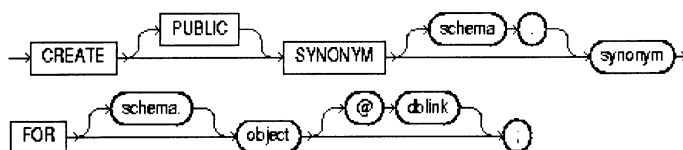
## Syntax Loops

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.

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

### Single-Row Character Functions

Single-row character functions that return character values

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

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

Single-row character functions that return numeric values

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

## Date Functions

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

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

## Date Truncation and Rounding

Function	Value Returned
ROUND( <i>d,fmt</i> )	<i>d</i> rounded as specified by rounding unit <i>fmt</i>
TRUNC( <i>d,fmt</i> )	<i>d</i> truncated as specified by <i>fmt</i>

## 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	First day of year (rounds up on July 1)
YYY or YY or Y	
Y,YYY or YEAR or SYEAR	
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.

<b>Function</b>	<b>Value Returned</b>
CHARTOROWID( <i>char</i> )	<i>char</i> , converted from a CHAR value to a ROWID value
CONVERT( <i>char</i> , <i>dest_char_set</i> [, <i>source_char_set</i> ])	<i>char</i> , converted from source_char_set representation to dest_char_set representation
HEXTORAW( <i>char</i> )	<i>char</i> , converted from a hexadecimal number to a binary RAW value
RAWTOTHEx( <i>raw</i> )	<i>raw</i> , converted from a binary value of datatype RAW to a hexadecimal number of datatype CHAR
ROWIDTOCHAR ( <i>rowid</i> )	<i>rowid</i> , converted from a ROWID value to a CHAR value.
TO_CHAR ( <i>expr</i> [, <i>fmt</i> [, <i>nls_num_fmt</i> ']])	<i>expr</i> , converted from a NUMBER or DATE value to a CHAR value in the format specified by <i>fmt</i> . If you omit <i>fmt</i> , ORACLE converts DATE values to default date format and NUMBER values to CHAR values exactly wide enough to hold all significant digits. <i>nls_num_fmt</i> sets the international number-format specifications. In Trusted Oracle, converts values of type MLS or MLS_LABEL to type VARCHAR2.
TO_DATE( <i>char</i> [, <i>fmt</i> [, <i>nls_lang</i> ']])	<i>char</i> , converted from a CHAR value in the format <i>fmt</i> to a DATE value. If you omit <i>fmt</i> , <i>char</i> must be in default date format. <i>nls_lang</i> specifies the language used for day and month
TO_MULTI_BYTE ( <i>char</i> )	<i>char</i> , with all single-byte characters that have a multibyte equivalent converted to their multibyte form
TO_NUMBER( <i>char</i> [, <i>fmt</i> [, <i>nls_lang</i> ']])	<i>char</i> , which is a character value containing a number in the format specified by the optional <i>fmt</i> , converted to a NUMBER value. <i>nls_lang</i> specifies the language used for numeric characters and currency symbols.
TO_SINGLE_BYTE ( <i>char</i> )	<i>char</i> , 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=1)	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-letter 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 9s 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 <sup><i>n</i></sup> , where <i>n</i> is the number of 9s after the <i>V</i>

EEEE	9.999EEE	Returns value in scientific notation. <i>fmt</i> must contain exactly four
	E	<i>Es</i>
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( <i>expr</i> , <i>search1</i> , <i>return1</i> , [ <i>search2</i> , <i>return2</i> ,]... [ <i>default</i> ])	If <i>expr</i> equals any search, returns the following return; if not, returns default
DUMP( <i>expr</i> [, <i>display_format</i> [, <i>start_position</i> [, <i>length</i> ]]])	<i>expr</i> in Oracle internal format
GREATEST( <i>expr</i> [, <i>expr</i> ]...)	<i>expr</i> with the greatest value
LEAST( <i>expr</i> [, <i>expr</i> ]...)	<i>expr</i> with the least value
NVL( <i>expr1</i> , <i>expr2</i> )	<i>expr2</i> , if <i>expr1</i> is null; otherwise returns <i>expr1</i>
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( <i>expr</i> )	The number of bytes in Oracle's internal representation of <i>expr</i> .

## Object Reference Functions

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

Function	Value Returned
DEREF( <i>e</i> )	The object reference of argument <i>e</i>
MAKE_REF( <i>tableview</i> , <i>key</i> [, <i>key</i> ]...)	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( <i>correlation_variable</i> )	REF takes as its argument a correlation variable (table alias) associated with a row of an object table or an object view.
REFTOHEX( <i>r</i> )	Converts argument <i>r</i> to a character value containing its hexadecimal equivalent
VALUE( <i>correlation_variable</i> )	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] <i>n</i> )	Average value of <i>n</i> , ignoring nulls
COUNT([ALL] *)	Number of rows returned by a query or subquery
COUNT([DISTINCT  ALL] <i>expr</i> )	Number of rows where <i>expr</i> is not null
MAX([DISTINCT ALL] <i>expr</i> )	Maximum value of <i>expr</i>
MIN([DISTINCT ALL] <i>expr</i> )	Minimum value of <i>expr</i>
STTDEV([DISTINCT ALL] <i>n</i> )	Standard deviation of <i>n</i> , ignoring null values
SUM([DISTINCT ALL] <i>n</i> )	Sum of values of <i>n</i>
VARIANCE ([DISTINCT  ALL] <i>n</i> )	Variance of <i>n</i> , 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
<i>sequence</i> .CURRVAL	Current value of sequence for the current session (sequence.NEXTVAL must be referenced first.)
<i>sequence</i> .NEXTVAL	Next value of sequence for the current session
<i>table</i> .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.
[ <i>table</i> ].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:

<b>Datatype</b>	<b>Explanation</b>
CHAR( <i>size</i> )	Fixed-length character data of length <i>size</i> bytes. Maximum <i>size</i> is 2000. Default and minimum <i>size</i> is 1 byte.
VARCHAR2( <i>size</i> )	Variable-length character string having maximum length <i>size</i> bytes. Maximum <i>size</i> is 4000, and minimum is 1. You must specify <i>size</i> .
NCHAR( <i>size</i> )	Fixed-length character data of length <i>size</i> characters or bytes, depending on the choice of national character set. Maximum <i>size</i> is determined by the number of bytes required to store each character, with an upper limit of 2000 bytes. Default and minimum <i>size</i> is 1 character or 1 byte, depending on the character set.
NVARCHAR2( <i>size</i> )	Variable-length character string having maximum length <i>size</i> characters or bytes, depending on the choice of national character set. Maximum <i>size</i> is determined by the number of bytes required to store each character, with an upper limit of 4000 bytes. You must specify a <i>size</i> .
NUMBER( <i>precision</i> [, <i>scale</i> ])	Number having <i>precision</i> and <i>scale</i> . 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( <i>size</i> )	Raw binary data of length <i>size</i> bytes. Maximum <i>size</i> is 2000 bytes. You must specify a <i>size</i> .
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( <i>size</i> )	Hexadecimal string representing the logical address of a row of an index-organized table. The optional <i>size</i> 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	EXISTS*	MODIFY	SET*
ADD	FILE	NOAUDIT	SHARE
ALL*	FLOAT*	NOCOMPRESS	SIZE
ALTER	FLOAT*	NOT*	SMALLINT*
AND*	FOR*	NOWAIT	START
ANY*	FROM*	NULL*	SUCCESSFUL
AS*	GRANT*	NUMBER	SYNONYM
ASC*	GROUP*	OF*	SYSDATE
AUDIT	HAVING*	OFFLINE	TABLE*
BETWEEN*	IDENTIFIED	ON*	THEN
BY*	IMMEDIATE	ONLINE	TO*
CHAR*	IN*	OPTION*	TRIGGER
CHECK*	INCREMENT	OR*	UID
CLUSTER	INDEX	ORDER*	UNION*
COLUMN	INITIAL	PCTFREE	UNIQUE*
COMMENT	INSERT*	PRIOR	UPDATE*
COMPRESS	INTEGER*	PRIVILEGES*	USER*
CONNECT	INTERSECT	PUBLIC*	VALIDATE
CREATE*	INTO*	RAW	VALUES*
CURRENT*	IS*	RENAME	VARCHAR
DECIMAL	LEVEL	RESOURCE	VARCHAR2
DEFAULT*	LIKE*	REVOKE	VIEW*
DELETE*	LOCK	ROW	WHENEVER
DESC*	LONG	ROWID	WHERE*
DISTINCT*	MAXEXTENTS	ROWNUM	WITH*
DROP	MINUS	ROWS	
ELSE	MLSLABEL	SELECT*	
EXCLUSIVE	MODE	SESSION	

## SQL Keywords

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

## 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*	DISTINCT*	LIKE*	POSITIVE
ALTER*	DO	LIMITED	POSITIVEN
AND*	DROP*	LOCK*	PRAGMA
ANY*	ELSE*	LONG*	PRIOR*
ARRAY	ELSIF	LOOP	PRIVATE
AS*	END	MAX	PROCEDURE
ASC*	EXCEPTION	MIN	PUBLIC*
AUTHID	EXCLUSIVE*	MINUS*	RAISE
ASSIGN	EXECUTE	MINUTE	RANGE
AVG	EXISTS*	MLSLABEL*	RAW*
BEGIN	EXIT	MOD	REAL
BETWEEN*	EXTENDS	MODE*	RECORD
BINARY_INTEGER	FALSE	MONTH	REF
BODY*	FETCH	NATURAL	RELEASE
BOOLEAN	FLOAT*	NATURALN	RETURN
BULK	FOR*	NEW	REVERSE
BY*	FORALL	NEXTVAL	ROLLBACK
CHAR*	FROM*	NOCOPY	ROW*
CHAR_BASE	FUNCTION	NOT*	ROWID*
CHECK*	GOTO	NOWAIT*	ROWLABEL*
CLOSE	GROUP*	NULL*	ROWNUM*
CLUSTER*	HAVING*	NUMBER*	ROWTYPE
COLLECT	HEAP	NUMBER_BASE	SAVEPOINT
COMMENT*	HOUR	OCIROWID	SECOND
COMMIT	IF	OF*	SELECT*
COMPRESS*	IMMEDIATE*	ON*	SEPARATE
CONNECT*	IN*	OPAQUE	SET*
CONSTANT	INDEX*	OPEN	SHARE*
CREATE*	INDICATOR	OPERATOR	SMALLINT*
CURRENT*	INSERT*	OPTION*	SPACE
CURRVAL	INTEGER*	OR*	SQL
CURSOR	INTERFACE	ORDER*	SQLCODE
DATE*	INTERSECT*	ORGANIZATION	SQLERRM
DAY	INTERVAL	OTHERS	START*
DECLARE	INTO*	OUT	STDDEV
DECIMAL*	IS*	PACKAGE	SUBTYPE
DEFAULT*	ISOLATION	PARTITION	SUCCESSFUL*
DELETE*	JAVA	PCTFREE*	SUM
DESC*	LEVEL*	PLS_INTEGER	SYNONYM*

<b>SYSDATE*</b>	<b>TYPE</b>	<b>VALUES*</b>	<b>WHILE</b>
<b>TABLE*</b>	<b>UID*</b>	<b>VARCHAR*</b>	<b>WITH*</b>
<b>THEN*</b>	<b>UNION*</b>	<b>VARCHAR2*</b>	<b>WORK</b>
<b>TIME</b>	<b>UNIQUE*</b>	<b>VARIANCE</b>	<b>WRITE</b>
<b>TIMESTAMP</b>	<b>UPDATE*</b>	<b>VIEW*</b>	<b>YEAR</b>
<b>TO*</b>	<b>USE</b>	<b>WHEN</b>	<b>ZONE</b>
<b>TRIGGER*</b>	<b>USER*</b>	<b>WHENEVER*</b>	
<b>TRUE</b>	<b>VALIDATE*</b>	<b>WHERE*</b>	

## SYSTEM PRIVILEGES

### INDEX

#### System Privilege

CREATE INDEX

CREATE ANY INDEX

ALTER ANY INDEX

DROP ANY INDEX

QUERY REWRITE

GLOBAL QUERY REWRITE

#### Allows Grantee To ...

Create in the grantee's schema an index on any table in the grantee's schema or a domain index

Create in any schema except SYS a domain index or an index on any table in any schema except SYS

Alter indexes in any schema except SYS

Drop indexes in any schema except SYS

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

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

ALTER ANY ROLE

DROP ANY ROLE

GRANT ANY ROLE

Create roles

Alter any role in the database

Drop roles

Grant any role in the database

### SEQUENCES

#### System Privileges

CREATE SEQUENCE

CREATE ANY SEQUENCE

ALTER ANY SEQUENCE

DROP ANY SEQUENCE

SELECT ANY SEQUENCE

Create a sequence in grantee's schema

Create sequences in any schema except SYS

Alter any sequence in the database

Drop sequences in any schema except SYS

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 <b>Note:</b> 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 <b>Note:</b> 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

NOT

#### Function

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

'Equal to any member of' test. Equivalent to '= ANY'.

Equivalent to '!=ALL'. Evaluates to FALSE if any member of the set is NULL.

IN

Compares a value to each value in a list or returned by a query. Must be preceded by =, !=, >, <, >=, <=.

NOT IN

Evaluates to FALSE if the query returns no rows

ANY

Compares a value to every value in a list or returned by a query. Must be preceded by =, !=, >, <, >=, <=.

SOME

Evaluates to TRUE if the query returns no rows

ALL

[Not] greater than or equal to *x* and less than or equal to *y*

[NOT] BETWEEN *x* AND *y*

TRUE if a subquery returns [does not return] at least one row

[NOT] EXISTS

TRUE if *x* does [not] match the pattern *y*. Within *y*, the character '%' matches any string of zero or more characters except null. The character '\_' matches any single character. Any character, except percent (%) and underbar (\_), may follow ESCAPE. A wildcard character

*x* [NOT] LIKE *y* [ESCAPE 'z']

IS [NOT] NULL

is treated as a literal if preceded by the character designated as the escape character.  
Tests for nulls. This is the only operator that you should use to test for nulls.

## Set Operators

### Operator

UNION  
UNION ALL  
INTERSECT  
MINUS

### Returns

All rows selected by either query  
All rows selected by either query, including all duplicates  
All distinct rows selected by both queries  
All distinct rows selected by the first query but not the second

### Operator

|| Concatenation operator  
(+) Outer Join operator

### Purpose

Concatenates character strings  
Indicates that the preceding column is an outer join column in a join  
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.

PRIOR

## 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

#### Operator

+, -  
\*, /  
+, -, ||  
=, !=, <, >, <=, >=, IS NULL, LIKE,  
BETWEEN, IN  
NOT  
AND  
OR

#### Operation

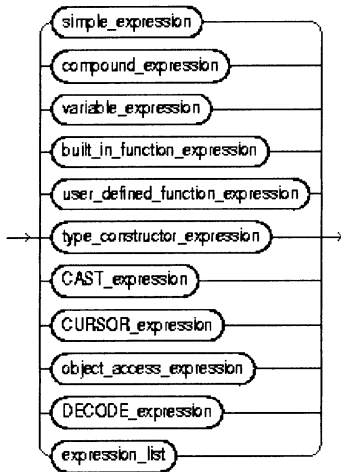
identity, negation  
multiplication, division  
addition, subtraction, concatenation  
comparison  
exponentiation, logical negation  
conjunction  
disjunction

## Expressions and Conditions

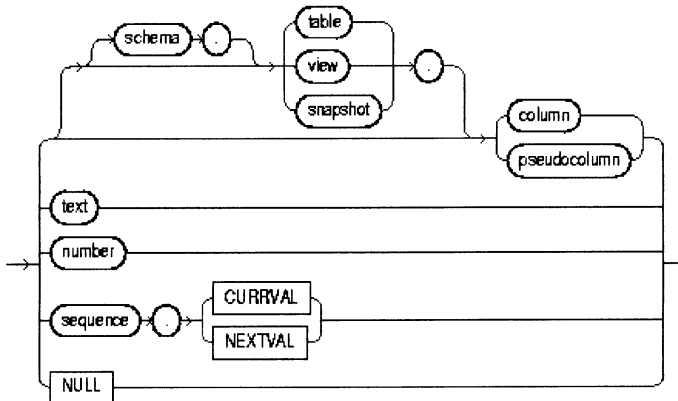
The following diagrams show the possible expressions and conditions denoted by “expr” and “condition” in the SQL command syntax diagrams.

### Expressions

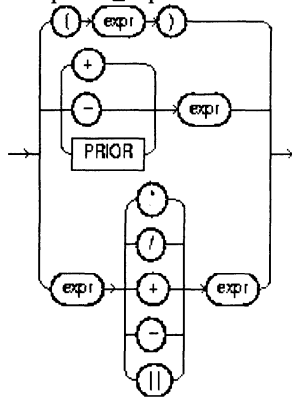
expr ::=



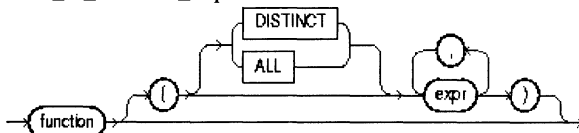
simple\_expression ::=



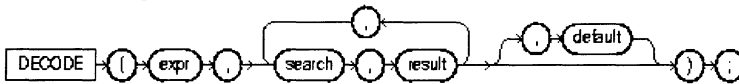
compound\_expression ::=



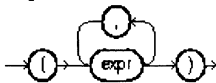
built\_in\_function\_expression ::=



DECODE\_expression ::=

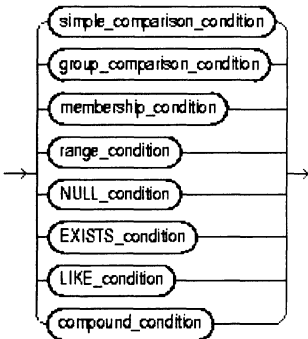


expression\_list ::=

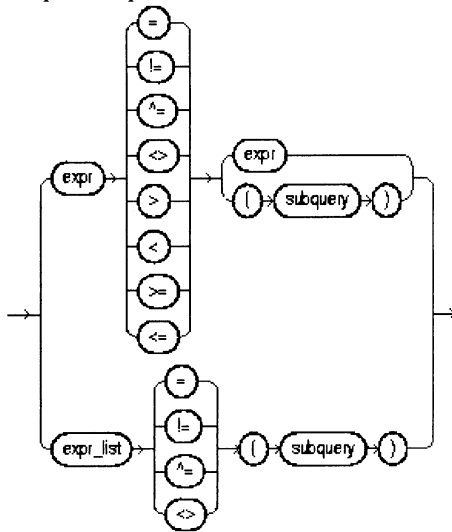


### Conditions

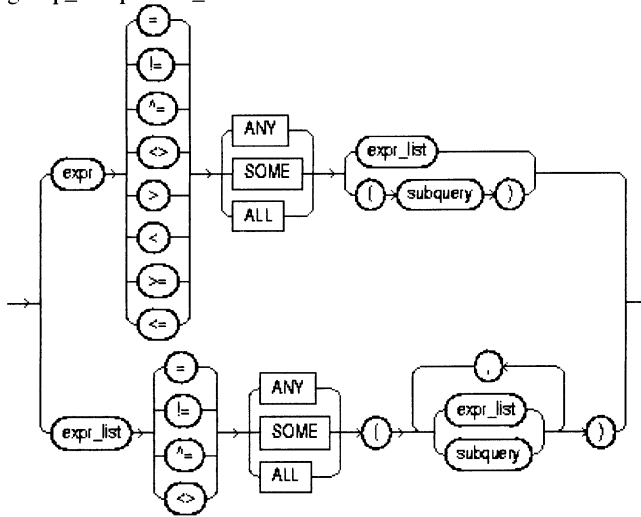
condition ::=



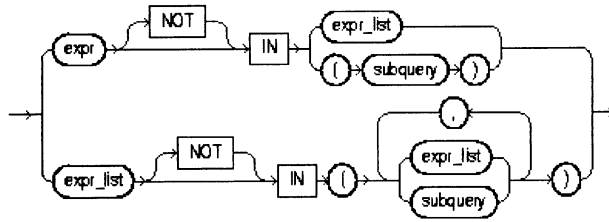
simple\_comparison\_condition ::=



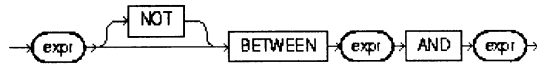
group\_comparison\_condition ::=



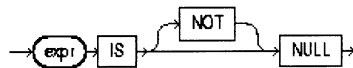
membership\_condition ::=



range\_conditions ::=



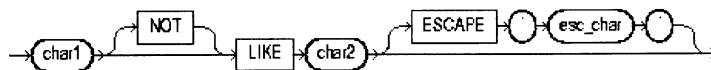
NULL\_condition ::=



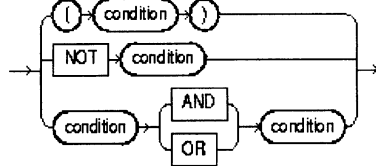
EXISTS\_condition ::=



LIKE\_condition ::=



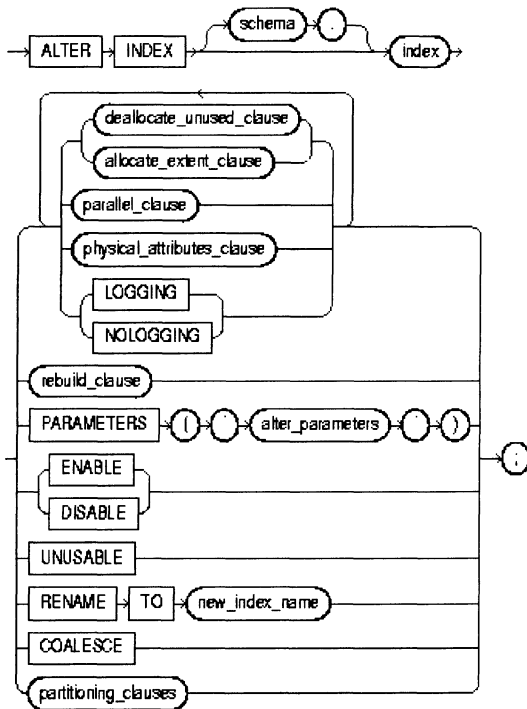
compound\_condition ::=



## SQL Commands Syntax

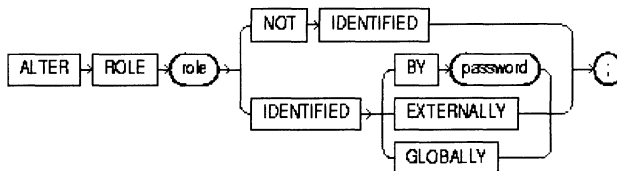
### ALTER INDEX

Use this command to change or rebuild an existing index.



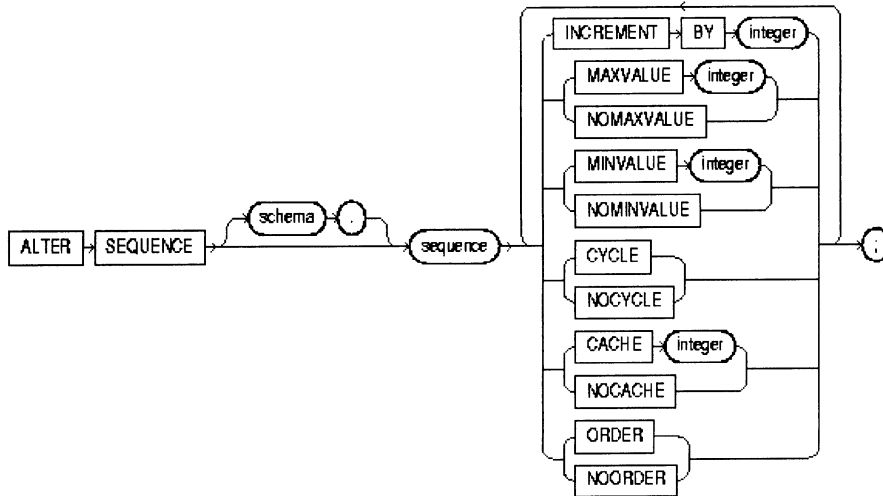
### ALTER ROLE

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



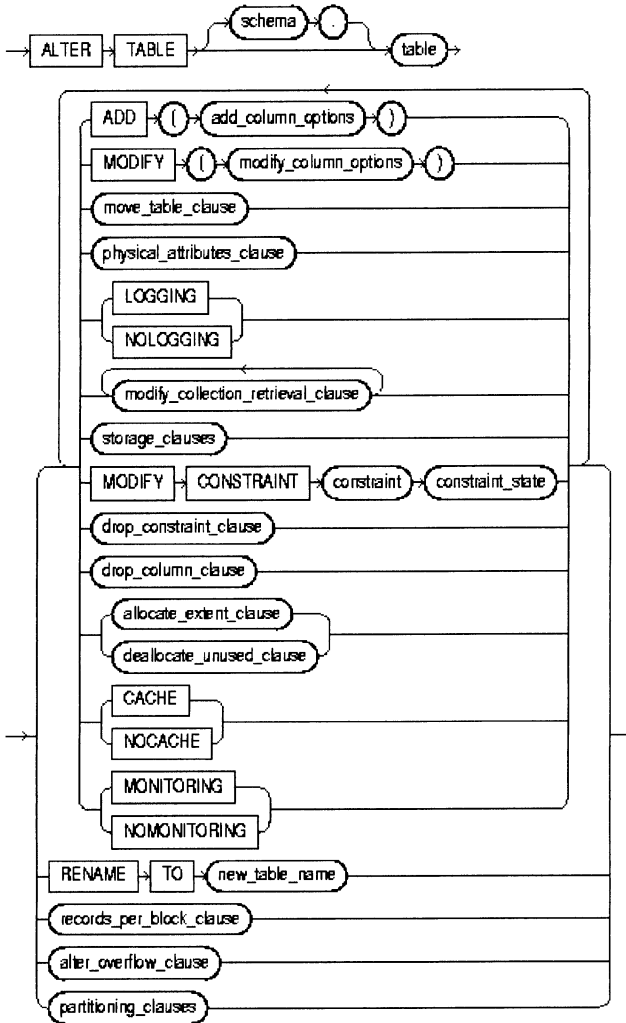
## ALTER SEQUENCE

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.

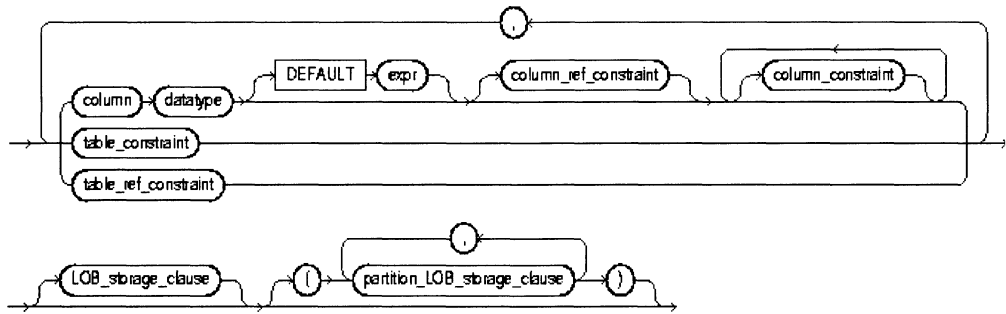


## ALTER TABLE

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

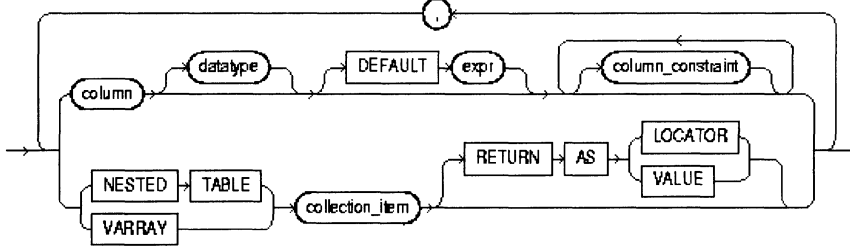


add\_column\_options ::=

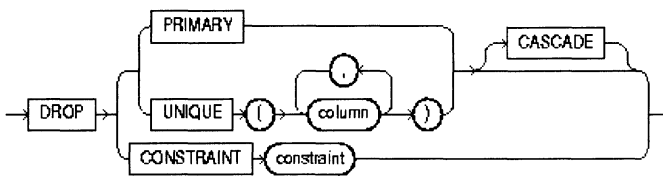


column\_constraint, table\_constraint, column\_ref\_constraint, table\_ref\_constraint, constraint\_state ::=  
See *constraint\_clause*

modify\_column\_options ::=



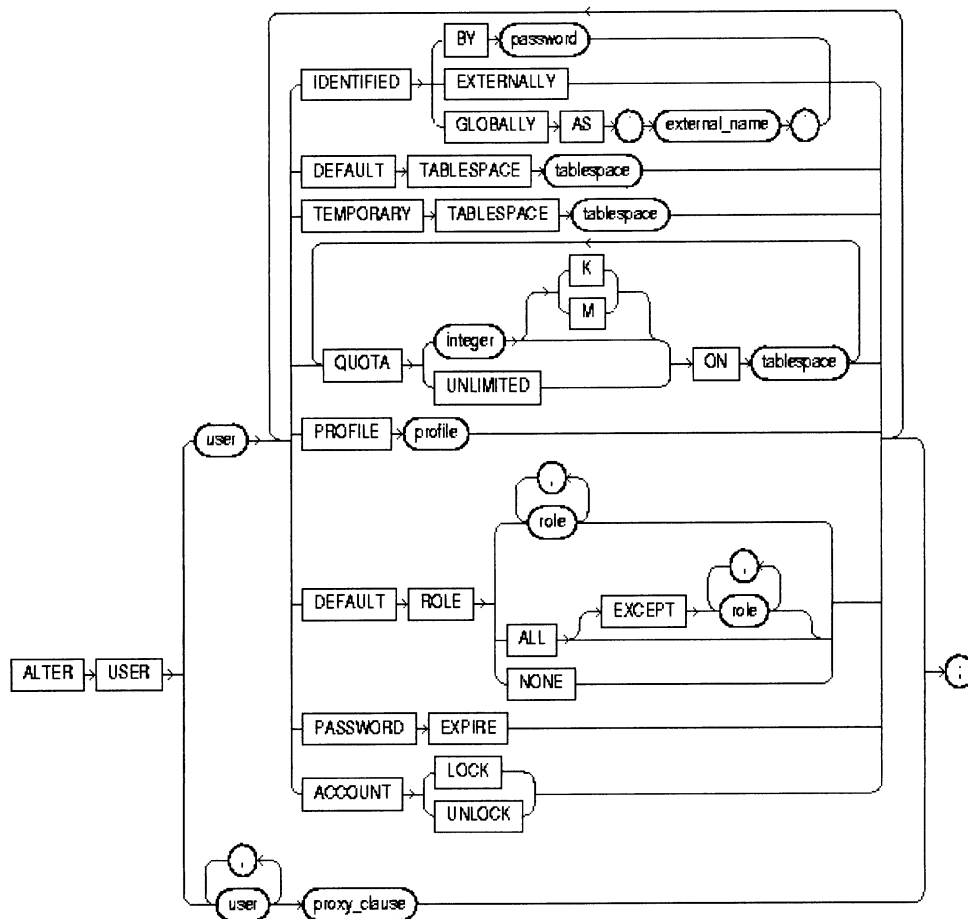
drop\_constraint\_clause ::=





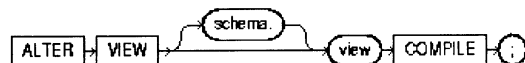
## ALTER USER

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



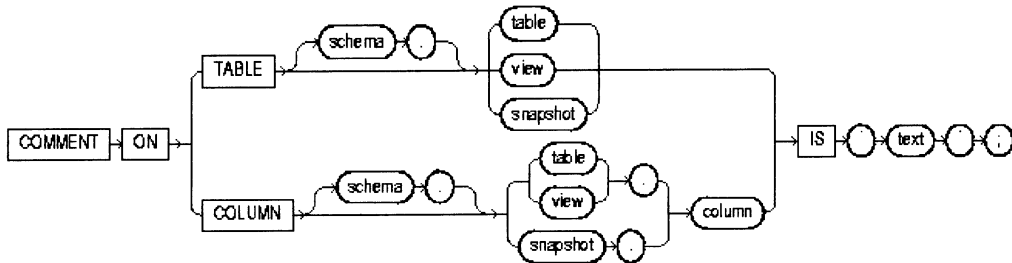
## ALTER VIEW

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



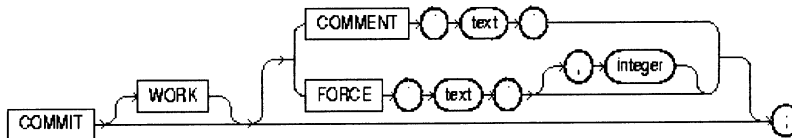
## COMMENT

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



## COMMIT

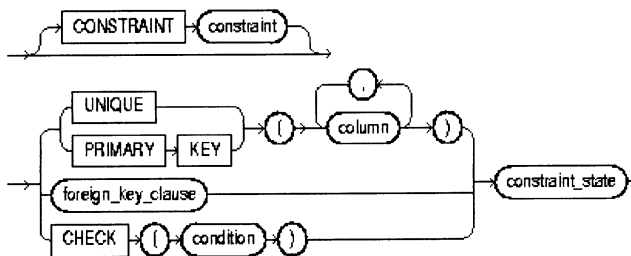
Use this command to end your current transaction and make permanent all changes performed in the transaction.



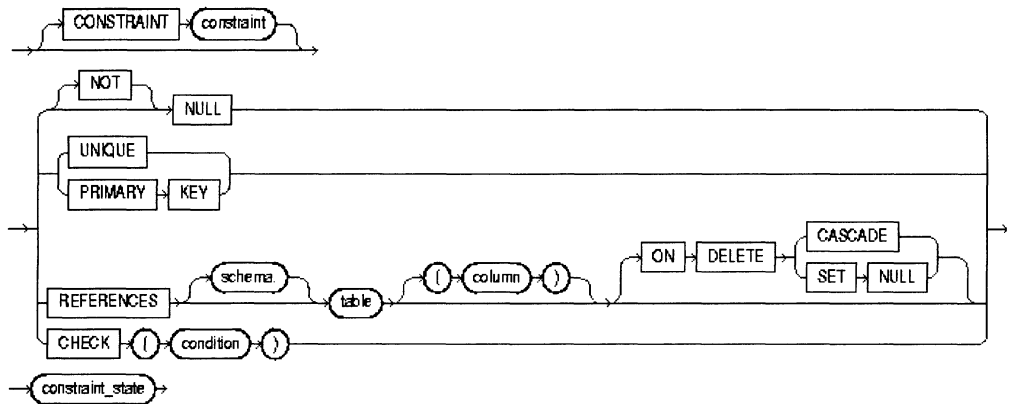
### *constraint\_clause*

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.

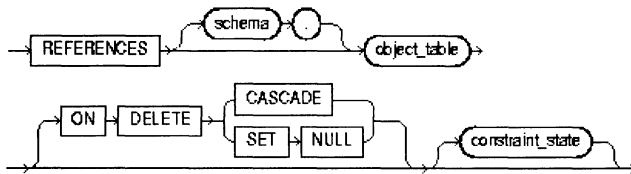
`table_constraint ::=`



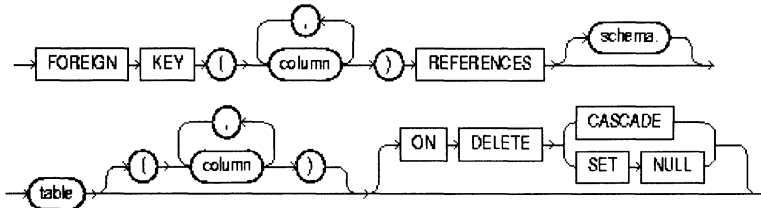
column\_constraint ::=



references\_clause ::=

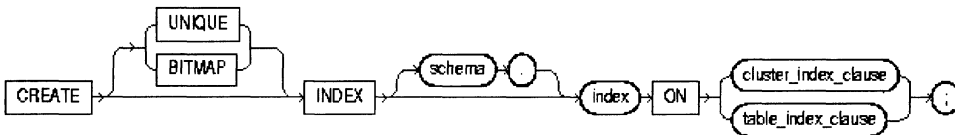


foreign\_key\_clause ::=

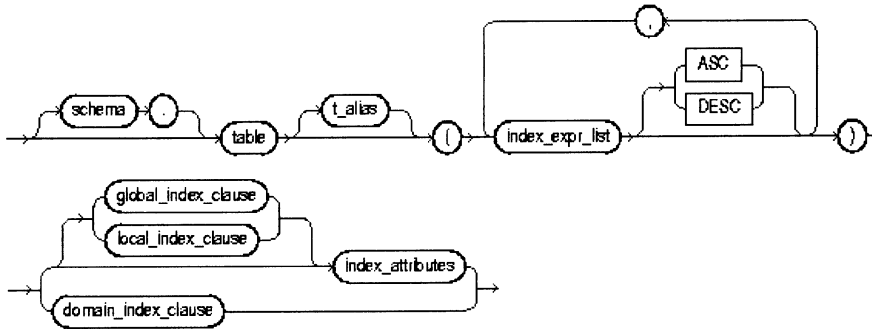


## CREATE INDEX

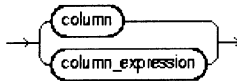
Use this command to create an index on one or more columns of a table, a partitioned table, an index-organized table, a cluster, one or more scalar typed object attributes of a table or a cluster, or a nested table storage table for indexing a nested table column.



table\_index\_clause ::=

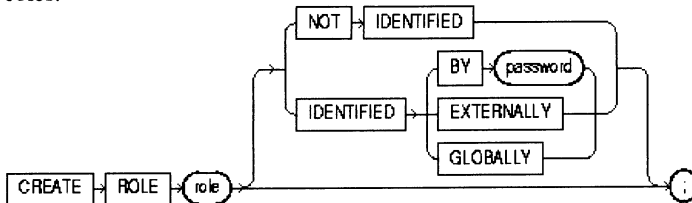


index\_expr\_list ::=



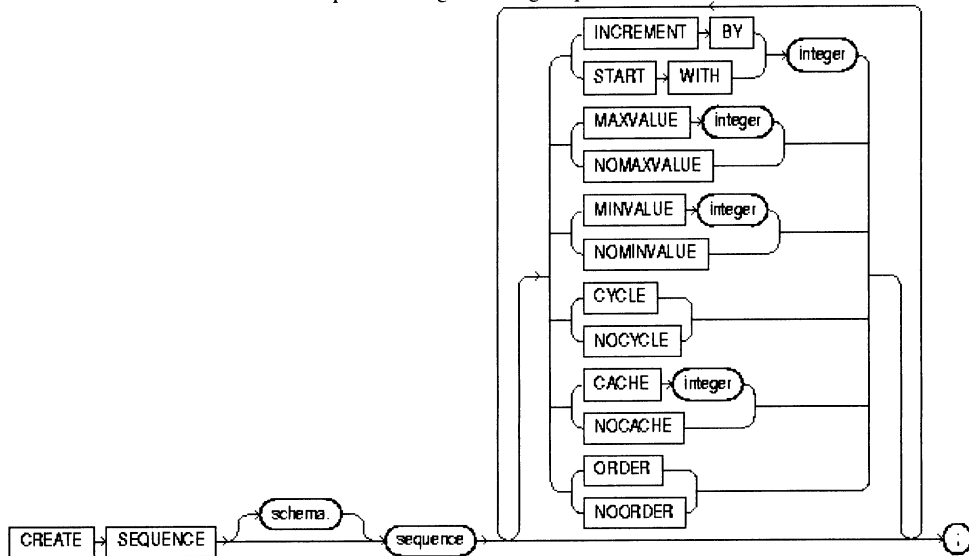
## CREATE ROLE

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



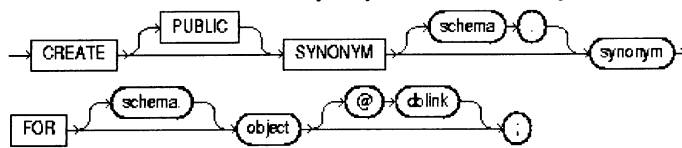
## CREATE SEQUENCE

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



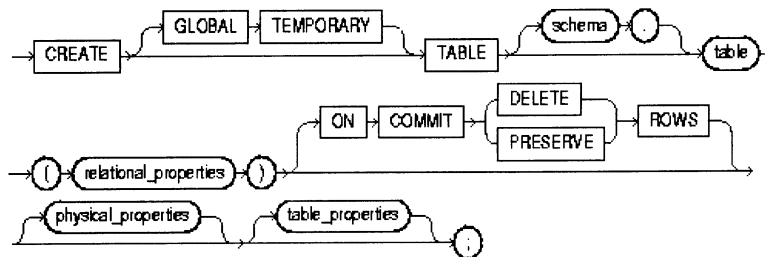
## CREATE SYNONYM

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

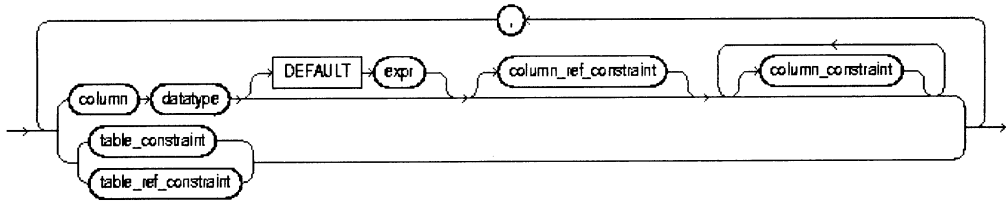


## CREATE TABLE

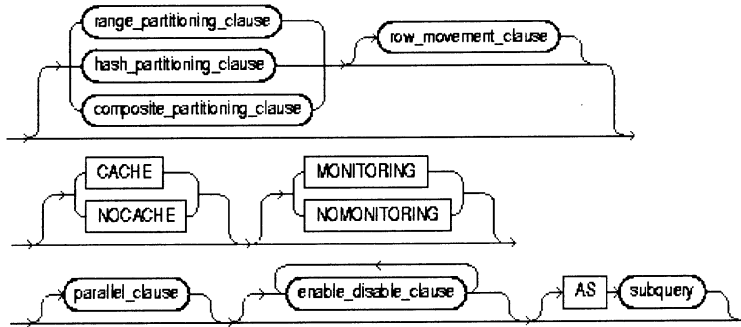
Use this command to create a relational table, the basic structure to hold user data, and to create an object table or a table that uses an object type for a column definition.



relational\_properties ::=



table\_properties ::=



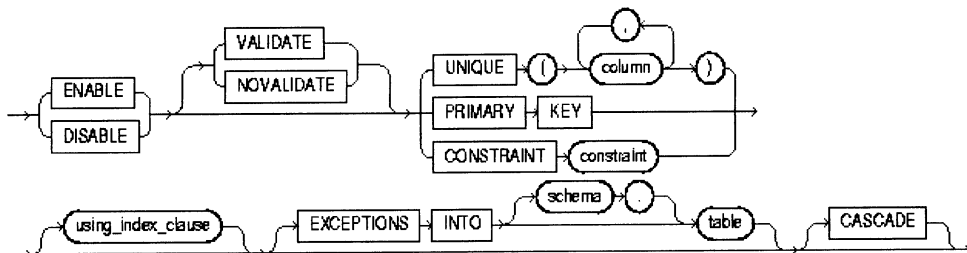
subquery

See *SELECT and subqueries*

table\_constraint, column\_constraint, table\_ref\_constraint, column\_ref\_constraint, constraint\_state

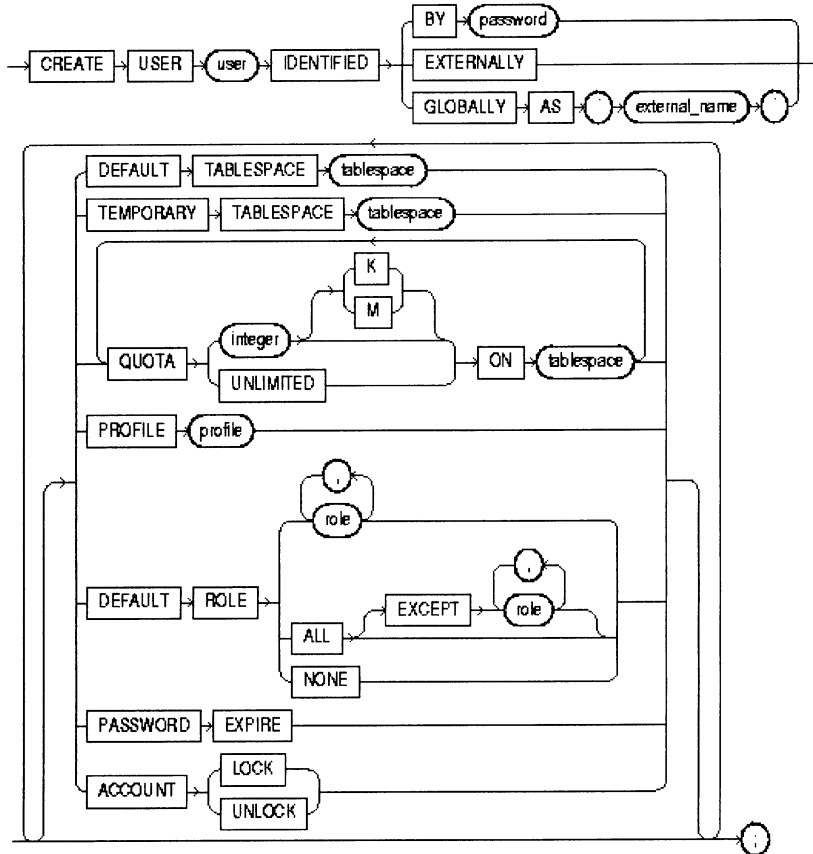
See *constraint\_clause*

enable\_disable\_clause ::=



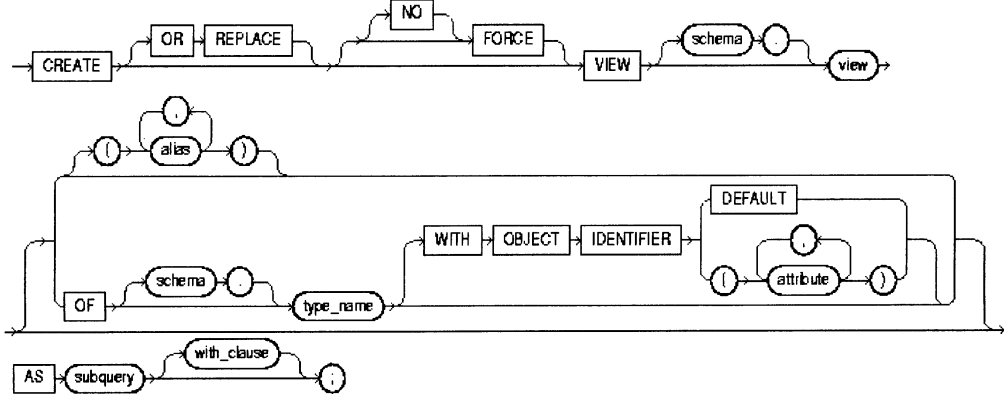
## CREATE USER

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

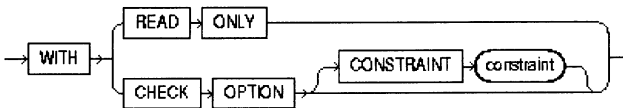
Use this command to define a view, a logical table based on one or more tables or views.



subquery

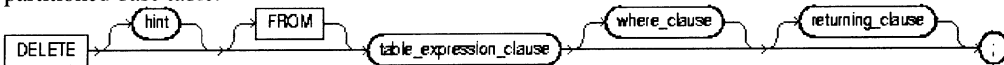
See *SELECT and subqueries*

with\_clause ::=

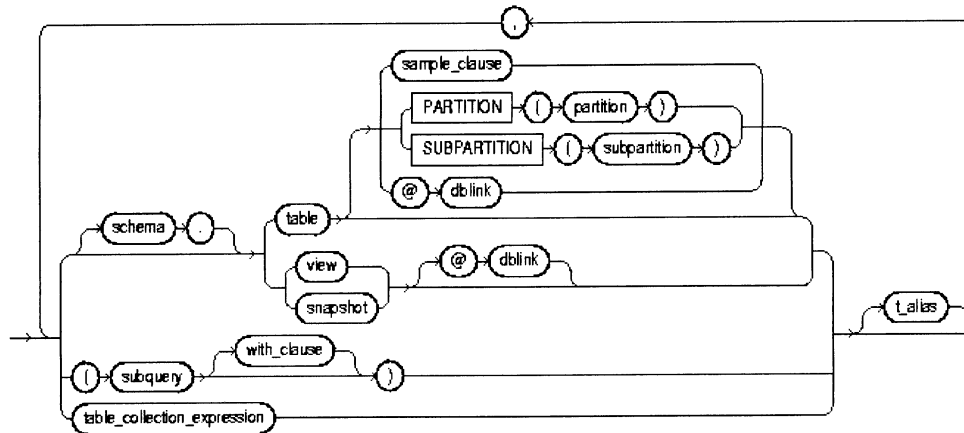


## DELETE

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



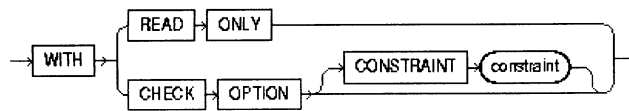
table\_expression\_clause ::=



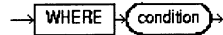
subquery

See *SELECT and subqueries*

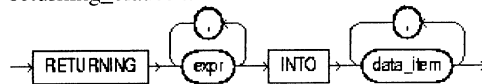
with\_clause ::=



where\_clause ::=

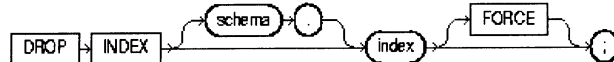


returning\_clause ::=



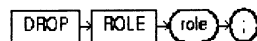
## DROP INDEX

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



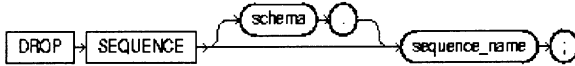
## DROP ROLE

Use this command to remove a role from the database.



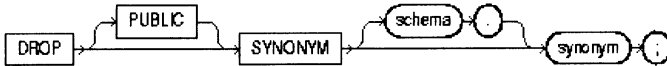
## DROP SEQUENCE

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.



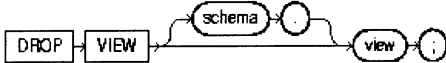
## DROP USER

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



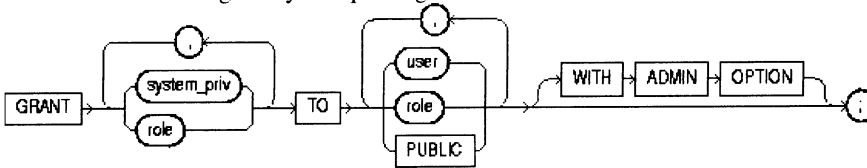
## DROP VIEW

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



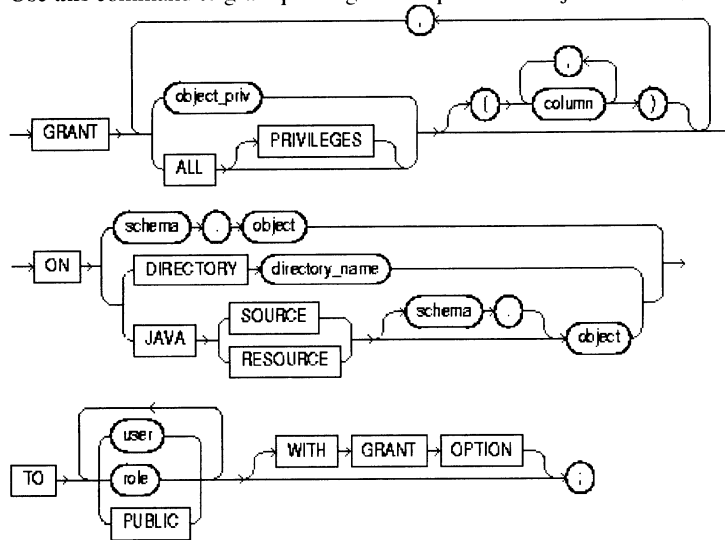
## GRANT (*system\_privileges\_and\_roles*)

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



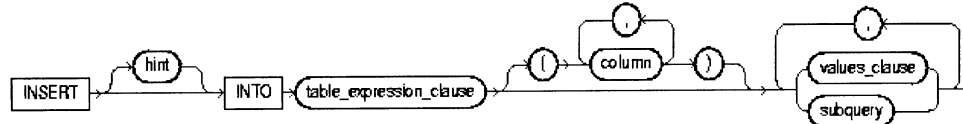
## GRANT (object\_privileges)

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

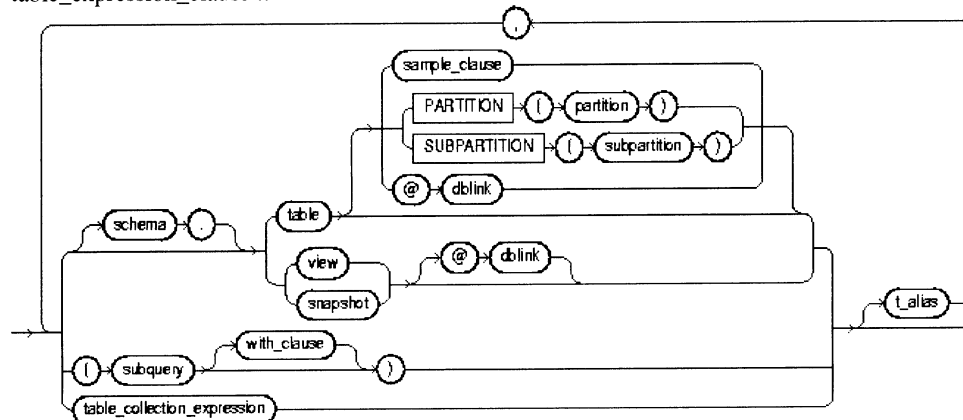


## INSERT

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



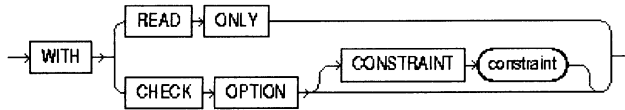
table\_expression\_clause ::=



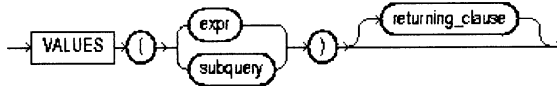
subquery

See *SELECT and subqueries*

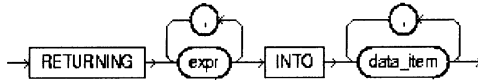
with\_clause ::=



values\_clause ::=

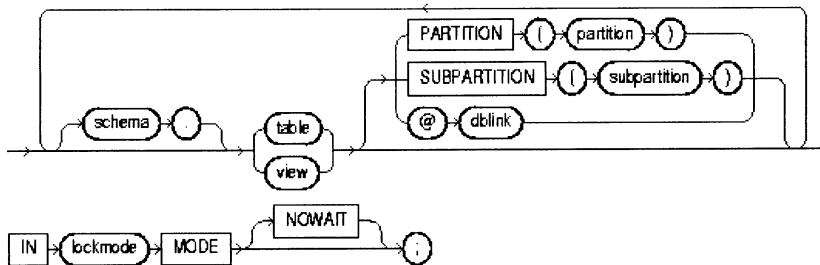
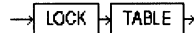


returning\_clause ::=



## 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.



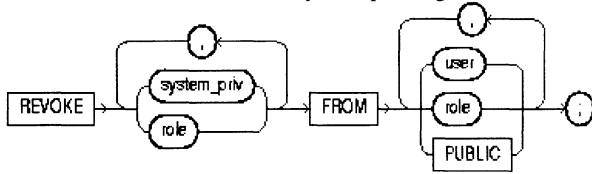
## RENAME

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



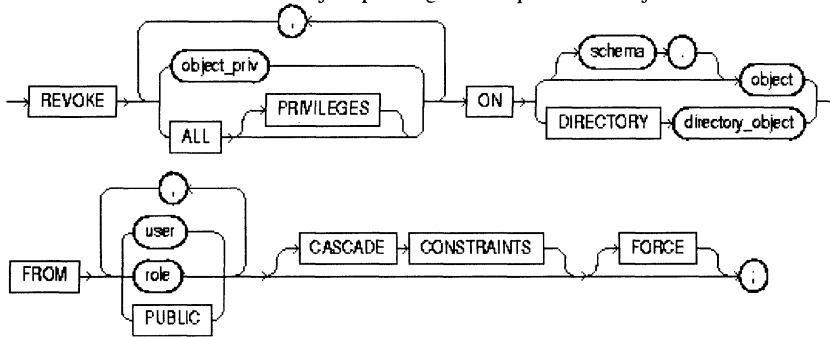
## REVOKE (system\_privileges\_and\_roles)

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



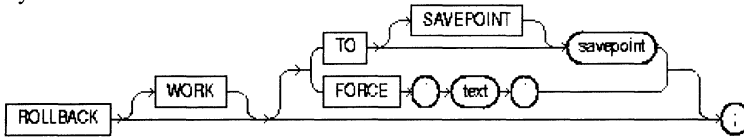
## REVOKE (object\_privileges)

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.



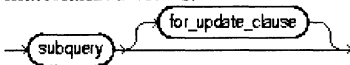
## SAVEPOINT

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

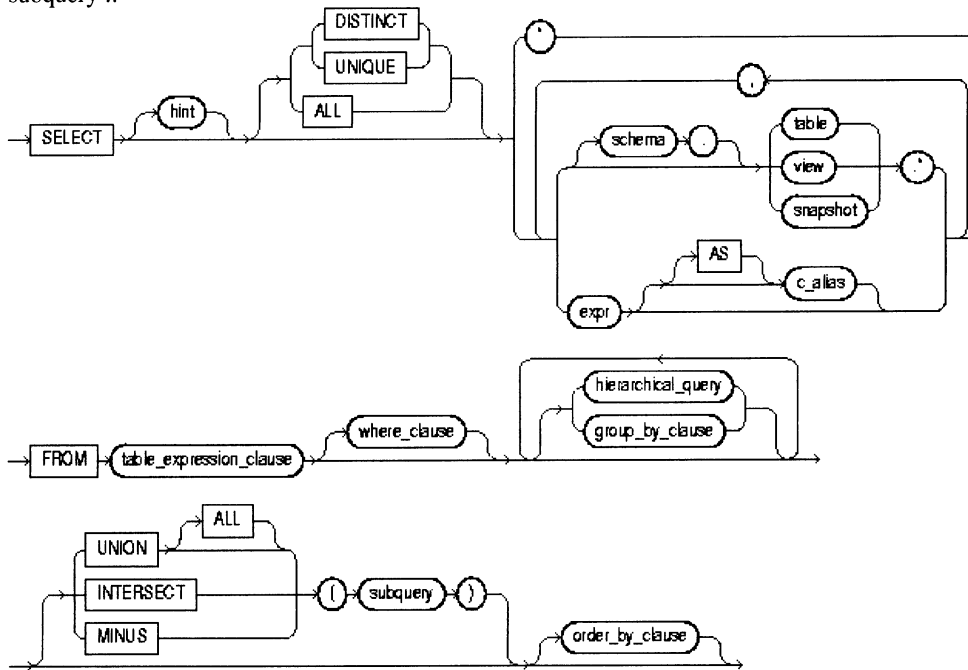


## SELECT

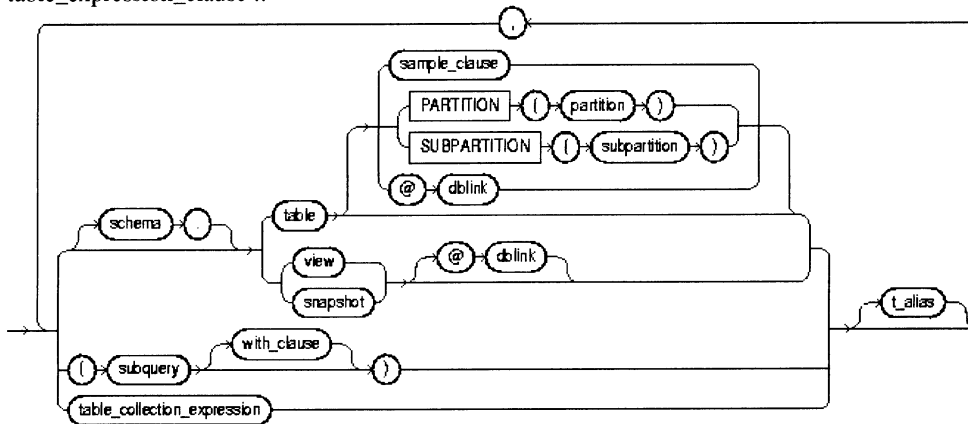
Use this command to retrieve data from one or more tables, object tables, views, object views, or materialized views.



subquery ::=



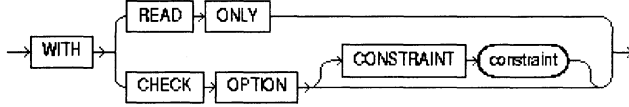
table\_expression\_clause ::=



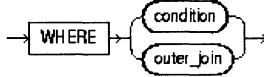
sample\_clause ::=



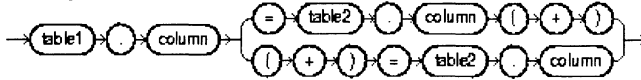
with\_clause ::=



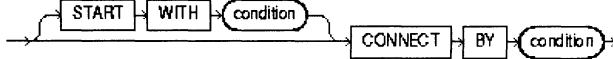
where\_clause ::=



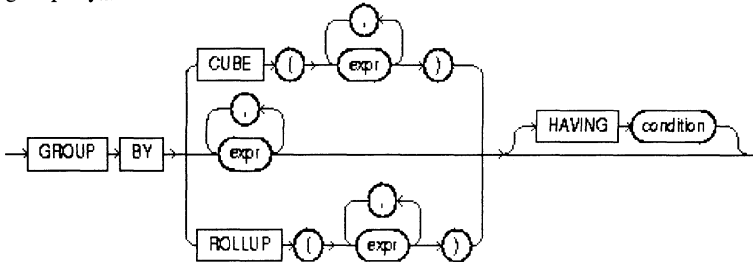
outer\_join ::=



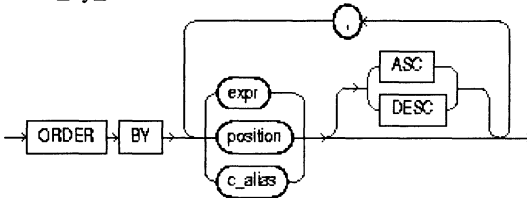
hierarchical\_query\_clause ::=



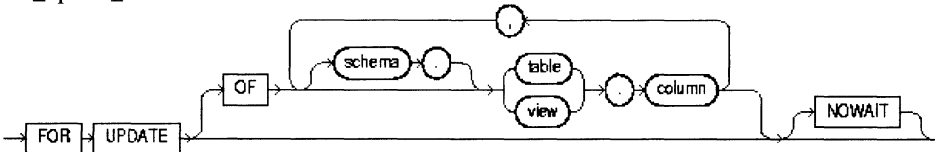
group\_by\_clause ::=



order\_by\_clause ::=

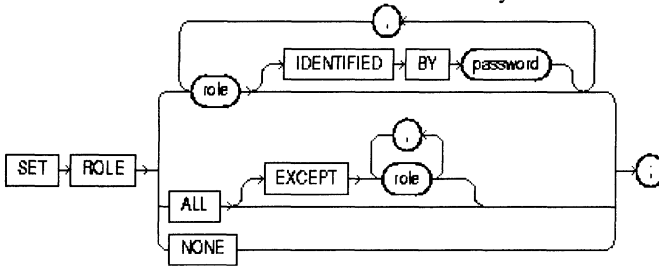


for\_update\_clause ::=



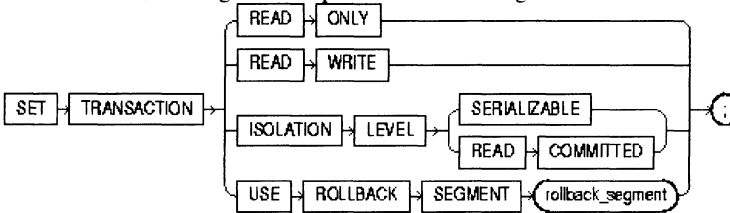
## SET ROLE

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



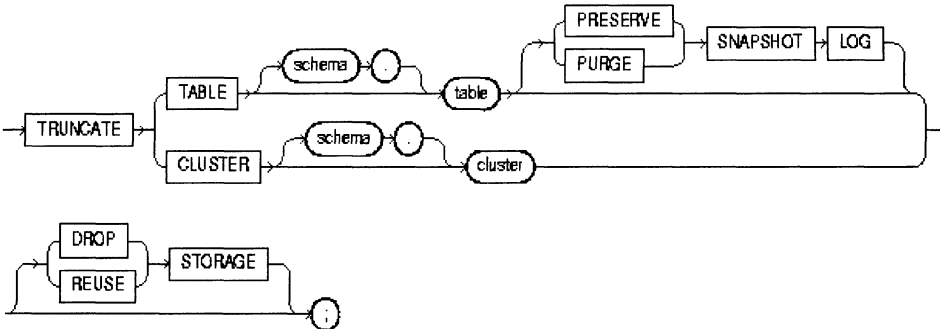
## SET TRANSACTION

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.



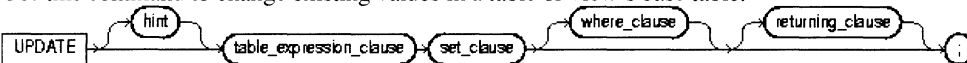
## TRUNCATE

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.

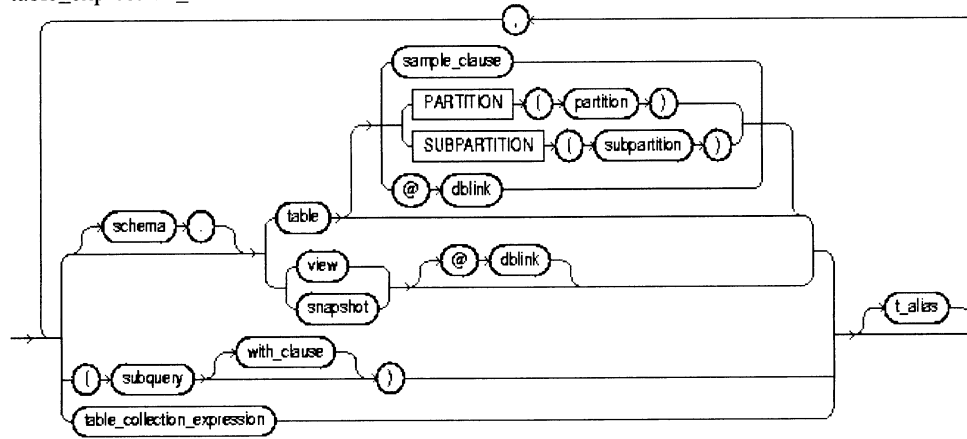


## UPDATE

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



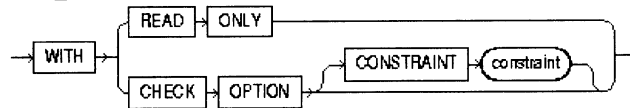
table\_expression\_clause ::=



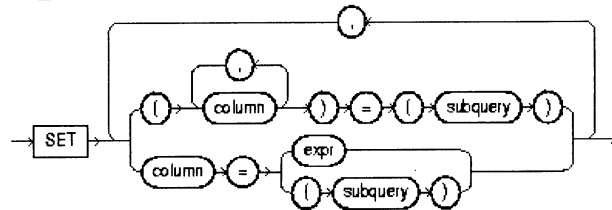
subquery

See *SELECT* and subqueries

with\_clause ::=



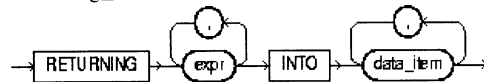
set\_clause ::=



where\_clause ::=



returning\_clause ::=



## Data Dictionary Views

Data dictionary view available to all users

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

**ORACLE<sup>®</sup>**

Oracle is a registered trademark of Oracle Corporation. All Oracle product names are trademarks or registered trademarks of Oracle Corporation. All other companies and product names mentioned are used for identification purposes only, and may be trademarks of their respective owner.

Copyright © Oracle Corporation 1998  
All Rights Reserved