SQL*Plus

Quick Reference

Release 9.0.1

June 2001

Part No. A88828-01

About this Quick Reference

This Quick Reference shows the syntax for SQL*Plus commands. For detailed information on each command, refer to the *SQL*Plus User's Guide and Reference.*

This Quick Reference discusses the following topics:

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. If users are expected to type them into the system, they are identified by the keyboard icon shown to the left of the following example. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

Similarly, output from an example is identified by a computer screen icon in the margin as shown in the margin following.

```
PAGESIZE 24
```

Where both icons occur together, it implies interative entry and output.

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[ ]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (*digits* [, *precision*])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ENABLE | DISABLE}
|	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE | DISABLE} [COMPRESS | NOCOMPRESS]

. . .	Horizontal ellipsis points indicate either:
	■ That we have omitted parts of the code that are not directly related to the example
	■ That you can repeat a portion of the code
		CREATE TABLE ... AS *subquery*;
		SELECT *col1*, *col2*, ... , *coln* FROM employees;

. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	

Convention	Meaning	Example

Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	`acctbal NUMBER(11,2);`
		`acct    CONSTANT NUMBER(4) := 3;`

`Italics`	Italicized text indicates placeholders or variables for which you must supply particular values.	`CONNECT SYSTEM/`*`system_password`*
		`DB_NAME = `*`database_name`*

`UPPERCASE`	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	`SELECT last_name, employee_id FROM employees;`
		`SELECT * FROM USER_TABLES;`
		`DROP TABLE hr.employees;`

`lowercase`	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.
	Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.
		`SELECT last_name, employee_id FROM employees;`
		`sqlplus hr/hr`
		`CREATE USER mjones IDENTIFIED BY ty3MU9;`

Starting and Leaving SQL*Plus

Use the following commands to log in to and out of SQL*Plus.

SQLPLUS [[*option*] [*logon*] [*start*]]

where *option* has the following syntax:

> -H[ELP] | -V[ERSION]
> | [[-M[ARKUP] "*mark_options*"] [-R[ESTRICT] {1|2|3}] [-S[ILENT]]]

> where *mark_options* has the following syntax:
>
> > HTML [ON|OFF] [HEAD *text*] [BODY *text*] [TABLE *text*]
> > [ENTMAP {ON|OFF}] [SPOOL {ON|OFF}] [PRE[FORMAT] {ON|OFF}]

where *logon* has the following syntax:

> {*username*[/*password*][@*connect_identifier*]|/} [AS {SYSOPER|SYSDBA}]
> |/NOLOG

> **Note:** On many operating systems, if you log in using the option, AS {SYSOPER|SYSDBA}, you need to contain the logon clause in quotes.

and where *start* has the following syntax:

> @{*uri*|*file_name*[.*ext*]} [*arg* ...]

{EXIT|QUIT} [SUCCESS|FAILURE|WARNING|*n*|*variable*
|:*BindVariable*] [COMMIT|ROLLBACK]

Commits all pending changes, terminates SQL*Plus, and returns control to the operating system.

Starting Up and Shutting Down a Database

STARTUP [FORCE] [RESTRICT] [PFILE=*filename*] [MOUNT [*dbname*] | OPEN [*open_options*] [*dbname*] | NOMOUNT]

where *open_options* has the following syntax:

READ {ONLY | WRITE [RECOVER]} | RECOVER

Starts an Oracle instance with several options, including mounting and opening a database.

SHUTDOWN [ABORT|IMMEDIATE|NORMAL|TRANSACTIONAL [LOCAL]]

Shuts down a currently running Oracle instance, optionally closing and dismounting a database.

Entering and Executing Commands

Use the following commands to execute and collect timing statistics on SQL commands and PL/SQL blocks.

/ (slash)

Executes the SQL command or PL/SQL block currently stored in the SQL buffer. Does not list the command.

EXEC[UTE] *statement*

Executes a single PL/SQL statement or runs a stored procedure.

R[UN]

Lists and executes the SQL command or PL/SQL block currently stored in the SQL buffer.

TIMI[NG] [START *text*|SHOW|STOP]

Records timing data for an elapsed period of time, lists the current timer's name and timing data, or lists the number of active timers.

Use the following command to access the help system.

HELP [*topic*]

Accesses help on SQL*Plus commands.

Use the following command to execute host operating system commands.

HO[ST] [*command*]

Executes a host operating system command without leaving SQL*Plus.

Note: With some operating systems, you can use another character instead of HOST such as "$" (VMS), "!" (UNIX) and "$" (Windows). See the Oracle installation and user's manuals provided for your operating system for details.

Manipulating SQL, SQL*Plus, and PL/SQL Commands

Use the following commands to edit SQL commands and PL/SQL blocks.

A[PPEND] *text*

Adds specified text to the end of the current line in the SQL buffer. To separate *text* from the preceding characters with a space, enter two spaces between APPEND and *text*. To append *text* that ends with a semicolon, end the command with two semicolons (SQL*Plus interprets a single semicolon as a command terminator).

C[HANGE] *sepchar old* [*sepchar* [*new* [*sepchar*]]]

Changes text on the current line in the SQL buffer. You can use any non-alphanumeric character such as "/" or "!" as a *sepchar*. You can omit the space between CHANGE and the first *sepchar*.

DEL [*n*|*n m*|*n* *|*n* LAST|*|* *n*|* LAST|LAST]

Deletes one or more lines of the SQL buffer ("*" indicates the current line). You can omit the space between DEL and *n* or *, but not between DEL and LAST. Enter DEL with no clauses to delete the current line of the buffer.

I[NPUT] [*text*]

Adds one or more new lines of text after the current line in the SQL buffer.

L[IST] [*n*|*n m*|*n* *|*n* LAST|*|* *n*|* LAST|LAST]

Lists one or more lines of the SQL buffer ("*" indicates the current line). You can omit the space between LIST and *n* or *, but not between LIST and LAST. Enter LIST with no clauses to list all lines.

Use the following commands to create and modify command files.

@ { *uri* | *file_name*[.*ext*] } [*arg* ...]

Runs the SQL*Plus statements in the specified command file. The command file can be called from the local file system or from a web server. You can pass values to script variables in the usual way.

uri is only supported on Windows platforms in this release.

@@ *file_name*[.*ext*]

Runs the specified command file. This command is identical to the @ (“at” sign) command. It is useful for running nested command files because it looks for the specified command file in the same path or *uri* as the command file from which it was called.

ED[IT] [*file_name*[.*ext*]]

Invokes a host operating system text editor on the contents of the specified file or on the contents of the SQL buffer. To edit the buffer contents, omit the file name.

GET *file_name*[.*ext*] [LIS[T]|NOL[IST]]

Loads a host operating system file into the SQL buffer.

REM[ARK]

Begins a comment in a command file. The REMARK command must appear at the beginning of a line, and the comment ends at the end of the line (a line cannot contain both a comment and a command). SQL*Plus does not interpret the comment as a command.

SAV[E] *file_name*[.*ext*] [CRE[ATE]|REP[LACE]|APP[END]]

Saves the contents of the SQL buffer into a host operating system file (a command file).

STORE SET *file_name*[.*ext*] [CRE[ATE]|REP[LACE]|APP[END]]

Saves the attributes of the current SQL*Plus environment in a host operating system file (a command file).

STA[RT] { *uri* | *file_name*[.*ext*] } [*arg* ...]

Runs the SQL*Plus statements in the specified command file. The command file can be called from the local file system or from a web server. You can pass values to script variables in the usual way.

uri is only supported on Windows platforms in this release.

WHENEVER OSERROR {EXIT [SUCCESS|FAILURE
|*n*|*variable*|*:BindVariable*] [COMMIT|ROLLBACK]|CONTINUE
[COMMIT|ROLLBACK|NONE]}

Exits SQL*Plus if an operating system error occurs (such as a file I/O error).

WHENEVER SQLERROR {EXIT [SUCCESS|FAILURE|WARNING
|*n*|*variable*|*:BindVariable*] [COMMIT|ROLLBACK]|CONTINUE
[COMMIT|ROLLBACK|NONE]}

Exits SQL*Plus if a SQL command or PL/SQL block generates an error.

Use the following commands to write interactive commands.

ACC[EPT] *variable* [NUM[BER]|CHAR|DATE] [FOR[MAT] *format*]
[DEF[AULT] *default*] [PROMPT *text*|NOPR[OMPT]] [HIDE]

Reads a line of input and stores it in a given user variable.

DEF[INE] [*variable*]|[*variable* = *text*]

Specifies a user variable and assigns it a CHAR value. Alternatively, lists the value and variable type of a single variable or all variables.

PAU[SE] [*text*]

Displays an empty line followed by a line containing text, then waits for the user to press [Return]. Alternatively, displays two empty lines and waits for the user's response.

PROMPT [*text*]

Sends the specified message or a blank line to the user's screen.

UNDEF[INE] *variable* ...

Deletes given user variables that you defined either explicitly (with the DEFINE command) or implicitly (with an argument to the START command).

Use the following commands to create and display bind variables.

PRI[NT] [*variable* ...]

Displays the current values of bind variables.

VAR[IABLE] [*variable* {NUMBER|CHAR|CHAR (*n* [CHAR|BYTE])|NCHAR
|NCHAR (*n*)|VARCHAR2 (*n* [CHAR|BYTE])| NVARCHAR2(*n*)|CLOB|NCLOB|REFCURSOR}]

Declares a bind variable which can then be referenced in PL/SQL. If no arguments are supplied, VARIABLE lists all declared bind variables.

Use the following symbols to create substitution variables and parameters for use in command files.

&*n*

Specifies a parameter in a command file you run using the START command. START substitutes values you list after the command file name as follows: the first for &1, the second for &2, and so on.

&*user_variable*, &&*user_variable*

Indicates a substitution variable in a SQL or SQL*Plus command. SQL*Plus substitutes the value of the specified user variable for each substitution variable it encounters. If the user variable is undefined, SQL*Plus prompts you for a value *each* time an “&” variable is found, and the *first* time an “&&” variable is found.

. (period)

Terminates a substitution variable followed by a character that would otherwise be part of the variable name.

Formatting Query Results

Use the following commands to format, store and print your query results.

ATTRIBUTE [*type_name*.*attribute_name* [*option*...]]

Specifies display attributes for a given object attribute, or lists the current display attributes for a single attribute or for all attributes; *option* represents one of the following clauses:

ALI[AS] *alias*
CLE[AR]
FOR[MAT] *format*
LIKE {*type_name*.*attribute_name*|*alias*}
ON|OFF

BRE[AK] [ON *report_element* [*action* [*action*]]] ...

Specifies where and how formatting will change in a report (for example, skipping a line each time a given column value changes). Enter BREAK with no clauses to list the current BREAK definition.

Where *report_element* has the following syntax:

{*column*|*expr*|ROW|REPORT}

and where *action* has the following syntax:

[SKI[P] *n*|[SKI[P]] PAGE] [NODUP[LICATES]|DUP[LICATES]]

BTI[TLE] [*printspec* [*text*|*variable*] ...]|[ON|OFF]

Places and formats the specified title at the bottom of each report page, or lists the current BTITLE definition. See TITLE for additional information on valid *printspec* clauses.

CL[EAR] *option* ...

Resets or erases the current value or setting for the specified option; *option* represents one of the following clauses:

BRE[AKS]
BUFF[ER]
COL[UMNS]
COMP[UTES]
SCR[EEN]
SQL
TIMI[NG]

COL[UMN] [{*column*|*expr*} [*option* ...]]

Specifies the display attributes for a given column, such as text for the column heading, or formats for CHAR, NCHAR, VARCHAR2 (VARCHAR), NVARCHAR2 (NCHAR VARYING), LONG, CLOB, NCLOB and NUMBER data; *option* represents one of the following clauses:

ALI[AS] *alias*
CLE[AR]
ENTMAP {ON|OFF}
FFOLD_A[FTER]
FOLD_B[EFORE]
FOR[MAT] *format*
HEA[DING] *text*
JUS[TIFY] {L[EFT]|C[ENTER]|C[ENTRE]|R[IGHT]}
LIKE {*expr*|*alias*}
NEWL[INE]
NEW_V[ALUE] *variable*
NOPRI[NT]|PRI[NT]
NUL[L] text
OLD_V[ALUE] *variable*
ON|OFF
WRA[PPED]|WOR[D_WRAPPED]|TRU[NCATED]

Enter COLUMN followed by *column* or *expr* and no other clauses to list the current display attributes for only the specified column or expression. Enter COLUMN with no clauses to list all current column display attributes.

Enter COLUMN [{*column* |*expr*} FORMAT *format*] where the *format* element specifies the display format for the column.

To change the display format of a NUMBER column, use FORMAT followed by one of the elements in the following table:

Element	**Example(s)**	**Description**
9	9999	Number of “9”s specifies number of significant digits returned. Blanks are displayed for leading zeroes. A zero (0) is displayed for a value of zero.
0	0999 9990	Displays a leading zero or a value of zero in this position as a 0.
$	$9999	Prefixes value with dollar sign.
B	B9999	Displays leading zero or a value of zero in this position as a blank, regardless of “0”s in the format model.

Element	Example(s)	Description
MI	9999MI	Displays "-" after a negative value. For a positive value, a trailing space is displayed.
S	S9999	Returns "+" for positive values and "-" for negative values in this position.
PR	9999PR	Displays a negative value in <angle brackets>. For a positive value, a leading and trailing space is displayed.
D	99D99	Displays the decimal character in this position, separating the integral and fractional parts of a number.
G	9G999	Displays the group separator in this position.
C	C999	Displays the ISO currency symbol in this position.
L	L999	Displays the local currency symbol in this position.
, (comma)	9,999	Displays a comma in this position.
. (period)	99.99	Displays a period (decimal point) in this position, separating the integral and fractional parts of a number.
V	999V99	Multiplies value by 10^n, where n is number of "9"s after "V."
EEEE	9.999EEEE	Displays value in scientific notation (format must contain exactly four "E"s).
RN or rn	RN	Displays upper- or lowercase Roman numerals. Value can be an integer between 1 and 3999.
DATE	DATE	Displays value as a date in MM/DD/YY format; used to format NUMBER columns that represent Julian dates.

COMP[UTE] [*function* [LAB[EL] *text*] ... OF {*expr*|*column*|*alias*} ...ON {*expr*|*column*|*alias*|REPORT|ROW} ...]

Calculates and prints summary lines, using various standard computations, on subsets of selected rows, or lists all COMPUTE definitions. The following table lists valid functions. All functions except NUMBER apply to non-null values only. COMPUTE functions are always executed in the following sequence AVG, COUNT, MINIMUM, MAXIMUM, NUMBER, SUM, STD, VARIANCE.

Function	Computes	Applies to Datatypes
AVG	Average of non-null values	NUMBER
COU[NT]	Count of non-null values	All types
MIN[IMUM]	Minimum value	NUMBER, CHAR, NCHAR, VARCHAR2 (VARCHAR), NVARCHAR2 (NCHAR VARYING)
MAX[IMUM]	Maximum value	NUMBER, CHAR, NCHAR, VARCHAR2 (VARCHAR), NVARCHAR2 (NCHAR VARYING)
NUM[BER]	Count of rows	All types
SUM	Sum of non-null values	NUMBER
STD	Standard deviation of non-null values	NUMBER
VAR[IANCE]	Variance of non-null values	NUMBER

REPF[OOTER] [PAGE] [*printspec* [*text*|*variable*] ...]|[ON|<u>OFF</u>]

Places and formats a specified report footer at the bottom of each report, or lists the current REPFOOTER definition. See REPHEADER for additional information on valid *printspec* clauses.

REPH[EADER] [PAGE] [*printspec* [*text*|*variable*] ...]|[ON|<u>OFF</u>]

Places and formats a specified report header at the top of each report, or lists the current REPHEADER definition. Use one of the following clauses in place of *printspec*:

COL *n*
S[KIP] [*n*]
TAB *n*
LE[FT]
CE[NTER]
R[IGHT]
BOLD
FORMAT *text*

SPO[OL] [*filename*[.*ext*]]|OFF|OUT]

Stores query results in an operating system file and, optionally, sends the file to a printer. OFF stops spooling. OUT stops spooling and sends the file to your host computer’s standard (default) printer. Enter SPOOL with no clauses to list the current spooling status. If no file extension is given, the default extension, *.lst* or *.lis*, is used

TTI[TLE] [*printspec* [*text*|*variable*] ...]|[ON|OFF]

Places and formats a specified title at the top of each report page, or lists the current TTITLE definition. Use one of the following clauses in place of *printspec*:

COL *n*
S[KIP] [*n*]
TAB *n*
LE[FT]
CE[NTER]
R[IGHT]
BOLD
FORMAT *text*

Accessing Databases

Use the following commands to access and copy data between tables on different databases.

CONN[ECT] [{*logon*|/} [AS {SYSOPER|SYSDBA}]]

where *logon* requires the following syntax:

> *username*[/*password*][@*connect_identifier*]

Connects a given username to Oracle. If you omit *connect_identifier,* connects you to the default database. If you omit *username* and / or *password*, SQL*Plus prompts you for them. CONNECT followed by a slash (/) connects you using a default (OPS$) logon.

DISC[ONNECT]

Commits pending changes to the database and logs the current username off Oracle, but does not exit SQL*Plus.

COPY {FROM *database* | TO *database* | FROM *database* TO *database*}
{APPEND|CREATE|INSERT|REPLACE} *destination_table*[(*column*, *column*, *column*, ...)]
USING *query*

where *database* has the following syntax:

> *username*[/*password*]@*connect_identifier*

Copies data from one Oracle *database* to a table in another. APPEND, CREATE, INSERT or REPLACE specifies how COPY treats the existing copy of the destination table (if it exists). USING *query* identifies the source table and determines which rows and columns COPY copies from it.

PASSW[ORD] [*username*]

Allows you to change a password without echoing the password on the terminal.

Miscellaneous

ARCHIVE LOG {LIST|STOP}|{START|NEXT|ALL|*integer*}[TO *destination*]

Starts or stops automatic archiving on online redo log files, manually (explicitly) archives specified redo log files, or displays information about redo log files.

DESC[RIBE] [*schema*.]*object*[*@connect_identifier*]

Lists the column definitions for the specified table, view or synonym or the specifications for the specified function or procedure.

RECOVER {*general* | *managed* | END BACKUP}

where the *general* clause has the following syntax:

[AUTOMATIC] [FROM *location*]
{ {*full_database_recovery* | *partial_database_recovery* |LOGFILE *filename*}
[{TEST | ALLOW integer CORRUPTION } [TEST | ALLOW integer CORRUPTION]...]
|CONTINUE [DEFAULT]|CANCEL}

where the *full_database_recovery* clause has the following syntax:

[STANDBY] DATABASE
[{UNTIL {CANCEL | TIME *date* | CHANGE *integer*} | USING BACKUP CONTROLFILE}
[UNTIL {CANCEL | TIME *date* | CHANGE *integer*} | USING BACKUP CONTROLFILE]...]

where the *partial_database_recovery* clause has the following syntax:

{TABLESPACE *tablespace* [, *tablespace*]... | DATAFILE *datafilename* [, *datafilename*]...
| STANDBY
{TABLESPACE *tablespace* [, *tablespace*]... | DATAFILE *datafilename* [, *datafilename*]...}
UNTIL [CONSISTENT] [WITH] CONTROLFILE }

where the *managed* clause has the following syntax:

MANAGED STANDBY DATABASE
[{NODELAY | [TIMEOUT] *integer* | CANCEL [IMMEDIATE] [NOWAIT]}
| [DISCONNECT [FROM SESSION]] [FINISH [NOWAIT]]]

Performs media recovery on one or more tablespaces, one or more datafiles, or the entire database.

SET *system_variable value*

Sets a system variable to alter the SQL*Plus environment for your current session, such as turning on HTML formatting, setting the display width for NUMBER data, or setting the number of lines per page. Enter a system variable followed by a value as shown below:

```
APPI[NFO]{ON|OFF|text}
ARRAY[SIZE] {15|n}
AUTO[COMMIT] {ON|OFF|IMM[EDIATE]|n}
AUTOP[RINT] {ON|OFF}
AUTORECOVERY {ON|OFF]
AUTOT[RACE] {ON|OFF|TRACE[ONLY]} [EXP[LAIN]] [STAT[ISTICS]]
BLO[CKTERMINATOR] {.|c}
CMDS[EP] {;|c|ON|OFF}
COLSEP {_|text}
COM[PATIBILITY] {V7|V8|NATIVE}
CON[CAT] {.|c|ON|OFF}
COPYC[OMMIT] {0|n}
COPYTYPECHECK {ON|OFF}
DEF[INE] {‘&’|c|ON|OFF}
DESCRIBE [DEPTH {1|n|ALL}][LINENUM {ON|OFF}][INDENT {ON|OFF}]
ECHO {ON|OFF}
EDITF[ILE] file_name[.ext]
EMB[EDDED] {ON|OFF}
ESC[APE] {\|c|ON|OFF}
FEED[BACK] {6|n|ON|OFF}
FLAGGER {OFF|ENTRY|INTERMED[IATE]|FULL}
FLU[SH] {ON|OFF}
HEA[DING] {ON|OFF}
HEADS[EP] {||c|ON|OFF}
INSTANCE [instance_path|LOCAL]
LIN[ESIZE] {80|n}
LOBOF[FSET] {n|1}
LOGSOURCE [pathname]
LONG {80|n}
LONGC[HUNKSIZE] {80|n}
MARK[UP] HTML [ON|OFF] [HEAD text] [BODY text] [TABLE text]
   [ENTMAP {ON|OFF}] [SPOOL {ON|OFF}] [PRE[FORMAT] {ON|OFF}]
NEWP[AGE] {1|n|NONE}
NULL text
NUMF[ORMAT] format
NUM[WIDTH] {10|n}
PAGES[IZE] {24|n}
PAU[SE] {ON|OFF|text}
RECSEP {WR[APPED]|EA[CH]|OFF}
```

RECSEPCHAR {_|*c*}
SERVEROUT[PUT] {ON|OFF} [SIZE *n*] [FOR[MAT] {WRA[PPED]
|WOR[D_WRAPPED]|TRU[NCATED]}]
SHIFT[INOUT] {VIS[IBLE]|INV[ISIBLE]}
SHOW[MODE] {ON|OFF}
SQLBL[ANKLINES] {ON|OFF}
SQLC[ASE] {MIX[ED]|LO[WER]|UP[PER]}
SQLCO[NTINUE] {> |*text*}
SQLN[UMBER] {ON|OFF}
SQLPLUSCOMPAT[IBILITY] {x.y[.z]}
SQLPRE[FIX] {#|*c*}
SQLP[ROMPT] {SQL>|*text*}
SQLT[ERMINATOR] {;|*c*|ON|OFF}
SUF[FIX] {SQL|*text*}
TAB {ON|OFF}
TERM[OUT] {ON|OFF}
TI[ME] {ON|OFF}
TIMI[NG] {ON|OFF}
TRIM[OUT] {ON|OFF}
TRIMS[POOL] {ON|OFF}
UND[ERLINE] {-|c|ON|OFF}
VER[IFY] {ON|OFF}
WRA[P] {ON|OFF}

SHO[W] [*option*]

Lists the value of a SQL*Plus system variable. Enter any system variable set by the SET command in place of *system_variable*. Use one of the following terms or clauses in place of *option*:

system_variable
ALL
BTI[TLE]
ERR[ORS] [{FUNCTION | PROCEDURE | PACKAGE | PACKAGE BODY | TRIGGER
| VIEW | TYPE | TYPE BODY | DIMENSION | JAVA CLASS} [*schema.*]*name*]
LNO
PARAMETERS [*parameter_name*]
PNO
REL[EASE]
REPF[OOTER]
REPH[EADER]
SGA
SPOO[L]
SQLCODE
TTI[TLE]
USER

SQL Command List

SQL commands were formerly documented in SQL*Plus documentation. You should now refer to the *Oracle9i SQL Reference* for full documentation of these commands and clauses. Below is a list of major SQL commands:

ALTER
ANALYZE
AUDIT
COMMENT
COMMIT
CREATE
DELETE
DROP
EXPLAIN
GRANT
INSERT
LOCK TABLE
NOAUDIT
RENAME
REVOKE
ROLLBACK
SAVEPOINT
SELECT
SET ROLE
SET TRANSACTION
TRUNCATE
UPDATE