SQL> COLUMN DEPTNO HEADING Department

SQL> COLUMN ENAME HEADING Employee

SQL> COLUMN SAL HEADING Salary

SQL> COLUMN COMM HEADING Commission

SQL> SELECT DEPTNO, ENAME, SAL, COMM

  2  FROM EMP

  3  WHERE JOB = 'SALESMAN';

SQL\*Plus displays the following output:

Department Employee       Salary Commission

---------- ---------- ---------- ----------

        30 ALLEN            1600        300

        30 WARD             1250        500

        30 MARTIN           1250       1400

        30 TURNER           1500          0

|  |
| --- |
| **Note:**  The new headings will remain in effect until you enter different headings, reset each column's format, or exit from SQL\*Plus. |

To change a column heading to two or more words, enclose the new heading in single or double quotation marks when you enter the COLUMN command. To display a column heading on more than one line, use a vertical bar (|) where you want to begin a new line. (You can use a character other than a vertical bar by changing the setting of the HEADSEP variable of the SET command. See the [SET](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch8.htm#1001850) command in [Chapter 8](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch8.htm#1010735) for more information.) 

***Example 4-2 Splitting a Column Heading***

To give the column ENAME the heading EMPLOYEE NAME and to split the new heading onto two lines, enter

SQL> COLUMN ENAME HEADING 'Employee|Name'

Now rerun the query with the slash (/) command:

SQL> /

SQL\*Plus displays the following output:

           Employee

Department Name           Salary Commission

---------- ---------- ---------- ----------

        30 ALLEN            1600        300

        30 WARD             1250        500

        30 MARTIN           1250       1400

        30 TURNER           1500          0

To change the character used to underline each column heading, set the UNDERLINE variable of the SET command to the desired character. 

***Example 4-3 Setting the Underline Character***

To change the character used to underline headings to an equal sign and rerun the query, enter the following commands:

SQL> SET UNDERLINE =

SQL> /

SQL\*Plus displays the following results:

           Employee

Department Name           Salary Commission

========== ========== ========== ==========

        30 ALLEN            1600        300

        30 WARD             1250        500

        30 MARTIN           1250       1400

        30 TURNER           1500          0

Now change the underline character back to a dash:

SQL> SET UNDERLINE '-'

|  |
| --- |
| **Note:**  You must enclose the dash in quotation marks; otherwise, SQL\*Plus interprets the dash as a hyphen indicating that you wish to continue the command on another line. |

**Formatting NUMBER Columns**

When displaying NUMBER columns, you can either accept the SQL\*Plus default display width or you can change it using the COLUMN command. The sections below describe the default display and how you can alter the default with the COLUMN command. 

**Default Display**

A NUMBER column's width equals the width of the heading or the width of the FORMAT plus one space for the sign, whichever is greater. If you do not explicitly use FORMAT, then the column's width will always be at least the value of SET NUMWIDTH. 

SQL\*Plus normally displays numbers with as many digits as are required for accuracy, up to a standard display width determined by the value of the NUMWIDTH variable of the SET command (normally 10). If a number is larger than the value of SET NUMWIDTH, SQL\*Plus rounds the number up or down to the maximum number of characters allowed. 

You can choose a different format for any NUMBER column by using a format model in a COLUMN command. A format model is a representation of the way you want the numbers in the column to appear, using 9's to represent digits. 

**Changing the Default Display**

The COLUMN command identifies the column you want to format and the model you want to use, as shown below:

COLUMN *column\_name* FORMAT *model*

Use format models to add commas, dollar signs, angle brackets (around negative values), and/or leading zeros to numbers in a given column. You can also round the values to a given number of decimal places, display minus signs to the right of negative values (instead of to the left), and display values in exponential notation. 

To use more than one format model for a single column, combine the desired models in one COLUMN command (see [Example 4-4](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch4.htm#1001255)). For a complete list of format models and further details, see the [COLUMN](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch8.htm#1012281) command in [Chapter 8](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch8.htm#1010735). 

***Example 4-4 Formatting a NUMBER Column***

To display SAL with a dollar sign, a comma, and the numeral zero instead of a blank for any zero values, enter the following command:

SQL> COLUMN SAL FORMAT $99,990

Now rerun the current query:

SQL> /

SQL\*Plus displays the following output:

           Employee

Department Name          Salary  Commission

---------- ---------- ---------  ----------

        30 ALLEN         $1,600         300

        30 WARD          $1,250         500

        30 MARTIN        $1,250        1400

        30 TURNER        $1,500           0

Use a zero in your format model, as shown above, when you use other formats such as a dollar sign and wish to display a zero in place of a blank for zero values. 

|  |
| --- |
| **Note:**  The format model will stay in effect until you enter a new one, reset the column's format with  COLUMN colname CLEAR  or exit from SQL\*Plus. |

**Formatting Datatypes**

When displaying datatypes, you can either accept the SQL\*Plus default display width or you can change it using the COLUMN command. Datatypes, in this manual, include the following variables:

* CHAR
* NCHAR
* VARCHAR2 (VARCHAR)
* NVARCHAR2 (NCHAR VARYING)
* DATE
* LONG
* CLOB
* NCLOB

|  |
| --- |
| **Note:**  The NCHAR, NVARCHAR2 (NCHAR VARYING), CLOB and NCLOB datatypes require Oracle8 or higher. |

**Default Display**

The default width of datatype columns is the width of the column in the database. 

The default width and format of unformatted DATE columns in SQL\*Plus is derived from the NLS parameters in effect. Otherwise, the default format width is A9. For more information on formatting DATE columns, see the FORMAT clause of the [COLUMN](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch8.htm#1012281) command in [Chapter 8](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch8.htm#1010735). 

|  |
| --- |
| **Note:**  The default justification for datatypes is left justification. |

**Changing the Default Display**

You can change the displayed width of a datatype or DATE, by using the COLUMN command with a format model consisting of the letter A (for alphanumeric) followed by a number representing the width of the column in characters. 

Within the COLUMN command, identify the column you want to format and the model you want to use:

COLUMN *column\_name* FORMAT *model*

If you specify a width shorter than the column heading, SQL\*Plus truncates the heading. If you specify a width for a LONG, CLOB, or NCLOB column, SQL\*Plus uses the LONGCHUNKSIZE or the specified width, whichever is smaller, as the column width. See the [COLUMN](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch8.htm#1012281) command in [Chapter 8](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch8.htm#1010735) for more details. 

***Example 4-5 Formatting a Character Column***

To set the width of the column ENAME to four characters and rerun the current query, enter

SQL> COLUMN ENAME FORMAT A4

SQL> /

SQL\*Plus displays the results:

           Empl

Department Name     Salary Commission

---------- ---- ---------- ----------

       30  ALLE     $1,600        300

           N

       30  WARD     $1,250        500

       30  MART     $1,250       1400

           IN

       30  TURN     $1,500          0

           ER

|  |
| --- |
| **Note:**  The format model will stay in effect until you enter a new one, reset the column's format with  COLUMN colname CLEAR  or exit from SQL\*Plus. |

Now return the column to its previous format:

SQL> COLUMN ENAME FORMAT A10

**Copying Column Display Attributes**

When you want to give more than one column the same display attributes, you can reduce the length of the commands you must enter by using the LIKE clause of the COLUMN command. The LIKE clause tells SQL\*Plus to copy the display attributes of a previously defined column to the new column, except for changes made by other clauses in the same command. 

***Example 4-6 Copying a Column's Display Attributes***

To give the column COMM the same display attributes you gave to SAL, but to specify a different heading, enter the following command:

SQL> COLUMN COMM LIKE SAL HEADING Bonus

Rerun the query:

SQL> /

SQL\*Plus displays the following output:

           Employee

Department Name           Salary      Bonus

---------- ---------- ---------- ----------

        30 ALLEN          $1,600       $300

        30 WARD           $1,250       $500

        30 MARTIN         $1,250     $1,400

        30 TURNER         $1,500         $0

**Listing and Resetting Column Display Attributes**

To list the current display attributes for a given column, use the COLUMN command followed by the column name only, as shown below:

COLUMN *column\_name*

To list the current display attributes for all columns, enter the COLUMN command with no column names or clauses after it:

COLUMN

To reset the display attributes for a column to their default values, use the CLEAR clause of the COLUMN command as shown below:

COLUMN *column\_name* CLEAR

To reset the attributes for all columns, use the COLUMNS clause of the CLEAR command. 

***Example 4-7 Resetting Column Display Attributes to their Defaults***

To reset all columns' display attributes to their default values, enter the following command:

SQL> CLEAR COLUMNS

columns cleared

**Suppressing and Restoring Column Display Attributes**

You can suppress and restore the display attributes you have given a specific column. To suppress a column's display attributes, enter a COLUMN command in the following form:

COLUMN *column\_name* OFF

The OFF clause tells SQL\*Plus to use the default display attributes for the column, but does not remove the attributes you have defined through the COLUMN command. To restore the attributes you defined through COLUMN, use the ON clause:

COLUMN *column\_name* ON

**Clarifying Your Report with Spacing and Summary Lines**

When you use an ORDER BY clause in your SQL SELECT command, rows with the same value in the ordered column (or expression) are displayed together in your output. You can make this output more useful to the user by using the SQL\*Plus BREAK and COMPUTE commands to create subsets of records and add space and/or summary lines after each subset. 

COMPUTE command functions are always executed in the sequence AVG, COUNT, MINIMUM, MAXIMUM, NUMBER, SUM, STD, VARIANCE, regardless of their order in the COMPUTE command. 

The column you specify in a BREAK command is called a *break column*. By including the break column in your ORDER BY clause, you create meaningful subsets of records in your output. You can then add formatting to the subsets within the same BREAK command, and add a summary line (containing totals, averages, and so on) by specifying the break column in a COMPUTE command. 

For example, the following query, without BREAK or COMPUTE commands,

SQL> SELECT DEPTNO, ENAME, SAL

  2  FROM EMP

  3  WHERE SAL < 2500

  4  ORDER BY DEPTNO;

produces the following unformatted results:

  DEPTNO  ENAME             SAL

--------  ----------  ---------

      10  CLARK            2450

      10  MILLER           1300

      20  SMITH             800

      20  ADAMS            1100

      30  ALLEN            1600

      30  JAMES             950

      30  TURNER           1500

      30  WARD             1250

      30  MARTIN           1250

9 rows selected.

To make this report more useful, you would use BREAK to establish DEPTNO as the break column. Through BREAK you could suppress duplicate values in DEPTNO and place blank lines or begin a new page between departments. You could use BREAK in conjunction with COMPUTE to calculate and print summary lines containing the total (and/or average, maximum, minimum, standard deviation, variance, or count of rows of) salary for each department and for all departments. 

**Inserting Space when a Break Column's Value Changes**

You can insert blank lines or begin a new page each time the value changes in the break column. To insert *n* blank lines, use the BREAK command in the following form:

BREAK ON *break\_column* SKIP *n*

To skip a page, use the command in this form:

BREAK ON *break\_column* SKIP PAGE

***Example 4-10 Inserting Space when a Break Column's Value Changes***

To place one blank line between departments, enter the following command:

SQL> BREAK ON DEPTNO SKIP 1

Now rerun the query:

SQL> /

SQL\*Plus displays the results:

    DEPTNO ENAME             SAL

---------- ----------- ---------

        10 CLARK            2450

           MILLER           1300

        20 SMITH             800

           ADAMS            1100

        30 ALLEN            1600

           JAMES             950

           TURNER           1500

           WARD             1250

           MARTIN           1250

9 rows selected.

**Inserting Space after Every Row**

You may wish to insert blank lines or a blank page after every row. To skip *n* lines after every row, use BREAK in the following form:

BREAK ON ROW SKIP *n*

To skip a page after every row, use

BREAK ON ROW SKIP PAGE

|  |
| --- |
| **Note:**  SKIP PAGE does not cause a physical page break unless you have also specified NEWPAGE 0. |

Now, to skip a page when the value of DEPTNO changes and one line when the value of JOB changes, enter the following command:

SQL> BREAK ON DEPTNO SKIP PAGE ON JOB SKIP 1

To show that SKIP PAGE has taken effect, create a TTITLE with a page number, enter

SQL> TTITLE COL 35 FORMAT 9 'Page:' SQL.PNO

Run the new query to see the results:

SQL> /

                                    Page: 1

    DEPTNO JOB       ENAME             SAL

---------- --------- ---------- ----------

        10 CLERK     MILLER            300

           MANAGER   CLARK            2450

                                    Page: 2

    DEPTNO JOB       ENAME             SAL

---------- --------- ---------- ----------

        20 CLERK     SMITH             800

                     ADAMS            1100

                                    Page: 3

    DEPTNO JOB       ENAME             SAL

---------- --------- ---------- ----------

        30 CLERK     JAMES             950

           SALESMAN  ALLEN            1600

                     TURNER           1500

                     WARD             1250

                     MARTIN           1250

9 rows selected.

**Listing and Removing Break Definitions**

Before continuing, turn off the top title display without changing its definition, using:

SQL> TTITLE OFF

You can list your current break definition by entering the BREAK command with no clauses:

BREAK

You can remove the current break definition by entering the CLEAR command with the BREAKS clause:

CLEAR BREAKS

You may wish to place the command CLEAR BREAKS at the beginning of every command file to ensure that previously entered BREAK commands will not affect queries you run in a given file. 

**Computing Summary Lines when a Break Column's Value Changes**

If you organize the rows of a report into subsets with the BREAK command, you can perform various computations on the rows in each subset. You do this with the functions of the SQL\*Plus COMPUTE command. Use the BREAK and COMPUTE commands together in the following forms:

BREAK ON *break\_column*

COMPUTE *function* LABEL *label\_name* OF *column column column*

... ON *break\_column*

You can include multiple break columns and actions, such as skipping lines in the BREAK command, as long as the column you name after ON in the COMPUTE command also appears after ON in the BREAK command. To include multiple break columns and actions in BREAK when using it in conjunction with COMPUTE, use these commands in the following forms:

BREAK ON *break\_column\_1* SKIP PAGE ON *break\_column\_2* SKIP 1

COMPUTE *function* LABEL *label\_name* OF *column column column*

... ON *break\_column\_2*

The COMPUTE command has no effect without a corresponding BREAK command. 

You can COMPUTE on NUMBER columns and, in certain cases, on all types of columns. For more information about the [COMPUTE](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch8.htm#1012747) command, see the "Command Reference" in [Chapter 8](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch8.htm#1010735) 

The following table lists compute functions and their effects 

***Table 4-1 Compute Functions***

| **Function** | **Effect** |
| --- | --- |
| SUM | Computes the sum of the values in the column. |
| MINIMUM | Computes the minimum value in the column. |
| MAXIMUM | Computes the maximum value in the column. |
| AVG | Computes the average of the values in the column. |
| STD | Computes the standard deviation of the values in the column. |
| VARIANCE | Computes the variance of the values in the column. |
| COUNT | Computes the number of non-null values in the column. |
| NUMBER | Computes the number of rows in the column. |

|  |
| --- |
|  |

The function you specify in the COMPUTE command applies to all columns you enter after OF and before ON. The computed values print on a separate line when the value of the ordered column changes. 

Labels for ON REPORT and ON ROW computations appear in the first column; otherwise, they appear in the column specified in the ON clause. 

You can change the compute label by using COMPUTE LABEL. If you do not define a label for the computed value, SQL\*Plus prints the unabbreviated function keyword. 

The compute label can be suppressed by using the NOPRINT option of the COLUMN command on the break column. See the [COMPUTE](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch8.htm#1012747) command in [Chapter 8](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch8.htm#1010735) for more details. 

SQL> COLUMN DUMMY NOPRINT

SQL> COMPUTE SUM OF SAL ON DUMMY

SQL> BREAK ON DUMMY SKIP 1

SQL> SELECT DEPTNO DUMMY, DEPTNO, ENAME, SAL

  2  FROM EMP

  3  WHERE DEPTNO <= 20

  4  ORDER BY DEPTNO;

SQL\*Plus displays the following output:

    DEPTNO ENAME             SAL

---------- ---------- ----------

        10 KING             5000

        10 CLARK            2450

        10 MILLER           1300

                      ----------

                            8750

        20 JONES            2975

        20 FORD             3000

        20 SMITH             800

        20 SCOTT            3000

        20 ADAMS            1100

                      ----------

                           10875

8 rows selected.

To compute the salaries at the end of the report:

SQL> COLUMN DUMMY NOPRINT

SQL> COMPUTE SUM OF SAL ON DUMMY

SQL> BREAK ON DUMMY

SQL> SELECT NULL DUMMY, DEPTNO, ENAME, SAL

  2  FROM EMP

  3  WHERE DEPTNO <= 20

  4  ORDER BY DEPTNO;

SQL\*Plus displays the following output:

    DEPTNO ENAME             SAL

---------- ---------- ----------

        10 KING             5000

        10 CLARK            2450

        10 MILLER           1300

        20 JONES            2975

        20 FORD             3000

        20 SMITH             800

        20 SCOTT            3000

        20 ADAMS            1100

                      ----------

                           19625

8 rows selected.

|  |
| --- |
| **Note:**  The format of the column SAL controls the appearance of the sum of SAL, as well as the individual values of SAL. When you establish the format of a NUMBER column, you must allow for the size of sums you will include in your report. |

**Listing and Removing COMPUTE Definitions**

You can list your current COMPUTE definitions by entering the COMPUTE command with no clauses:

COMPUTE

***Example 4-16 Removing COMPUTE Definitions***

To remove all COMPUTE definitions and the accompanying BREAK definition, enter the following commands:

SQL> CLEAR BREAKS

breaks cleared

SQL> CLEAR COMPUTES

computes cleared

You may wish to place the commands CLEAR BREAKS and CLEAR COMPUTES at the beginning of every command file to ensure that previously entered BREAK and COMPUTE commands will not affect queries you run in a given file. 

**Defining Page and Report Titles and Dimensions**

The word *page* refers to a screenful of information on your display or a page of a spooled (printed) report. You can place top and bottom titles on each page, set the number of lines per page, and determine the width of each line. 

The word *report* refers to the complete results of a query. You can also place headers and footers on each report and format them in the same way as top and bottom titles on pages. 

**Setting the Top and Bottom Titles and Headers and Footers**

As you have already seen, you can set a title to display at the top of each page of a report. You can also set a title to display at the bottom of each page. The TTITLE command defines the top title; the BTITLE command defines the bottom title. 

You can also set a header and footer for each report. The REPHEADER command defines the report header; the REPFOOTER command defines the report footer. 

A TTITLE, BTITLE, REPHEADER or REPFOOTER command consists of the command name followed by one or more clauses specifying a position or format and a CHAR value you wish to place in that position or give that format. You can include multiple sets of clauses and CHAR values:

TTITLE *position\_clause(s) char\_value position\_clause(s)* *char\_value* ...

BTITLE *position\_clause(s) char\_value position\_clause(s)* *char\_value* ...

REPHEADER *position\_clause(s) char\_value position\_clause(s)* *char\_value* ...

REPFOOTER *position\_clause(s) char\_value position\_clause(s)* *char\_value* ...

For descriptions of all TTITLE, BTITLE, REPHEADER and REPFOOTER clauses, see the [TTITLE](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch8.htm#1003070) and [REPHEADER](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch8.htm#1001637) commands in [Chapter 8](https://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch8.htm#1010735). 

***Example 4-17 Placing a Top and Bottom Title on a Page***

To put titles at the top and bottom of each page of a report, enter

SQL> TTITLE CENTER -

>    'ACME WIDGET SALES DEPARTMENT PERSONNEL REPORT'

SQL> BTITLE CENTER 'COMPANY CONFIDENTIAL'

Now run the current query:

SQL> /

SQL\*Plus displays the following output:

              ACME WIDGET SALES DEPARTMENT PERSONNEL REPORT

    DEPTNO ENAME             SAL

---------- ---------- ----------

        30 JAMES             950

        30 WARD             1250

        30 MARTIN           1250

        30 TURNER           1500

        30 ALLEN            1600

        30 BLAKE            2850

                             COMPANY CONFIDENTIAL

6 rows selected.

***Example 4-18 Placing a Header on a Report***

To put a report header on a separate page, and to center it, enter

SQL> REPHEADER PAGE CENTER 'ACME WIDGET'

Now run the current query:

SQL> /

SQL\*Plus displays the following output on page one

              ACME WIDGET SALES DEPARTMENT PERSONNEL REPORT

                               ACME WIDGET

                           COMPANY CONFIDENTIAL

and the following output on page two

              ACME WIDGET SALES DEPARTMENT PERSONNEL REPORT

    DEPTNO ENAME             SAL

---------- ---------- ----------

        30 JAMES             950

        30 WARD             1250

        30 MARTIN           1250

        30 TURNER           1500

        30 ALLEN            1600

        30 BLAKE            2850

                             COMPANY CONFIDENTIAL

6 rows selected.

To suppress the report header without changing its definition, enter

SQL> REPHEADER OFF

***Example 4-24 Setting Page Dimensions***

To set the page size to 66 lines, clear the screen (or advance the printer to a new sheet of paper) at the start of each page, and set the linesize to 32, enter the following commands:

SQL> SET PAGESIZE 66

SQL> SET NEWPAGE 0

SQL> SET LINESIZE 32

Now enter and run the following commands to see the results:

SQL> TTITLE CENTER 'ACME WIDGET PERSONNEL REPORT' SKIP 1 -

>  CENTER '10-JAN-99' SKIP 2

SQL> COLUMN DEPTNO HEADING DEPARTMENT

SQL> COLUMN ENAME HEADING EMPLOYEE

SQL> COLUMN SAL FORMAT $99,999 HEADING SALARY

SQL> SELECT DEPTNO, ENAME, SAL

  2  FROM EMP

  3  ORDER BY DEPTNO;

SQL\*Plus displays a formfeed followed by the query results:

  ACME WIDGET PERSONNEL REPORT

              10-JAN-99

DEPARTMENT EMPLOYEE       SALARY

---------- ---------- ----------

        10 CLARK          $2,450

        10 KING           $5,000

        10 MILLER         $1,300

        20 SMITH            $800

        20 ADAMS          $1,100

        20 FORD           $3,000

        20 SCOTT          $3,000

        20 JONES          $2,975

        30 ALLEN          $1,600

        30 BLAKE          $2,850

        30 MARTIN         $1,250

        30 JAMES            $950

        30 TURNER         $1,500

        30 WARD           $1,250

14 rows selected.

Now reset PAGESIZE, NEWPAGE, and LINESIZE to their default values:

SQL> SET PAGESIZE 24

SQL> SET NEWPAGE 1

SQL> SET LINESIZE 80

To list the current values of these variables, use the SHOW command:

SQL> SHOW PAGESIZE

pagesize 24

SQL> SHOW NEWPAGE

newpage 1

SQL> SHOW LINESIZE

linesize 80

Through the SQL\*Plus command SPOOL, you can store your query results in a file or print them on your computer's default printer. 