

SQL*Plus Formatting and Documentation Notes

You may easily become confused about formatting columns in SQL*Plus with the COLUMN command. The options for this are numerous and are described in great detail in the "SQL*Plus User's Guide and Reference," which is available online by typing 'sqlplusref' in the 'crldata' window. Take a close look at the section called "Formatting Columns" in Chapter 4.

While this documentation can be a little hard to figure out, anything you ever wanted to know (and plenty you didn't) is in there. I also recommend Chapter 7 ("SQL Statements") in the "Oracle8i SQL Reference." This is the definitive reference on the SQL statements that Oracle 8i supports. In particular, the section on the SELECT statement gives an exhaustive discussion of what you can do in a SELECT statement.

I should point out the distinction between SQL statements and SQL*Plus commands. SQL*Plus has a number of commands it uses to control your session and format your results. SQL*Plus commands we've encountered include:

```
SET LINESIZE linewidth
SET PAUSE text
COLUMN colname FORMAT colformat
DESCRIBE tablename
EXIT
```

Note that SQL*Plus commands do NOT need to be terminated by a semicolon.

SQL statements, alternatively, are commands that are processed by the SQL database engine on the remote server. SQL statements we've encountered include:

```
SELECT
CREATE TABLE
DROP TABLE
```

Later, we'll work with the SQL statements INSERT, DELETE, UPDATE, CREATE VIEW, and DROP VIEW. COMMIT and ROLLBACK may also come into play. SQL statements should be terminated with a semicolon at the end of a line OR a forward slash (/) on a line by itself.

Lastly, I should mention something about type conversions. Oracle tries to be clever about converting numbers into text strings and vice versa. Oracle may SILENTLY convert a number into a string that looks like a number but is actually a sequence of text characters that just happens to be digits if it thinks it is necessary. Similarly, it may silently convert a string of digits into a bona-fide number if it sees fit. An example of where this silent conversion occurs is with the string concatenation operator (the two

vertical bars, ||). The PARCEL table has two numeric columns, PID and WPB. The concatenation operator works only on strings, however. So, let's try

```
SELECT PID || WPB  
FROM PARCEL;
```

Oracle converts PID and WPB from numbers to strings before applying the concatenation operator (which only understands strings). The concatenation operator combines these two newly-made strings and produces a string as its result. Hence, any formatting on this value must be done as a string, not as a number, **EVEN THOUGH THE STRING LOOKS JUST LIKE A NUMBER**. This issue is important, because after you work with databases and GIS for awhile, you will encounter many instances of strings that look like numbers. Sometimes this is useful, sometimes a hindrance, but it always requires your attention.

Let's say we want to format a string column, S, to display 15 characters. We can use the following SQL*Plus command:

```
COLUMN S FORMAT A15
```

Henceforward, any time in this session that we SELECT a column called S, it will appear with 15 characters. This will also apply to any column alias named S that we use for an expression. Thus, the SQL statement

```
SELECT PID || WPB S  
FROM PARCEL;
```

assigns the column alias S to the expression 'PID || WPB', and hence we will get the following result:

```
S  
-----  
101224  
109292  
103112  
104267  
191209  
324342  
101265  
106234  
108453  
123287  
109381  
110250  
183236  
121550  
142231  
103276  
131283  
145229  
127653  
129543
```

Note that the header for column S is 15 characters wide. A 15-column numeric format such as 999999999999990 will not work here, because it applies only to numbers, not strings. Please consult the SQL*Plus reference above for detailed information on numeric formats.

N.B.: To assign a format to an expression in your SELECT list, you **MUST** use a column alias (**S** in this example) and reference it in a COLUMN command.

The Athena command 'sqlhelp' can be used in the 'crldata' window to bring up the "Oracle8i SQL Reference" mentioned earlier. Chapter 4 of this manual, "Operators, Functions, Expressions, and Conditions" contains a lot of useful information about the functions that are available in SQL statements in Oracle. These functions offer a lot of power, but many of them are unique to Oracle and cannot be used with other database systems.
