**CS 60 Project 8 Anonymous Oracle PL/SQL Programs**

Due Monday December 11 for Section 4502

Due Monday December 12 for Section 4127

In a script named **CS60\_8\_*Lastname\_Firstname*.SQL**, write, debug, and execute anonymous programs that

/\* Step 1. In the Declaration section, stores SYSDATE in a symbolic constant named con\_today and stores your full name in a variable named var\_fullname. Then in the executable section that begins with the keyword BEGIN, use the procedure dbms\_output.put\_line(*string)* to output the variable and separately output the symbolic constant. Just after the BEGIN, include dbms\_output.enable; \*/

/\* Step 2a. Create a table named LFM\_Course(where LFM are your initials) and insert the three rows below:

|  |  |
| --- | --- |
| Course\_Number | Course\_Name |
| CS60 | **Database Concepts and Applications** |
| CS61 | Microsoft SQL Server Database |
| CS65 | Oracle Programming |

\*/

/\* Step 2b. In an anonymous program, declare a variable named var\_Course\_Name with datatype varchar2(40) and another variable named var\_rowcount with datatype number(1). In the executable section, from the table select the Course\_Name for Course\_Number CS60 (a string) into the variable you declared above. Then output the variable to the screen. From the table, select the count(\*) into the variable var\_rowcount that you declared above. Then output this total number of rows (which will be 3) to the screen.

In a PL/SQL program, you cannot include an ordinary select, but you can select a value into a variable, then output the variable with dbms\_output.put\_line(*string).*  If you want to output a number to the screen, convert it into a string using the function TO\_CHAR(*number*), then output the string to the screen. \*/

/\* Step 2c. Drop your table. \*/

/\* Step 3a. Copy the anonymous program in Step 1 and remove all references to the symbolic constant con\_today and its output. In your declaration section,declare variables var\_reversed\_fullname with datatype varchar2(30) and var\_letter with datatype char(1).

Below the output of your name in your executable section, code a FOR loop that stores your full name in reverse order in the variable var\_reversed\_fullname. The loop moves the last letter of your var\_fullname into the first character of your var\_reversed\_fullname, then the next-to-last letter into the 2nd character of your variable, and so on until the first letter of your var\_fullname is the last letter stored in var\_reversed\_fullname. Then output the reversed name. A sketch may help you plan your program.

You’ll need to use **concatenation** (shown on page 218 so you don’t lose the part of the string you’d built up in earlier cycles of the loop) and an **assignment statement** (page 217) to build up your reversed name. Oracle has a built-in function named REVERSE to reverse a string, but don’t use this function.

Two built-in functions will be useful:

1. Function LENGTH(*string*) returns the length of the string as a number. For example, LENGTH('Harold') returns 6. With a variable, LENGTH(var\_fullname) returns the number of characters stored in that variable.
2. Function SUBSTR(*string, number of the beginning character, number of characters*) pulls a substring out of the *string*. The substring begins at the *number of the beginning character*, and it pulls off the *number of characters* in the 3rd argument. For example, SUBSTR('H**aro**ld', 2, 3) returns the substring 'aro'.

Copy your script to Zeus/data/rogler\_harold/CS60 Database Concepts and Applications.