Homework Week #7
PL/SQL Virtual Training

1. First some questions about packages, what they can contain, what they are made up of and how we find out which ones we have access to in our accounts.

   A. Name at least four objects that can be put into a package.

   B. Name the two components of a package and explain the difference between them.

   C. Which of the two components must be created first, and why?

   D. Write a query against a Data Dictionary view to show you a list of packages you own.

2. Create the specification for the check_emp_pkg which you studied in this lesson. The specification should declare a constant and two procedures, as follows:

   g_max_length_of_service, datatype NUMBER, initialized to 100

   chk_hiredate with one input parameter having the same datatype as employees.hire_date

   chk_dept_mgr with two input parameters having the same datatypes as employees.employee_id and employees.manager_id.

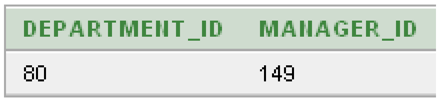
3. Create the package body for check_emp_pkg. Remember that the names and parameters of the procedures in the body must be identical to those in the specification, or the body will not compile.

   The code for chk_hiredate should RAISE_APPLICATION_ERROR if the employee was hired more than 100 years ago (hint: use MONTHS_BETWEEN, comparing with g_max_length_of_service * 12).

   The second procedure, chk_dept_mgr, accepts two input parameters: an employee_id and a manager_id. The code should find the manager of the employee's department and check whether this manager has the same manager_id as the second parameter. If the manager_id is the same, display a suitable "success" message; if they are different, raise an application error. Include an exception handler for NO_DATA_FOUND.

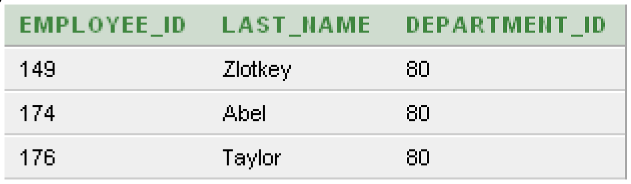
The following sample data from the employees and departments tables may help you:

Departments:



DEPARTMENT_ID	MANAGER_ID
80	149

Employees:



EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
149	Zlotkey	80
174	Abel	80
176	Taylor	80

Passing parameters (174,149) would be successful, while (174,176) would raise an error.

4. Procedure chk_hiredate:
   A. Test the chk_hiredate procedure using input value 17-Jan-87 (it should succeed).

   B. Test the chk_dept_mgr procedure twice using input values (174,149) and (174,176). The first should succeed while the second should fail.

5. Now you want to modify the package so that the chk_dept_mgr procedure displays a different error message if the two manager_ids are different. What do you need to recreate: the Specification, the Body, or both? Make the change and test it again, using (174,176) as before.

6. Create a package called overload. The package should contain three procedures all called what_am_i. The first procedure should accept a VARCHAR2 as an IN parameter, the second a NUMBER and the third a DATE. Each procedure should display a simple message to show which datatype was passed to it. For example, the first procedure could display "Here I am a Varchar2". Save your work for later. When you are done, describe the package.

7. Test the overload package by calling it and passing in a character string, a number and a date respectively. You should see the different messages returned. – Note: you must use the TO_DATE function to pass a value in as a date.

8. Now modify the overload package to have a fourth procedure in it, again called what_am_i, accepting two parameters: p_in NUMBER and p_in2 VARCHAR2. Test the new procedure to verify that it works.

9. Modify the overload package specification again to add a fifth procedure called what_am_i with the same two IN parameters as the fourth procedure. Try to recreate the specification. What happens and why?

10. Suppose you write a packaged function that returns a BOOLEAN value of TRUE or FALSE. Does the BOOLEAN limit the possible places from which you can call the packaged function?