

Home assignment #6  
Due date 6/3 5:30PM

Q1. (15 points)

In this question, you are required to create a function that will convert temperature from Celsius to Fahrenheit. In a PL/SQL block, you will call this function with a provided temperature in Celsius, say 21 degree, the program will print out the degree equivalent in Fahrenheit.

The formula: Fahrenheit degree = (degree in Celsius) \* 1.8 + 32.

Q2. (15 points)

This question is based on table employees.

Create a function called hwf2. That function has one IN parameter, which has the same data type as employee\_id. The function uses the employee\_id to retrieve the employee's salary and commission\_pct, then calculates/returns the bonus according the rules as below:

bonus = 500, if total income is equal or less than \$10,000;  
bonus = 100, if total income is more than \$10,000.  
the total income = salary \* ( 1 + NVL (commission\_pct, 0) )

In the PL/SQL block, your program will call the function two times, first time with ID = 170, second time ID = 160; prints out the value of bonus returned by the function.

Q3A. Function returns a data type of record. This question is based on table employees.  
(20 points).

Write a PL/SQL block, in that program, define a data type of RECORD named dept\_mgr, that has the fields of

Department ID,  
The manager ID of this department (notice, manager ID's are in the same domain of employee ID's in the employees table),  
The last name of the manager,  
The number of employees belong to that department.

Then we define a nested function named F6, that function F6 will accept the department ID as IN parameter, returns a record of type dept\_mgr (just defined above).

In the executable section, the program will invoke this function F6 for all the departments, print out the related info from the records.

Q3B. Function overload. (15 points) refer to page 9, 2.7 note7b\_functions.

This question is based on your code of Q3A. We will define the same function F6 as in Q3A, the only difference is that, the IN parameter will be the department name rather department ID.

In the executable section, the program will invoke this function F6 (this time, using department name as IN parameter) for all the departments, print out the related info from the records.

Q3C (0 point), Exercise for next week. Modify your codes from Q3A and Q3B, define a package call the func\_rec. In the package specification, we declare the Type dept\_mgr as record, then declare the F6 function twice, one uses department ID as IN parameter, the second uses department name as IN parameter. In the package body, we define the F6 programs. Then we can write a PL/SQL block to call the F6 twice, one by ID, another by name. Basically, this exercise is an editor job. Have fun.

Q4. (15 points) Trigger.

This question is based on table employees. To keep the history of the change of the salary of each employee, we will create a table that includes information (columns) of employee\_id, salary before and after change, who/when did the change, etc. We need to create a trigger for the event of any update of the salary (not include insert, delete). Once an update/change of salary occurs, the trigger will add one record into that log table.

Once created the trigger, write some update command to test your trigger. such as:

```
UPDATE employees set salary = salary +200 where employee_id = 105;
```

Then look at the log table, there should be one more row after that update.

Q5. Trigger (20 points)

This question is based on the table departments.

To monitor the table of departments, we will create a table called access\_dept\_log and a row-level triggers called access\_dept. Whenever there is a command of insert, or delete, or update/change on column of manager\_id, the trigger will be fired, and add one row into a table called access\_dept\_log.

The record should contain the information about who is the login user, when that happens, information related to the affected record.

Once you have successfully built the trigger, you are required to run a few suggested commands listed below.

```
Select * from departments;
SELECT * from access_dept_log ;
INSERT INTO departments VALUES (999, 'FAKE', 161, 1700);
UPDATE departments set manager_id = 203 WHERE department_id = 70;
DELETE departments where department_id = '999';
SELECT * from access_dept_log ;
rollback;
```