Home assignment #5 (352 &452),
Due date 5/27 5:30PM (points = 100);

Q1. (15 points).
Define a user-defined record type called Emp_phone that has three components: last_name, first_name and phone. You program will define their data types (such as varchar2) correspondently.
In a PL/SQL anonymous block, from table employees, your program will retrieve the info of the employee whose ID is 160, then insert his last name, first name and phone number into that user-defined record. Then your program will print out the contents of this record.

Q2. (20 points) In this question, you will do the following requirements/steps:
(If needed, there are some hints at the end of this file)

(a) Create a table as below:

```
-- DROP TABLE PHONEBOOK; if not first time run this command
CREATE TABLE PHONEBOOK(
    Last_name   VARCHAR2(20),
    First_name  VARCHAR2(20),
    Area_code   VARCHAR2(3),
    Prefix      VARCHAR2(3),
    Num         VARCHAR2(4)
);
```

(b) Write a PL/SQL block. Define a user-define TYPE RECORD called phone_num, with five components similar/same to the columns in the table Phonebook in step (a).
(Of course, your program needs to declare a variable for the TYPE)

(c) Declare/define a cursor that will retrieve last_name, first_name, phone information from the table employees (only use the info from those employees who work in department ID 20, 80 and 90) (hint: where department_id in (20,80,90)).

(d) It will be helpful to have a quick look at all phone numbers in the Employees table. There are two kinds of phone numbers in the table employees, such as:

011.44.1644.429262
650.507.9876

If the phone number starts with 011, that is for international, your program will ignore these phone numbers;
Otherwise, the phone numbers are US domestic. Use function SUBSTR to extract the three parts from the phone number(s), each part correspondent to the field of the RECORD type - phone_num you just defined.

(e) Populate the record phone_num with the domestic phone information you get from the query as in step (c), then insert that record into the table phonebook.

(f) Using “select * from phonebook” to print out the result.

Q3. (20 points)

We will repeat the code that defines the record type of phone_num, in Q2 (b). In this question #3, we will define a nested table type which use the used-defined record phone_num as the element of that NT. Recall that we use NT_variable (index) to reference the NT element; now we need to use NT_variable (index).record_field to reference the element's field.

(1) As step 2 in Question 2, declare/define the record type phone_num with five fields.
(2) You need to define a NT type call NTphonebook, that uses phone_num as its element; then declare a NT variable v1.
(3) Same/similar as in question #2, retrieve data from table employees for those people who work in department 20, 80, 90. Populate this NY variable v1, and print out the data.

Q4. (15 points) Cursor Variable

Define a strong cursor variable type, name it as empcurtyp. Its return type is employees%ROWTYPE, based on table Employees.
Then declare a cursor variable of this type.

In your program, first open the cursor variable for a select statement as

```
SELECT *  FROM employees
   WHERE department_id = 100
   ORDER BY last_name;
```

print out the employees' full names.
(please pay attention, the Cursor-For loop does NOT work for cursor variable)

After this, open that cursor variable again, this time for the statement of

```
SELECT *  FROM employees
   WHERE department_id = 30
   ORDER BY last_name;
```

then print out the employees' full names.
Then close the cursor variable.

Q5. (15 points)

Write a PL/SQL block, in that program, define a procedure named hw5 (so this procedure is a procedure just lives in that block), that procedure will accept the department ID as IN parameter, send back the number of employees that belong to this department, and the average salary of these employee salary.
In the executable section, the program will invoke this procedure hw5, print out the number of employees that belong to this department, and the average salary of these employee salary.
that belong to department (ID) 100.

Q6. (15 points)
You may re-use some of codes in Q5.
In this question, we will create a procedure named hw5, that will accept the department ID as IN parameter, send back the number of employees that belong to this department, and the average salary of these employee salary.

In the PL/SQL block, the program will invoke this procedure hw5 for all the departments that have employee(s), print out the number of employees that belong to this department and the average salary o employees in that department.

(If needed, there are some hints at the end of this file)

Hint for Q2:
(assuming indx is the dummy index in the for loop)
For cases where number starts with 011,

```
IF  SUBSTR (indx.phone_number, 1, 3 ) = '011' then
    null;
```

For US phone numbers, these codes might be helpful:

```
variable_name.area_code := SUBSTR (indx.phone_number, 1, 3 );
variable_name.Prefix    := SUBSTR (indx.phone_number, 5, 3 );
variable_name.num       := SUBSTR (indx.phone_number, 9, 4 );
```

You may try this kind command to understand how to use function `SUBSTR`:

```
select SUBSTR ('abcdefg_12345', 1, 3) from dual;
```

You can change the number 1, 3 to find out how that function works.

You may use the method “Insert Records into Tables” (follow the sample) on notes6b, section 4, page 7.

You may check the result by SQL query,
`SELECT * FROM` phonebook;

Hint for Q6.
In the invoker program, you my use a cursor loop, each time the cursor get a department ID, then call the procedure hw5 to get the related info.
We need to exclude the employee that has no department assigned yet (department_id should not be null).