Chapter 6 Practice

Firstly create a new table called RETIRED_EMPS like in the Chapter 6.
Here is the code:

```
CREATE TABLE retired_emps (
empno       NUMBER(4),
ename       VARCHAR2(20),
job         VARCHAR2(20),
mgr         NUMBER(4),
hiredate    DATE,
leavedate   DATE,
sal         NUMBER(7,2),
comm        NUMBER(7,2),
deptno      NUMBER(4)   )
```

1) Write PL/SQL block that will be used to ADD row to the new table and then will inspect the content of the table. You will add 3 rows by providing valid employee numbers (who are about to retire). As retirement date always use "Yesterday's Date".

After running this block 3 times with Id's of (174,201,141) here is the table output:

EMPNO	ENAME	JOB	MGR	HIREDATE	LEAVEDATE	SAL	COMM	DEPTNO
174	Abel	SA_REP	149	11-MAY-96	28-SEP-09	11000	.3	80
201	Hartstein	MK_MAN	100	17-FEB-96	28-SEP-09	13000		20
141	Rajs	ST_CLERK	124	17-OCT-95	28-SEP-09	3500		50

2) Write a PL/SQL block that will be used to MODIFY the WHOLE row in the new table and then will inspect the content of the table. You will modify just one row by providing valid employee number. As retirement date use "Tomorrow's Date", the person to retire will get $5,000 on the top of his/her salary. If he/she is eligible for the commission, that value should be increased by 50%.

After providing value 174 here is the output.

Please provide the employee number: 174

EMPNO	ENAME	JOB	MGR	HIREDATE	LEAVEDATE	SAL	COMM	DEPTNO
174	Abel	SA_REP	149	11-MAY-96	30-SEP-09	1600	.45	80

						0		
201	Hartstein	MK_MAN	100	17-FEB-96	28-SEP-09	13000		20
141	Rajs	ST_CLERK	124	17-OCT-95	28-SEP-09	3500		50

3) Write a PL/SQL block that will use RECORD type in order to display message like shown:

Today is : 29-SEP-09
It is : TUESDAY
Time is: 15:07:13
PL/SQL procedure successfully completed.

4) Write a PL/SQL block that will use NESTED RECORD type of 2 components (Name and Address) in order to display message about Students's full name and complete address like shown below:

Enter a valid Student Number: 102

Full Name is : Fred Crocitto
Full Address is : 101-09 120th St. Richmond Hill NY 11419
PL/SQL procedure successfully completed.

5)) Write a PL/SQL block that will use PL/SQL (INDEX BY) table to display department names and their location numbers, when user provides just the First Letter of the department. Do NOT use an explicit cursor to solve this problem. Here is the output:

Please provide the First Letter of Department Name E

Executive in location 1700
PL/SQL procedure successfully completed.

Please provide the First Letter of Department Name S

ERROR at line 1:
ORA-01422: exact fetch returns more than requested number of rows
ORA-06512: at line 13

If we have MORE THAN ONE department starting on that letter, then the error happens and the ONLY WAY to solve that kind of problem is to use EXPLICIT CURSORS → See Chapter 7 Practice

ANSWERS

1)

```
SET  SERVEROUTPUT ON
SET  VERIFY OFF
ACCEPT  empnum  PROMPT 'Please provide the employee number: '
DECLARE
    emp_rec     employees%ROWTYPE;
BEGIN
    SELECT  * INTO  emp_rec
    FROM     employees
    WHERE  employee_id = &empnum;

    INSERT INTO  retired_emps  VALUES (emp_rec.employee_id, emp_rec.last_name,
                                emp_rec.job_id, emp_rec.manager_id, emp_rec.hire_date,
       SYSDATE -1,emp_rec.salary, emp_rec.commission_pct, emp_rec.department_id);
END;
/
SELECT * FROM retired_emps;
```

2)

```
SET  SERVEROUTPUT ON
SET  VERIFY OFF
ACCEPT  empnum  PROMPT 'Please provide the employee number: '
DECLARE
    oldemp_rec     retired_emps%ROWTYPE;
BEGIN
    SELECT  * INTO  oldemp_rec
    FROM    retired_emps
    WHERE  empno = &empnum;

   oldemp_rec.leavedate := SYSDATE +1;
   oldemp_rec.sal := oldemp_rec.sal + 5000;

   IF  oldemp_rec.comm IS NOT NULL THEN
   oldemp_rec.comm := oldemp_rec.comm*1.5;
   END  IF;

   UPDATE  retired_emps  SET ROW =  oldemp_rec
   WHERE   empno = &empnum;

END;
/
```

```
SELECT * FROM retired_emps;
```

3)

```
SET  SERVEROUTPUT ON
DECLARE
   TYPE   day_rec_type IS RECORD (
      curr_date       DATE,
      curr_day        VARCHAR(12),
      curr_time       VARCHAR2(8) := '00:00:00') ;
   day_rec            DAY_REC_TYPE;
BEGIN
    SELECT  TRUNC(sysdate), TO_CHAR(sysdate,'DAY'), TO_CHAR(sysdate,'HH24:MI:SS')
    INTO  day_rec
    FROM     dual;
DBMS_OUTPUT.PUT_LINE('Today is : ' || day_rec.curr_date);
DBMS_OUTPUT.PUT_LINE('It is : ' || day_rec.curr_day);
DBMS_OUTPUT.PUT_LINE('Time is: ' || day_rec.curr_time);
END;
```

4)

```
SET  SERVEROUTPUT ON
ACCEPT  studentnum PROMPT 'Enter a valid Student Number: '
SET  VERIFY OFF
DECLARE
```

```
    TYPE   name_type IS RECORD (
        fname  VARCHAR2(20),
        lname   VARCHAR2(20)    );
    TYPE   address_type IS RECORD (
        street  VARCHAR2(50),
        city     VARCHAR2(20),
        state   CHAR(2),
        zip      VARCHAR2(6));
    TYPE   person_type IS RECORD (
        name   NAME_TYPE,
        address   ADDRESS_TYPE);
  person_rec  PERSON_TYPE;
BEGIN
    SELECT first_name, last_name, street_address, city, state, z.zip
    INTO       person_rec.name.fname, person_rec.name.lname, person_rec.address.street,
               person_rec.address.city, person_rec.address.state, person_rec.address.zip
    FROM     Student  s , Zipcode  z
    WHERE    s.zip= z.zip
    AND       student_id = &studentnum;
  DBMS_OUTPUT.PUT_LINE('Full Name is : ' || person_rec.name.fname || ' ' ||
person_rec.name.lname);
   DBMS_OUTPUT.PUT_LINE('Full Address is : ' || person_rec.address.street || ' ' ||
person_rec.address.city || ' ' ||  person_rec.address.state || ' ' || person_rec.address.zip);
END;
```

Note: ***If you follow the style of Example 3 and provide in the INTO clause just the RECORD NAME like shown, you will end up with the error – for Nested Records you must specify EACH COMPONENT separately***

SELECT first_name, last_name, street_address, city, state, z.zip

INTO person_rec

```
INTO        person_rec
                *
```

ERROR at line 20:
ORA-06550: line 20, column 21:
PLS-00597: expression 'PERSON_REC' in the INTO list is of wrong type

5)

```
SET  SERVEROUTPUT ON

SET  VERIFY OFF

ACCEPT  dept  PROMPT 'Please provide the First Letter of Department Name: '

DECLARE

    TYPE   dept_tab_type IS TABLE OF departments%ROWTYPE

              INDEX BY BINARY_INTEGER;

     dept_tab    dept_tab_type;

     v_bound     NUMBER(3);

     a     NUMBER(3) := 0;

BEGIN

    SELECT  COUNT(*)  INTO  v_bound

    FROM     departments

    WHERE   department_name LIKE '&dept%' ;

   FOR  i  IN   1..v_bound LOOP

        a := a + 1;

        SELECT   * INTO dept_tab(a)

        FROM      departments

        WHERE   department_name LIKE '&dept%' ;
```

```
        DBMS_OUTPUT.PUT_LINE(dept_tab(a).department_name || ' in location ' ||
dept_tab(a).location_id);

    END LOOP;

 END;
```