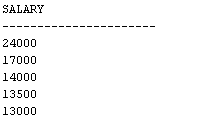
**Practice 7**

1. Create a PL/SQL block that determines the top *n* salaries of the employees.

1. Execute the lab\_07\_01.sql script to create a new table, top\_salaries, for storing the salaries of the employees.

CREATE TABLE top\_salaries(SALARY NUMBER(8,2));

1. In the declarative section, declare a variable v\_num of type NUMBER that holds a number n representing the number of top *n* earners from the employees table. For example, to view the top five salaries, enter 5. Declare another variable sal of type employees.salary. Declare a cursor, c\_emp\_cursor, that retrieves the salaries of employees in descending order.
2. In the executable section, open the loop and fetch top *n* salaries and insert them into top\_salaries table. You can use a simple loop to operate on the data. Also, try and use %ROWCOUNT and %FOUND attributes for the exit condition.
3. After inserting into the top\_salaries table, display the rows with a SELECT statement. The output shown represents the five highest salaries in the employees table.



1. Test a variety of special cases, such as v\_num = 0 or where v\_num is greater than the number of employees in the employees table. Empty the top\_salaries table after each test.

DECLARE

v\_num NUMBER := &n;

v\_sal TBLEMPLOYEES.SALARY%TYPE;

CURSOR c\_emp\_sal IS

SELECT SALARY FROM TBLEMPLOYEES

ORDER BY SALARY DESC;

BEGIN

OPEN c\_emp\_sal;

LOOP

FETCH c\_emp\_sal INTO v\_sal;

EXIT WHEN (NOT c\_emp\_sal%FOUND OR c\_emp\_sal%rowcount=v\_num+1);

INSERT INTO top\_salaries VALUES (v\_sal);

END LOOP;

CLOSE c\_emp\_sal;

END;

/

select \* from top\_salaries;

delete from top\_salaries;

2. Create a PL/SQL block that does the following:

1. In the declarative section, declare a variable v\_deptno of type NUMBER and assign a value that holds department ID.
2. Declare a cursor, c\_emp\_cursor, that retrieves the last\_name, salary, and manager\_id of the employees working in the department specified in v\_deptno.
3. In the executable section, use the cursor FOR loop to operate on the data retrieved. If the salary of the employee is less than 5,000 and if the manager ID is either 101 or 124, display the message <<*last\_name*>> Due for a raise. Otherwise, display the message <<*last\_name*>> Not due for a raise.
4. Test the PL/SQL block for the following cases:



SET SERVEROUTPUT ON;

DECLARE

v\_deptno NUMBER := &dept\_id;

CURSOR c\_emp\_cursor IS

SELECT LAST\_NAME , SALARY, MANAGER\_ID

FROM TBLEMPLOYEES

WHERE DEPARTMENT\_ID = v\_deptno;

BEGIN

FOR emp\_row in c\_emp\_cursor

LOOP

IF (emp\_row.SALARY<5000 AND emp\_row.MANAGER\_ID IN (101,124)) THEN

DBMS\_OUTPUT.PUT\_LINE(emp\_row.LAST\_NAME || ' Due for a raise');

ELSE

DBMS\_OUTPUT.PUT\_LINE(emp\_row.LAST\_NAME || ' Not Due for a raise');

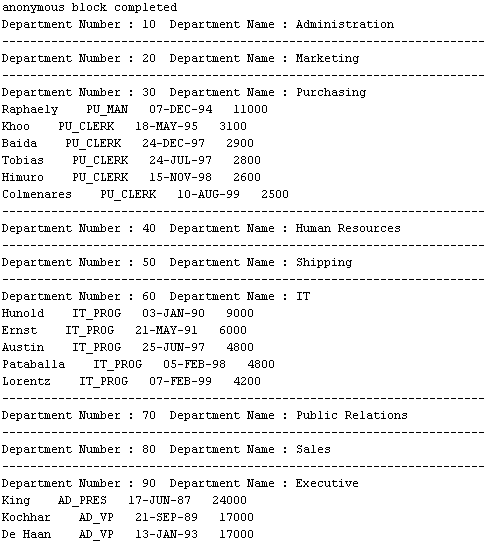
END IF;

END LOOP;

END;

/

1. Write a PL/SQL block that declares and uses cursors with parameters.  
   In a loop, use a cursor to retrieve the department number and the department name from the departments table for a department whose department\_id is less than 100. Pass the department number to another cursor as a parameter to retrieve from the employees table the details of employee last name, job, hire date, and salary of those employees whose employee\_id is less than 120 and who work in that department.
2. In the declarative section, declare a cursor dept\_cursor to retrieve department\_id and department\_name for those departments with department\_id less than 100. Order by department\_id.
3. Declare another cursor emp\_cursor that takes the department number as parameter and retrieves last\_name, job\_id, hire\_date, and salary of those employees whose employee\_id is less than 120 and who work in that department.
4. Declare variables to hold the values retrieved from each cursor. Use the %TYPE attribute while declaring variables.
5. Open the dept\_cursor, use a simple loop, and fetch values into the variables declared. Display the department number and department name.
6. For each department, open emp\_cursor by passing the current department number as a parameter. Start another loop and fetch the values of emp\_cursor into variables and print all the details retrieved from the employees table.  
   **Note:** You may want to print a lineafter you have displayed the details of each department.Use appropriate attributes for the exit condition. Also, determine whether a cursor is already open before opening the cursor.
7. Close all the loops and cursors, and then end the executable section. Execute the script.



SET SERVEROUTPUT ON;

DECLARE

CURSOR dept\_cursor IS

SELECT DEPARTMENT\_ID , DEPARTMENT\_NAME

FROM TBLDEPARTMENTS

WHERE DEPARTMENT\_ID < 100

ORDER BY DEPARTMENT\_ID;

CURSOR emp\_cursor(dept\_no NUMBER) IS

SELECT LAST\_NAME, JOB\_ID, HIRE\_DATE, SALARY

FROM TBLEMPLOYEES

WHERE EMPLOYEE\_ID <120 AND DEPARTMENT\_ID = dept\_no;

dept\_row dept\_cursor%ROWTYPE;

emp\_row emp\_cursor%ROWTYPE;

BEGIN

OPEN dept\_cursor;

LOOP

FETCH dept\_cursor INTO dept\_row;

EXIT WHEN dept\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('DEPARTMENT NUMBER: '||dept\_row.DEPARTMENT\_ID || ' DEPARTMENT NAME: '|| dept\_row.DEPARTMENT\_NAME);

OPEN emp\_cursor(dept\_row.DEPARTMENT\_ID);

LOOP

FETCH emp\_cursor INTO emp\_row;

EXIT WHEN emp\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(emp\_row.LAST\_NAME || ' '|| emp\_row.JOB\_ID||' '||emp\_row.HIRE\_DATE||' '||emp\_row.SALARY);

END LOOP;

CLOSE emp\_cursor;

DBMS\_OUTPUT.PUT\_LINE('----------------------------------------------------------------------------------------------------');

END LOOP;

CLOSE dept\_cursor;

END;

/

**Practice 6**

1. Write a PL/SQL block to print information about a given country.

* 1. Declare a PL/SQL record based on the structure of the countries table.
  2. Declare a variable v\_countryid. Assign CA to v\_countryid.
  3. In the declarative section, use the %ROWTYPE attribute and declare the v\_country\_record variable of type countries.
  4. In the executable section, get all the information from the countries table by using countryid. Display selected information about the country. Sample output is as follows.



* 1. You may want to execute and test the PL/SQL block for the countries with the IDs DE, UK, US.

SET SERVEROUTPUT ON;

DECLARE

v\_countryid TBLCOUNTRIES.COUNTRY\_ID%TYPE := 'CA';

v\_country\_record TBLCOUNTRIES%ROWTYPE;

BEGIN

SELECT \*

INTO v\_country\_record

FROM TBLCOUNTRIES

WHERE COUNTRY\_ID = v\_countryid;

DBMS\_OUTPUT.PUT\_LINE('Country Id: '||v\_country\_record.COUNTRY\_ID||' Country Name: '||v\_country\_record.COUNTRY\_NAME||' Region : '||v\_country\_record.REGION\_ID);

END;

2. Create a PL/SQL block to retrieve the name of some departments from the departments table and print each department name on the screen, incorporating an INDEX BY table. Save the script as lab\_06\_02\_soln.sql.

* 1. Declare an INDEX BY table dept\_table\_type of type departments.department\_name. Declare a variable my\_dept\_table of type dept\_table\_type to temporarily store the name of the departments.
  2. Declare two variables: loop\_count and deptno of type NUMBER. Assign 10 to loop\_count and 0 to deptno.
  3. Using a loop, retrieve the name of 10 departments and store the names in the INDEX BY table. Start with department\_id 10. Increase deptno by 10 for every iteration of the loop. The following table shows the department\_id for which you should retrieve the department\_name and store in the INDEX BY table.



* 1. Using another loop, retrieve the department names from the INDEX BY table and display them.
  2. Execute and save your script as lab\_06\_02\_soln.sql. The output is as follows:



SET SERVEROUTPUT ON;

DECLARE

TYPE dept\_table\_type IS TABLE OF TBLDEPARTMENTS.DEPARTMENT\_NAME%TYPE

INDEX BY PLS\_INTEGER;

my\_dept\_table dept\_table\_type;

loop\_count NUMBER := 10;

deptno NUMBER :=0;

BEGIN

FOR i IN 1..loop\_count

LOOP

deptno := deptno + 10;

SELECT DEPARTMENT\_NAME

INTO my\_dept\_table(i)

FROM TBLDEPARTMENTS

WHERE DEPARTMENT\_ID = deptno;

END LOOP;

FOR i IN 1..loop\_count

LOOP

DBMS\_OUTPUT.PUT\_LINE(my\_dept\_table(i));

END LOOP;

END;

3. Modify the block that you created in question 2 to retrieve all information about each department from the departments table and display the information. Use an INDEX BY table of records.

a. Load the lab\_06\_02\_soln.sql script.

b. You have declared the INDEX BY table to be of type departments.department\_name. Modify the declaration of the INDEX BY table, to temporarily store the number, name, and location of the departments. Use the %ROWTYPE attribute.

c. Modify the SELECT statement to retrieve all department information currently in the departments table and store it in the INDEX BY table.

d. Using another loop, retrieve the department information from the INDEX BY table and display the information. Sample output is as follows.

anonymous block completed

Department Number: 10 Department Name: Administration Manager Id: 200 Location Id: 1700

Department Number: 20 Department Name: Marketing Manager Id: 201 Location Id: 1800

Department Number: 30 Department Name: Purchasing Manager Id: 114 Location Id: 1700

Department Number: 40 Department Name: Human Resources Manager Id: 203 Location Id: 2400

Department Number: 50 Department Name: Shipping Manager Id: 121 Location Id: 1500

Department Number: 60 Department Name: IT Manager Id: 103 Location Id: 1400

Department Number: 70 Department Name: Public Relations Manager Id: 204 Location Id: 2700

Department Number: 80 Department Name: Sales Manager Id: 145 Location Id: 2500

Department Number: 90 Department Name: Executive Manager Id: 100 Location Id: 1700

Department Number: 100 Department Name: Finance Manager Id: 108 Location Id: 1700

SET SERVEROUTPUT ON;

DECLARE

TYPE dept\_table\_type IS TABLE OF TBLDEPARTMENTS%ROWTYPE

INDEX BY PLS\_INTEGER;

my\_dept\_table dept\_table\_type;

loop\_count NUMBER := 10;

deptno NUMBER :=0;

BEGIN

FOR i IN 1..loop\_count

LOOP

deptno := deptno + 10;

SELECT \*

INTO my\_dept\_table(i)

FROM TBLDEPARTMENTS

WHERE DEPARTMENT\_ID = deptno;

END LOOP;

FOR i IN 1..loop\_count

LOOP

DBMS\_OUTPUT.PUT\_LINE('Department Number: ' || my\_dept\_table(i).DEPARTMENT\_ID ||' Department Name: '||my\_dept\_table(i).DEPARTMENT\_NAME||' Manager Id: '||my\_dept\_table(i).MANAGER\_ID||' Location Id: '||my\_dept\_table(i).LOCATION\_ID);

END LOOP;

END;