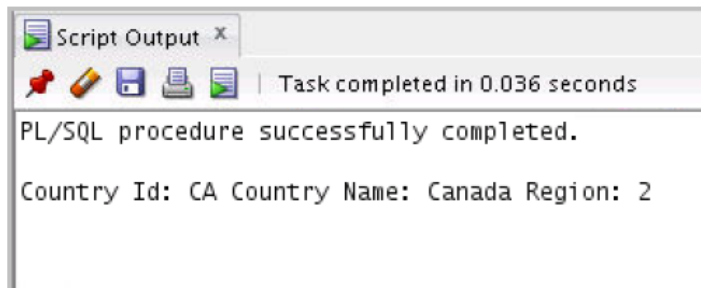


1.

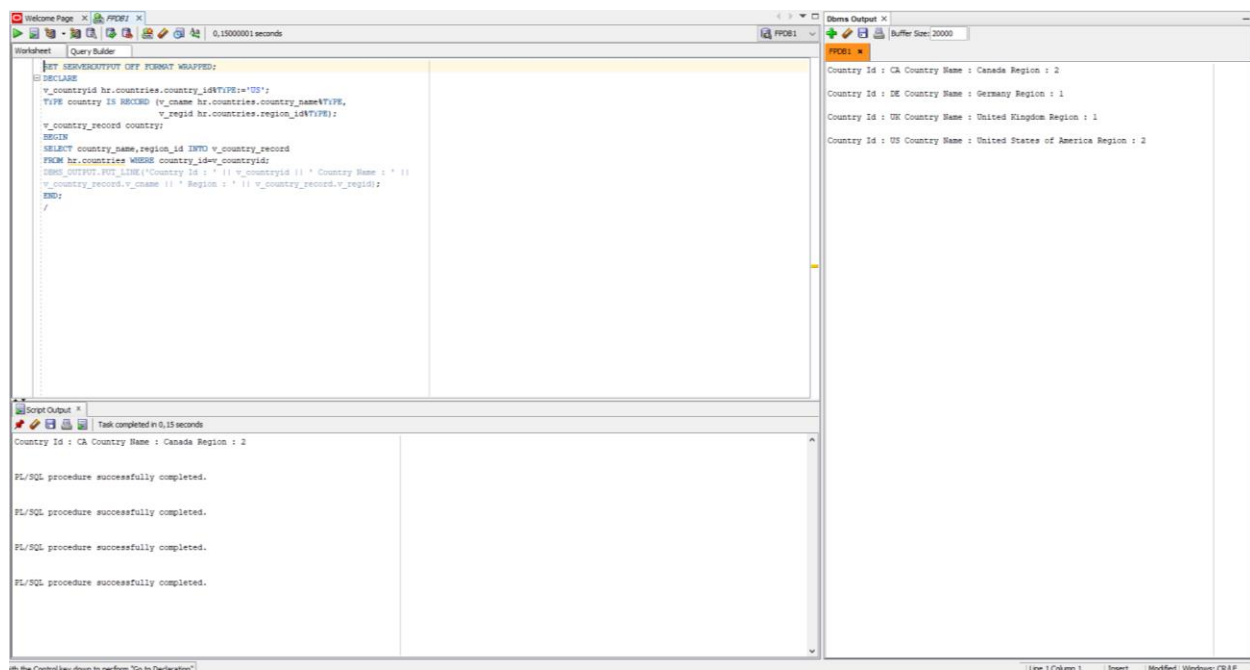
Write a PL/SQL block to print information about a given country.

- Declare a PL/SQL record based on the structure of the `COUNTRIES` table.
- Declare a variable `v_countryid`. Assign CA to `v_countryid`.
- In the declarative section, use the `%ROWTYPE` attribute and declare the `v_country_record` variable of type `countries`.
- In the executable section, get all the information from the `COUNTRIES` table by using `v_countryid`. Display selected information about the country. The sample output is as follows:



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

1-



2.

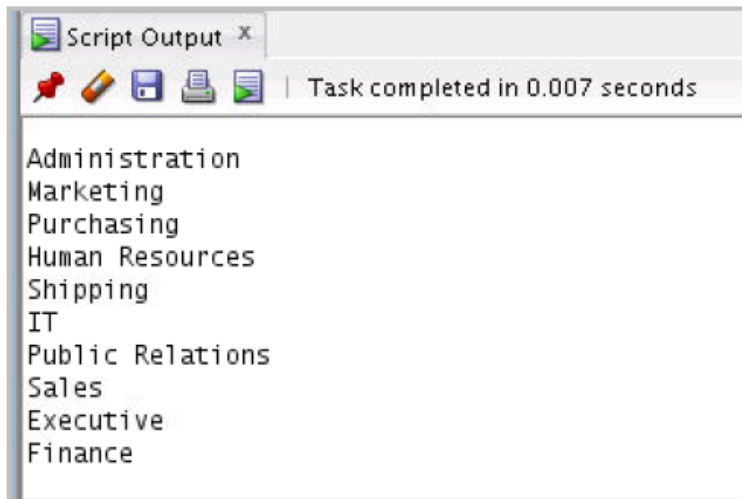
Create a PL/SQL block to retrieve the names of some departments from the `DEPARTMENTS` table and print each department name on the screen, incorporating an associative array. Save the script as `lab_07_02_soln.sql`.

- a. 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 names of the departments.
- b. Declare two variables: `f_loop_count` and `v_deptno` of type `NUMBER`. Assign 10 to `f_loop_count` and 0 to `v_deptno`.
- c. Using a loop, retrieve the names of 10 departments and store the names in the associative array. Start with `department_id` 10. Increase `v_deptno` by 10 for every loop iteration. The following table shows the `department_id` for which you should retrieve the `department_name`.

DEPARTMENT_ID	DEPARTMENT_NAME
10	Administration
20	Marketing
30	Purchasing
40	Human Resources

50	Shipping
60	IT
70	Public Relations
80	Sales
90	Executive
100	Finance

- d. Using another loop, retrieve the department names from the associative array and display them.
- e. Execute and save your script as `lab_07_02_soln.sql`. The output is as follows:



```

Administration
Marketing
Purchasing
Human Resources
Shipping
IT
Public Relations
Sales
Executive
Finance

```

2.1-

Worksheet | Query Builder | 0,083 seconds

```
SET SERVEROUTPUT ON FORMAT WRAPPED;
DECLARE
  TYPE dept_table_type IS TABLE OF hr.departments.department_name%TYPE
  INDEX BY simple_integer;
  my_dept_table dept_table_type;
  f_loop_count NUMBER := 10;
  v_deptno NUMBER := 0;
BEGIN
  LOOP
    v_deptno := v_deptno + f_loop_count;
    SELECT department_name INTO my_dept_table(v_deptno) FROM hr.departments WHERE department_id=v_deptno;
    DBMS_OUTPUT.PUT_LINE(my_dept_table(v_deptno));
    EXIT WHEN v_deptno=100;
  END LOOP;
END;
```

Script Output | Task completed in 0,083 seconds

Administration
Marketing
Purchasing
Human Resources
Shipping
IT
Public Relations
Sales
Executive
Finance

PL/SQL procedure successfully completed.

At the Control key down to perform "Go to Declaration"

Line 18 Column 2 | Insert | Modified | Windows: CRJF

2.2-

Worksheet | Query Builder | 0,06 seconds

```
SET SERVEROUTPUT ON FORMAT WRAPPED;
DECLARE
  TYPE rec IS TABLE OF hr.departments.department_name%TYPE
  INDEX BY simple_integer;
  v_rec rec;
  v_min hr.departments.department_id%TYPE;
  v_max hr.departments.department_id%TYPE;
  v_i NUMBER := 0;
BEGIN
  SELECT min(department_id),max(department_id) INTO v_min,v_max FROM hr.employees;
  LOOP
    v_i:=v_i + 10;
    SELECT department_name INTO v_rec(v_i) FROM hr.departments WHERE department_id=v_i;
    EXIT WHEN v_i=v_max;
    DBMS_OUTPUT.PUT_LINE(v_rec(v_i));
  END LOOP;
END;
```

Script Output | Task completed in 0,06 seconds

Administration
Marketing
Purchasing
Human Resources
Shipping
IT
Public Relations
Sales
Executive
Finance

PL/SQL procedure successfully completed.

At the Control key down to perform "Go to Declaration"

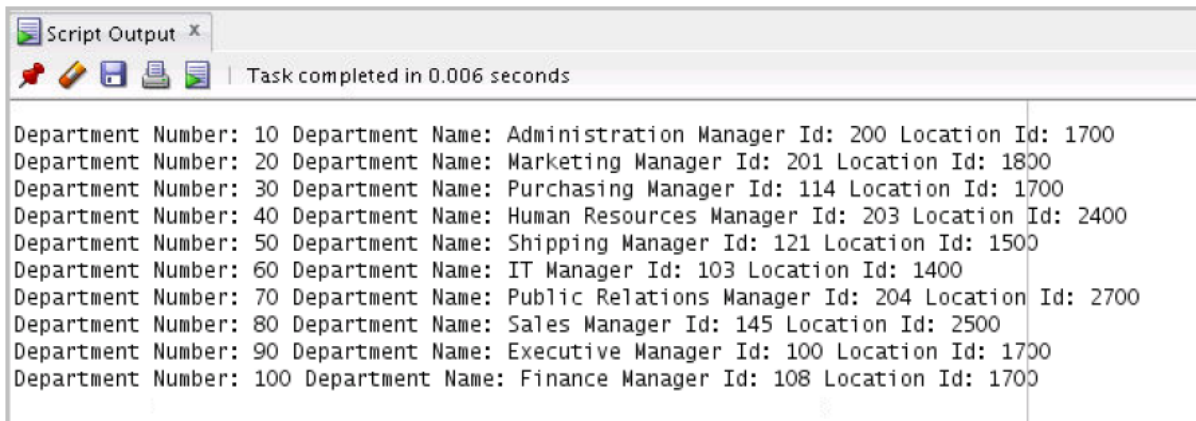
Line 18 Column 2 | Insert | Modified | Windows: CRJF

3.

Modify the block that you created in Task 2 to retrieve all information about each department from the `DEPARTMENTS` table and display the information. Use an associative array with the `INDEX BY` table of records method.

- Load the `lab_07_02_soln.sql` script.
- You have declared the associative array to be of type `departments.department_name`. Modify the declaration of the associative array to temporarily store the number, name, and location of all the departments. Use the `%ROWTYPE` attribute.
- Modify the `SELECT` statement to retrieve all department information currently in the `DEPARTMENTS` table and store it in the associative array.
- Using another loop, retrieve the department information from the associative array and display the information.

The sample output is as follows:



```
Script Output x
Task completed in 0.006 seconds

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
```

3.1-

SQL Script:

```
SET SERVEROUTPUT ON FORMAT WRAPPED;
DECLARE
  TYPE rec IS RECORD (
    v_dept hr.departments.department_id%TYPE,
    v_deptname hr.departments.department_name%TYPE,
    v_mgr hr.departments.manager_id%TYPE,
    v_locid hr.departments.location_id%TYPE);

  v_rec rec;
  v_i NUMBER := 0;
  v_max NUMBER;
  BEGIN
    SELECT max(department_id) INTO v_max FROM hr.departments;
    LOOP
      v_i := v_i + 1;
      SELECT
        department_id,
        department_name,
        manager_id,
        location_id INTO v_rec
      FROM hr.departments
      WHERE department_id = v_i;

      DBMS_OUTPUT.PUT_LINE('Department Number : ' || v_rec.v_dept ||
        ' Department Name : ' || v_rec.v_deptname ||
        ' Manager Id : ' || v_rec.v_mgr ||
        ' Location Id : ' || v_rec.v_locid);

      EXIT WHEN v_i = v_max;
    END LOOP;
  END;
```

Script Output: Task completed in 0.11 seconds

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
 Department Number : 110 Department Name : Accounting Manager Id : 205 Location Id : 1700
 Department Number : 120 Department Name : Treasury Manager Id : Location Id : 1700
 Department Number : 130 Department Name : Corporate Tax Manager Id : Location Id : 1700
 Department Number : 140 Department Name : Control And Credit Manager Id : Location Id : 1700
 Department Number : 150 Department Name : Shareholder Services Manager Id : Location Id : 1700
 Department Number : 160 Department Name : Benefits Manager Id : Location Id : 1700
 Department Number : 170 Department Name : Manufacturing Manager Id : Location Id : 1700
 Department Number : 180 Department Name : Construction Manager Id : Location Id : 1700
 Department Number : 190 Department Name : Contracting Manager Id : Location Id : 1700
 Department Number : 200 Department Name : Operations Manager Id : Location Id : 1700
 Department Number : 210 Department Name : IT Support Manager Id : Location Id : 1700
 Department Number : 220 Department Name : MOC Manager Id : Location Id : 1700
 Department Number : 230 Department Name : IT Helpdesk Manager Id : Location Id : 1700
 Department Number : 240 Department Name : Government Sales Manager Id : Location Id : 1700
 Department Number : 250 Department Name : Retail Sales Manager Id : Location Id : 1700
 Department Number : 260 Department Name : Recruiting Manager Id : Location Id : 1700

3.2-

SQL Script: (Same as in 3.1)

Script Output: Task completed in 0.018 seconds

Department Number : 130 Department Name : Corporate Tax Manager Id : Location Id : 1700
 Department Number : 140 Department Name : Control And Credit Manager Id : Location Id : 1700
 Department Number : 150 Department Name : Shareholder Services Manager Id : Location Id : 1700
 Department Number : 160 Department Name : Benefits Manager Id : Location Id : 1700
 Department Number : 170 Department Name : Manufacturing Manager Id : Location Id : 1700
 Department Number : 180 Department Name : Construction Manager Id : Location Id : 1700
 Department Number : 190 Department Name : Contracting Manager Id : Location Id : 1700
 Department Number : 200 Department Name : Operations Manager Id : Location Id : 1700
 Department Number : 210 Department Name : IT Support Manager Id : Location Id : 1700
 Department Number : 220 Department Name : MOC Manager Id : Location Id : 1700
 Department Number : 230 Department Name : IT Helpdesk Manager Id : Location Id : 1700
 Department Number : 240 Department Name : Government Sales Manager Id : Location Id : 1700
 Department Number : 250 Department Name : Retail Sales Manager Id : Location Id : 1700
 Department Number : 260 Department Name : Recruiting Manager Id : Location Id : 1700

PL/SQL procedure successfully completed.

4.

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.”

Department ID	Message
10	Whalen Due for a raise
20	Hartstein Not Due for a raise Fay Not Due for a raise
50	Weiss Not Due for a raise Fripp Not Due for a raise Kaufling Not Due for a raise Vollman Not Due for a raise. OConnell Due for a raise Grant Due for a raise
80	Russell Not Due for a raise Partners Not Due for a raise Errazuriz Not Due for a raise Cambrault Not Due for a raise . . . Livingston Not Due for a raise Johnson Not Due for a raise

4-

[illegible]