**Quiz**

Identify situations in which you can use the %ROWTYPE attribute.

1. When you are not sure about the structure of the underlying database table

2. When you want to retrieve an entire row from a table

3. When you want to declare a variable according to another previously declared variable or database column

**Answer: 1, 2**

**Practice 6: Working with Composite Data Types**

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

a)  Declare a PL/SQL record based on the structure of the countries table.

b)  Declare a variable v\_countryid. Assign CA to v\_countryid.

c)  In the declarative section, use the %ROWTYPE attribute and declare the v\_country\_record variable of type countries.

d)  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:

**SET SERVEROUTPUT ON**

**DECLARE**

**v\_countryid VARCHAR2(30) := 'CA';**

**v\_country\_record countries%ROWTYPE;**

**BEGIN**

**SELECT \* INTO v\_country\_record**

**FROM countries WHERE country\_id = v\_countryid;**

**DBMS\_OUTPUT.PUT\_LINE('Country Id: ' || v\_country\_record.country\_id ||**

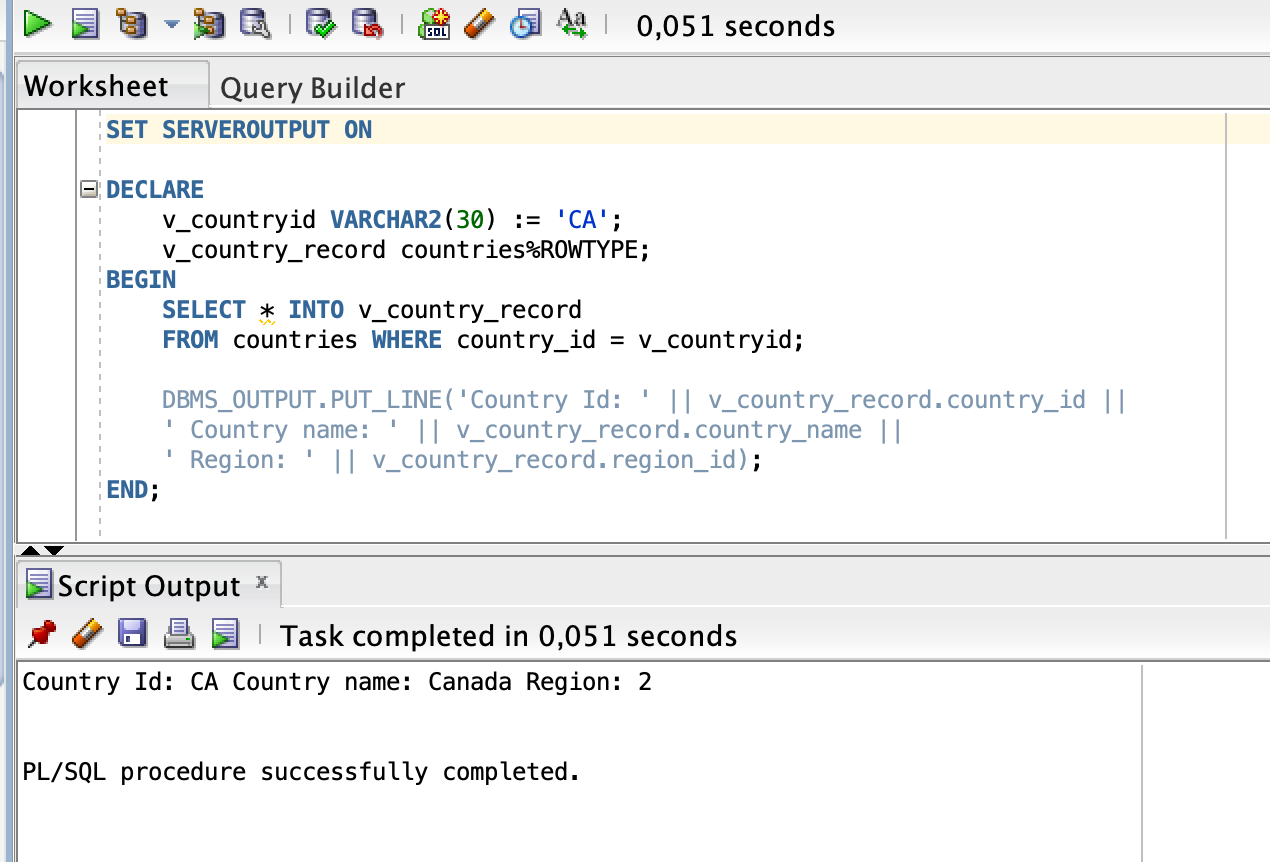
**' Country name: ' || v\_country\_record.country\_name ||**

**' Region: ' || v\_country\_record.region\_id);**

**END;**

Изображение выглядит как текст

Автоматически созданное описание



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

**SET SERVEROUTPUT ON**

**DECLARE**

**v\_countryid VARCHAR2(30) := 'CA';**

**v\_country\_record countries%ROWTYPE;**

**BEGIN**

**FOR v\_country\_record IN**

**(SELECT \* FROM countries WHERE country\_id IN ('DE', 'UK', 'US'))**

**LOOP**

**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 LOOP;**

**END;**

Изображение выглядит как текст

Автоматически созданное описание

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\_06\_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.

**DECLARE**

**TYPE dept\_table\_type IS TABLE OF**

**departments.department\_name%TYPE**

**INDEX BY PLS\_INTEGER;**

**my\_dept\_table dept\_table\_type;**

b)  Declare two variables: f\_loop\_count and v\_deptno of type NUMBER. Assign 10 to f\_loop\_count and 0 to v\_deptno.

**f\_loop\_count NUMBER := 10;**

**v\_deptno NUMBER := 0;**

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.

**BEGIN**

**FOR i IN 1..f\_loop\_count LOOP**

**v\_deptno := v\_deptno + 10;**

**SELECT department\_name**

**INTO my\_dept\_table(i)**

**FROM departments WHERE department\_id = v\_deptno;**

**END LOOP;**

Изображение выглядит как стол

Автоматически созданное описание

d) Using another loop, retrieve the department names from the associative array and display them.

**FOR i IN 1..f\_loop\_count LOOP**

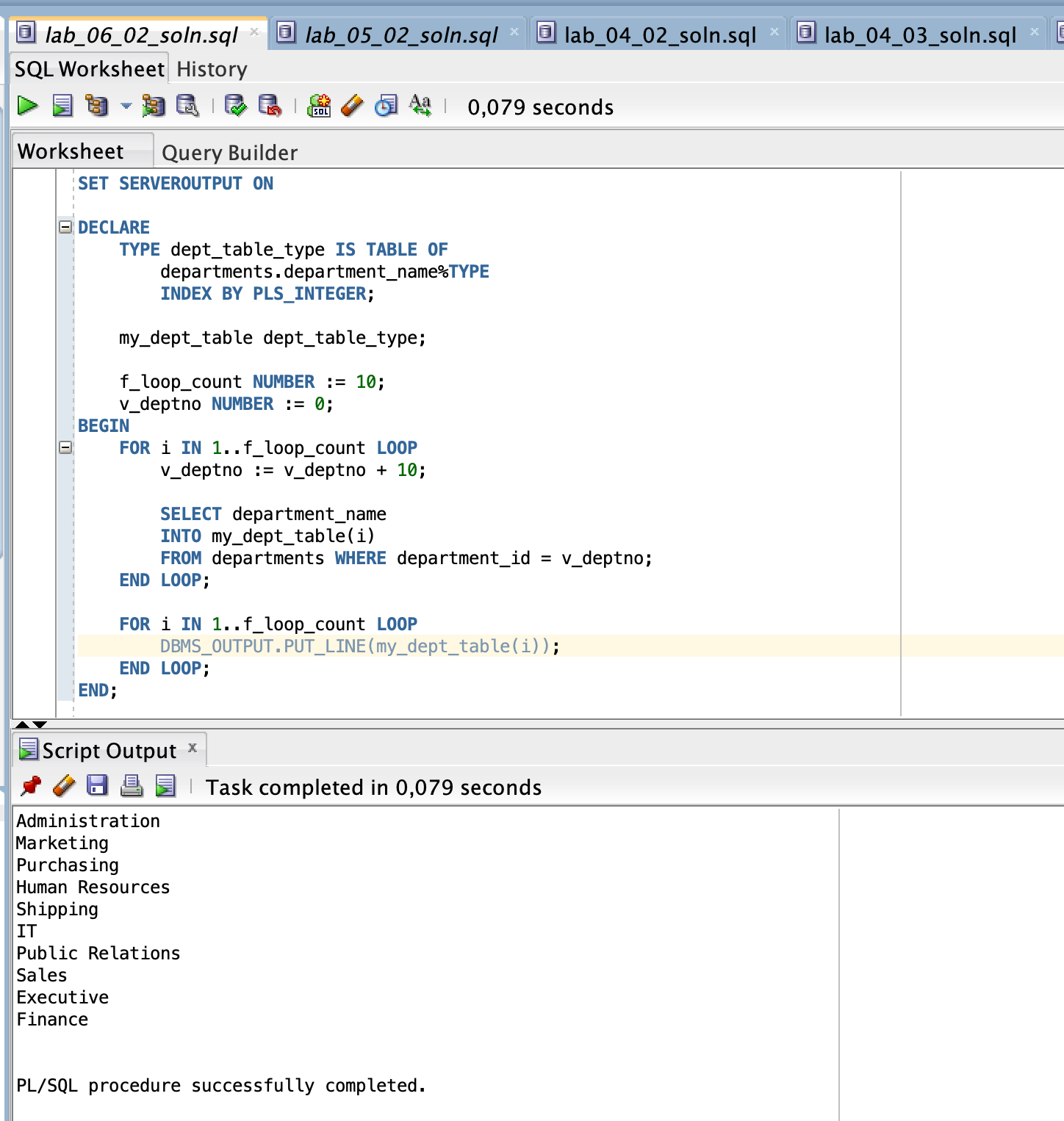
**DBMS\_OUTPUT.PUT\_LINE(my\_dept\_table(i));**

**END LOOP;**

e)  Execute and save your script as lab\_06\_02\_soln.sql. The output is as follows:

Изображение выглядит как стол

Автоматически созданное описание



3) Modify the block that you created in Practice 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.

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

b)  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.

**DECLARE**

**TYPE dept\_table\_type IS TABLE OF**

**departments%ROWTYPE**

**INDEX BY PLS\_INTEGER;**

c)  Modify the SELECT statement to retrieve all department information currently in the departments table and store it in the associative array.

**SELECT \* INTO my\_dept\_table(i)**

**FROM departments WHERE department\_id = v\_deptno;**

d)  Using another loop, retrieve the department information from the associative array and display the information.

**FOR i IN 1..f\_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;**

The sample output is as follows:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание