Database Programming with PL/SQL

4-2: Conditional Control: Case Statements

Practice Activities

Vocabulary

Identify the vocabulary word for each definition below:

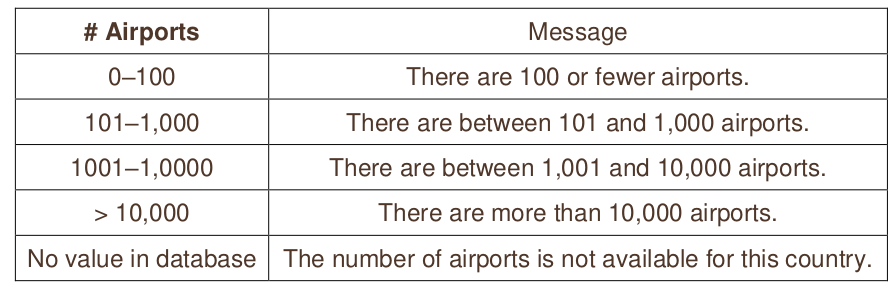
|  |  |
| --- | --- |
| CASE expression | An expression that selects a result and returns it into a variable. |
| Logic tables | Shows the results of all possible combinations of two conditions. |
| CASE statement | A block of code that performs actions based on conditional tests. |

Try It / Solve It

1. Write a PL/SQL block:

A. To find the number of airports from the countries table for a supplied country\_name. Based on

this number, display a customized message as follows

Use a CASE statement to process your comparisons.

You can use the following code to get started:

DECLARE

v\_country\_name countries.country\_name%TYPE := '<country\_name>';

v\_airports

countries.airports%TYPE;

BEGIN

SELECT airports INTO v\_airports

FROM countries

WHERE country\_name = v\_country\_name;

CASE

WHEN v\_airports < 101 then print(mesaj1);

when v<airports > 100 and v\_airports <1001 then print(mesaj2)

when v\_airports > 1000 and v\_airports <10001 then print(mesaj3)

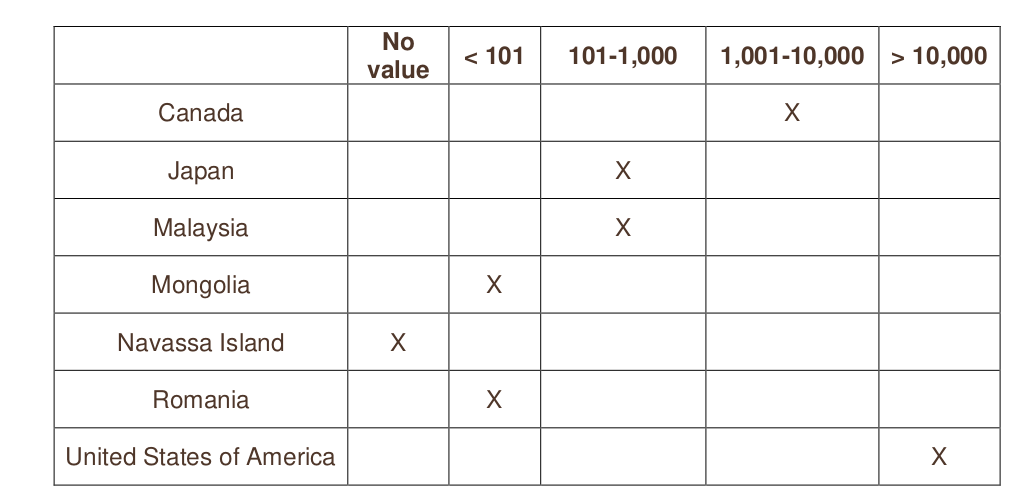
when v\_airports > 10000 then print(mesaj4)

else print(mesaj5)

END CASE;

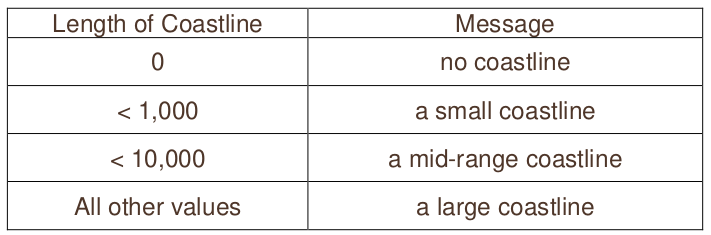
END;

B. Test your code for the following countries and confirm the results.

2. Write a PL/SQL block:

A. To find the amount of coastline for a supplied country name. Use the countries table. Based on

the amount of coastline for the country, display a customized message as follows:

Use a CASE expression.

Use the following code to get started:

DECLARE

v\_country\_name

countries.country\_name%TYPE := '<country name>';

v\_coastline

countries.coastline %TYPE;

v\_coastline\_description VARCHAR2(50);

BEGIN

SELECT coastline INTO v\_coastline

FROM countries

WHERE country\_name = v\_country\_name;

v\_coastline\_description :=

CASE

when v\_coastline = 0 then print(no coastline);

when v\_coastline<1000 then print(a small coastline);

when v\_coastline<10000 then print(a mid-range coastline);

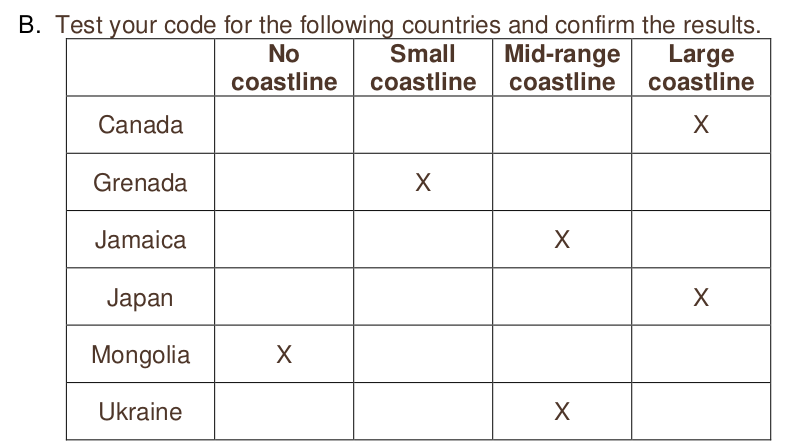
else print(a large coastline);

END;

DBMS\_OUTPUT.PUT\_LINE('Country ' || v\_country\_name || ' has '

|| v\_coastline\_description);

END;

3. Use a CASE statement:

A. Write a PL/SQL block to select the number of countries using a supplied currency name. If the

number of countries is greater than 20, display “More than 20 countries”. If the number of

countries is between 10 and 20, display “Between 10 and 20 countries”. If the number of

countries is less than 10, display “Fewer than 10 countries”. Use a CASE statement.

Declare v\_currency wf\_countries.currency\_code%type:=’USD’;

v\_result varchar2(50);

v\_count number(3);

BEGIN

select count(country\_id) into v\_count

from wf\_countries

where currency\_code=v\_currency;

v\_result:=

CASE

when v\_count<10 then mesaj1 –se atrivbuie lui v\_result

when v\_count>=10 and v\_count<21 then mesaj2

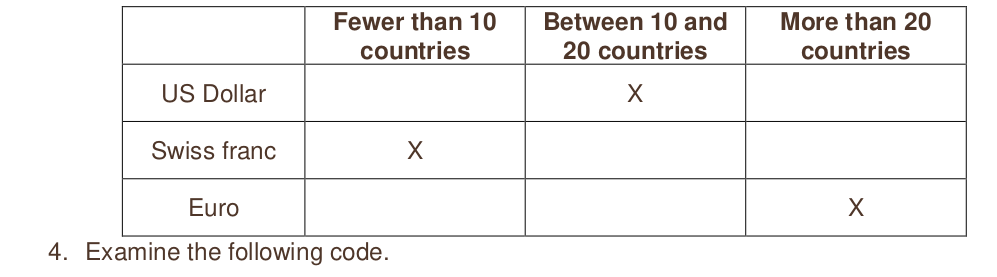
else mesaj3

END;

DBMS...(v\_result);

END;

B. Test your code using the following data:

A. What do you think the output will be? Test your prediction by running the code.

**DECLARE**

**x BOOLEAN := FALSE;**

**y BOOLEAN;**

**v\_color VARCHAR(20) := 'Red';**

**BEGIN**

**IF (x OR y)**

**THEN v\_color := 'White';**

**ELSE**

**v\_color := 'Black';**

**END IF;**

**DBMS\_OUTPUT.PUT\_LINE(v\_color);**

**END;**

Black

B. Change the declarations to x and y as follows. What do you think the output will be? Test your

prediction by running the code again.

x BOOLEAN ;

y BOOLEAN ;

Black

C. Change the declarations to x and y as follows. What do you think the output will be? Test your

prediction by running the code again.

x BOOLEAN := TRUE;

y BOOLEAN := TRUE;

White

D. Experiment with changing the OR condition to AND.

La fel da, black, pt ca avem NULL pe y, deci unknown;