ORACLE Academy

academy.oracle.com

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 Table	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:

# Airports	Message
0–100	There are 100 or fewer airports.
101–1,000	There are between 101 and 1,000 airports.
1001–1,0000	There are between 1,001 and 10,000 airports.
> 10,000	There are more than 10,000 airports.
No value in database	The number of airports is not available for this country.

Declararea variabilelor

v_nume_tara stocheaz numele rii introduse de utilizator.
v_aeroporturi va stoca numrul de aeroporturi preluat din baza de date.
Interogarea bazei de date

Se folosete SELECT INTO pentru a prelua numrul de aeroporturi corespunztor rii introduse.
Utilizarea instruciunii CASE pentru afiarea mesajului potrivit
Dac numrul de aeroporturi este între 0 i 100, se afieaz mesajul: „Exist 100 sau mai puine aeroporturi.”
Dac este între 101 i 1.000, mesajul este: „Exist între 101 i 1.000 de aeroporturi.”
Dac este între 1.001 i 10.000, mesajul este: „Exist între 1.001 i 10.000 de aeroporturi.”
Dac este mai mare de 10.000, se afieaz „Exist mai mult de 10.000 de aeroporturi.”
Tratarea excepiei NO_DATA_FOUND
Dac ara introdus nu exist în baza de date sau nu are un numr de aeroporturi asociat, se afieaz mesajul: „Numrul de aeroporturi nu este disponibil pentru aceast tara.”

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

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 ...
    …
  END CASE;
END;
```

```
WHEN v_aeroporturi BETWEEN 0 AND 100 THEN
   DBMS_OUTPUT.PUT_LINE('Exist 100 sau mai puine aeroporturi.');
WHEN v_aeroporturi BETWEEN 101 AND 1000 THEN
   DBMS_OUTPUT.PUT_LINE('Exist între 101 i 1.000 de aeroporturi.');
WHEN v_aeroporturi BETWEEN 1001 AND 10000 THEN
   DBMS_OUTPUT.PUT_LINE('Exist între 1.001 i 10.000 de aeroporturi.');
WHEN v_aeroporturi > 10000 THEN
   DBMS_OUTPUT.PUT_LINE('Exist mai mult de 10.000 de aeroporturi.');
```

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

	No value	**< 101**	**101-1,000**	**1,001-10,000**	**> 10,000**
Canada				X	
Japan			X		
Malaysia			X		
Mongolia		X			
Navassa Island	X				
Romania		X			
United States of America					X

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

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:

Length of Coastline	Message
0	no coastline
< 1,000	a small coastline
< 10,000	a mid-range coastline
All other values	a large coastline

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
    ...
    END;
  DBMS_OUTPUT.PUT_LINE('Country ' || v_country_name || ' has '
      || v_coastline_description);
END;
```

```
WHEN v_coastline = 0 THEN 'no coastline' -- 0 km de litoral
        WHEN v_coastline < 1000 THEN 'a small coastline' -- mai mic de 1.000 km
        WHEN v_coastline < 10000 THEN 'a mid-range coastline' -- între 1.000 i
10.000 km
        ELSE 'a large coastline' -- alte valori
```

Variabilele declarate:

v_country_name: Reine numele rii furnizat de utilizator.

v_coastline: Reine lungimea litoralului pentru ara respectiv.

v_coastline_description: Reine descrierea personalizat a litoralului.

SELECT INTO:

Execut o interogare SQL pentru a obine valoarea coastline pentru ara specificat.

Expresia CASE:

Determin mesajul personalizat pe baza valorii v_coastline.

DBMS_OUTPUT.PUT_LINE:

Afieaz mesajul rezultat în consol.

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

	No coastline	**Small coastline**	**Mid-range coastline**	**Large coastline**
Canada				X
Grenada		X		
Jamaica			X	
Japan				X
Mongolia	X			
Ukraine			X	

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

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.

B. Test your code using the following data:

	Fewer than 10 countries	Between 10 and 20 countries	More than 20 countries
US Dollar		X	
Swiss franc	X		
Euro			X

4. Examine the following code.

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;
```

```
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;
```

Analiz: Variabila x este iniializat cu FALSE. Variabila y nu este iniializat, iar o valoare nedefinit pentru BOOLEAN este tratat ca NULL. Într-o expresie OR, dac una dintre variabile este NULL, expresia devine NULL, iar condiia IF este considerat FALSE.

Rezultat ateptat: v_color va fi setat la 'Black', iar acest mesaj va fi afiat.

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 ;
```

Analiz: Atât x, cât i y sunt neiniializate (ambele sunt NULL). Într-o expresie OR, dac ambele valori sunt NULL, expresia rezult NULL, ceea ce face ca condiia IF s fie evaluat ca FALSE.

Rezultat ateptat: v_color va fi setat la 'Black', iar acest mesaj va fi afiat.

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;
```

Analiz: Atât x, cât i y sunt iniializate cu TRUE. Într-o expresie OR, dac cel puin una dintre valori este TRUE, expresia rezult TRUE. Astfel, condiia IF va fi evaluat ca TRUE.

Rezultat ateptat: v_color va fi setat la 'White', iar acest mesaj va fi afiat.

D. Experiment with changing the OR condition to AND.

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

În cazul în care ambii sunt TRUE, expresia x AND y va fi TRUE, iar v_color va fi 'White'.

Dac unul dintre ele este FALSE sau NULL, expresia x AND y va fi FALSE, iar v_color va fi 'Black'.