 



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:
   1. 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. |

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 wf\_countries

WHERE country\_name = v\_country\_name; CASE

WHEN …

…

END CASE; END;

DECLARE

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

v\_airports countries.airports%TYPE;

BEGIN

SELECT airports INTO v\_airports FROM wf\_countries

WHERE country\_name = v\_country\_name;

CASE

WHEN v\_airports > 0 AND v\_airports <= 100 THEN DBMS\_OUTPUT.PUT\_LINE('There are 100 or fewer airports');

WHEN v\_airports > 100 AND v\_airports <= 1000 THEN DBMS\_OUTPUT.PUT\_LINE('There are between 101 and 1000 airports');

WHEN v\_airports > 1000 AND v\_airports <=10000 THEN DBMS\_OUTPUT.PUT\_LINE('There are between 1001 and 10000 airports');

WHEN v\_airports > 10000 THEN DBMS\_OUTPUT.PUT\_LINE('There are more than 10000 airports');

ELSE

DBMS\_OUTPUT.PUT\_LINE('The number of airports is not available for this country.');

END CASE;

END;

* 1. 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 |

1. Write a PL/SQL block:
   1. 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;

DECLARE

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

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 'no coastline'

WHEN v\_coastline > 0 and v\_coastline < 1000 THEN 'a small coastline'

WHEN v\_coastline >= 1000 and v\_coastline < 10000 THEN' a mid-range coastline'

ELSE 'a large coastline'

END;

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

|| v\_coastline\_description);

END;

* 1. 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 |  |

1. Use a CASE statement:
   1. 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\_number countries.currency\_code%TYPE;

BEGIN

SELECT COUNT(currency\_code) INTO v\_number

FROM countries

WHERE currency\_code = 'EUR';

CASE

WHEN v\_number >=0 and v\_number < 10 THEN DBMS\_OUTPUT.PUT\_LINE('Fewer than 10 countries');

WHEN v\_number >=10 and v\_number < 20 THEN DBMS\_OUTPUT.PUT\_LINE('Between 10 and 20 countries');

ELSE DBMS\_OUTPUT.PUT\_LINE('More than 20 countries');

END CASE;

END;

Currency\_code USD si CHF (Swiss franc)

* 1. 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 |

1. Examine the following code.
   1. 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;

Cum y se initializeaza la NULL, rezultatul dintre FALSE or NULL este NULL, si se va executa ramura a 2-a, tiparind rezultatul Black.

* 1. 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 ;

Vom obtine acelasi rezultat, x fiind si el acum NULL, astfel rezultatul dintre NULL si NULL este NULL.

* 1. 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;

Rezultatul va fi WHITE, executandu-se a prima ramura a lui IF.

* 1. Experiment with changing the OR condition to AND.

DECLARE

x BOOLEAN := FALSE ;

y BOOLEAN ;

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

IF (x AND y)

THEN v\_color := 'White'; ELSE

v\_color := 'Black'; END IF;

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

FALSE AND NULL este FALSE, se va executa a 2-a ramura

DECLARE

x BOOLEAN := FALSE ;

y BOOLEAN ;

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

IF (x AND y)

THEN v\_color := 'White'; ELSE

v\_color := 'Black'; END IF;

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

NULL AND NULL este NULL, se va executa a 2-a ramura

DECLARE

x BOOLEAN := TRUE;

y BOOLEAN := TRUE;

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

IF (x AND y)

THEN v\_color := 'White'; ELSE

v\_color := 'Black'; END IF;

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

TRUE AND TRUE este TRUE se va executa prima ramura.