 



Database Programming with PL/SQL 4-1: Conditional Control: IF Statements Practice Activities

# Vocabulary

Identify the vocabulary word for each definition below:

|  |  |
| --- | --- |
| IF Statements Structure | Statement that enables PL/SQL to perform actions selectively based on conditions. |
| LOOP Control Structures | Control structures – Repetition statements that enable you to execute statements in a PL/SQL block repeatedly. |
| Conditia | An expression with a TRUE or FALSE value that is used to make a decision. |
|  | An expression that determines a course of action based on conditions and can be used outside a PL/SQL block in a SQL statement. |

# Try It / Solve It

1. What is the purpose of a conditional control structure in PL/SQL?

Scopul unor structuri de control este de a executa diferite instructiuni pentru diferite date.

1. List the three categories of control structures in PL/SQL.

IF-ELSE Statements

CASE Statements

LOOP Statemens

1. List the keywords that can be part of an IF statement.

IF, ELSE, ELFSIF, END.

1. List the keywords that are a required part of an IF statement.

IF, END;

1. Write a PL/SQL block to find the population of a given country in the countries table. Display a message indicating whether the population is greater than or less than 1 billion (1,000,000,000). Test your block twice using India (country\_id = 91) and United Kingdom (country\_id = 44). India’s population should be greater than 1 billion, while United Kingdom’s should be less than 1 billion.

DECLARE

v\_country\_name countries.country\_name%TYPE;

v\_population countries.population%TYPE;

BEGIN

SELECT population, country\_name INTO v\_population, v\_country\_name

FROM countries

WHERE country\_id = 91;

IF v\_population < 1000000000 THEN

DBMS\_OUTPUT.PUT\_LINE('Populatie peste un milion de locuitori:' || v\_population || ' in ' || v\_country\_name);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Populatia sub un milion de locuitori:'|| v\_population || ' in ' || v\_country\_name);

END IF;

END;

DECLARE

v\_country\_name countries.country\_name%TYPE;

v\_population countries.population%TYPE;

BEGIN

SELECT population, country\_name INTO v\_population, v\_country\_name

FROM countries

WHERE country\_id = 44;

IF v\_population < 1000000000 THEN

DBMS\_OUTPUT.PUT\_LINE('Populatie peste un milion de locuitori:' || v\_population || ' in ' || v\_country\_name);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Populatia sub un milion de locuitori:'|| v\_population || ' in ' || v\_country\_name);

END IF;

END;

1. Modify the code from the previous exercise so that it handles all the following cases:
   1. Population is greater than 1 billion.
   2. Population is greater than 0.
   3. Population is 0.
   4. Population is null. (Display: No data for this country.)

Run your code using the following country ids. Confirm the indicated results.

* + - China (country\_id = 86): Population is greater than 1 billion.
    - United Kingdom (country\_id = 44): Population is greater than 0.
    - Antarctica (country\_id = 672): Population is 0.
    - Europa Island (country\_id = 15): No data for this country.

DECLARE

v\_country\_name countries.country\_name%TYPE;

v\_population countries.population%TYPE;

BEGIN

SELECT population, country\_name INTO v\_population, v\_country\_name

FROM countries

WHERE country\_id = 86;

IF v\_population = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Populatia este zero' || v\_population || ' in ' || v\_country\_name);

ELSIF v\_population IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Populatia este necunoscuta:'|| v\_population || ' in ' || v\_country\_name);

ELSIF v\_population > 1000000000 THEN

DBMS\_OUTPUT.PUT\_LINE('Populatia mai mare decat 1 milion:'|| v\_population || ' in ' || v\_country\_name);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Populatie mai mare decat 0:' ||v\_population || ' in ' || v\_country\_name);

END IF;

END;

1. Examine the following code: DECLARE

v\_country\_id countries.country\_name%TYPE := <a value>; v\_ind\_date countries.date\_of\_independence%TYPE; v\_natl\_holiday countries.national\_holiday\_date%TYPE;

BEGIN

SELECT date\_of\_independence, national\_holiday\_date INTO v\_ind\_date, v\_natl\_holiday

FROM countries

WHERE country\_id = v\_country\_id; IF v\_ind\_date IS NOT NULL THEN

DBMS\_OUTPUT.PUT\_LINE('A');

ELSIF v\_natl\_holiday IS NOT NULL THEN DBMS\_OUTPUT.PUT\_LINE('B');

ELSIF v\_natl\_holiday IS NULL AND v\_ind\_date IS NULL THEN DBMS\_OUTPUT.PUT\_LINE('C');

END IF; END;

* 1. What would print if the country has an independence date equaling NULL and a national holiday date equaling NULL?

Va printa C .

* 1. What would print if the country has an independence date equaling NULL and a national holiday date containing a value?

Va printa B.

* 1. What would print if the country has an independence date equaling a value and a national holiday date equaling NULL?

Va printa A.

* 1. Run a SELECT statement against the COUNTRIES table to determine whether the following countries have independence dates or national holiday dates, or both. Predict the output of running the anonymous block found at the beginning of this question.

DECLARE

v\_country\_id countries.country\_name%TYPE := 1;

v\_ind\_date countries.date\_of\_independence%TYPE;

v\_natl\_holiday countries.national\_holiday\_date%TYPE;

BEGIN

SELECT date\_of\_independence, national\_holiday\_date INTO v\_ind\_date, v\_natl\_holiday

FROM countries

WHERE country\_id = v\_country\_id;

IF v\_ind\_date IS NOT NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Cazul 1:' || v\_ind\_date ||' '|| v\_natl\_holiday);

ELSIF v\_natl\_holiday IS NOT NULL THEN DBMS\_OUTPUT.PUT\_LINE('Cazul 2:' || v\_ind\_date || ' ' || v\_natl\_holiday);

ELSIF v\_natl\_holiday IS NULL AND v\_ind\_date IS NULL THEN DBMS\_OUTPUT.PUT\_LINE('Cazul 3:' || v\_ind\_date ||' ' || v\_natl\_holiday);

END IF;

END;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Country** | **Country\_ID** | **Independence Date** | **National**  **Holiday Date** | **Output should be** |
| Antarctica | 672 | NULL | NULL | C |
| Iraq | 964 | 28-Jun-2004 | NULL | A |
| Spain | 34 | NULL | 12-Jun | B |
| United States | 1 | 4-Jul-1776 | 4-Jul | A |

* 1. Finally, run the anonymous block found at the beginning of this question using each of the above country ids as input. Check whether your output answers are correct.

1. Examine the following code. What output do you think it will produce? DECLARE

v\_num1 NUMBER(3) := 123;

v\_num2 NUMBER;

BEGIN

IF v\_num1 <> v\_num2 THEN

DBMS\_OUTPUT.PUT\_LINE('The two numbers are not equal'); ELSE

DBMS\_OUTPUT.PUT\_LINE('The two numbers are equal'); END IF;

END;

Run the code to check if your prediction was correct. What was the result and why? Modify the code to use various comparison operators to see different results.

Rezultatul va fi ca cele 2 numere sunt egale pentru ca avem o comparatie cu NULL care este interpretata ca fiind FALSE, ca urmare se executa linia de pe ELSE .

DECLARE

v\_num1 NUMBER(3) := 123;

v\_num2 NUMBER;

BEGIN

IF v\_num1 > v\_num2 THEN

DBMS\_OUTPUT.PUT\_LINE('The two numbers are not equal'); ELSE

DBMS\_OUTPUT.PUT\_LINE('The two numbers are equal'); END IF;

END;

1. Write a PL/SQL block to accept a year and check whether it is a leap year. For example, if the year entered is 1990, the output should be “1990 is not a leap year.”

Hint: A leap year should be exactly divisible by 4, but not exactly divisible by 100. However, any year exactly divisible by 400 is a leap year.

Test your solution with the following years:

|  |  |
| --- | --- |
| **Year** | **Result Should Be** |
| 1990 | Not a leap year |
| 2000 | Leap year |
| 1996 | Leap year |
| 1900 | Not a leap year |
| 2016 | Leap year |
| 1884 | Leap year |

DECLARE

year NUMBER(4) := 1990;

BEGIN

IF MOD(year , 4)= 0 AND MOD(year , 4)!= 0 THEN

DBMS\_OUTPUT.PUT\_LINE('This is a leap year!');

ELSE

DBMS\_OUTPUT.PUT\_LINE('This is NOT a leap year!');

END IF;

END;