[www.oracle.com/academy](http://www.oracle.com/academy)

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

## Vocabulary

Identify the vocabulary word for each definition below:

|  |  |
| --- | --- |
| IF | Statement that enables PL/SQL to perform actions selectively based on conditions. |
| LOOP | Control structures – Repetition statements that enable you to execute statements in a PL/SQL block repeatedly. |
| IF | An expression with a TRUE or FALSE value that is used to make a decision. |
| CASE | 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 inPL/SQL?

Determinar que bloques se ejecutan.

1. List the three categories of control structures inPL/SQL.

IF, CASE, LOOP

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

IF, THEN, ELSE, END IF, elsif

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

IF, END IF

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 1b i l I o .(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\_india wf\_countries.population%TYPE ; v\_nume wf\_countries.COUNTRY\_NAME%TYPE;

BEGIN SELECT population,COUNTRY\_NAME INTO v\_india,v\_nume FROM wf\_countries

WHERE country\_id=91; IF v\_india>1000000000

THEN DBMS\_OUTPUT.PUT\_LINE('populatia mai mare de 1.000.000.000 este ' ||v\_nume);

END IF;

END;

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

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

* + - China (country\_id = 86): Population is greater than 1billion.
    - United Kingdom (country\_id = 44): Population is greater than0.
    - Antarctica (country\_id = 672): Population is0.
    - Europa Island (country\_id = 15): No data for thiscountry.

DECLARE v\_india wf\_countries.population%TYPE ; v\_nume wf\_countries.COUNTRY\_NAME%TYPE;

BEGIN SELECT population,COUNTRY\_NAME INTO v\_india,v\_nume FROM wf\_countries

WHERE country\_id=91; IF v\_india>1000000000

THEN DBMS\_OUTPUT.PUT\_LINE('populatia mai mare de 1.000.000.000 este ' ||v\_nume);

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\_holidaycountries.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 equalingNULL? C
  2. What would print if the country has an independence date equaling NULL and a national holiday date containing avalue? B
  3. What would print if the country has an independence date equaling a value and a national holiday date equalingNULL? A
  4. 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 thisquestion.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Country** | **Country\_ID** | **Independence Date** | **National Holiday Date** | **Output should be** |
| Antarctica | 672 | NO | NO | C |
| Iraq | 964 | SI | NO | A |
| Spain | 34 | NO | SI | B |
| United States | 1 | SI | NO | A |

1. Examine the following code. What output do you think it willproduce?

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.

The two numbers are equal. Porque el IF está mal puesto.

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 leapyear.”

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 |

[www.oracle.com/academy](http://www.oracle.com/academy)

Database Programming with PL/SQL 4-2:Conditional Control: Case StatementsPractice Activities

## Vocabulary

Identify the vocabulary word for each definition below:

|  |  |
| --- | --- |
| CASE (exp) | An expression that selects a result and returns it into a variable. |
| Tablas lógicas | Shows the results of all possible combinations of two conditions. |
| CASE (secuencia) | A block of code that performs actions based on conditional tests. |

## Try It / Solve It

1. Write a PL/SQLblock:
   1. To find the number of airports from the countries table for a supplied country\_name. Based on this number, display a customized message asfollows:

|  |  |
| --- | --- |
| **# 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;

* 1. Test your code for the following countries and confirm theresults.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **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 |

Todos están correctos excepto Malaysia que es <101.

1. Write a PL/SQLblock:
   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 := '<country name>';v\_coastline countries.coastline%TYPE;v\_coastline\_description VARCHAR2(50);BEGINSELECT coastline INTO v\_coastlineFROM countriesWHERE country\_name = v\_country\_name;v\_coastline\_description :=CASEWHEN v\_coastline=0 then 'No hay linea costera'

WHEN v\_coastline>=1 and v\_coastline<1000 then 'Pequeña linea costera' WHEN v\_coastline>=1000 and v\_coastline<10000 then 'Mediana linea costera' WHEN v\_coastline>=10000 then 'Larga linea costera'END;DBMS\_OUTPUT.PUT\_LINE('Country ' || v\_country\_name || ' has '|| v\_coastline\_description);

END;

* 1. Test your code for the following countries and confirm theresults.

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

Todo está correcto.

1. Use a CASEstatement:
   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 20countries”.

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 CASEstatement.

DECLARE

v\_currency\_name currencies.currency\_name%TYPE :='Euro';

v\_contar NUMBER;

v\_currency\_description VARCHAR2(50);

BEGINselect COUNT(countries.country\_id) INTO v\_contar FROM currencies inner join countries ON currencies.currency\_code=countries.currency\_codewhere currencies.currency\_name=v\_currency\_name;

v\_currency\_description :=CASEWHEN v\_contar<10 THEN

'Menos de diez paises'

WHEN v\_contar>=10 AND v\_contar<=20 THEN

'Entre 10 y 20 paises'

WHEN v\_contar>20 THEN

'Mas de 20 paises'

END;

DBMS\_OUTPUT.PUT\_LINE('Moneda: '||v\_currency\_name||' Cuantos paises lo utilizan: '||v\_contar||' Y esta: '||v\_currency\_description);

END;

* 1. Test your code using the followingdata:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **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 thecode.

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

* 1. Change the declarations to x and y as follows. What do you think the output will be? Test your prediction by running the codeagain.

x BOOLEAN ; y BOOLEAN ;

WHITE

* 1. Change the declarations to x and y as follows. What do you think the output will be? Test your prediction by running the codeagain.

x BOOLEAN := TRUE; y BOOLEAN := TRUE;

RED

* 1. Experiment with changing the OR condition toAND.

TODOS LOS RESULTADOS CAMBIAN.

[www.oracle.com/academy](http://www.oracle.com/academy)

# Database Programming with PL/SQL

4-4:Iterative Control: WHILE and FOR Loops

Practice Activities

## Vocabulary

Identify the vocabulary word for each definition below:

|  |  |
| --- | --- |
| WHILE | Repeats a sequence of statements until the controlling condition is no longer TRUE. |
| FOR | Repeats a sequence of statements until a set number of iterations have been completed. |

## Try It / Solve It

1. Write a PL/SQL block to display the country\_id and country\_name values from the COUNTRIES table for country\_id whose values range from 51 through 55. Use a WHILE loop. Increment a variable from 51 through 55. Test your variable to see when it reaches 55. EXIT the loop after you have displayed the 5countries.

DECLARE

v\_countryid wf\_countries.country\_id%TYPE;

v\_countryname wf\_countries.country\_name%TYPE;

v\_ctr NUMBER(3) := 51;

BEGIN

WHILE v\_ctr <= 55 LOOP

SELECT country\_name, country\_id

INTO v\_countryname, v\_countryid

FROM wf\_countries

WHERE country\_id = v\_ctr; -- uses country id with same number as couner

DBMS\_OUTPUT.PUT\_LINE(v\_countryname||' has an ID of '||v\_countryid||'.');

v\_ctr := v\_ctr + 1;

END LOOP;

END;

1. Write a PL/SQL block to display the country\_id and country\_name values from the COUNTRIES table for country\_id whose values range from 51 through 55*in the reverse order*. Use a FORloop.

DECLAREv\_id wf\_countries.country\_id%TYPE; v\_nume wf\_countries.country\_name%TYPE; BEGIN SELECT country\_id,country\_name INTO v\_id,v\_nume FROM wf\_countries WHERE country\_id=v\_id; FOR i IN REVERSE 51..55 LOOP DBMS\_OUTPUT.PUT\_LINE(v\_nume||v\_id);

END LOOP; END;

1. Execute the following statements to build a new\_empstable.

DROP TABLE new\_emps;

CREATE TABLE new\_emps AS SELECT \* FROM employees;

ALTER TABLE new\_emps ADD stars VARCHAR2(50);

* 1. Create a PL/SQL block that inserts an asterisk in the stars column for everywhole

$1,000 of an employee’s salary. For example, if an employee has salary of $7,800, the string “\*\*\*\*\*\*\*” would be inserted, and, if an employee has salary of $3,100, the string “\*\*\*” would be inserted. Use the following code as a starting point.

DECLARE

v\_empno new\_emps.employee\_id%TYPE :=<employee\_id>;

v\_asterisk new\_emps.stars%TYPE := NULL; v\_sal\_in\_thousands new\_emps.salary%TYPE;

BEGIN

SELECT NVL(TRUNC(salary/1000), 0) INTO v\_sal\_in\_thousands FROM new\_emps WHERE employee\_id = v\_empno;

FOR

UPDATE new\_emps SET stars = v\_asterisk WHERE employee\_id = v\_empno;

END;

Testyour code using employee\_ids 124 and 142, then confirm theresults.

[www.oracle.com/academy](http://www.oracle.com/academy)

# Database Programming with PL/SQL 4-5: Iterative Control: Nested Loops Practice Activities

## Vocabulary

*No new vocabulary for this lesson*

## Try It / Solve It

1. Write a PL/SQL block to produce a list of available vehicle license plate numbers. These numbers must be in the following format: NN-MMM, where NN is between 60 and 65, and MMM is between 100 and 110. Use nested FOR loops. The outer loop should choose numbers between 60 and 65. The inner loop should choose numbers between 100 and 110, and concatenate the two numberstogether.

BEGINFOR v\_licenta IN 60..65 LOOPFOR v\_licentav IN 100..110 LOOPDBMS\_OUTPUT.PUT\_LINE(v\_licenta ||'-'|| v\_licentav);END LOOP;END LOOP;END;

1. Modify your block from question 1 to calculate the sum of the two numbers on each iteration of the inner loop (for example, 62-107 sums to 169), and exit from the OUTER loop if the sum of the two numbers is greater than 172. Use loop labels. Test your modified code.

DECLARED

v\_sum NUMBER(5);BEGINFOR v\_licenta IN 60..65 LOOPFOR v\_licentav IN 100..110 LOOPv\_sum:=v\_licenta+v\_licentav;DBMS\_OUTPUT.PUT\_LINE(v\_licenta ||'-'||v\_licentav|| '=' ||v\_sum);

END LOOP;

EXIT WHEN v\_sum>172;

END LOOP;

END;