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.

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

   IF, CASE, LOOP

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

   IF, THEN, ELSE, END IF, elsif

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

   IF, END IF

5. 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 billo. (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;
```

6. Modify the code from the previous exercise so that it handles all the following cases:

   A. Population is greater than 1 billion.
   B. Population is greater than 0.
   C. Population is 0.
   D. 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_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;
```

7. 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;
```

A. What would print if the country has an independence date equaling NULL and a national holiday date equalingNULL? C

B. What would print if the country has an independence date equaling NULL and a national holiday date containing avalue? B

C. What would print if the country has an independence date equaling a value and a national holiday date equalingNULL? A

D. 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

8. 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.

9. 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

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:

   A. 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;
```

B. 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.

2. Write a PL/SQLblock:

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;
```

```
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_descripti
on :=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_LI
NE('Country ' || v_country_name || '
has '|| v_coastline_description);
END;
```

B. 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.

3. Use a CASEstatement:

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 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;
```

B. 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

4. Examine the following code.

A. 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

B. 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

C. 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

D. Experiment with changing the OR condition toAND.

TODOS LOS RESULTADOS CAMBIAN.

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;
```

2. 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 FOR loop.

```
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;
```

3. Execute the following statements to build a new_emps table.

```
DROP TABLE new_emps;

CREATE TABLE new_emps AS SELECT * FROM employees;

ALTER TABLE new_emps ADD stars VARCHAR2(50);
```

A. Create a PL/SQL block that inserts an asterisk in the stars column for every whole $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;
```

Test your code using employee_ids 124 and 142, then confirm the results.

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;
```

2. 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;
```