Database Programming with PL/SQL

2-4: Using Scalar Data Types

Practice Activities

Vocabulary

Identify the vocabulary word for each definition below:

Boolean	A datatype that stores one of the three possible values used for logical calculations: TRUE, FALSE, or NULL.
%TYPE	Attribute used to declare a variable according to another previously declared variable or database column.

Try It / Solve It

1. Declarations:

A. Which of the following variable declarations are valid?

	Declaration	Valid or Invalid
a	number_of_students PLS_INTEGER;	**VALID**
b	STUDENT_NAME VARCHAR2(10) = Johnson;	**INVALID**
c	stu_per_class CONSTANT NUMBER;	**INVALID**
d	tomorrow DATE := SYSDATE+1;	**VALID**

B. For the invalid declarations above, describe why they are invalid.

B) Esta mal la forma de asignar el valor, además el valor debe ir entre comillas simples.

C) No se asignó un valor a la constante.

C. Write an anonymous block in which you declare and print (on the screen) each of the variables in 1A above, correcting the invalid declarations and adding information as needed.

```
DECLARE
number_of_students    PLS_INTEGER:= 10;
STUDENT_NAME  VARCHAR2(10) := 'Johnson';
stu_per_class  CONSTANT NUMBER(10):= 10;
tomorrow  DATE := SYSDATE+1;
BEGIN
  DBMS_OUTPUt.PUT_LINE('number_of_students'|| number_of_students);
  DBMS_OUTPUt.PUT_LINE('Student_name'|| student_name);
  DBMS_OUTPUt.PUT_LINE('stu_per_class'|| stu_per_class);
  DBMS_OUTPUt.PUT_LINE('tomorrow'|| tomorrow);
END;
```

2. Evaluate the variables in the following code. Answer the following questions about each variable. Is it named well? Why or why not? If it is not named well, what would be a better name and why?

```
DECLARE
    country_name   VARCHAR2(50); median_age
    NUMBER(6, 2);
BEGIN
    SELECT country_name, median_age INTO country_name, median_age
        FROM countries
        WHERE country_name = 'Japan';
    DBMS_OUTPUT.PUT_LINE('The median age in '|| country_name || ' is '
```

```
        || median_age || '.');
END;
```

3. Change the declarations in #2 above so they use the %TYPE attribute.

```
DECLARE
   country_name   countries.country_name%type;
   median_age  countries.median_age%type;
BEGIN
    SELECT country_name, median_age INTO country_name, median_age
   FROM countries
   WHERE country_name = 'Japan';
    DBMS_OUTPUT.PUT_LINE('The median age in '|| country_name || ' is '
   || median_age || '.');
END;
```

4. In your own words, describe why using the %TYPE attribute is better than hard-coding data types. Can you explain how you could run into problems in the future by hard-coding the data types of the country_name and median_age variables in question 2?

Es mejor, porque si el tipo de dato de la columna cambia, no habría ningún problema porque usando %type se estaría haciendo una copia exacta del tipo de dato que contiene la columna.

5. Create the following anonymous block:

```
BEGIN
  DBMS_OUTPUT.PUT_LINE('Hello World'); END;
```

A. Add a declarative section to this PL/SQL block. In the declarative section, declare the following variables:

- A variable named TODAY of datatype DATE. Initialize TODAY with SYSDATE.
- A variable named TOMORROW with the same datatype as TODAY. Use the %TYPE attribute to declare this variable.

```
 DECLARE
TODAY DATE:= SYSDATE;
TOMORROW TODAY%TYPE;
BEGIN


END;
```

B. In the executable section, initialize the TOMORROW variable with an expression that calculates tomorrow's date (add 1 to the value in TODAY). Print the value of TODAY and TOMORROW after printing 'Hello World'.

```
 DECLARE
TODAY DATE:= SYSDATE;
TOMORROW TODAY%TYPE:= SYSDATE+1;


BEGIN
  DBMS_OUTPUT.PUT_LINE('Hello World');
  DBMS_OUTPUT.PUT_LINE('TODAY '|| TODAY);
  DBMS_OUTPUT.PUT_LINE('TOMORROW'|| TOMORROW);

END;
```