



EJERCICIOS PL/SQL

1. Evaluar cada una de las siguientes declaraciones. Determinar cuáles *no* son válidas y explique por qué.

- a. DECLARE
v_id NUMBER(4);
- b. DECLARE
v_x, v_y, v_z VARCHAR2(10);
- c. DECLARE
v_birthdate DATE NOT NULL ;
- d. DECLARE
v_in_stock BOOLEAN := 1;
- e. DECLARE
v_name TABLA1.NAME%TYPE;
- f. DECLARE
PiezaX NumPieza%TYPE;
NumPieza Pieza.Codigo%TYPE;

2. Examinar el bloque PL/SQL siguiente y determinar el tipo de dato y el valor de cada una de las siguientes variables, de acuerdo con las reglas del ámbito.

- a. El valor de V_WEIGHT en la posición 1 es:
- b. El valor de V_NEW_LOCN en la posición 1 es:
- c. El valor de V_WEIGHT en la posición 2 es:
- d. El valor de V_MESSAGE en la posición 2 es:
- e. El valor de V_NEW_LOCN en la posición 2 es:



```
DECLARE
    v_weight    NUMBER(3) := 600;
    v_message    VARCHAR2(255) := 'Product 10012';
BEGIN
    /*SUBBLOQUE*/
    DECLARE
        v_weight    NUMBER(3) := 1;
        v_message    VARCHAR2(255) := 'Product 11001';
        v_new_locn    VARCHAR2(50) := 'Europe';
    BEGIN
        v_weight := v_weight + 1;
        v_new_locn := 'Western ' || v_new_locn;
    ① → END;

        v_weight := v_weight + 1;
        v_message := v_message || ' is in stock';
        v_new_locn := 'Western ' || v_new_locn;
    ② → END;
    /
```

3. Crea un bloque anónimo para hacer que aparezca la frase "Hola Mundo" en la pantalla.
4. Construir un bloque PL/SQL que extraiga el apellido y el oficio del empleado cuyo código es el 7839.
5. Crea un bloque PL/SQL que seleccione el número de departamento más alto en la tabla DEPART. Imprima el resultado en pantalla.
6. Obtener a través de un bloque de PL/SQL el numero total de empleados. Imprimir el resultado.
7. Crea un bloque anónimo PL/SQL que inserta un nuevo departamento en la tabla DEPART. Sigue los pasos que se te indican:
Llame al nuevo departamento Educacion.



El número del nuevo departamento será el número de departamento más alto de la tabla DEPART mas 10.
Deja la localidad como nulo.

Comprueba el nuevo departamento que ha creado.

8. Crea un bloque PL/SQL que actualice la localidad del nuevo departamento que agregaste en la práctica anterior.
9. Crear un bloque PL/SQL que suprima el departamento que has creado en el ejercicio 7. Deseamos obtener por pantalla el número de filas afectadas. Se deben tener controlados los posibles errores de ejecución.
10. Crear un bloque PL/SQL que incremente el salario de los empleados del departamento 30 tanto como el salario que cobre el empleado que menos cobra de todos los departamentos.
11. Crear un bloque PL/SQL que incremente en un 10% el salario de los empleados del cuyo salario sea menor que el salario medio de todos los empleados de la empresa.
12. Crear la tabla MENSAJES, tabla con un único campo de tipo Number. Escribir un bloque PL/SQL para insertar números en dicha tabla. Insertaremos números del 1 al 10, excluyendo el 6 y el 8.
13. Crear un bloque PL/SQL que calcule la comisión del empleado 7521 basándose en el sueldo de dicho empleado.
 - Si el sueldo del empleado es menor de 200.000, muestre la bonificación del empleado como un 10% del sueldo.
 - Si el sueldo del empleado se encuentra entre 200.000 y 300.000, muestre la bonificación del empleado como un 15% del sueldo.
 - Si el sueldo del empleado supera los 300.000, muestre la bonificación del empleado como un 20% del sueldo.
 - Si el sueldo del empleado es NULL, muestre la bonificación del empleado como 0.
14. Crear una tabla EMP que sea una réplica de la tabla EMPLE. Agregar a la tabla EMP una nueva columna, STARS, del tipo de dato VARCHAR2 y con una longitud 50 para almacenar asteriscos (*).

Crear un bloque PL/SQL que recompense a un empleado agregando un asterisco a la columna STARS por cada 100000 del sueldo del empleado. Por ejemplo, si el empleado tiene un sueldo de 200000, la cadena de asteriscos debería contener dos asteriscos y si el empleado tiene un sueldo de 285000, la cadena de asteriscos debería contener 3 asteriscos (redondear el salario). El empleado en cuestión es el 7902.



-
15. A través de un bloque PL/SQL deseamos visualizar el apellido y la fecha de alta de todos los empleados ordenados por apellido. Teniendo en cuenta que la sentencia SELECT va a devolver mas de un valor. Realizalo de tres formas.
 16. Escribir un procedimiento que reciba dos números y visualice su suma.
 17. Codificar un procedimiento que reciba una lista de hasta 5 números y visualice su suma.
 18. Codificar un procedimiento que permita borrar un empleado cuyo número se pasará en la llamada.
 19. Escribir un procedimiento que modifique la localidad de un departamento. El procedimiento recibirá como parámetros el número del departamento y la localidad nueva.
 20. Visualizar todos los procedimientos y funciones del usuario almacenados en la base de datos y su situación (valid o invalid).
 21. Codificar un procedimiento que incremente el salario de un empleado, pasando como parámetros de dicho procedimiento el número del empleado y el incremento del sueldo.
 22. Crear una función denominada SALARIO_ANUAL para devolver el sueldo anual aceptando dos parámetros: el sueldo mensual y la comisión de un empleado. La función debería tratar valores NULL. Cualquiera de los dos valores que se han transferido, o los dos, puede ser NULL, pero la función debería seguir devolviendo el sueldo anual, que no es NULL. El sueldo anual se define por medio de la fórmula básica:
$$(\text{suelo} + \text{comisión}) * 12$$

Ayuda función NVL de SQL.

Utilizar la función en una sentencia SELECT en la tabla EMPLE del departamento 30
 23. Crear la función FACTORIAL que calcule el factorial de un número.
 24. Crear un procedimiento, NUEVO_EMP, para insertar un nuevo empleado en la tabla EMPLE. El procedimiento debería contener una llamada a la función VALIDAR_DEPT_NO para comprobar si el identificador de departamento especificado para el nuevo empleado existe en la tabla DEPART.

La función debería devolver un valor booleano (BOOLEAN).

Si la función devuelve FALSE, el procedimiento debería alertar al usuario con el mensaje adecuado.



-
- 25.** Programar una función que tenga como argumento el código de un Empleado. La función devolverá:
- La diferencia entre su Salario y el Salario más grande de entre todos los empleados de su departamento.
 - La función devolverá -1 si no existe ese empleado.
- 26.** Escribir un procedimiento PL/SQL que seleccione el apellido del empleado que tiene un sueldo determinado.
El salario se pasará al procedimiento como parámetro.
- Si el sueldo que ha introducido devuelve más de una fila, maneje la excepción con el manejador de excepciones adecuado e incluya en la tabla MENSAJES el mensaje "Más de un empleado con el salario <sueldo>."
 - Si el sueldo que ha introducido no devuelve ninguna fila, maneje la excepción con el manejador de excepciones adecuado e incluya en la tabla MENSAJES el mensaje "No hay empleados con el salario de <sueldo>."
 - Si el sueldo que ha introducido sólo devuelve una fila, incluya en la tabla MENSAJES el apellido del empleado y su sueldo.
 - Manejar cualquier otra excepción con el manejador de excepciones adecuado e incluya en la tabla MENSAJES el mensaje "Cualquier otro error producido ."
- 27.** Escribir un procedimiento PL/SQL que imprima el número de empleados que ganan 1000 más o menos que el valor del salario que se ha pasado al procedimiento como parámetro.
- a. Si no hay ningún empleado que tenga un sueldo en ese rango, imprima un mensaje para el usuario que le indique la situación. Utilice una excepción para este caso.
 - b. Si hay varios empleados en ese rango, el mensaje debería indicar cuántos empleados se encuentran en ese rango de sueldo. Utilice una excepción para este caso.
 - c. Manejar cualquier otra excepción con el manejador de excepciones adecuado. El mensaje debería indicar que se ha producido otro tipo de error.
- 28.** Escribir un procedimiento que reciba todos los datos de un nuevo empleado procese la transacción de alta, gestionando posibles errores. Los errores son: si no existe el departamento, sacar el mensaje 'Departamento inexistente'; si no existe el jefe, sacar el mensaje 'Jefe inexistente'; si el número del empleado que pasamos en la llamada está duplicado, 'Código de empleado duplicado'.



-
- 29.** Crear un procedimiento que reciba el código de un empleado y controle si en el departamento que trabaja dicho empleado hay demasiados empleados. Esto ocurre cuando el número supera los 5 individuos. Las excepciones que debéis controlar son:
- Que el empleado no exista.
 - Que haya demasiados empleados en el departamento.
 - En ambos casos, incluir un mensaje de error en la tabla Tabla_Avisos.
- 30.** Crear una nueva tabla para almacenar los sueldos de los empleados. Crear un procedimiento PL/SQL que determine cuáles son los empleados que tienen los sueldos más altos.
- a. Utilizar como parámetro de entrada un número n , donde n representa el número de los n que más ganan de la tabla EMPLE. Por ejemplo, escriba 5 para ver a los cinco empleados que más ganan.
 - b. Los sueldos no deberían estar duplicados. Si dos empleados ganan el mismo sueldo, este sólo se debería recopilar una vez.
 - c. Almacenar los sueldos en la tabla.
- Probar diversos casos especiales como, por ejemplo, $n = 0$, o donde n es mayor que el número de empleados de la tabla EMPLE. Vaciar la tabla después de cada prueba.