

Ejercicios PL/SQL

1. ¿Funcionaria el siguiente código?. Explicar por qué y cómo solucionarlo (si se os ocurren varias formas de arreglarlo, explicarlas todas).

```
DECLARE
    base NUMBER:=100;
BEGIN
    FOR dept IN 1..10 LOOP
        UPDATE dept SET nom_dept=base+dept
        WHERE cod_dept=base+dept;
    END LOOP;
END;
```

2. ¿Qué ocurriría al ejecutar el siguiente código?, ¿Por qué?, ¿Cómo arreglarlo?

(Si existen varias posibilidades, comentarlas e indicar cual sería más eficiente).

```
DECLARE
    ap1_emp VARCHAR2(40):='Fernandez';
BEGIN
    DELETE FROM emp WHERE ap1_emp=ap1_emp;
    COMMIT;
END;
```

3. ¿Es correcto el siguiente código en PL/SQL?, ¿Por qué?

Nota: Ignorar el funcionamiento del código (no hace nada), ceñirse exclusivamente a la sintaxis válida en PL/SQL.

```
FOR ctr IN 1..10 LOOP
    IF NOT fin THEN
        INSERT INTO temp VALUES (ctr, 'Hola');
        COMMIT;
        factor:=ctr*2;
    ELSE
        ctr:=10;
    END IF;
END LOOP;
```

4. ¿Qué resultado provoca la ejecución del siguiente código PL/SQL?

```
DECLARE
    variable NUMBER:=1;
    almacenamos NUMBER:=1;
BEGIN
    FOR i IN 5..variable LOOP
```

```

        almacenamos:=almacenamos+i;
    END LOOP;
    INSERT INTO traza VALUES(TO_CHAR(almacenamos));
    COMMIT;
END;

```

5. ¿Qué resultado proporciona la ejecución de este bloque?

```

DECLARE
    V_num NUMBER;
BEGIN
    SELECT COUNT(*) INTO V_num
    FROM productos;
    DBMS_OUTPUT.PUT_LINE(V_num);
END;

```

6. Crea una tabla MENSAJES con un solo campo VALOR, de tipo VARCHAR2(5).

Crea un bloque que inserte 8 elementos en la tabla con valores del 1 al 10, excepto el 4 y 5.

7. Vuelve a usar la tabla anterior con los datos ya rellenos, y si el número que insertas es menor que 4 entonces debes sacar un mensaje de error si ya existe, si el número es mayor o igual que 4, debes insertarlo si no existe y actualizarlo sumándole 1 si ya existe.

8. Crea una tabla PRODUCTO (CODPROD, NOMPROD, PRECIO), usando SQL (no uses un bloque PL/SQL).

Añade un producto a la tabla usando una sentencia insert dentro de un bloque PL/SQL.

9. Añade otro producto, ahora utilizando una lista de variables en la sentencia insert.

10. Añade, ahora usando un registro PL/SQL, dos productos más.

11. Borra el primer producto insertado, e incrementa el precio de los demás en un 5%.

12. Obtén y muestra por pantalla el número de productos que hay almacenados, usando select ... Into y el mensaje "Hay n productos".

13. Obtén y muestra todos los datos de un producto (busca a partir de la clave primaria, escogiendo un código de producto existente.
Usa select ... into y un registro PL/SQL.

14- Haz un bloque PL/SQL, de forma que se indique que no hay productos, que hay pocos (si hay entre 1 y 3) o el número exacto de productos existentes (si hay más de 3).