

Ejercicios de Introducción - PL-SQL

1. ¿Es correcta la siguiente sintaxis General de la sentencia IF-THEN ELSE?, ¿Por qué?, ¿Cómo la escribirías?

```
BEGIN
    IF condicion1 THEN
        BEGIN
            secuencia_de_instrucciones1;
        ELSE
            secuencia_de_instrucciones2;
        ENDIF;
    END;
```

2. ¿Qué resultado nos daría la siguiente comparación?

```
DECLARE
    identificador1 VARCHAR2(10):='Hola Pepe';
    identificador2 VARCHAR2(10):='Hola pepe';
BEGIN
    IF identificador1<>identificador2 THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;
END;
```

3. Indicar que errores existen en el siguiente código fuente:

```
DECLARE
    a NUMBER:=1;
    b NUMBER:=6;
    salida_bucle BOOLEAN;
BEGIN
    salida_bucle:='FALSE';
    WHILE NOT salida_bucle LOOP
        BEGIN
            IF a>=b THEN
                salida_bucle:='TRUE';
            ELSE
                a:=(a+1);
            END IF;
        END LOOP;
    END;
```

4. ¿Qué valor contendrá la variable 'sumador' al salir del bucle?, ¿Por qué?

```
DECLARE
    sumador NUMBER;
BEGIN
    FOR i IN 1..100 LOOP
        sumador:=sumador+i;
    END LOOP;
END;
```

5. ¿Qué resultado dará la ejecución del siguiente código?

```
DECLARE
    temp NUMBER(1,0);
    SUBTYPE numero IS temp%TYPE;
    valor numero;
BEGIN
    WHILE valor<20 LOOP
        valor:=valor+1;
    END LOOP;
END;
```

6. ¿Funcionaría el siguiente trozo de código?, ¿Por qué?, ¿Cómo arreglarlo?

```
DECLARE
    mi_valor NUMBER;
    cierto BOOLEAN:=FALSE;
BEGIN
    WHILE NOT cierto LOOP
        IF mi_valor=NULL THEN
            mi_valor:=1;
        ELSE
            mi_valor:=mi_valor+1;
        END IF;
        IF mi_valor>100 THEN
            cierto:=TRUE;
        END IF;
        EXIT WHEN cierto;
    END LOOP;
END;
```

7. Escribir la sintaxis General de un código que evalúe si se cumple una condición, en caso de cumplirse que ejecute una serie de sentencias, en caso contrario que evalúe otra, que de cumplirse ejecute otras instrucciones, si ésta no se cumple que evalúe una tercera condición.. y así N veces. En caso de existir varias soluciones, comentarlas y escribir la más óptima o clara.

8. Implementar en PL/SQL un bucle infinito que vaya sumando valores en una variable de tipo NUMBER.

```
DECLARE
    num NUMBER:=0;
BEGIN
    WHILE TRUE
    LOOP
        num = num + 1;
    END LOOP;
END;
```

9. En base al bucle anterior, añadirle la condición de que salga cuando la variable sea mayor que 10.000.

```
DECLARE
    num NUMBER:=0;
BEGIN
    WHILE TRUE LOOP
        num := num + 1;
        IF num > 10000 THEN
            EXIT;
        END IF;
    END LOOP;
END;
```

10. Implementar un bucle en PL/SQL mediante la sentencia WHILE, en el cual vayamos sumando valores a una variable mientras ésta sea menor que 10, y asegurándonos de que el bucle se ejecuta por lo menos una vez.

11. Implementar en PL/SQL, el código necesario de un programa que al final de su ejecución haya almacenado en una variable llamada 'cadena', el siguiente valor:

```
cadena:='10*9*8*7*6*5*4*3*2*1'
```