

# BASES DE DATOS TEMA 15

PL/SQL

**Índice**

**BASES DE DATOS TEMA 15 1**

[**1. Código para trabajar en clase 3**](#_heading=h.tyjcwt)

[**2. Conceptos teóricos. Desarrollar cada concepto brevemente. 7**](#_heading=h.3dy6vkm)

[**2.1. PL/SQL 7**](#_heading=h.1t3h5sf)

[**2.2. Imprimir por pantalla: 7**](#_heading=h.4d34og8)

[**2.3. Unidades léxicas: 7**](#_heading=h.2s8eyo1)

[**3. Ejercicio 1: Código PL/SQL 7**](#_heading=h.17dp8vu)

[**4. Ejercicio 2: Imprimir por pantalla 8**](#_heading=h.3rdcrjn)

[**5. Ejercicio 3: Hola mundo 8**](#_heading=h.26in1rg)

[**6. Ejercicio 4: SELECT en PL/SQL 9**](#_heading=h.lnxbz9)

[**7. Ejercicio 5: Condicionales con IF 9**](#_heading=h.35nkun2)

[**8. Ejercicio 6: Condicionales con CASE 9**](#_heading=h.1ksv4uv)

[**9. Ejercicio 7: Optimizar estructuras de control 9**](#_heading=h.44sinio)

[**10. Ejercicio 8: Bucles con WHILE 10**](#_heading=h.2jxsxqh)

[**11. Ejercicio 9: Bucles con FOR 10**](#_heading=h.z337ya)

[**12. Ejercicio 10: Bucle sencillo LOOP 10**](#_heading=h.3j2qqm3)

[**13. Ejercicio 11: Batería de ejercicios 10**](#_heading=h.1y810tw)

PL/SQL

# Código para trabajar en clase

Lenguaje SQL para preparar la base de datos:

DROP TABLE fabricante CASCADE CONSTRAINTS;

DROP TABLE producto CASCADE CONSTRAINTS;

DROP TABLE reposicion CASCADE CONSTRAINTS;

CREATE TABLE fabricante (

id INT PRIMARY KEY,

nombre VARCHAR(100) NOT NULL

);

INSERT INTO fabricante VALUES(1, 'Asus');

INSERT INTO fabricante VALUES(2, 'Lenovo');

INSERT INTO fabricante VALUES(3, 'Hewlett-Packard');

INSERT INTO fabricante VALUES(4, 'Samsung');

INSERT INTO fabricante VALUES(5, 'Seagate');

INSERT INTO fabricante VALUES(6, 'Crucial');

INSERT INTO fabricante VALUES(7, 'Gigabyte');

INSERT INTO fabricante VALUES(8, 'Huawei');

INSERT INTO fabricante VALUES(9, 'Xiaomi');

INSERT INTO fabricante VALUES (10, 'Amazon');

CREATE TABLE producto (

id INT PRIMARY KEY,

nombre VARCHAR(100) NOT NULL,

precio NUMBER NOT NULL,

id\_fabricante INT,

FOREIGN KEY (id\_fabricante) REFERENCES fabricante(id)

);

INSERT INTO producto VALUES(1, 'Disco duro SATA3 1TB', 86.99, 5);

INSERT INTO producto VALUES(2, 'Memoria RAM DDR4 8GB', 120, 6);

INSERT INTO producto VALUES(3, 'Disco SSD 1 TB', 49.99, 4);

INSERT INTO producto VALUES(4, 'GeForce GTX 1050Ti', 185, 7);

INSERT INTO producto VALUES(5, 'GeForce GTX 1080 Xtreme', 755, 6);

INSERT INTO producto VALUES(6, 'Monitor 24 LED Full HD', 202, 1);

INSERT INTO producto VALUES(7, 'Monitor 27 LED Full HD', 245.99, 1);

INSERT INTO producto VALUES(8, 'Portátil Yoga 520', 559, 2);

INSERT INTO producto VALUES(9, 'Portátil Ideapd 320', 444, 2);

INSERT INTO producto VALUES(10, 'Impresora HP Deskjet 3720', 59.99, 3);

INSERT INTO producto VALUES(11, 'Impresora HP Laserjet Pro M26nw', 180, 3);

INSERT INTO producto VALUES(12, 'Raton Optico', 20, NULL);

INSERT INTO producto VALUES (13, 'Alexa Echo Dot', 69.99, 10);

INSERT INTO producto VALUES (14, 'ThinkPad E16 Gen 1', 943.49, 2);

CREATE TABLE reposicion(

id\_reposicion INT PRIMARY KEY,

id\_producto INT NOT NULL,

unidades INT DEFAULT(0),

fecha DATE,

CONSTRAINT FK\_ReposicionProducto FOREIGN KEY

(id\_producto) REFERENCES producto(id)

);

INSERT INTO reposicion VALUES(1, 1, 2, '25/01/2023');

INSERT INTO reposicion VALUES(2, 2, 5, '15/08/2005');

INSERT INTO reposicion VALUES(3, 3, 3, '20/04/2010');

INSERT INTO reposicion VALUES(4, 4, 1, '03/11/2015');

INSERT INTO reposicion VALUES(5, 5, 2, '10/02/2003');

INSERT INTO reposicion VALUES(6, 6, 4, '28/06/2018');

INSERT INTO reposicion VALUES(7, 7, 3, '15/09/2009');

INSERT INTO reposicion VALUES(8, 8, 6, '22/07/2012');

INSERT INTO reposicion VALUES(9, 9, 8, '08/10/2017');

INSERT INTO reposicion VALUES(10, 10, 10, '17/12/2006');

INSERT INTO reposicion VALUES(11, 11, 2, '05/01/2022');

INSERT INTO reposicion VALUES(12, 2, 3, '10/04/2002');

INSERT INTO reposicion VALUES(13, 3, 6, '01/09/2018');

INSERT INTO reposicion VALUES(14, 4, 2, '12/11/2007');

INSERT INTO reposicion VALUES(15, 5, 4, '25/02/2013');

INSERT INTO reposicion VALUES(16, 6, 1, '18/07/2004');

INSERT INTO reposicion VALUES(17, 7, 5, '30/10/2010');

INSERT INTO reposicion VALUES(18, 8, 7, '05/06/2015');

INSERT INTO reposicion VALUES(19, 9, 9, '22/12/2006');

INSERT INTO reposicion VALUES(20, 10, 2, '08/03/2021');

INSERT INTO reposicion VALUES(21, 11, 8, '14/08/2009');

# Conceptos teóricos. Desarrollar cada concepto brevemente.

## PL/SQL

Definición: Lenguaje de programación procedimental (bucles, funciones…).

PL/SQL (Procedural Language/Structured Query Language).

Lenguaje de programación estructurado en bloques que se puede combinar con sentencias SQL.

Hay otros lenguajes procedimentales que usan SQL.

Ejemplo de sintaxis de código PL/SQL:

* DECLARE
* BEGIN
* END

**Normas básicas:**

* La palabra clave, nombres de tabla, funciones, etc. no distinguen entre mayúsculas y minúsculas.
* Las variables, constantes y excepciones se definen en el bloque DECLARE.
* Todas las instrucciones finalizan con punto y coma, a excepción de los encabezados.
* Las instrucciones pueden ocupar varias líneas.
* Los comentarios comienzan or /\* y finalizan por \*/. También se puede utilizar el doble guión (--) para una sola línea.

## Imprimir por pantalla:

Preparar la sesión: SET SERVEROUTPUT ON

Comando: DBMS\_OUTPUT.PUT\_LINE(Num);

## Unidades léxicas:

**Delimitadores: (NO CAE)**

**Identificadores:**

* Se utilizan para proporcionar un nombre a los elementos (variables, funciones, bloques, etc.).
* Se suelen utilizar letras que pueden combinarse con numeros o signos como $, # o \_
* Hay cieras palabras reservadas que no se pueden utilizar (type, if, then, else..).

**Literales:**

Se utilizan para asignar un valor a un identificador. Pueden ser valores lógicos, numéricos, de tipo carácter…

**Comentarios:**

Hay dos clases de comentarios.

* Entre los simbolos /\* y \*/ cuando son de varias lineas.
* A continuación del doble guion (--), cuando el comentario es una sola línea.

# Ejercicio 1: Código PL/SQL

El siguiente código, ¿crees que puede dar error?

DECLARE

Num NUMBER := 5; -- Comentario

/\* A continuación comienza el bloque,

Mediante un BEGIN, y finaliza mediante un END \*/

BEGIN

Num := num\*2;

END;

# Ejercicio 2: Imprimir por pantalla

Vamos a sacar el resultado por pantalla.

SET SERVEROUTPUT ON

DECLARE

Num NUMBER := 5; -- Comentario

/\* A continuacion comienza el bloque,

Mediante un BEGIN, y finaliza mediante un END \*/

BEGIN

Num := num\*2;

DBMS\_OUTPUT.PUT\_LINE(Num);

END;

# Ejercicio 3: Hola mundo

Declarar una variable tipo varchar que contenga la cadena “Hola Mundo” y en el bloque BEGIN imprimir por consola un mensaje con el contenido de dicha variable.

# Ejercicio 4: SELECT en PL/SQL

Escribir un bloque PL/SQL que busque el producto con el precio más alto, guarde el nombre en una variable llamada “v\_nombre” e imprima el siguiente mensaje por consola:

El producto más caro es [v\_nombre]

# Ejercicio 5: Condicionales con IF

Realizar una sentencia SELECT que obtenga el precio de un producto cuyo identificador se pida por consola. Crear una estructura if donde se compruebe si el precio es superior, inferior o igual a 185€ y salga un mensaje por pantalla de la siguiente forma:

El precio del producto es superior/igual/inferior a 185€

# Ejercicio 6: Condicionales con CASE

Realizar una sentencia SELECT que obtenga el id de fabricante de un producto cuyo identificador se meta por consola. Haciendo uso de una estructura CASE con los id y nombre de fabricantes sacar el siguiente mensaje:

El producto v\_nombre pertenece al fabricante v\_fabricante.

# Ejercicio 7: Optimizar estructuras de control

Repite el ejercicio anterior de forma más óptima.

# Ejercicio 8: Bucles con WHILE

Realizar un bucle que vaya consultando todos los productos desde el 1 hasta que encuentre un producto sin fabricante. En cada iteración se deberá imprimir el siguiente mensaje:

El producto v\_nombre cuesta v\_precio€

# Ejercicio 9: Bucles con FOR

Realizar un bucle que vaya consultando todos los productos desde el 1 hasta el 14 e imprimir el siguiente mensaje:

El producto v\_nombre cuesta v\_precio€

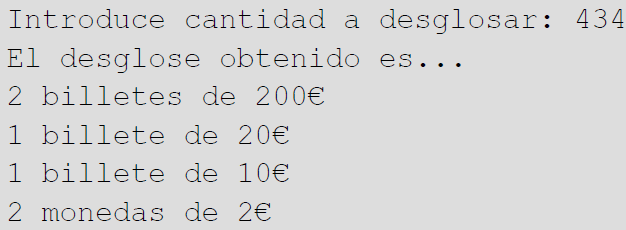
# Ejercicio 10: Bucle sencillo LOOP

Repetir ejercicio 15.8 con un bucle sencillo LOOP.

# Ejercicio 11: Batería de ejercicios

Relación de ejercicios para practicar PL/SQL:

1. Declarar una variable ‘a’ y una variable ‘b’. Ambas numéricas y constantes con valores 10 y 20 respectivamente.
   1. Sumar ambos valores en una variable llamada “result” y sacarla por consola.
   2. Sacar por consola el valor más alto de las dos variables, usando una estructura IF.
2. Solicita dos números por pantalla. Muestra por consola la suma, la resta, la multiplicación, la potencia y la división de ambos números. Usa la siguiente plantilla:
   1. La suma de \_\_ y \_\_ es:
   2. La resta de \_\_ y \_\_ es:
   3. La multiplicación de \_\_ y \_\_ es:
   4. El cociente de \_\_ entre \_\_ da:
3. Crea un algoritmo que calcule la media de 5 números que se le pasan por pantalla.
4. Pedir por teclado un número entero de 4 cifras.
   1. Devolver un mensaje si el número tiene más o menos de 4 cifras.
   2. Mostrar cada una de sus cifras (una debajo de otra).
   3. Crear un nuevo número con las cifras del primero pero al revés.
5. Solicitar que el usuario introduzca una clave dos veces. Mostrar un mensaje indicando si las claves son iguales o si son diferentes.
6. Realiza un programa que dada una cantidad de dinero en Euros, realice un desglose en billetes y monedas. Ej:



Los billetes disponibles son de 500, 200, 100, 50, 20, 10 y 5€ y las monedas de 2 y 1€.

1. Escribe un programa que lea de teclado 2 números enteros y saque en pantalla todos los números que estén entre ellos.
2. Modifica el ejercicio 7 para que solo escriba en pantalla los números pares del intervalo
3. Diseña un algoritmo que calcule el factorial de un número pedido por teclado. El factorial de un número es la multiplicación desde el 1 hasta ese número.
4. Realizar una sentencia SELECT que obtenga el nombre del producto con identificador 5 y lo asigne a una variable llamada “v\_nombre”. Sacar el nombre por consola.
5. Realizar una búsqueda de productos según un identificador que se introduzca por pantalla. Sacar su nombre y el nombre de su fabricante. Se deberá imprimir un mensaje:

El producto v\_nombre pertenece al fabricante v\_fabricante.

En caso de que no tenga fabricante el mensaje deberá ser:

El producto v\_nombre no tiene fabricante.