

# **TABD: Practica 1 Resumen**

- **Ejercicio 1:**

Escribe un script PL/SQL anónimo que reciba como entradas:

- Un valor real que representa una temperatura.
- Un carácter que indica la escala en la que está medida dicha temperatura ('C' para Celsius o 'F' para Fahrenheit).

Solución:

--Autores: Adriana Maña Watson | Francisco Javier Molina Rojas

SET SERVEROUTPUT ON;

DECLARE

    Temperatura REAL := &temp;

    Escala CHAR := &esc;

    tcalculada REAL;

BEGIN

    IF Escala = 'C' THEN

        tcalculada := (9/5)\*Temperatura+32;

        dbms\_output.put\_line('La temperatura en Farenheit es: ' || tcalculada);

    ELSIF Escala = 'F' THEN

        tcalculada := (5/9)\*(Temperatura-32);

        dbms\_output.put\_line('La temperatura en Celcius es: ' || tcalculada);

    ELSE

        dbms\_output.put\_line('ERROR: ESCALA NO ACEPTADA');

    END IF;

EXCEPTION WHEN OTHERS THEN

    dbms\_output.put\_line('ERROR: EXCEPCION');

END;

---

Cosas a tener en cuenta:

- Para poder mostrar los dbms\_output.put\_line('') hay que poner al principio de los scripts SET SERVEROUTPUT ON;
- Secciones principales (4): DECLARE, BEGIN, EXCEPTION y END
- Declare contiene las variables | si quieres inicializarlas, hacer: Nombre CHAR := &nombre
- El resto es sintaxis básica

- **Ejercicio 2:**

Ejecuta los siguientes scripts PL/SQL anónimos. Copia el resultado de ejecutarlos a un fichero y explica por qué muestran resultados diferentes aun cuando la variable v\_num toma el mismo valor en ambos scripts.

Solución:

Bloque 1:

Salida obtenida: v\_num is not greater than 0

Esto sucede debido a que NULL no es mayor que 0, es más, no es 0 ni un tipo Numerico. null no puede ser mayor/igual/menor que nada.

Bloque 2:

Salida obtenida: (vacía)

Esto es debido a que, primero se está comprobando si v\_num (NULL) es mayor que 0, y luego se esta comprobando si v\_num (NULL) NO es mayor que 0 (es decir, que es menor o igual a 0). Llegando a la conclusion del apartado anterior.

---

Cosas a tener en cuenta:

- NULL no posee tipo (no es un int/char...)
- No se puede comparar con otros datos ya que no pertenece a ellos (siempre dará false)

- **Ejercicio 3:**

Escribe un script PL/SQL anónimo para llevar el control de las ventas de los artículos de la base de datos, siguiendo los siguientes pasos:

- 1.Crea una tabla llamada Inventario donde guardaremos, para cada artículo el número de unidades existentes en el almacén y la fecha en que se actualizó por última vez esa información.
- 2.Crea una tabla denominada ControlVentas donde para cada venta guardaremos el artículo, el número de unidades vendidas o solicitadas, la fecha de venta o solicitud y un comentario que indicará si se pudo satisfacer el pedido o si, por el contrario, no hubo existencias disponibles.
- 3.Introduce algunas filas en las tablas.
- 4.Escribe un script PL/SQL anónimo que procese la orden de compra de cualquier número de unidades de un artículo dado.Si no hay cantidad disponible, se comunicará a través de un mensaje al usuario.

Solución:

--Autores: Adriana Maña Watson | Francisco Javier Molina Rojas

SET SERVEROUTPUT ON;

CREATE TABLE Inventario --Creacion tabla Inventario --

```
(
  NombreArt VARCHAR(15),
  NumExistencias INT,
  FechaUltAct DATE
);
```

CREATE TABLE ControlVentas --Creacion tabla ControlVentas --

```
(
  Nombreart VARCHAR(15),
  UnidVendidas INT,
  FechaVenta DATE,
  Comentario CHAR
);
```

-- INSERCIÓN DE DATOS --

INSERT INTO Inventario VALUES

```
(
  'Galletas',
  24,
  TO_DATE('2023-01-21', 'YYYY-MM-DD')
);
```

INSERT INTO Inventario VALUES

Francisco Javier Molina Rojas

```
(
    'Pelota',
    0,
    TO_DATE('2023-01-10', 'YYYY-MM-DD')
);
```

```
INSERT INTO ControlVentas VALUES
```

```
(
    'Galletas',
    6,
    TO_DATE('2022-12-30', 'YYYY-MM-DD'),
    'S'
);
```

```
INSERT INTO ControlVentas VALUES
```

```
(
    'Pelota',
    1,
    TO_DATE('2023-01-15', 'YYYY-MM-DD'),
    'S'
);
```

```
-- FIN DE INSERCIÓN DE DATOS --
```

```
DECLARE
```

```
    articulo VARCHAR(15) := &art; --Nombre del articulo a comprar
```

```
    uds INT := &uds; --unidades a comprar
```

```
    stockdisp INT; --Stock del articulo disponible
```

```
BEGIN
```

```
    -- 1er bloque: Consulta SQL --
```

```
    -- Objetivo: Conseguir el numero de existencias (stock) que posee el articulo a comprar --
```

```
    SELECT NumExistencias
```

```
    INTO stockdisp
```

```
    FROM Inventario
```

```
    WHERE NombreArt = articulo;
```

```
    -- Fin Consulta SQL --
```

```
    -- 2do bloque Hay que comprobar que el stock conseguido puede satisfacer el numero de unidades que se desea comprar --
```

```

IF stockdisp >= uds THEN -- Si se puede realizar la compra (stock disp >= uds pedidas) --

    -- Primero restamos del inventario las uds pedidas --
    UPDATE Inventario
    SET NumExistencias = stockdisp - uds, FechaUltAct = SYSDATE
    WHERE NombreArt = articulo;

    -- Segundo creamos una instancia de la venta con el nombre del articulo, las unidades, la fecha de hoy y el caracter 'S' que
    indica que la venta ha sido exitosa --
    INSERT INTO ControlVentas VALUES
    (
        articulo,
        uds,
        SYSDATE,
        'S'
    );

ELSE -- Si la venta no se a podido realizar --
    dbms_output.put_line('ERROR: EL NUMERO DE UNIDADES PEDIDAS ES MAYOR QUE EL STOCK DISPONIBLE'); --
    Lo indicamos --

    INSERT INTO ControlVentas VALUES -- Realizamos un registro con el nombre del articulo, las unidades, la fecha de hoy y
    el caracter 'N' que indica que la venta NO ha sido exitosa --
    (
        articulo,
        uds,
        SYSDATE,
        'N'
    );
END IF;

EXCEPTION

WHEN NO_DATA_FOUND THEN --En el caso de que el objeto a buscar NO este en la base de datos
    dbms_output.put_line('ERROR: EL ARTICULO SELECCIONADO NO EXISTE');

END;

```

---

Cosas a tener en cuenta:

- Para crear tabla usar CREATE TABLE Nombre (NombreVar TipoVar ...)
- Para insertar datos usar INSERT INTO NombreTabla VALUES (Variables...)
- Para guardar un dato de una consulta en una variable usar SELECT dato INTO variable FROM tabla Where condición
- Para lanzar una excepción cuando no se encuentra el dato, EXCEPTION WHEN NO\_DATA\_FOUND THEN ...