

Programación de bases de datos

Ejercicios en Oracle:

```
set serveroutput ON
begin
    DBMS_OUTPUT.PUT_LINE('hola desde Oracle PL/SQL');
END;
/

SET SERVEROUTPUT ON
--este código nos pide introducir el valor de altura, base y calcula el área del triángulo
DECLARE
    Altura INT;
    Base INT;
BEGIN
    Altura:=&Introduce_el_valor_de_altura;
    Base:=&Introduce_el_valor_de_base;
    DBMS_OUTPUT.PUT_LINE('Un triángulo de base: '||Base||' y altura: '||Altura||' tiene un área de: '||Base*Altura/2);
END;
/

SET SERVEROUTPUT ON
DECLARE
    nota NUMBER(2);
BEGIN
    nota := 7;
    IF nota = 10 OR nota = 9 THEN
        DBMS_OUTPUT.PUT_LINE('Sobresaliente');
    ELSIF nota = 8 OR nota = 7 THEN
        DBMS_OUTPUT.PUT_LINE('Notable');
    ELSIF nota = 6 THEN
        DBMS_OUTPUT.PUT_LINE('Bien');
    ELSIF nota = 5 THEN
        DBMS_OUTPUT.PUT_LINE('Suficiente');
    ELSIF nota < 5 AND nota >= 0 THEN
        DBMS_OUTPUT.PUT_LINE('Insuficiente');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Nota no válida');
    END IF;
END;
/
```

Ejercicios en MySQL:

```
CREATE DATABASE IF NOT EXISTS TEST; /* CREAMOS LA BBDD DE TEST SI NO EXISTE */

DELIMITER $$ /* La palabra DELIMITER indica qué caracter usar para inicio y fin del script,
              dado que el ; se usa en SQL, indicamos que vamos a usar $$ */

DROP PROCEDURE IF EXISTS test.hola_mundo$$ /* Borramos el procedimiento si no existe */

CREATE PROCEDURE test.hola_mundo() /* creamos el procedimiento y le damos nombre indicando donde lo creamos (test) */
BEGIN /* Indicamos el comienzo de las sentencias SQL del procedimiento */
    SELECT "HOLA MUNDO desde procedure"; /* Conjunto de sentencia SQL a ejecutar, en este caso escribir el texto */
END$$ /* Indicamos el fin del procedimiento con el END y con $$ */

#para ejecutarlo--> CALL test.hola_mundo(); /* Una vez creado el procedimiento, para llamarlo desde otra ventana escribir CALL nombre */
```

```

DELIMITER $$
CREATE FUNCTION test.estado(in_estado CHAR(1)) /*creamos la función y le decimos
que tiene un parámetro de tipo carácter*/
RETURNS VARCHAR(20) /* Y que cuando termine devolverá una cadena
de texto de tamaño máximo 20*/
BEGIN
DECLARE estado VARCHAR(20); /*Declaramos la variable a devolver
del mismo tipo que la que se devuelve
donde asignaremos el resultado*/

IF in_estado = 'C' THEN
SET estado= 'caducado';
ELSEIF in_estado = 'A' THEN
SET estado = 'activo';
ELSEIF in_estado = 'N' THEN
SET estado = 'nuevo';
END IF;
RETURN(estado);
END;$$

```

```

DELIMITER $$
CREATE FUNCTION esimpar(numero INT)
RETURNS INT
BEGIN
DECLARE impar INT;
IF MOD(numero,2)=0 THEN
SET impar=FALSE; /*FALSE=0*/
ELSE
SET impar=TRUE; /*TRUE=1*/
END IF;
RETURN(impar);
END;$$

```

```

DELIMITER $$
DROP PROCEDURE IF EXISTS muestra_estado$$
CREATE PROCEDURE test.muestra_estado(in numero int) /*parámetro de entrada*/
BEGIN
IF (esimpar(numero)) THEN /*llamada a la función*/
SELECT CONCAT(numero, " es impar");
ELSE
SELECT CONCAT(numero, " es par");
END IF;
END;$$

```