
 <b>cpifp</b> <b>Bajo Aragón</b>	<b>DAM - BASES DE DATOS DISTANCIA-</b>		 <b>IFC</b> <b>INFORMÁTICA</b> <b>Y COMUNICACIONES</b>
	<b>2ª EVALUACION</b>		
	<b>FECHA: 07 / 05 / 2024</b>	<b>CURSO: 1º</b>	<b>NOTA</b>
<b>NOMBRE:</b>			<b>/ 10</b>

### **Ejercicio Práctico: Administración y Programación en bases de datos (10 ptos)**

Una empresa de venta de productos por internet nos pide implementar ciertas operaciones y funcionalidades sobre su base de datos de gestión. Nuestros diseñadores tras un análisis han realizado el esquema lógico que encontramos al final de este documento. Además, nuestros diseñadores también han elaborado el script SQL que permite la creación del esquema físico sobre un SGBD Oracle.

Una vez creado el esquema, **nos piden implementar cierta funcionalidad en la base de datos y crear tres scripts sql (archivo con extensión .sql)** con las instrucciones que implementen lo siguiente

A. **Un script con las sentencias DML** del lenguaje SQL que resuelven lo siguiente (3,25 ptos):

- 1.1 Dar de alta como mínimo cinco registros en la tabla de Clientes (0.75 ptos).
- 2.1 Dar de alta como mínimo cinco registros en la tabla de Productos (1 ptos).
- 3.1 Actualizar el IVA de todos los productos de la categoría "bebidas alcohólicas". El nuevo valor del IVA será del 30% (0.5 ptos).
- 4.1 Realizar un descuento del 18% sobre el precio de los productos pedidos por clientes pertenecientes a provincias cuya longitud de nombre es mayor de 5 caracteres y el nombre no empieza ni por I ni por U. (0.5 ptos).
- 5.1 Eliminar los videos de los productos que no se han pedido nunca (0.5 ptos).



**Importante:** Las instrucciones no pueden dar fallo, han de hacer lo que se pide. Si no lo hacen se valoran con 0.

B. **Un script con las sentencias PL/SQL** que permiten la creación de las estructuras que permitan implementar los mecanismos de control siguientes (1.75 ptos):

- 1.1 Al dar de alta un pedido la fecha de este no puede ser mayor que la fecha actual, además de que un pedido solo puede servirse (servido=1) si el pedido está pagado (0.75 ptos).
- 2.1 No se puede dar de alta líneas de pedido (pedidos\_detalle) con unidades menor que 1, con el precio en blanco o con valor 0 ni con un IVA distinto de 21.(0.75 ptos).
- 3.1 El precioOferta de un producto siempre tiene que ser inferior al precio normal del mismo. (0.25 ptos).

En caso de que cualquiera de estas condiciones no se cumplan, se lanzará un mensaje de error con información del error/restricción que no se cumple.

C. Un script con las **sentencias PL/SQL** que permiten la creación de las estructuras que permiten implementar la siguiente funcionalidad (5 ptos):



 <b>cpifp</b> <b>Bajo Aragón</b>	<b>DAM - BASES DE DATOS DISTANCIA</b>		 <b>IFC</b> <b>INFORMÁTICA</b> <b>Y COMUNICACIONES</b>
	<b>2ª EVALUACION</b>		
	<b>FECHA: 07 / 05 / 2024</b>	<b>CURSO: 1º</b>	

- 1.1 Procedimiento a la que pasado como parámetro el nombre de un producto y una categoría nueva (nombre) permita cambiar el producto de una categoría a otra, verificando que esta existe y que es distinta a la actual del producto. **(0.75 ptos).**
- 2.1 Procedimiento o función que para una categoría de producto y un rango de fechas (fecha inicio y fecha fin) pasadas como parámetro, permita aplicar un % de descuento al precio de los productos pedidos de dicha categoría. **(0.75 ptos).**
- 3.1 Procedimiento o función que permita dar de baja todos los productos que no han sido nunca pedidos por ningún cliente. **(0.75 ptos).**
- 4.1 Procedimiento o función que permita, pasando como parámetro el nombre de una categoría obtener el importe total facturado en productos de dicha categoría. **(0.75 ptos).**
- 5.1 Función a la que pasando como parámetro el nombre de una provincia devuelva si algún cliente de dicha provincia ha realizado algún pedido. **(0.75 ptos).**
- 6.1 Función pedidos a la que se le pueden pasar dos parámetros y que tiene el siguiente comportamiento **(1.25 ptos)**. El parámetro1 siempre hay que pasarlo y en función de su valor hará los siguiente:

- 'provincias' me devuelve el importe total de los pedidos de todas las provincias
- 'clientes' devuelve el número de pedidos realizados por los clientes
- 'categorías' me devuelve el importe total de los pedidos de todas las categorías.

Si, además, como parámetro2 recibe el valor de un elemento correspondiente a la tabla del parámetro1 entonces filtra por dicho valor. Posibles ejemplos de esta función serían:

- ✓ `select pedidos('provincias', '') from provincias;` → mostraría importe total de todos los pedidos realizados por clientes de cualquier provincia.
- ✓ `select nombre, pedidos('provincias', cod_provincia) from provincias;` → mostraría cada provincia el importe de todos los pedidos realizados por clientes de la provincia pasada como parámetro 2.
- ✓ `select cod_cliente, nombre, apellido1, pedidos('clientes', '') from clientes;` → mostraría para cada cliente, el nº total de pedidos realizados por todos los clientes.
- ✓ `select cod_cliente, nombre, apellido1, pedidos('clientes', cod_cliente) from clientes;` → mostraría para cada cliente, el nº total de pedidos realizados por el cliente pasado como parámetro 2.
- ✓ `select nombre, pedidos('categorias', nombre) from categorias;` → mostraría para cada categoría el importe de todos los pedidos realizados de artículos pertenecientes a dicha categoría.

 <b>cpifp</b> <b>Bajo Aragón</b>	<b>DAM - BASES DE DATOS DISTANCIA</b>		 <b>IFC</b> INFORMÁTICA Y COMUNICACIONES
	<b>2ª EVALUACION</b>		
	FECHA: 07 / 05 / 2024	CURSO: 1º	

Firma del Alumno: \_\_\_\_\_

**Nota: cada apartado tiene que entregarse en un SCRIPT (3 scripts)**  
**en ellos se ha de hacer SIEMPRE referencia al punto que se está resolviendo.**

Esquema relacional usando la notación de Martin, con tablas y los atributos que las componen.

