PRACTICAS PL-SQL AVANZADO

1) Estructuras de control:

a. Escribir un script de pl/sql que imprima la tabla de códigos ascii con el siguiente formato:

033:! 034:" 035:# 036:$ 037:% 038:& 039:' 040:(
041:) 042:* 043:+ 044:, 045:- 046:. 047:/ 048:0
049:1 050:2 051:3 052:4 053:5 054:6 055:7 056:8
057:9 058:: 059:; 060:< 061:= 062:> 063:? 064:@
065:A 066:B 067:C 068:D 069:E 070:F 071:G 072:H

Empezar en el char 33 y terminar en el 256.

Para obtener el carácter ascii y formatear números con ceros, podemos utilizar:

```
dbms_output.put_line('Char 65: ' || chr(65));
dbms_output.put_line(to_char(1, '000'));
```

b. Empresa, case: Una sentencia SQL para obtener el número de pedidos que realiza cada compañía de envío y añadir una columna calculada que nos indique lo siguiente:

i. 150 a 250 → Simple, 250 a 300 → Normal, 300 a 350 → Super
ii. Se verá algo así:

ID	NOMBRE	CUENTA	CLASIFICACION
1	Speedy Express	248	Simple
2	United Package	327	Super
3	Federal Shipping	255	Normal

3 rows selected

c. Fechas, while: A partir de dos variables: mes y año de tipo number generar las fechas correspondientes de ese mes. Imprimirlas, teniendo en cuenta el número de días del mes, si el año es bisiesto, etc.

i. Para imprimir dbms_output.putline('texto' || var);

2) Cursores implícitos:

a. Escribir sentencias SQL e imprimir las propiedades de los cursores: Sentencias SQL que devuelvan o no resultados, y el número de filas afectadas.

3) Cursores explícitos:

a. Mostrar el id del cliente y % de ganancia que la empresa tiene por cada cliente. Teniendo en cuenta el importe total de todos los pedidos y de cada cliente. NO tener en cuenta el cargo del pedido, sólo operar con los detalles. mediante un bloque de PL/SQL. Utilizar un cursor parametrizado, capturar excepciones e imprimir por pantalla una pantalla similar a esta (utilizar los métodos open, close, fetch):

anonymous block completed
ALFKI:,7%
ANATR:,41%
ANTON:,48%
AROUT:,62%
BERGS:3,57%
BLAUS:,64%
BLONP:2,2%
….
….
WARTH:1,61%
WELLI:,86%
WHITC:2,35%
WILMK:,69%
WOLZA:,43%
Ganancia Total: 1304759
Número total de clientes: 89
Media por Cliente: 14660,21

4) Cursores de actualización:

Sobre la tabla de pedidos, incrementar en un 5% los pedidos de los clientes cuyo idcliente empiece por H.

5) Procedimientos Almacenados:

a. Escribir un procedimiento que intercambie dos variables. Escribir un bloque anónimo para probarlo.

b. Empresa: Definir un procedimiento que recibe como parámetro el id de un pedido y lo elimina. Utilizar transacciones y control de excepciones.
c. Implementar un procedimiento almacenado que genere n registros de notas, con la estructura de la tabla siguiente:

```
/**CREACION DE TABLA**/

CREATE TABLE "NOTAS" (

        "NOTA_ID" NUMBER NOT NULL ENABLE,

        "PRIMER_PARCIAL"  NUMBER,

        "SEGUNDO_PARCIAL" NUMBER,

        "EXAMEN_FINAL"    NUMBER,

        CONSTRAINT "NOTAS_PK" PRIMARY KEY ("NOTA_ID")

);
```

Para la generación de la notas de forma aleatoria se pueden utilizar las siguientes funciones. La clave la generamos de forma secuencial (utilizar un contador).

trunc(dbms_random.value(0,10),2) : Genera un valor entre 0 y 10, y lo trunca a 2 decimales.

d. Después del anterior implementar un procedimiento que devuelva cual es la nota (el id) más alta. También debe devolver la nota.

e. Otro procedimiento que imprima la notas que están aprobadas. Se tienen que dar las siguientes condiciones: que el examen final sea >= 5 o que la media de los dos parciales sea >= 5.

6) Funciones definidas por el usuario:

a. Implementar una función que compruebe si un año es bisiesto o no. Se cumple cuando es múltiplo de 400 o es múltiplo de 4 pero no de 100.

b. Escribir una función que reciba el idpedido y calcule el importe total. Tener en cuenta el cargo y el cálculo de los detalles.

c. Crear con SQL Developer una tabla Socios(login, pass): Escribir una función que compruebe si existe un socio en la base de datos. La función recibirá el login y las password por parámetro, devolverá un 0 cuando el usuario exista y la password sea correcta, 1 cuando solo exista el login y un 2 en caso contrario, no existe el socio.

d. Academia: Calcular el sueldo final de cada profesor sabiendo que tienen 12 pagas + 500 € por beneficios. El cálculo se realiza en una función. La función recibirá el código del profesor, si no existe devolverá -1. Después en un script escribir una consulta que utilice la función.

e. Academia: Calcular el sueldo de los profesores de la siguiente manera: si el sueldo es mayor o igual que 1000 se aumenta un 5% y si es menor 8%. El cálculo se realiza en una función, si no existe devolverá -1. Después en un script escribir una consulta que utilice la función.

7) **Triggers**:
   a. Empresa: Definir un Trigger que se dispare cuando se inserta un nuevo detalle será el encargado de actualizar las unidades en existencia de la tabla de productos.
   b. Hacer una modificación del anterior de tal forma que cuando se descuenten las existencias en la tabla de productos si las unidades en existencia tienen un valor interior a 10, se genera un registro en una tabla llamada tbpedido_proveedores ¿Qué información tendría esta tabla?.

8) **Paquetes**:
   a. Escribir un paquete: Gestion_pedidos, tendrá los siguientes procedimientos almacenados:
      i. Añadir un pedido, con los siguientes parámetros:
         1. Idcliente, nombre del empleado y el nombre de la compañía y el país.
            a. el cargo será un valor fijo de 30 €.
            b. La fecha pedido, la de hoy.
            c. La fecha envío pedido + 7 días.
            d. La fecha de entrega pedido + 15 días.
      ii. Añadir un detalle a un pedido: Devolverá un mensaje si lo ha conseguido agregar o no.
      iii. Eliminar un detalle de un pedido.
         1. Devolverá un mensaje si lo ha conseguido o no.
      iv. Añadir también a este paquete la función que calcula el importe de un pedido.
   b. Escribir un paquete y declarar un tipo record con los mismos campos que la tabla de Tbclientes, un cursor asociado a todos los clientes y una función que devuelva el número de clientes total. Escribir un bloque anónimo para probarlo: utilizar el cursor y llamar a la función.

9) **Estructuras:**
   a. a. Registros. Definir un registro con los campos de la tabla de ac_asignaturas, declarar variables y copiarlas. Insertar en la tabla a partir de la variable creada y después recuperar con una consulta, imprimir los resultados.

10) **Funciones predefinidas de Oracle:**
   a. Cadenas: Definir una variable, se le asigna un nombre (nombre apellido1 apellido2), escribir un bloque de código PL-SQL que reparta cada parte en una variable distinta. Después imprimirlas por pantalla. Siempre existen los dos espacios en blanco.

11) **EXECUTE IMMEDIATE**: A partir de la tabla pedidos crear una tabla para cada país, repartir los registros a cada tabla según corresponda.
Utilizar un cursor estático para obtener los distintos países. Luego mediante SQL Dinámico, crear un tabla para cada país seleccionando los registros: **create tabla TbpedidosAlemania as select * from Tbpedidos where paisdestinatario = 'Alemania'; Para las comillas simples utilizar el Chr(39).**
Tener en cuenta los espacios en blanco de los países.
El cursor estático utilizarlo con la sentencia **FOR IN.**
El script debe emitir el siguiente informe:
anonymous block completed
Se ha creado la tabla TbPedidosAlemania con 122 registros
Se ha creado la tabla TbPedidosArgentina con 16 registros
Se ha creado la tabla TbPedidosAustria con 40 registros
Se ha creado la tabla TbPedidosBelgica con 19 registros
Se ha creado la tabla TbPedidosBrasil con 83 registros
Se ha creado la tabla TbPedidosCanada con 30 registros
Se ha creado la tabla TbPedidosDinamarca con 18 registros
Se ha creado la tabla TbPedidosEspanya con 23 registros
Se ha creado la tabla TbPedidosEstadosUnidos con 122 registros
Se ha creado la tabla TbPedidosFinlandia con 23 registros
Se ha creado la tabla TbPedidosFrancia con 76 registros
Se ha creado la tabla TbPedidosIrlanda con 19 registros
Se ha creado la tabla TbPedidosItalia con 28 registros
Se ha creado la tabla TbPedidosMexico con 28 registros
Se ha creado la tabla TbPedidosNoruega con 6 registros
Se ha creado la tabla TbPedidosPolonia con 6 registros
Se ha creado la tabla TbPedidosPortugal con 13 registros
Se ha creado la tabla TbPedidosReinoUnido con 55 registros
Se ha creado la tabla TbPedidosSuecia con 38 registros
Se ha creado la tabla TbPedidosSuiza con 18 registros
Se ha creado la tabla TbPedidosVenezuela con 46 registros

Se puede implementar otro parecido para borrar.

12) **UTL_FILE**: Utilizando un cursor exportar a un fichero de texto el contenido de la tabla de clientes.

13) **Dbms_sql**: Utilizando este paquete exportar a un fichero de texto (CSV) como en la práctica anterior, pero en este caso la consulta será cualquiera y se escribirán los nombres de las columnas de los campos en la primera línea del fichero. Separar las columnas con un **;**