**ALUMNO: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ FECHA: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**NOTAS:**

1. Debéis adicionar los siguientes atributos en la Tabla CLIENTES de **DISTRIBUIDOR**:

NUMEROPEDIDOS, IMPORTE, ESTATUS. Acordaos de poner valor por defecto 0(cero) a los atributos numéricos.

1. Como creo que sabéis en estas bases de datos nos podemos encontrar con: clientes que no han comprado(pedido), productos que no se han vendido, empleados que no han vendido, “saltos” en los codigos, etc, etc.
2. **Debéis crear un único script llamado PL-SQL\_APELLIDOSNOMBRE.SQL con todos los ejercicios.** En caso de crear funciones y/o procedimientos adicionales debéis indicarlo y explicar los pasos necesarios para la realización de ese ejercicio como comentario(s).
3. **Los ejercicios** debéis nombrarlos como se exponen en el enunciado y ¡atención! **deben funcionar SI o SI y los datos deben ser correctos**.
4. Sólo podéis disponer para la realización del control lo entregado por el profesor debiendo seguir obligatoriamente las indicaciones del mismo, en cualquier otro caso tendréis un cero en el control y la posible pérdida de evaluación continua.
5. El profesor se reserva el derecho de exigir posteriormente la verificación de los ejercicios realizados.
6. **Es obligatorio que el script diseñado se encargue de borrarse a si mismo en caso de existir y de ejecutarse automáticamente**.

**EJERCICIOS:**

NOTA: las “rutinas” se deben crear independientemente del momento, es decir deben ser válidas para épocas y años venideros, debiendo mantener la integridad y funcionalidad de la BD en todo momento.

1.- (2p) Diseña una vista llamada “VISTA” de tal forma que nos salgan todos los empleados y el jefe a su cargo con los siguientes datos:

DATOSdelEMPLEADO(idempleado, tratamiento, nombre, apellidos, cargo),

DATOSdelJEFE(idempleado, tratamiento, nombre, apellidos, cargo)

Ordenados por jefe y empleado.

NOTA: en esta vista también debe aparecer el “mandamás” aunque no tenga jefe a su cargo.

2.- (3p) Crea un procedimiento llamado “PROCEDIMIENTO” de forma que se encargue de obtener y almacenar el NUMEROPEDIDOS e IMPORTE para cada CLIENTE asignándole el siguiente ESTATUS según el importe de sus pedidos:

IMPORTE < 5.000: MALO

IMPORTE entre 5.000 y 20.000: NORMAL

IMPORTE entre 20.000 y 50.000: BUENO

IMPORTE >= 50.000: EXCELENTE

3.- (2p) Crea ahora un Trigger o triggers llamado “DISPARADOR” de forma que a la hora de introducir un nuevo pedido se encargue de realizar lo siguiente:

1. almacenar automáticamente el PRECIOUNIDAD en LINEASPEDIDO.
2. Actualizar los atributos NUMEROPEDIDOS e IMPORTE en la tabla CLIENTES, debiendo obligatoriamente adicionar y NO recalcular y debiendo reasignar su posible nuevo ESTATUS
3. actualizar automáticamente el stock(UnidadesEnExistencia) de ese producto

4.- (3p) Crea ahora una función llamada “FUNCION” de forma que cuando se le invoque con el código de una Categoria(IdCategoria) nos devuelva igual que en el ejemplo la posición que ocupa en el ranking de ventas(en base a su importe de ventas/pedidos):

Ejemplo: CATEGORIA: 2 Condimentos RANKING: 6 IMPORTE: 113524