

## TEMA 8 (BUENO):

- 55) Selecciona los datos de los productos de los que haya como mucho 10 unidades en stock**  
**-- pero con el precio redondeado a un decimal.**

```
select stock, round(precio_venta,1) from productos
where stock <=10;
```

```
select round(precio_venta,1) from productos
where stock in (select stock from productos
where stock <=10) ;
```

- 56) Crea una consulta que muestre las dos soluciones de la siguiente ecuación de segundo grado**  
**-- grado  $3x^2+2x-5=0$ . Recordar que las soluciones de una ecuación de segundo grado son**

```
select (-2+(SQRT(POWER(2,2)-4*3*-5)))/(2*3) as x1,
(-2-(SQRT(POWER(2,2)-4*3*-5)))/(2*3) as x2 from dual;
```

- 57) Crea una consulta que calcule la raíz cuadrada de 49 y le sume el valor absoluto de (-5).**

```
select SQRT(49)+ABS(-5) from dual;
```

-- 59) Muestra el nombre y apellidos, ordenados alfabéticamente, de los empleados que han  
 -- vendido productos cuyo precio sea mayor que el precio medio de todos los productos.  
 -- Deben mostrarse los datos completamente en mayúsculas.

--MANERA 1:

```
SELECT UPPER(NOMBRE), UPPER(APELLIDOS)
FROM EMPLEADOS
  WHERE ID_EMPLEADO
        IN (SELECT ID_EMPLEADO
              FROM PEDIDOS
                WHERE NUMERO_PEDIDO
                      IN (SELECT NUMERO_PEDIDO
                            FROM DETALLES_PEDIDOS
                              WHERE NUMERO_PRODUCTO
                                    IN (SELECT
NUMERO_PRODUCTO
                                      FROM PRODUCTOS
                                        WHERE
PRECIO_VENTA >
                                      (SELECT
AVG(PRECIO_VENTA)
                                      FROM
PRODUCTOS)))) ORDER BY UPPER(APELLIDOS);
```

--MANERA 2:

```
SELECT DISTINCT UPPER(E.NOMBRE) , UPPER (E.APELLIDOS) FROM EMPLEADOS E
JOIN PEDIDOS P ON P.ID_EMPLEADO = E.ID_EMPLEADO
JOIN DETALLES_PEDIDOS DP ON P.NUMERO_PEDIDO = DP.NUMERO_PEDIDO
JOIN PRODUCTOS PD ON PD.NUMERO_PRODUCTO = DP.NUMERO_PRODUCTO
  WHERE PD.PRECIO_VENTA>(SELECT AVG(PRECIO_VENTA)
FROM PRODUCTOS)
  ORDER BY UPPER (E.APELLIDOS);
```

-- 63) Muestra los productos que pertenezcan a categorías en las que la longitud de su columna  
 -- descripción sea mayor de 8.

```
select ID_CATEGORIA from categorias where length(descripcion) >=8;
select * from productos where id_categoria in(select ID_CATEGORIA from categorias where
length(descripcion) >=8);
```

**-- 64) Muestra los nombres de los empleados con longitud máxima 10 y rellena los que sean**  
**-- menor que 10 con \* por la derecha hasta que tengan longitud 10**

```
select nombre from empleados where length(nombre)<=10;
select RPAD(nombre, 10, '*') from empleados where nombre in (select nombre from empleados where length(nombre)<=10) ;
```

**-- 67) Para cada pedido muestra el numero del pedido, el identificador del cliente, el nombre del**  
**-- cliente y el mes en que fue realizado el pedido (solo el mes).**

```
--select ID_CLIENTE, NOMBRE from clientes;
--select NUMERO_PEDIDO, fecha_pedido, id_cliente from pedidos;
select p.NUMERO_PEDIDO, p.id_cliente , TO_CHAR(p.fecha_pedido, 'MON') , c.nombre
from pedidos p
join clientes c
on (p.id_cliente =c.id_cliente);
```

**-- 68) Ahora muestra las mismas columnas que en la consulta anterior, pero la fecha de pedido**  
**-- debe aparecer en el siguiente formato ('dd-mm-yyyy'). Hay que utilizar la función**  
**-- TO\_CHAR.**

```
select p.NUMERO_PEDIDO, p.id_cliente , TO_CHAR(p.fecha_pedido, 'dd-mm-yyyy') ,
c.nombre
from pedidos p
join clientes c
on (p.id_cliente =c.id_cliente );
```

**-- 69) Obtener qué fecha será el próximo miércoles.**

```
select NEXT_DAY(sysdate, 3) from dual;
```

**-- 70) Obtener los nombres de los productos (nombre) ordenados por el número de caracteres que**  
**-- tienen.**

```
select LENGTH(nombre), nombre from PRODUCTOS order by length(nombre);
```