1. **Selecciona los datos de los productos de los que haya como mucho 10 unidades en stock pero con el precio redondeado a un decimal.**
2. **Crea una consulta que muestre las dos soluciones de la siguiente ecuación de segundo grado 3X2+2x-5=0. Recordar que las soluciones de una ecuación de segundo grado son**

…,,l,l,l,

1. **Crea una consulta que calcule la raíz cuadrada de 49 y le sume el valor absoluto de (-5).**
2. **Muestra el precio de venta medio de todos los productos.**

select avg(precio\_venta) from productos;

1. **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.**

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))));

1. **Calcula el número de productos que hay en la categoría 3.**

select \* from productos where id\_categoria = ‘3’;

1. **Muestra el precio de venta mayor y el precio de venta menor de los productos.**

select min(precio\_venta), max(precio\_venta) from productos;

1. **Muestra los datos del producto o productos que tienen el precio más alto.**

select \* from productos where precio\_venta=(select max(precio\_venta) from productos);

1. **Muestra los productos que pertenezcan a categorías en las que la longitud de su columna descripción sea mayor de 8.**

select \* from productos where id\_categoria in(select id\_categoria from categorias where length(descripcion) > 8);

1. **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 RPAD(nombre, 10,'\*') from clientes;

1. **Actualiza los datos de la tabla PRODUCTOS de manera que en la columna descripción de los productos que pertenecen a la categoría 1 ponga 'Es un accesorio'.**

update productos set descripcion='Es un accesorio' where id\_categoria=1;

**Actualiza los datos de la tabla PRODUCTOS de manera que en la columna descripción de los productos que pertenecen a la categoría 2 ponga 'Es una bicicleta'.**

update productos set descripcion='Es una bicicleta' where id\_categoria=2;

**Muestra ahora el nombre, descripción y el precio de venta de todos los productos, en caso de que la descripción tenga valor nulo debe aparecer el texto 'PRODUCTO SIN DESCRIPCIÓN'.**

select nombre, nvl(descripcion,'PRODUCTO SIN DESCRIPCIÓN'), precio\_venta from productos;

1. **Calcula el número de productos que no tienen descripción (valor nulo en esa columna).**

select \* from productos where descripcion is null;

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

select DISTINCT p.numero\_pedido, p.id\_cliente, c.nombre,

substr(p.fecha\_pedido,4,2) as "MES" from pedidos p, clientes c

where p.id\_cliente=c.id\_cliente;

1. **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 DISTINCT p.numero\_pedido, p.id\_cliente, c.nombre,

to\_char(p.fecha\_pedido,('dd-mm-yyyy'))AS "FECHA" from pedidos p, clientes c

where p.id\_cliente=c.id\_cliente;

1. **Obtener qué fecha será el próximo miércoles.**

select DISTINCT next\_day(sysdate,'Miércoles') from clientes;

1. **Obtener los nombres de los productos ordenados por el número de caracteres que tienen.**

select nombre from productos order by length(nombre);