

1ºParte XML a neodatis

Consulta 1

Consola:

El producto con mayor cantidad de ventas en unidades es: Lapiz con 30 unidades.

Código:

```
private static void encontrarProductoConMayorCantidadDeVentas() {
    ODB odb = null;
    try {
        odb = ODBFactory.open("NeodatisDB/ExamDB1.odb");
        Objects<Articulo> articulos = odb.getObjects(Articulo.class);
        Map<String, Integer> ventasPorProducto = new HashMap<>();

        // Se suma las ventas por producto
        while (articulos.hasNext()) {
            Articulo articulo = articulos.next();
            ventasPorProducto.put(articulo.getProducto(),
                ventasPorProducto.getOrDefault(articulo.getProducto(), 0) + articulo.getCantidad());
        }

        // Se encuentra el producto con la mayor cantidad de ventas
        String productoConMasVentas = null;
        int maxVentas = 0;
        for (Map.Entry<String, Integer> entry : ventasPorProducto.entrySet()) {
            if (entry.getValue() > maxVentas) {
                maxVentas = entry.getValue();
                productoConMasVentas = entry.getKey();
            }
        }

        // Se muestra el resultado
        if (productoConMasVentas != null) {
            System.out.println("\nEl producto con mayor cantidad de ventas en unidades es: " + productoConMasVentas
                + " con " + maxVentas + " unidades.");
        } else {
            System.out.println("No se encontraron ventas.");
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (odb != null) {
            odb.close();
        }
    }
}
```

Consulta 2

Consola:

```
Total de unidades vendidas por tipo Neodatis:
- tipo: Dibujo, Total Unidades: 10
- tipo: Escritura, Total Unidades: 95
- tipo: Manualidades, Total Unidades: 10
- tipo: Almacenamiento, Total Unidades: 17
```

Codigo:

```
private static void listarTotalUnidadesVendidasPorTipo() {
    ODB odb = null;
    try {
        odb = ODBFactory.open("NeodatisDB/ExamDB1.odb");

        // Se crea una consulta que agrupa las ventas por color y suma las cantidades.
        IValuesQuery query = new ValuesCriteriaQuery(Articulo.class).field("tipo").sum("cantidad").groupBy("tipo");
        Values values = odb.getValues(query);

        System.out.println("\nTotal de unidades vendidas por tipo Neodatis:");
        while (values.hasNext()) {
            ObjectValues value = values.next();
            String tipo = (String) value.getByAlias("tipo");
            BigDecimal totalUnidades = (BigDecimal) value.getByAlias("cantidad");
            System.out.println("- tipo: " + tipo + ", Total Unidades: " + totalUnidades);
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (odb != null) {
            odb.close();
        }
    }
}
```

Consulta 3

Consola:

```
costeTotal=290.0, €
```

Codigo :

```
private static void consultaInventario() {
    // Abre la conexión con la base de datos NeoDatis
    ODB odb = ODBFactory.open("NeodatisDB/ExamDB1.odb");

    // Crea la consulta para calcular el inventario total
    IValuesQuery query = new ValuesCriteriaQuery(Articulo.class).sum("costeTotal");

    // Ejecuta la consulta y obtén los resultados
    Values values = odb.getValues(query);

    // Obtiene el resultado de la consulta
    Object result = values.getFirst();

    // Muestra el resultado
    if (result != null) {
        System.out.println("\n " + result + " €");
    } else {
        System.out.println("No se encontraron resultados para la consulta.");
    }

    // Cierra la conexión con la base de datos
    odb.close();
}
```

2ºParte neodatis a SQL

Consulta 4:

Consola:

```
El producto con la mayor cantidad de ventas en unidades es: Lapiz con 30 unidades.
```

Codigo:

```
public static void encontrarProductoConMayorVentaSQLite() {
    try {
        // Se crea y ejecuta la consulta SQL para encontrar el producto con la mayor cantidad de unidades vendidas
        String sql = "SELECT producto, SUM(cantidad) AS total_unidades FROM Articulos GROUP BY producto ORDER BY total_unidades DESC LIMIT 1";
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(sql);

        // Verificar si hay resultado y mostrar el producto con la mayor cantidad de unidades vendidas
        if (rs.next()) {
            String producto = rs.getString("producto");
            int totalUnidades = rs.getInt("total_unidades");
            System.out.println("\nEl producto con la mayor cantidad de ventas en unidades es: " + producto + " con " + totalUnidades + " unidades.");
        } else {
            System.out.println("\nNo se encontraron ventas de articulo.");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Consulta 5

Consola:

```
Ventas por tipo:  
- Tipo: Almacenamiento, Total Unidades: 17  
- Tipo: Dibujo, Total Unidades: 10  
- Tipo: Escritura, Total Unidades: 95  
- Tipo: Manualidades, Total Unidades: 10
```

Codigo:

```
public static void encontrarProductoConMayorVentaSQLite() {  
    try {  
  
        // Se crea y ejecuta la consulta SQL para encontrar el producto con la mayor cantidad de unidades vendidas  
        String sql = "SELECT producto, SUM(cantidad) AS total_unidades FROM Articulos GROUP BY producto ORDER BY total_unidades DESC LIMIT 1";  
        Statement stmt = con.createStatement();  
        ResultSet rs = stmt.executeQuery(sql);  
  
        // Verificar si hay resultado y mostrar el producto con la mayor cantidad de unidades vendidas  
        if (rs.next()) {  
            String producto = rs.getString("producto");  
            int totalUnidades = rs.getInt("total_unidades");  
            System.out.println("\nEl producto con la mayor cantidad de ventas en unidades es: " + producto + " con " + totalUnidades + " unidades.");  
        } else {  
            System.out.println("\nNo se encontraron ventas de articulo.");  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Consulta 6

Consola:

```
Ventas :  
290.0€
```

Codigo:

```
}  
public static void calcularInventarioSql() {  
    try {  
  
        // Crear y ejecutar la consulta SQL para sumar la columna cantidad agrupada por tipo  
        String sql = "SELECT SUM(costeTotal) AS total FROM Articulos GROUP BY costeTotal";  
        Statement stmt = con.createStatement();  
        ResultSet rs = stmt.executeQuery(sql);  
        double totalisimo = 0;  
        // Mostrar el total de unidades vendidas por cada color  
        System.out.println("\nVentas :");  
        while (rs.next()) {  
  
            double total = rs.getDouble("total");  
            totalisimo += total;  
            //System.out.println("- Total: " + total);  
        }  
        System.out.println("\ntotalisimo + "€");  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```
