

EJERCICIO RESUELTO

Módulo: Programación

Gestión BD de Coches

Descripción:

En la presente unidad hemos visto como conectarnos con distintos SGBD como puede ser MySQL y como realizar sobre el operaciones de DML.

Vamos a crear una aplicación que nos permita gestionar una base de datos de para gestionar vehículos de manera funcional.

Por lo tanto en este ejercicio vamos a realizar las siguientes tareas:

- Crear una BD en MySQL de coches.
- Realizar una clase que conecte con el SGBD MySQL.
- Realizar una clase que nos permitan realizar las siguientes operaciones:
 - Mostrar todos los coches
 - Crear un nuevo coche
 - Actualizar la información de un coche
 - Eliminar un coche
 - Buscar un coche por su matrícula.
- Los datos que deseamos almacenar para cada coche son:
 - Matrícula
 - Precio
 - Color
 - Marca

Objetivos:

- Usar sentencias SQL para el DML (Data Manipulation Lenguaje).
- Establecer la conexión.
- Realizar una aplicación completa para la gestión de una BD.

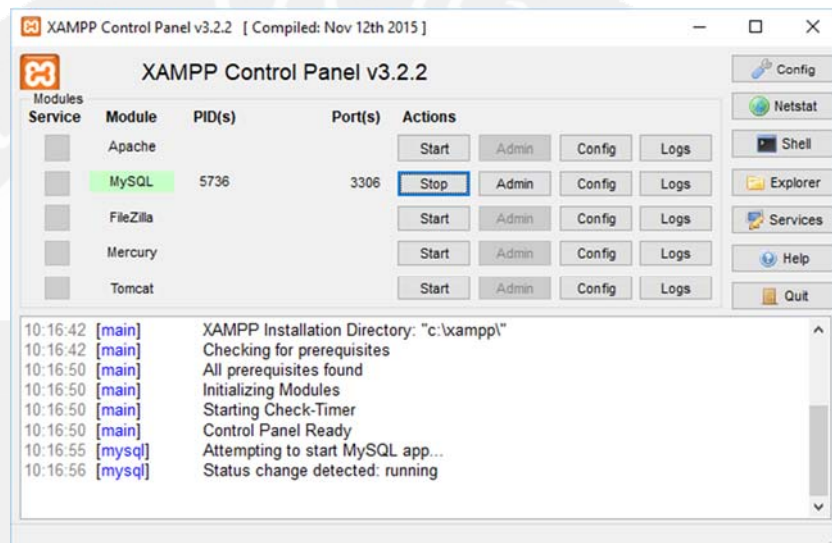
Recursos:

- Acceso a Internet.

- Netbeans.
- JDBC.
- Driver MySQL.

Resolución:

Lo primero que debemos realizar es levantar el un servicio SQL, el de XAMPP, por ejemplo.



Después creamos la base de datos y la tabla:

```
MariaDB [(none)]> CREATE DATABASE COCHES;
Query OK, 1 row affected (0.01 sec)
```

```
MariaDB [(none)]> use coches;
Database changed
```

```
MariaDB [coches]> CREATE TABLE coche
-> (
-> matricula VARCHAR(20) PRIMARY KEY,
-> precio FLOAT(7,2),
-> color VARCHAR(20),
-> marca VARCHAR(20)
-> );
Query OK, 0 rows affected (0.41 sec)
```

Ahora, creamos un proyecto Java y añadimos en el la librería de cliente de mariadb. Creamos una clase "Database" para la interacción con la base de datos, que sería así:

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;

public class Database {
    private Connection conn;

    public Database() {
        try {
            Class.forName("org.mariadb.jdbc.Driver");

            conn = DriverManager.getConnection(
                "jdbc:mariadb://localhost/COCHES",
                "root", ""
            );
            System.out.println("\n-----Conexión "
                + "establecida-----");
        } catch (ClassNotFoundException | SQLException e) {
            System.err.println("No se puede establecer la "
                + "conexión");
            System.exit(0);
        }
    }

    public void showAll() {
        System.out.println("-----Mostrando todos"
            + " los vehiculos-----");
        String q = "SELECT * from coche";
        executeSelect(q);
    }

    public void addCar(String matricula, double precio,
        String color, String marca) {
        System.out.println("-----Añadiendo el vehiculo"
            + " con matricula "+matricula+"-----");
        String query = "INSERT INTO coche values ('"
            +matricula+"', '"+precio+", '"
            +color+"', '"+marca+"')";
        executeQuery(query, "insertado");
    }

    public void updateCar(String matricula, double precio,
        String color, String marca) {
        System.out.println("-----Modificando el vehiculo"
            + " con matricula "+matricula+"-----");
        String query = "update coche set "
            + "precio="+precio+", " + "color='"+color+"', "
            + "marca='"+marca+" ' " + "where matricula='"+matricula+"'; ";
        executeQuery(query, "actualizado");
    }

    public void deleteCar(String matricula) {
        System.out.println("-----Eliminado el vehiculo"
            + " con matricula "+matricula+"-----");
        String query = "delete from coche "
            + "where matricula='"+matricula+"'; ";
        executeQuery(query, "eliminado");
    }
}

```

```

public void searchCar(String matricula){
    System.out.println("-----Mostrando el vehiculo"
        + "con matricula "+matricula+"-----");
    String q = "SELECT * from coche where matricula='"+matricula+"'";
    executeSelect(q);
}

private void executeQuery(String q, String action){
    try{
        int n = conn.prepareStatement(q).executeUpdate();
        System.out.println("\nSe han "+action+" "+n+" vehiculos");
    }catch(SQLException e){
        System.err.println("\nNo se pudo realizar la operacion");
    }
}

private void executeSelect(String q){
    try{
        ResultSet result = conn.createStatement().executeQuery(q);
        System.out.println("\nCoches encontrados en la BD: ");
        while(result.next()){
            System.out.println("\nMatricula: "+result.getString(1)+
                "\nPrecio: "+result.getDouble(2)+
                "\nColor: "+result.getString(3)+
                "\nMarca: "+result.getString(4));
        }
    }catch(SQLException e){
        System.err.println("No se pudo realizar la consulta");
    }
}
}

```

Después, esta sería la clase principal con el main:

```

package gestion_bd_coches;

public class Gestion_BD_Coches {

    public static void main(String[] args) {
        Database db = new Database();
        db.addCar("A 1234 AB", 14534.5, "rojo", "Seat");
        db.addCar("B 5678 CD", 12514.5, "amarillo", "Ford");
        db.addCar("E 9012 FG", 10514.5, "amarillo", "Audi");

        db.showAll();

        db.updateCar("A 1234 AB", 4534.5, "verde", "seat");

        db.searchCar("A 1234 AB");

        db.deleteCar("A 1234 AB");
    }
}

```

Finalmente, el resultado sería el siguiente:

```
run:

-----Estableciendo conexión-----

-----Insertando vehiculos-----

Se han insertado 1 vehiculos

Se han insertado 1 vehiculos

Se han insertado 1 vehiculos

-----Recuperando todos los vehiculos-----

Coches encontrados en la BD:

Matricula: A 1234 AB
Precio: 14534.5
Color: rojo
Marca: Seat

Matricula: B 5678 CD
Precio: 12514.5
Color: amarillo
Marca: Ford

Matricula: E 9012 FG
Precio: 10514.5
Color: amarillo
Marca: Audi

-----Actualizando el coche con matricula 'A 1234 AB'-----

Se han actualizado 1 vehiculos

-----Buscando el coche con matricula 'A 1234 AB'-----

Coches encontrados en la BD:

Matricula: A 1234 AB
Precio: 4534.5
Color: verde
Marca: seat

-----Eliminado el coche con matricula 'A 1234 AB'-----

Se han eliminado 1 vehiculos
BUILD SUCCESSFUL (total time: 0 seconds)
```