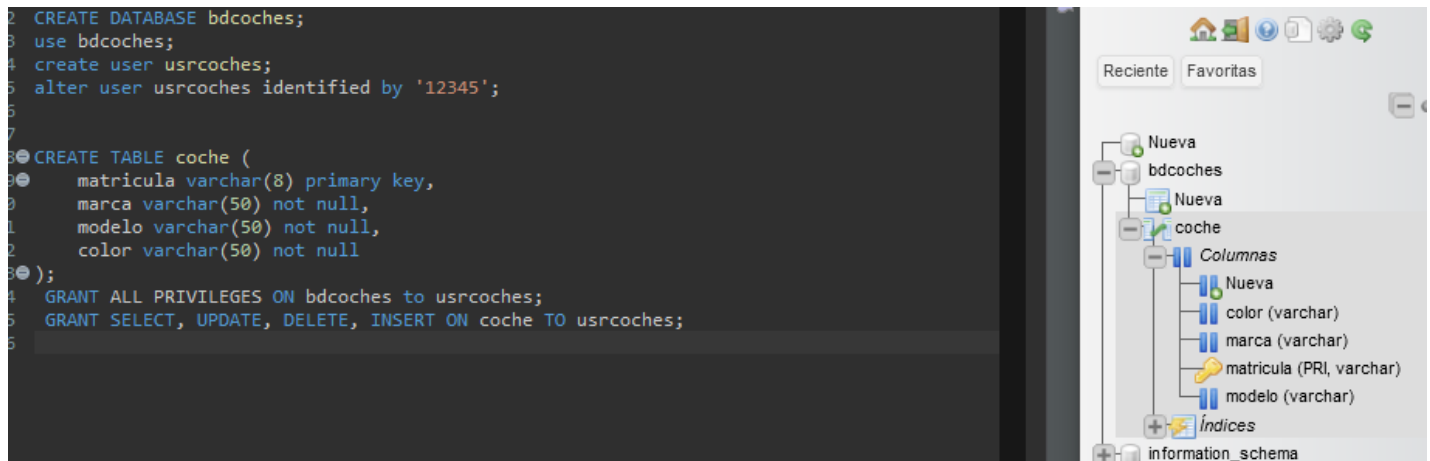


CICLO FORMATIVO DE GRADO SUPERIOR: DESARROLLO DE  
APLICACIONES MULTIPLATAFORMA  
2º CURSO  
MODULO : Acceso a Datos  
UT02  
EXAMEN 1ª Evaluación  
ALUMNO: Ginés Gutiérrez Saura

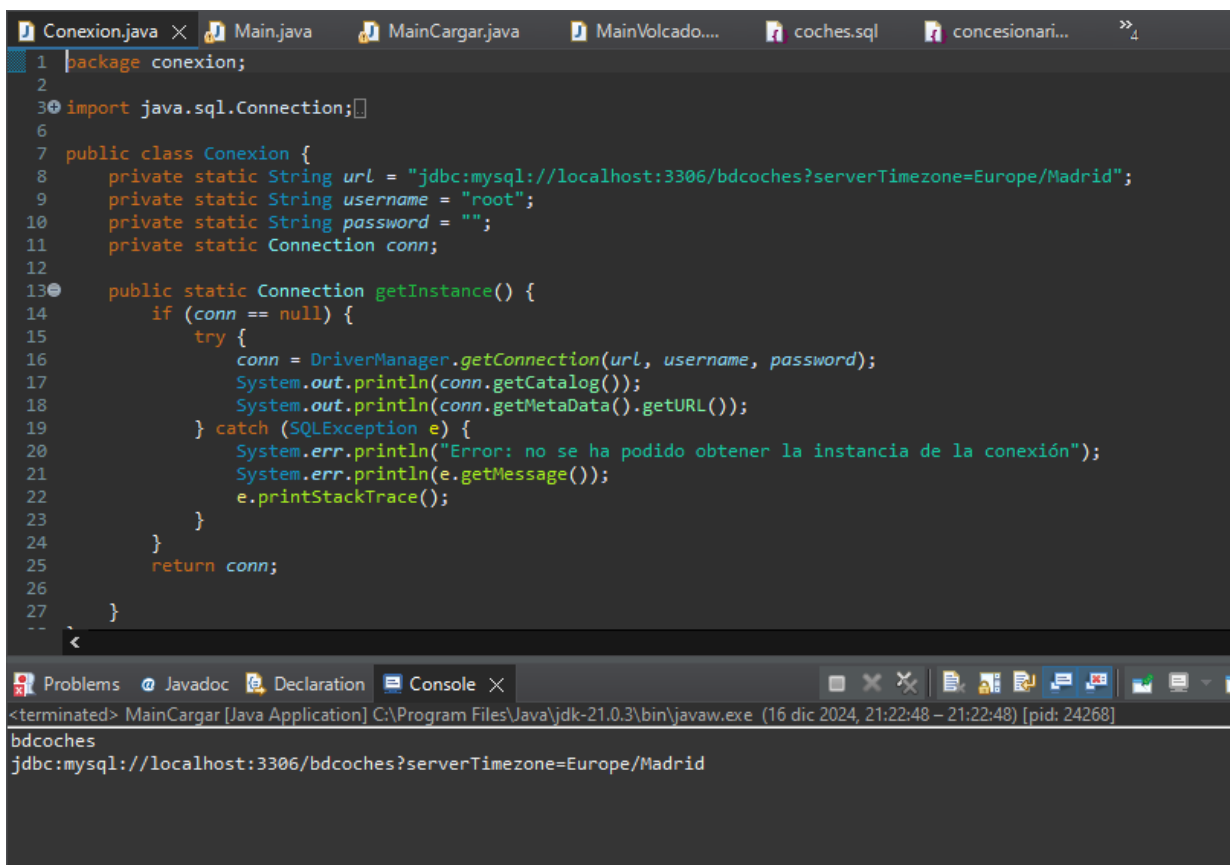
a) Ejecución coches.sql:

Vista bd/tabla una vez ejecutado el comando en phpmyadmin



b) Crear conexión a BD MySQL

La conexión está creada y se usa siguiendo el patrón singleton, usa el nuevo usuario creado (en la versión inicial si está, en esta no lo pongo por si da algún error) en "coches.sql", he hecho que imprima por consola la información: nombre de la base de datos y url. Al usar el patrón singleton, esta información sólo se muestra la primera vez que se interactúa con la conexión, es decir, la primera vez que se realiza una operación.



c) Creación CRUD siguiendo pautas de la asignatura: creado bajo el nombre de la clase “CocheRepository” en el paquete reposirot, implementa la interfaz “Repository” .

## INTERFACE REPOSITORY

```
1 package repository;
2
3 public interface Repository<Coche> {
4     void insertar(Coche coche);
5     void actualizar(String matricula, String marca, String modelo);
6     void eliminar(String matricula);
7 }
8
```

## Operaciones CRUD + Consola realizando todas las operaciones

```
1 package repository;
2
3 import java.sql.Connection;
4
5 public class CocheRepository implements Repository<Coche> {
6     private static Connection getConnection() throws SQLException {
7         return Conexion.getInstance();
8     }
9
10    @Override
11    public void insertar(Coche coche) {
12        String sql = "INSERT INTO coche (matricula, marca, modelo, color) VALUES (?, ?, ?, ?)";
13        try (PreparedStatement ps = getConnection().prepareStatement(sql)) {
14            ps.setString(1, coche.getMatricula());
15            ps.setString(2, coche.getMarca());
16            ps.setString(3, coche.getModelo());
17            ps.setString(4, coche.getColor());
18            ps.executeUpdate();
19        } catch (SQLException e) {
20            System.err.println("Error: al insertar coche");
21            System.err.println(e.getMessage());
22        }
23    }
24
25    @Override
26    public void actualizar(String matricula, String marca, String modelo) {
27        String sql = "UPDATE coche SET marca=?, modelo=? WHERE matricula=?";
28        try (PreparedStatement ps = getConnection().prepareStatement(sql)) {
29            ps.setString(1, marca);
30            ps.setString(2, modelo);
31            ps.setString(3, matricula);
32            ps.executeUpdate();
33        } catch (SQLException e) {
34            System.err.println("Error: al actualizar coche con matricula=" + matricula);
35            System.err.println(e.getMessage());
36        }
37    }
38
39    @Override
40    public void eliminar(String matricula) {
41        String sql = "DELETE FROM coche WHERE matricula=?";
42        try (PreparedStatement ps = getConnection().prepareStatement(sql)) {
43            ps.setString(1, matricula);
44            ps.executeUpdate();
45        } catch (SQLException e) {
46            System.err.println("Error: error al eliminar coche con matricula=" + matricula);
47            System.err.println(e.getMessage());
48        }
49    }
50 }
```

```
Main (6) [Java Application] C:\Program Files\Java\jdk-21.0.3\bin\javaw.exe (16 dic 2024, 21:33)
[1] Insertar Coche
[2] Actualizar Coche
[3] Eliminar Coche
[0] Salir
1
Inserta a continuación: Matricula + Marca + Modelo + Color
111
PRIMER
Corolla
Rojo
[1] Insertar Coche
[2] Actualizar Coche
[3] Eliminar Coche
[0] Salir
1
Inserta a continuación: Matricula + Marca + Modelo + Color
222
SEGUNDO
Scenic
Azul
[1] Insertar Coche
[2] Actualizar Coche
[3] Eliminar Coche
[0] Salir
2
Inserta a continuación: Matricula + Marca + Modelo
111
Actualizado
ModeloNuevo
[1] Insertar Coche
[2] Actualizar Coche
[3] Eliminar Coche
[0] Salir
3
Insertar Matricula:
222
[1] Insertar Coche
[2] Actualizar Coche
[3] Eliminar Coche
[0] Salir
```

## BDD CON DOS PRIMEROS INSERT

	matricula	marca	modelo	color
ar 111		PRIMER	Corolla	Rojo
ar 222		SEGUNDO	Scenic	Azul

-----

## BDD DESPUÉS UPDATE Y DELETE

	matricula	marca	modelo	color
ir 111			Actualizado	ModeloNuevo
				Rojo

#### d) Volcado de coches con una marca en común (volvo)

He capturado tanto el código, como la bdd y el fichero txt con los datos ya volcados para que quede constancia de que son los mismos datos y la clase funciona

The screenshot displays a Java IDE with the following components:

- MainVolcado.java:** Contains a `main` method that connects to a database, executes a query `"SELECT * FROM coche WHERE marca = ?"` with the parameter `"Volvo"`, and writes the results to `ficheros\coche.txt`. The code uses `PreparedStatement` and `ResultSet` to fetch data and `BufferedWriter` to write it.
- Console:** Shows the output `Volcado completado en: ficheros\coche.txt`.
- coche.txt:** A text file containing the exported data:

```
1 333,Volvo,ModeloGuapo,Rojo
2 666,Volvo,Modelo9,Azul
3 777,Volvo,Tercer Volvo,Rosa
4 |
```
- Database Query Result:** A table showing the results of the SQL query. The table has columns: `matricula`, `marca`, `modelo`, and `color`. The results are:

matricula	marca	modelo	color
111	Actualizado	ModeloNuevo	Rojo
222	SEGUNDO	Scenic	Azul
333	Volvo	ModeloGuapo	Rojo
444	Coche	Común	Amarillo
666	Volvo	Modelo9	Azul
777	Volvo	Tercer Volvo	Rosa

Handwritten annotations in pink include "BDD" (Database) above the query result table, "CONFIRMACIÓN" (Confirmation) next to the console output, and "Volcado" (Export) next to the text file content.

e) Con respecto a la primera entrega, lo único que se modifica en el proyecto es **este apartado**,  
 Anotación de la entrega en el examen:  
 Tabla y Clase creadas (archivo sql dentro del proyecto)  
 El main está medio a implementar, no es funcional, no he sido capaz de acabarlo

Comparación entre clases  
 (la diferencia son un par de líneas que no puede acabar en el exámen por tiempo)

Acabada (subida nota)

“Casi” implementada (examen)

```

13 public class MainCargar {
14     private static Connection getConnection() throws SQLException {
15         return Conexion.getInstance();
16     }
17     public static void main(String[] args) throws FileNotFoundException {
18         FileReader filePath = new FileReader("concesionario.dat");
19         String linea;
20         String[] propiedades;
21         String sql = "Insert INTO concesionario (id, nombreConcesionario, dir, te
22
23         Concesionario concesionario = new Concesionario();
24         try (PreparedStatement ps = getConnection().prepareStatement(sql);
25             BufferedReader br = new BufferedReader(filePath)) {
26
27             while((linea = br.readLine())!=null) {
28
29                 propiedades=linea.split("#");
30                 ps.setInt(1, Integer.parseInt(propiedades[0].trim()));
31                 ps.setString(2, propiedades[1].trim());
32                 ps.setString(3, propiedades[2].trim());
33                 ps.setString(4, propiedades[3].trim());
34                 ps.setString(5, propiedades[4].trim());
35                 ps.executeUpdate();
36
37             }
38         } catch (Exception e) {
39             System.err.println("Error: al conectar con la base de datos: ");
40             System.err.println(e.getMessage());
41         }
42     }
43 }
44 }
45 }
  
```

```

1 package vista;
2
3 import java.io.BufferedReader;
4
5 public class MainCargar {
6     private static Connection getConnection() throws SQLException {
7         return Conexion.getInstance();
8     }
9     public static void main(String[] args) throws FileNotFoundException {
10         FileReader filePath = new FileReader("concesionario.dat");
11         String linea;
12         String[] propiedades;
13         String sql = "Insert INTO concesionario (id, nombreConcesionario, dir, te
14
15         Concesionario concesionario = new Concesionario();
16         try (PreparedStatement ps = getConnection().prepareStatement(sql);
17             BufferedReader br = new BufferedReader(filePath)) {
18
19             linea = br.readLine();
20             propiedades=linea.split(" ");
21             ps.setInt(1, Integer.parseInt(propiedades[0]));
22             ps.setString(2, propiedades[1]);
23             ps.setString(3, propiedades[2]);
24             ps.setString(4, propiedades[3]);
25             ps.setString(5, propiedades[4]);
26             ResultSet rs = ps.executeQuery();
27
28         } catch (Exception e) {
29             System.err.println("Error: al conectar con la base de datos: ");
30             System.err.println(e.getMessage());
31         }
32     }
33 }
34 }
35 }
  
```

## DATOS CARGADOS EN LA BDD

	id	nombreConcesionario	dir	telefono	poblacion
Borrar	1	Maven Coches	Avda estacion S/N	924021111	Almendralejo
Borrar	2	Hermanos Herrera	Avda España S/N	924020303	Badajoz
Borrar	3	CiberCoches SA	C/ internet	655888111	Badajoz