Rapport Mini-projet

Gestion d'un cabinet médical

Préparé par :

Wissal Ajbabdi

Introduction

D’après cette base de données, on crée une application qui gère un ensemble des clients et des médecins après une authentification.

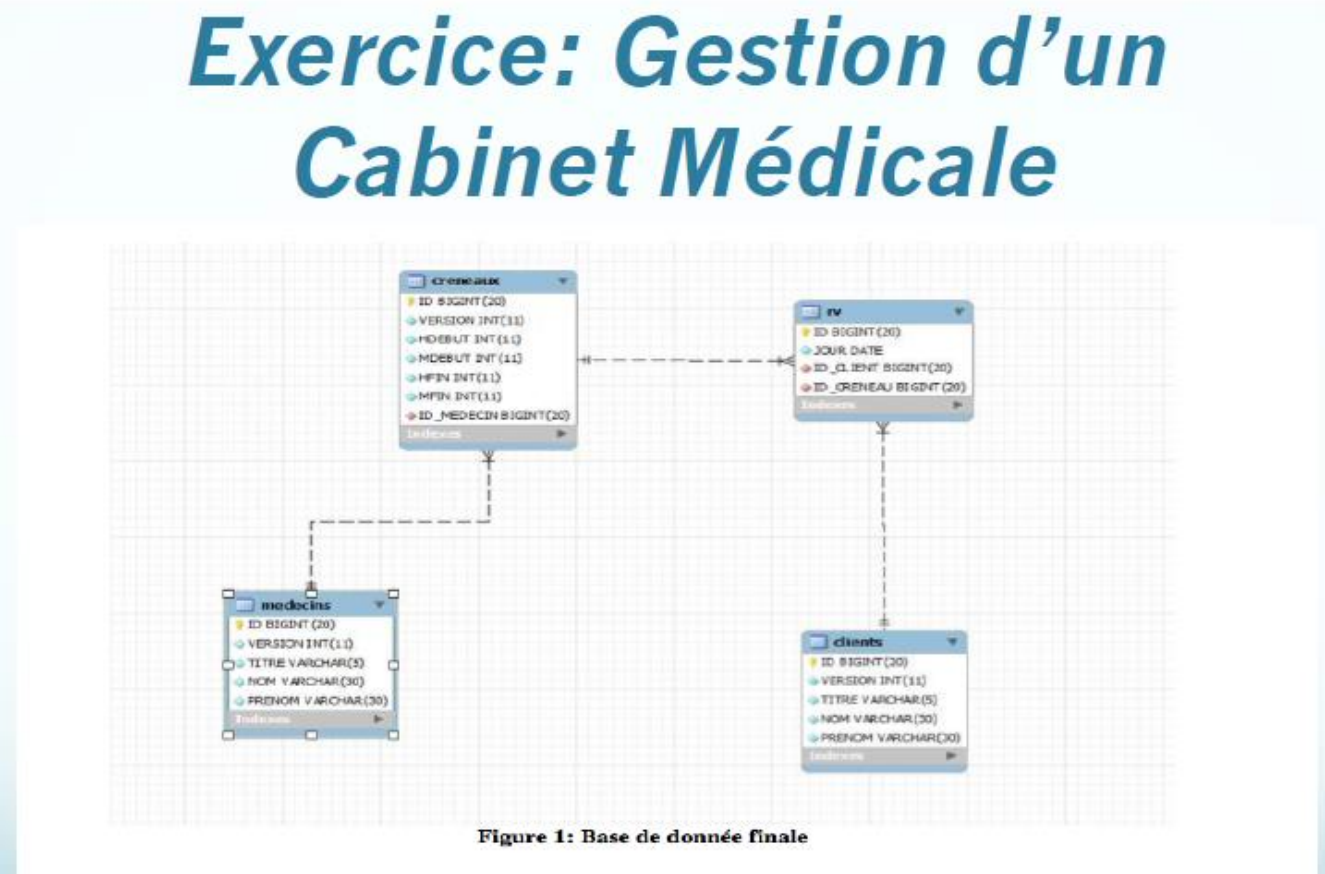


Figure 1: Base de donnée finale

On commence par les classes :

Classe client.java :

```
package my.projet.classes;

public class Client {

	private int id;
	private String version;
	private String sexe;
	private String nom;
	private String prenom;

	public Client(int id,String version, String sexe,String nom, String prenom) {
		this.id=id;
		this.version=version;
		this.sexe=sexe;
		this.nom=nom;
		this.prenom=prenom;
	}

	public Client() {}
```

```
	public int getId() {
		return id;
	}

	public void setId(int id) {
		this.id = id;
	}

	public String getVersion() {
		return version;
	}

	public void setVersion(String version) {
		this.version = version;
	}

	public String getSexe() {
		return sexe;
	}

	public void setSexe(String sexe) {
		this.sexe = sexe;
	}

	public String getNom() {
		return nom;
	}

	public void setNom(String nom) {
		this.nom = nom;
	}

	public String getPrenom() {
		return prenom;
	}

	public void setPrenom(String prenom) {
		this.prenom = prenom;
	}


}
```

Interface DAO.java :

```
package DAO;

import java.util.List;

public interface IDAO<T> {

	boolean create(T obj);
	boolean update(T obj);
	boolean delete(T obj);
	T findById(int id);
	List<T> findAll();

}
```

Classe ClientService.java qui implémente l'interface DAO.java :

```
package my.projet.service;


import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;


import DAO.IDAO;

import my.projet.classes.Client;

import my.projet.connection.Connexion;


public class ClientService implements IDAO<Client>{


	//@Override

	public boolean create(Client obj) {

		try {

			//Création d'une requete SQL

			String req="insert into client (version,titre,nom,prenom)
values("+obj.getVersion()+","+obj.getSexe()+","+obj.getNom()+","+obj.getPrenom()+")";

			//Création d'un statement

			Statement st= Connexion.getConnection().createStatement();

			//Exécution de la requete

			if(st.executeUpdate(req)==1) {

				return true;

			}

		}catch(SQLException e) {

			System.err.println("Erreur SQL!!");

		}

		return false;

	}


	public boolean update(Client obj) {

		try {

			String req="update client set version="+obj.getVersion()+", titre= ="+obj.getSexe()+", nom
="+obj.getNom()+", prenom ="+obj.getPrenom();
```

```
                Statement st=Connexion.getConnection().createStatement();
                if(st.executeUpdate(req)==1) {
                        return true;
                }
            }catch(SQLException e) {
                System.err.println("Erreur SQL!!");
            }
            return false;
        }

        public boolean delete(Client obj) {
            try {
                String req="delete from client where id="+obj.getId();
                Statement st=Connexion.getConnection().createStatement();
                if(st.executeUpdate(req)==1) {
                        return true;
                }
            }catch(SQLException e) {
                System.err.println("Erreur SQL!!");
            }
            return false;
        }

        public Client findById(int id) {
            try {
                String req="select * from client where id="+id;
                Statement st=Connexion.getConnection().createStatement();
                ResultSet rs=st.executeQuery(req);
                if(rs.next()) {
                        return new
Client(rs.getInt("ID"),rs.getString("VERSION"),rs.getString("SEXE"),rs.getString("NOM"),rs.getString("PRENOM"));
                }
            }catch(SQLException e) {
                System.err.println("Erreur SQL!!");
            }
            return null;
        }
```

```
	public List<Client> findAll(){
		List<Client> c=new ArrayList<>();
		try {
			String req="select * from client";
			Statement st=Connexion.getConnection().createStatement();
			ResultSet rs=st.executeQuery(req);
			while(rs.next()) {
				c.add(new
Client(rs.getInt("ID"),rs.getString("VERSION"),rs.getString("SEXE"),rs.getString("NOM"),rs.getString("PRENOM")));
			}
		}catch(SQLException e) {
			System.err.println("Erreur SQL!!");
		}
		return c;
	}

}
```

Classe Connexion.java :

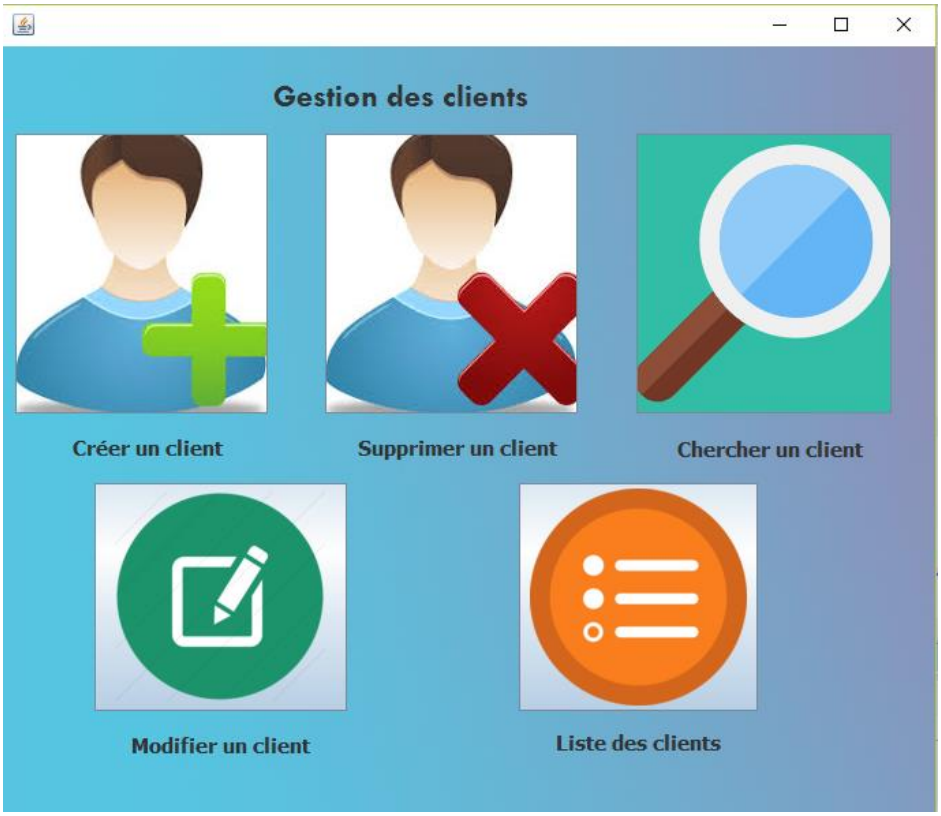
```
package my.projet.connection;

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

public class Connexion {

	private static Connection con;
	static {
		try {
			//Charger le driver
			Class.forName("com.mysql.cj.jdbc.Driver");
			//Authentification auprès de la base de données
			con= DriverManager.getConnection("jdbc:mysql://localhost/cabinet", "root", "");
		}catch(ClassNotFoundException e) {
			System.err.println("Impossible de charger le driver!!");
		}catch(SQLException e) {
```

```
                System.err.println("Erreur de connexion à la base de données!!");
            }
        }
        public static Connection getConnection() {
            return con;
        }
}
```

L'interface de l'authentification :

Cette page permet à un administrateur de se connecter pour gérer les clients du cabinet.

L'interface de gestion des clients :



L'interface d'ajouter un client :