

Compte Rendu TP8



Anthony Afonso

Base de données pour le répertoire :

Partie 3:

```
public class Connect {
    public static void main(String[] args) {
        // Chargement du driver
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
        }
        catch(ClassNotFoundException e) {
            System.err.println("Erreur de chargement du driver : "+e);
        }

        Connection conn = null;
        Statement statement = null;
        ResultSet res = null;

        try {
            // Création de la connexion
            conn = DriverManager.getConnection("jdbc:mysql://localhost/repertoire","root","");
            // Création de l'état tampon
            statement = conn.createStatement();
            // Execution de la requête, et stockage dans un ResultSet
            res = Statement.executeQuery("SELECT * FROM contact");
            // Parcours de la variable res, afin de traiter les informations retournées
            // par la base de données
            while(res.next()) {
                System.out.println(res.getInt("id")+" "+res.getString("nom")+" "+
                    res.getString("prenom"));
            }
            catch(SQLException e) {
                System.err.println("Erreur de connexion "+e);
            }
            catch(Exception e) {
                System.err.println(e);
            }
        }
    }
}
```

Ici j'ai essayé de reproduire une connexion entre mon répertoire et ma base de données en m'inspirant de ma classe TestJDBC

Pour ce faire j'ai :

- Créé ma connexion
- Affiché une erreur dans le cas où la connexion rate
- Adapté mon code afin que la base de données enregistre les informations récupérées par le répertoire
- Affiché ces données

Création de la classe passerelle :

```
import java.sql.*;
import java.util.ArrayList;

public class Passerelle {

    // Méthode qui charge toutes les personnes de la base dans une ArrayList
    public static ArrayList<Personne> chargeRepertoire() {
        ArrayList<Personne> repertoire = new ArrayList<>();
        String sql = "SELECT id, name, email, phone FROM contacts";

        try (Connection conn = Connect.get();
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(sql)) {

            while (rs.next()) {
                String prenom = rs.getString("First name");
                String nom = rs.getString("Last name");
                String email = rs.getString("email");
                String telephone = rs.getString("phone");

                Personne p = new Personne(id, nom, email, telephone);
                repertoire.add(p);
            }

        } catch (SQLException e) {
            e.printStackTrace();
        }

        return repertoire;
    }

    // Méthode qui ajoute une personne dans la base
    public static void ajoutePersonne(Personne p) {
        String sql = "INSERT INTO contacts (name, email, phone) VALUES (?, ?, ?)";

        try (Connection conn = Connect.get();
            PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setString(1, p.getNom());
            stmt.setString(2, p.getEmail());
            stmt.setString(3, p.getTelephone());

            stmt.executeUpdate();
            System.out.println("Personne ajoutée avec succès !");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Ici l'objectif était de créer une passerelle entre l'ajout de chaque données afin de les recevoir dans le répertoire et donc pouvoir par la suite les enregistrer et les afficher dans la base de données.

Aperçu en base de données :

Nom d'utilisateur	Nom d'hôte	Mot de passe	Privilèges globaux	Groupe d'utilisateurs	« Grant »	Action
<input type="checkbox"/> N'importe quel	%	Non	USAGE		Non	Éditer les privilèges Exporter Verrouiller
<input type="checkbox"/> pma	localhost	Non	USAGE		Non	Éditer les privilèges Exporter Verrouiller
<input type="checkbox"/> root	127.0.0.1	Non	ALL PRIVILEGES		Oui	Éditer les privilèges Exporter Verrouiller
<input type="checkbox"/> root	:::1	Non	ALL PRIVILEGES		Oui	Éditer les privilèges Exporter Verrouiller
<input type="checkbox"/> root	localhost	Non	ALL PRIVILEGES		Oui	Éditer les privilèges Exporter Verrouiller

Ici on représente le résultat en base de données sur phpmyadmin qui permet de contrôler les droits de chaque personne enregistré en base de données ce qui est un principe essentiel pour la cybersécurité. Cette gestion des droits permet de s'assurer de la confidentialité des informations personnelles des clients, pouvoir garder leur intégrité, maintenir la base de données disponible, et pouvoir tracer un potentiel attaquant.

Ici le résultat n'est pas convaincant. Je n'ai pas réussi à effectuer correctement les demandes ce qui donne un résultat presque vide.

L'une des raisons de ce raté est le fait que mon JDK était en version 21 et non 23 comme demandé; J'ai tenté d'arranger ce problème mais en vain.

Conclusion :

J'ai appris à relier une base de données et avec du code Java afin d'afficher et d'ajouter un client à celle-ci.

Pour le prochain travail, je devrais avoir une meilleure gestion du temps, apprendre encore beaucoup car j'en ai besoin et mieux savoir mieux comprendre mes erreurs afin de rendre un travail fonctionnel et complet en temps et en heure.