

TP 6 : Java-JDBC

Pierre Lefebvre

- 1) Tester un programme JDBC donné en cours pour valider la configuration de votre environnement
- 2) A partir de la maquette d'IHM suivante (source fourni), compléter l'implémentation de ce programme pour permettre une connexion et une consultation de la base de données Mysql. Vous vous limiterez à la consultation de tables.

Driver :

URL jdbc :

login :

password :

Connection

requete

Execution

- 3) L'objectif de ces exercices est de développer des méthodes de la classe Java *ExoJDBC* pour extraire et mettre à jour des données des tables du schéma *Parc Informatique*.

Curseur statique

Ecrire les méthodes :

- *ArrayList getSalles()* qui retourne sous la forme d'une liste les enregistrements de la table *Salle*.
- *Main* qui se connecte à la base, appelle la méthode *getSalles* et affiche les résultats (exemple donné ci-dessous)

nSalle	nomSalle	nbPoste	indIP
s01	Salle 1	3	130.120.80
s02	Salle 2	2	130.120.80
...			

Ajouter une nouvelle salle dans la table *Salle* dans l'interface de commande, et lancez à nouveau le programme pour vérifier

Curseur modifiable

Ecrivez la méthode *void deleteSalle(int)* qui supprime de la table *Salle* l'enregistrement de rang passé en paramètre. Vous utiliserez la méthode *deleteRow* appliquée à un curseur modifiable. Appelez dans le *main* cette méthode pour supprimer l'enregistrement de la table *Salle* que vous avez ajouté en test, dans l'exercice précédent. Si l'enregistrement est rattaché à un enregistrement fils, ne forcez pas la contrainte référentielle, contentez-vous d'afficher le message 1451 renvoyé par MySQL, dans le bloc des exceptions.

Source de l'interface graphique

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class JDBCTest extends Panel
implements ActionListener
{
    TextField    nomDriver;
    TextField    urlConnection;
    TextField    nomLogin;
    TextField    motPasse;
    Button        boutonConnection;
    TextField    requeteSQL;
    List          resultatRequete;
    Button        boutonExecuter;

    public JDBCTest()
    {
        Panel haut;
        Panel bas;

        haut = new Panel();
        bas = new Panel();

        boutonConnection = new Button("Connection");
        boutonConnection.addActionListener(this);

        boutonExecuter = new Button("Execution");
        boutonExecuter.addActionListener(this);

        Panel p1 = new Panel();
        p1.setLayout(new GridLayout(4, 2));
        p1.add(new Label("Driver :"));
        p1.add(nomDriver = new TextField(32));
        p1.add(new Label("URL jdbc :"));
        p1.add(urlConnection = new TextField(32));
        p1.add(new Label("login :"));
        p1.add(nomLogin = new TextField(32));
        p1.add(new Label("password :"));
        p1.add(motPasse = new TextField(32));

        haut.setLayout(new BorderLayout());
        haut.add(p1, BorderLayout.NORTH);
        haut.add(boutonConnection, BorderLayout.SOUTH);

        Panel p2 = new Panel();
```

```

        p2.setLayout(new BorderLayout());
        p2.add(new Label("requete"), BorderLayout.WEST);
        p2.add(requeteSQL = new TextField(32), BorderLayout.CENTER);
        Panel p3 = new Panel();
        p3.setLayout(new BorderLayout());
        p3.add(p2, BorderLayout.NORTH);
        p3.add(boutonExecuter, BorderLayout.SOUTH);

        bas.setLayout(new BorderLayout());
        bas.add(p3, BorderLayout.NORTH);
        bas.add(resultatRequete = new List(20));

        setLayout(new BorderLayout());
        add(haut, BorderLayout.NORTH);
        add(bas, BorderLayout.CENTER);
    }
    public void actionPerformed(ActionEvent evt)
    {
        //System.out.println("Not implemented !!");
        resultatRequete.add("au boulot les gars !!!");
    }

    public static void main(String[] arg)
    {
        JDBCTest    test;

        Frame f = new Frame();
        f.setSize(500, 400);
        test = new JDBCTest( );
        f.add(test, BorderLayout.CENTER);
        f.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
        });
        f.setVisible(true);
    }
}

```

Solution du TP : Java et JDBC

1)

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

class Test extends JDBCTest {
    Connection cnx = null;
    Test() { this("", ""); }

    Test(String driver, String url) {
        nomDriver.setText(driver);
        urlConnection.setText(url);
        motPasse.setEchoChar('*');
    }

    public void actionPerformed(ActionEvent e) {
        Button b = (Button)(e.getSource());
        try {
            if((b==boutonConnection) && (cnx == null)) {
                System.out.println("dans CONNECT");
                String driver=new String(nomDriver.getText());
                String url=new String(urlConnection.getText());
                String login=new String(nomLogin.getText());
                String passwd=new String(motPasse.getText());

                Class.forName(driver);
                cnx=DriverManager.getConnection(url,login,passwd);
            }
            else if ((b == boutonConnection) && (cnx !=null))
                System.out.println("connection deja etablie");
            else if ((b == boutonExecuter) && (cnx != null)) {
                System.out.println("dans EXECUTE");
                Statement stmt = cnx.createStatement();
                resultatRequete.removeAll();
                String requete=new String(requeteSQL.getText());
                ResultSet rs = stmt.executeQuery(requete);
                ResultSetMetaData rsmd=rs.getMetaData();

                int nbcol = rsmd.getColumnCount();
                String titre="";
                String resultat;
                for(int i = 0; i < nbcol; i++)
                    titre+=" : "+rsmd.getColumnName(i+1);
                resultatRequete.add(titre);
                while(rs.next()) {
```

```

        resultat = new String();
        for(int i=0; i < nbcol; i++)
        {
            resultat+=" : "+rs.getString(i+1);
            System.out.println(rs.getString(i+1));
        }
        resultatRequete.add(resultat);
        System.out.println("requete = " + resultat);

    }
    rs.close();
    stmt.close();
}
else System.out.println("connection non etablie ou cassee");
}
catch(SQLException ei) {
    ei.printStackTrace();
}
catch(ClassNotFoundException ei2) {
    ei2.printStackTrace();
} }
public static void main(String[] arg)
{
    Test test;
    Frame f = new Frame();
    f.setSize(500, 400);
    String driver="com.mysql.jdbc.Driver";
    String url = "jdbc:mysql://localhost/livre";
    //String url = "jdbc:oracle:thin:@pc401.gts93.greta.fr:1521:ora";
    test = new Test(driver, url);
    f.add(test, BorderLayout.CENTER);
    f.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e)
        {
            System.exit(0);
        }
    });
    f.setVisible(true);
}
}

```

3)

```

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

public class ExoJDBC {
    public static Connection cx;
    public static ResultSet rs, rs2;
    public static Statement etat;
    public static Statement etatModifiable;
    public static CallableStatement cetat;
}

```

```

public static ArrayList getSalles()
{
    ArrayList tableauRésultat = new ArrayList();
    try {
        etat = cx.createStatement();
        rs = etat.executeQuery("SELECT * FROM Salle");
        String [] ligne = null;
        while (rs.next()) {
            ligne = new String[4];
            ligne[0] = rs.getString(1);
            ligne[1] = rs.getString(2);
            ligne[2] = (new Integer(rs.getInt(3))).toString();
            ligne[3] = rs.getString(4);
            tableauRésultat.add(ligne);
        }
        rs.close();
        etat.close();
    }
    catch (SQLException ex) {
        while (ex != null) {
            System.out.println ("Statut SQL : "+ex.getSQLState());
            System.out.println ("Message      :
"+ex.getMessage());
            System.out.println ("Code erreur :
"+ex.getErrorCode());
            ex = ex.getNextException();
        }
    }
    return tableauRésultat;
}

public static void deleteSalle(int nl)
{
    try {
        etatModifiable =
cx.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
                    ResultSet.CONCUR_UPDATABLE);

        cx.setAutoCommit(false);
        rs2 = etatModifiable.executeQuery("SELECT s.* FROM Salle s");
        if (rs2.absolute(nl))
        {
            rs2.deleteRow(); cx.commit();
            System.out.println("Salle supprimée");
        }
        else System.out.println("Désolé, pas de "+ nl +" ème salle !");
        rs2.close();
        etatModifiable.close();
    }

    catch (SQLException ex) {
        while (ex != null) {
            System.out.println ("Statut SQL : "+ex.getSQLState());
            System.out.println ("Message      :
"+ex.getMessage());
            System.out.println ("Code erreur :
"+ex.getErrorCode());
            ex = ex.getNextException();
        }
    }
}

public static void main(String args[]) throws SQLException, Exception
{
    try {

```

```

        System.out.println("Initialisation de la connexion");
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        } catch (ClassNotFoundException ex)
            { System.out.println ("Problème au chargement"+ex.toString());
        }
        try
        {
            cx =
DriverManager.getConnection("jdbc:mysql://localhost/bdsoutou?user=soutou&pa
ssword=iut");

            ArrayList lignes = getSalles();
            System.out.println("Liste des salles :\n");
            System.out.println("nSalle\tnomSalle \tnbPoste\tindIP");
            System.out.println("-----");
            String[] lig;
            for (int i=0;i<lignes.size();i++)
            {
                lig=(String [])lignes.get(i);
                System.out.println(lig[0]+" \t"+lig[1]+" \t"+lig[2]+"
\t"+lig[3]);
            }
            System.out.println();

//Pb intégrité
// System.out.println("Salle 21 à supprimer");
// deleteSalle(6);

// dernière nouvelle salle pas rattachée
// System.out.println("Salle 77 à supprimer");
// deleteSalle(7);

        }
        catch (SQLException ex) {
            while (ex != null) {
                System.out.println ("Statut SQL : "+ex.getSQLState());
                System.out.println ("Message : "+ex.getMessage());
                System.out.println ("Code erreur : "+ex.getErrorCode());
                ex = ex.getNextException(); } }
    }
}

```