

Aubrée Romain  
21/01/2026  
SLAM

Beauvallet

Groupe 1

## **Bloc 2 TD 3**

|                   |           |
|-------------------|-----------|
| <b>Exercice 1</b> | <b>2</b>  |
| <b>Exercice 2</b> | <b>2</b>  |
| <b>Exercice 3</b> | <b>2</b>  |
| <b>Exercice 4</b> | <b>3</b>  |
| <b>Exercice 5</b> | <b>3</b>  |
| <b>Exercice 6</b> | <b>5</b>  |
| <b>Exercice 7</b> | <b>7</b>  |
| <b>Exercice 8</b> | <b>8</b>  |
| <b>Exercice 9</b> | <b>10</b> |
| <b>Conclusion</b> | <b>10</b> |

## Exercice 1

Installation du JDBC

Ajout dans pom.xml (projet IntelliJ avec Maven) :

```
<dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <version>9.5.0</version>
</dependency>
```

Puis reload Maven

## Exercice 2

Il faut allumer **XAMPP** et **MySQL**, puis dans la **cmd**, il faut allé dans “`cd C:\xampp\mysql\bin`” puis exécuté “`mysql_upgrade -u root -p --force`”, et ensuite redémarrer **MySQL**

## Exercice 3

Créer une nouvelle table dans “`phpmyadmin`” et importer le **.sql** fournie pour y intégrer toute la **BDD**.

```
-- phpMyAdmin SQL Dump -- version 4.5.5.1 -- http://www.phpmyadmin.net -- -- Client : 127.0.0.1 --
Généré le : Lun 13 Mars 2017 à 15:32 -- Version du serveur : 5.7.11 -- Version de PHP : 5.6.19 SET
SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s))

SET time_zone = "+00:00";

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s))

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]
```

## Exercice 4

Exercice déjà complété (dans le TD2).

## Exercice 5

```
@FXML

public void initialize() {

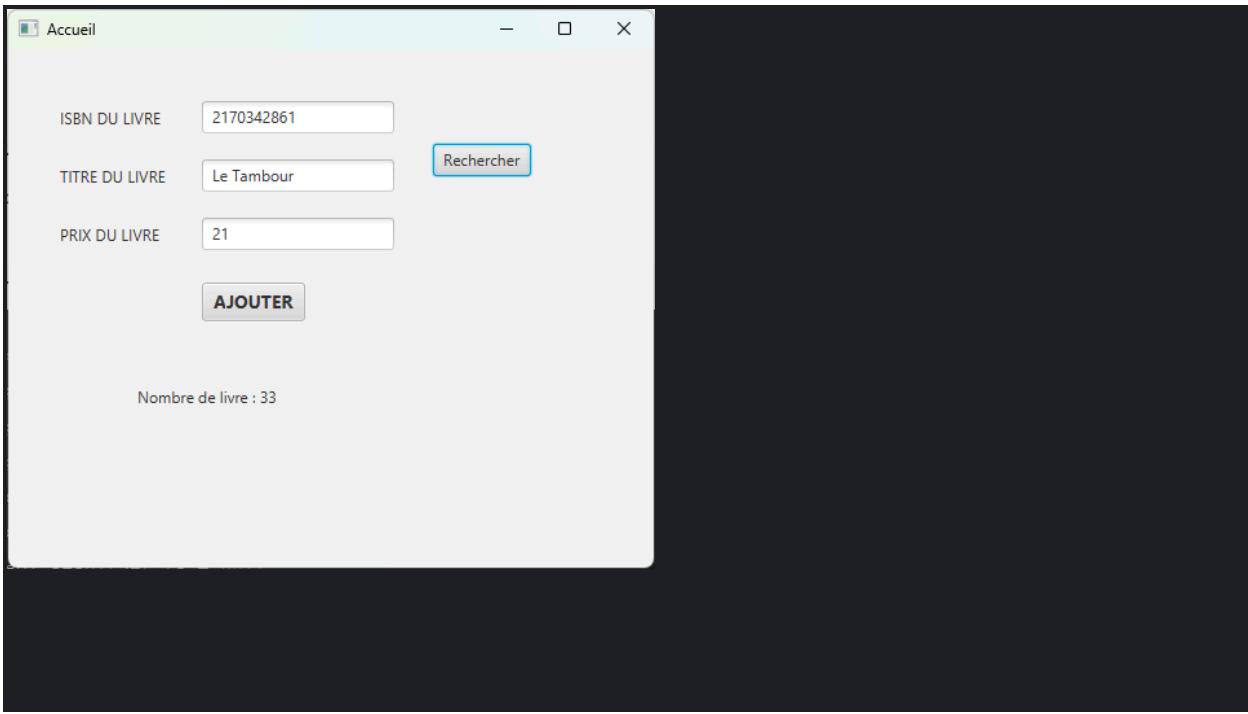
    String url = "jdbc:mysql://localhost:3306/slamp_td2";
    String login = "root";
    String passwd = "";
    java.sql.Connection cn = null;
    java.sql.Statement st = null;
    java.sql.ResultSet rs = null;
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        cn = java.sql.DriverManager.getConnection(url, login, passwd);
        st = cn.createStatement();
        String sql = "SELECT * FROM Livre";
        rs = st.executeQuery(sql);
        while (rs.next()) {
            String isbn = rs.getString("ISBN");
            String titre = rs.getString("titre");
            String auteur = rs.getString("auteur");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

```
        int prix = rs.getInt("prix");
        al.add(new Livre(isbn, titre, auteur, prix));
    }

} catch (java.sql.SQLException e) {
    e.printStackTrace();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
} finally {
    try {
        if (cn != null) cn.close();
        if (st != null) st.close();
        if (rs != null) rs.close();
    } catch (java.sql.SQLException e) {
        e.printStackTrace();
    }
}

for (Livre l : al) {
    l.Afficher();
    System.out.println();
}

lblCompteur.setText("Nombre de livre : " + al.size());
}
```



## Exercice 6

```
@FXML  
protected void onAjouter() {  
    String isbn = tfISBN.getText();  
    String titre = tfTitre.getText();  
    String auteur = "Inconnu";  
    int prix = 0;  
    try {  
        prix = Integer.parseInt(tfPrix.getText());  
    } catch (NumberFormatException e) {  
        System.out.println("Prix invalide, on met 0");  
    }  
    Livre nouveau = new Livre(isbn, titre, auteur, prix);  
    al.add(nouveau);  
    lblCompteur.setText("Nombre de livre : " + al.size());  
    String url = "jdbc:mysql://localhost:3306/slamp_td2";  
    String login = "root";  
    String passwd = "";  
    try {
```

```

        Class.forName("com.mysql.cj.jdbc.Driver");

        java.sql.Connection cn = java.sql.DriverManager.getConnection(url, login,
passwd);

        java.sql.Statement st = cn.createStatement();

        String sql = "INSERT INTO Livre (ISBN, titre, auteur, prix) VALUES ('"
                + isbn + "', '" + titre + "', '" + auteur + "', " + prix + ")";

        st.executeUpdate(sql);

        System.out.println("Succès ! Livre ajouté en BDD.");

        cn.close();

        st.close();

    } catch (Exception e) {

        e.printStackTrace();

        System.out.println("Erreur lors de l'enregistrement SQL");

    }

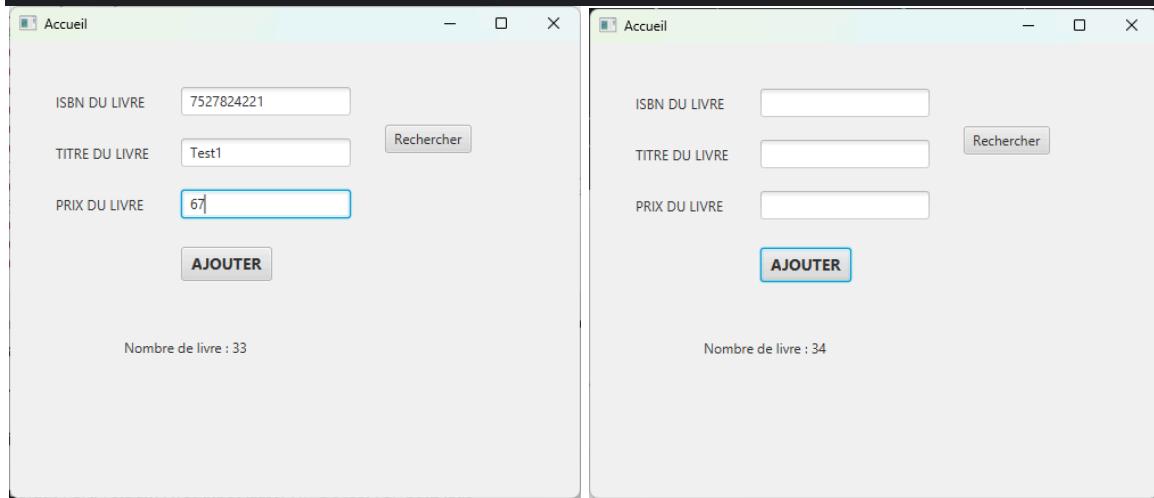
    tfISBN.clear();

    tfTitre.clear();

    tfPrix.clear();

}

```



## Exercice 7

Accueil

|  |   |   |
|--|---|---|
| ISBN DU LIVRE                          | <input type="text" value="1317442277"/> | <input type="button" value="Rechercher"/> |
| TITRE DU LIVRE                         | <input type="text" value="Contes"/>     |   |
| PRIX DU LIVRE                          | <input type="text" value="10"/>         |   |
| <input type="button" value="AJOUTER"/> |   |   |

Nombre de livre : 34

Run    Launcher    HelloApplication

ISBN: 210342801 | Titre: Le Tambour, Auteur: Gunter Grass, Prix: 21€  
ISBN: 7527824221 | Titre: Test1, Auteur: Inconnu, Prix: 67€

WARNING: A terminally deprecated method in sun.misc.Unsafe has been called  
WARNING: sun.misc.Unsafe::allocateMemory has been called by com.sun.marlin.OffHeap  
WARNING: Please consider reporting this to the maintainers of class com.sun.marlin  
WARNING: sun.misc.Unsafe::allocateMemory will be removed in a future release  
Livre existant trouvé : Mise à jour...

Accueil

|  |   |   |
|--|---|---|
| ISBN DU LIVRE                          | <input type="text" value="1317442277"/> | <input type="button" value="Rechercher"/> |
| TITRE DU LIVRE                         | <input type="text" value="Contes"/>     |   |
| PRIX DU LIVRE                          | <input type="text" value="10"/>         |   |
| <input type="button" value="AJOUTER"/> |   |   |

Nombre de livre : 34

## Exercice 8

```
package org.example.td219012026;

import java.sql.Statement;

public class Livre {
    private String ISBN, titre, Auteur;  5 usages
    private int prix;  5 usages

    public String getTitre() {return titre;}  1 usage
    public void setTitre(String titre) {this.titre = titre;}  1 usage
    public String getAuteur() {return Auteur;}  no usages
    public void setAuteur(String auteur) {Auteur = auteur;}  no usages
    public int getPrix() {return prix;}  1 usage
    public void setPrix(int prix) {this.prix = prix;}  1 usage
    public String getISBN() {return ISBN;}  2 usages
    public void setISBN(String ISBN) {ISBN = ISBN;}  no usages

    public Livre(String ISBN, String titre, String auteur, int prix) { 2 usages
        this.ISBN = ISBN;
        this.titre = titre;
        this.Auteur = auteur;
        this.prix = prix;
    }

    public Livre(String ISBN, String titre, String auteur, int prix, Statement st) {  no usage
        this.ISBN = ISBN;
        this.titre = titre;
        this.Auteur = auteur;
        this.prix = prix;

        try {
            String sql = "INSERT INTO Livre (ISBN, titre, auteur, prix) VALUES ('"
                + ISBN + "', '" + titre + "', '" + auteur + "', " + prix + ")";
            st.executeUpdate(sql);
            System.out.println("Q8 : Sauvegarde automatique via le constructeur réussie !");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void Afficher(){  1 usage
        System.out.print("ISBN: " + ISBN + " | ");
        System.out.print("Titre: " + titre + ", ");
        System.out.print("Auteur: " + Auteur + ", ");
        System.out.println("Prix: " + prix + "€");
    }
}
```

```

@FXML
protected void onAjouter() {
    String isbn = tfISBN.getText();
    String titre = tfTitre.getText();
    int prix = 0;
    try {
        prix = Integer.parseInt(tfPrix.getText());
    } catch (Exception e) {
        System.out.println("Prix incorrect, mis à 0");
    }
    String auteur = "Inconnu";
    String url = "jdbc:mysql://localhost:3306/slam_td2";
    String login = "root";
    String passwd = "";
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        java.sql.Connection cn = java.sql.DriverManager.getConnection(url, login, passwd);
        java.sql.Statement st = cn.createStatement();
        String checkSql = "SELECT * FROM Livre WHERE ISBN = '" + isbn + "'";
        java.sql.ResultSet rs = st.executeQuery(checkSql);
        if (rs.next()) {
            System.out.println("Livre existant trouvé : Mise à jour...");
            String updateSql = "UPDATE Livre SET titre = '" + titre + "', prix = " + prix + " WHERE ISBN = '" + isbn + "'";
            st.executeUpdate(updateSql);
            for (Livre l : al) {
                if (l.getISBN().equals(isbn)) {
                    l.setTitre(titre);
                    l.setPrix(prix);
                    break;
                }
            }
        } else {
            System.out.println("Nouveau livre : Création via Constructeur Q8...");
            Livre nouveau = new Livre(isbn, titre, auteur, prix, st);
            al.add(nouveau);
        }
        cn.close();
        st.close();
        rs.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

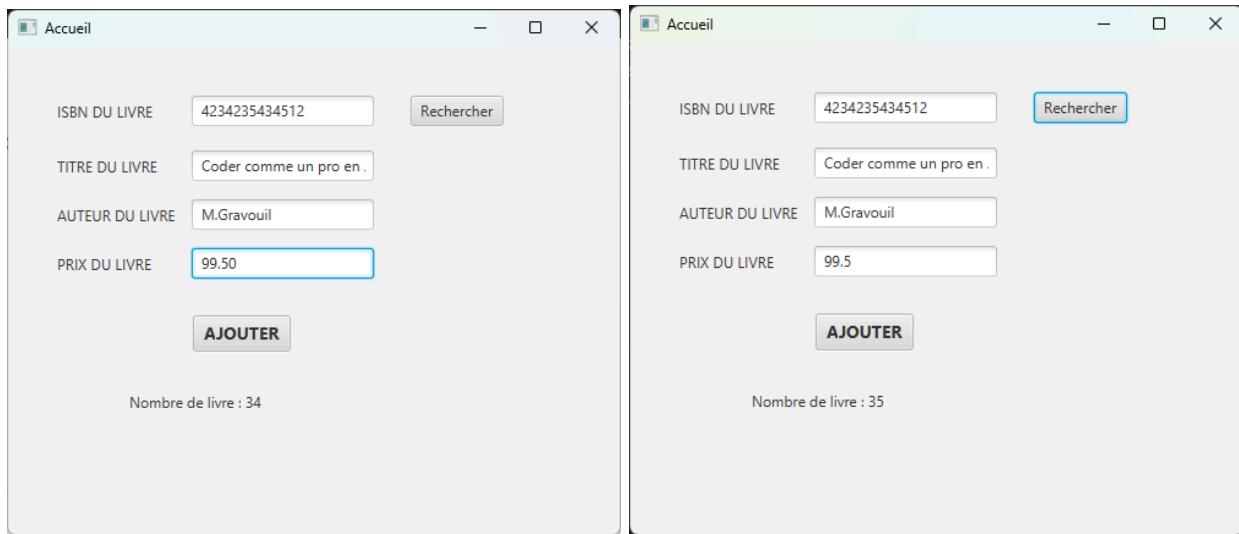
lblCompteur.setText("Nombre de livre : " + al.size());
tfISBN.clear();
tfTitre.clear();
tfPrix.clear();
}

```

## Exercice 9

J'ai du rajouter un ligne auteur :

J'ai aussi modifier le type de la valeur "prix" car elle était en **Int** depuis le début, donc j'ai tout convertie en **Float**.



## Conclusion

Ce TP nous a introduit à utiliser MySQL dans un programme Java.