

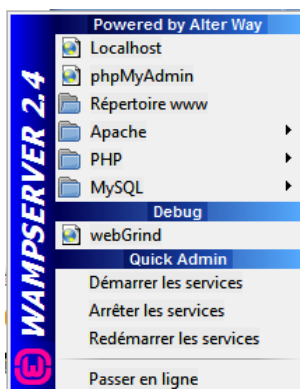
TP1_BDA_MYSQL_GRP1

Installer le *WampServer* 2.4:

- Lancer *WampServer*, vous verrez une petite icône apparaître en bas de votre écran, ressemblant à ça :



- Si vous cliquez la souris dessus, vous verrez apparaître une fenêtre :

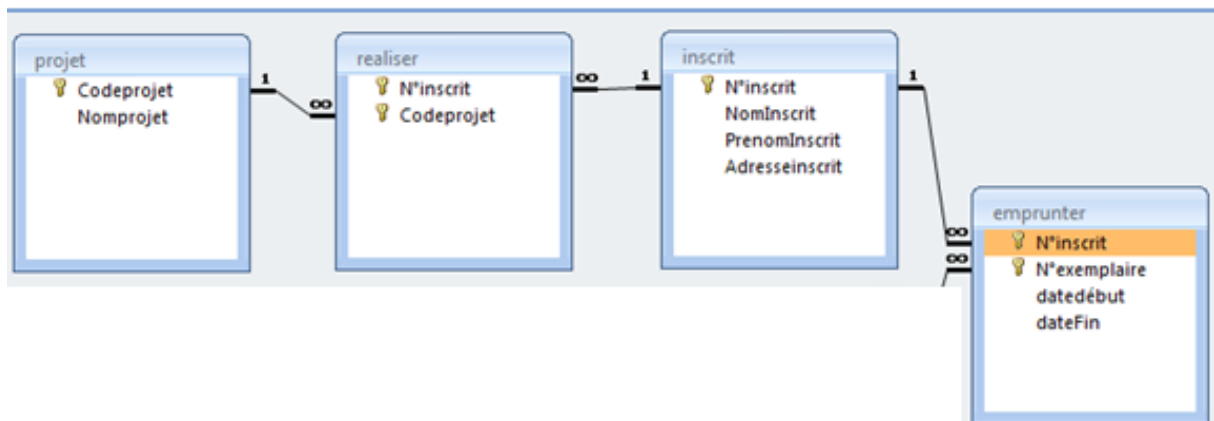


- Cliquez sur *phpMyAdmin*, pour que vous puissiez accéder à l'interface de gestion de MySQL.
- Vous remarquerez qu'il y'a déjà, 2 bases de données système qui sont présentes (*Information_schema*, *mysql*).
- Cliquez sur l'onglet *Base de données*, commencez à créer une nouvelle base des données, que nous allons appeler *Gestion_Emprunt1*.



- Une fois que vous aurez cliqué sur le bouton *Créer*, la base est créée, mais elle est vide.
- Cliquez sur *Gestion_Emprunt2*, ensuite rendez vous à l'onglet *SQL*, à fin de commencer à créer les tables de la base avec les requêtes correspondantes.

- Voici le modèle Physique des données de la Gestion d'emprunt1 :



- Voici la syntaxe pour une requête de création de table avec MYSQL :

```

CREATE TABLE [IF NOT EXISTS] [nomBase.] nomTable
(
    colonne1 type1
    [NOT NULL | NULL] [DEFAULT valeur1] [COMMENT 'chaine1']
    [, colonne2 type2
    [NOT NULL | NULL] [DEFAULT valeur2] [COMMENT 'chaine2'] ]
    [CONSTRAINT nomContraintel typeContraintel] ...)
[ENGINE= InnoDB];

```

- En voici un exemple de création de la table *Pilote* avec cette même requête, en définissant toutes les contraintes (Clé primaire, clé étrangère, ...) :

```

CREATE TABLE Pilote
(
    brevet INT(6), nom CHAR(15) NOT NULL,
    nbHVol DECIMAL(7,2), compa CHAR(4),
    PRIMARY KEY(brevet) /*clé primaire*/,
    FOREIGN KEY (compa) /*clé étrangère*/
    REFERENCES Compagnie(compa) /* table référence*/;

```

TAF: Créer toutes les tables de cette base des données, en définissant les requêtes de création pour chaque table, ainsi que leurs contraintes, et en exécutant ces requêtes dans MYSQL.

TP JAVA/Mysql JDBC

(Java database connectivity)

L'objectif de ce TP est de vous faire créer et manipuler manuellement une couche de persistance simple en utilisant JDBC au-dessus d'une base de données gérée par le SGDB MySQL. La base de données sera créée et manipulée via l'interface Phpadmin de MySQL. La programmation se fera sous NetBeans IDE.

Les documents et fichiers nécessaires au bon déroulement de ce TP sont disponibles à l'adresse :

- Mysql-connector-j-8.0.32.zip: <https://www.dev.mysql.com/downloads/connector/j/>

The screenshot displays the phpMyAdmin web interface. The top navigation bar shows the current server (MySQL:3306), database (mydatabase), and table (userlogin). The main content area shows a successful login message: "Affichage des lignes 0 - 0 (total de 1, traitement en 0,00)". Below this, a SQL query is displayed: `SELECT * FROM `userlogin``. A table of results is shown with columns: id, Prenom, Nom, mot_de_passe. The table contains one row: 100, Mohamed, Abdourahman, med. A modal dialog titled "Login" is open, showing the login form with fields for "Prenom" (Mohamed) and "Mot de Passe" (med), and a "Login" button. A smaller "Message" dialog is also open, displaying "Login Successful !" with an "OK" button.

id	Prenom	Nom	mot_de_passe
100	Mohamed	Abdourahman	med

I- Classe de Connexion

Apache NetBeans IDE

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config> 392.6/1034MB

Projects Files Services

- Microsoft SQL Server
- MySQL (Connector/J driver)
- MySQL (Connector/J driver) (1)
- Oracle OCI
- Oracle Thin
- PostgreSQL
- wampmysql**
 - mydatabase
 - Other databases
 - achat
 - information_schema
 - l3_infor
 - med
 - mysql
 - performance_schema
 - sys

Mavenprojectwithdatabase - Navigator

Members <empty>

- Mavenprojectwithdatabase
 - Mavenprojectwithdatabase()
 - conn() : Connection

wampmysql - Properties

Properties

Display name	wampmysql
Database URL	jdbc:mysql://localhost:3306/mydata...
Driver	MySQL (Connector/J driver) (1)
Driver class	com.mysql.cj.jdbc.Driver
Schema	
User	admin
Remember password	<input checked="" type="checkbox"/>
Show system tables separately	<input type="checkbox"/>
Use scrollable cursors	<input type="checkbox"/>
Connection properties	No properties set
Database product name	MySQL
Database product version	8.3.0
Read only	<input type="checkbox"/>
Mixed-case identifiers	<input type="checkbox"/>
Mixed-case quoted identifiers	<input type="checkbox"/>
Alter table with add column	<input checked="" type="checkbox"/>
Alter table with drop column	<input checked="" type="checkbox"/>
Convert	<input type="checkbox"/>
Table correlation names	<input checked="" type="checkbox"/>
Expressions in order by	<input checked="" type="checkbox"/>

wampmysql
Database Connection

Close Help



File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help mavenprojectwithdatabase - Apache NetBeans IDE 22 Search (Ctrl+I)

Projects Files Services

- mavenprojectwithdatabase
 - Source Packages
 - com.mycompany.mavenprojectwithdatabase
 - Mavenprojectwithdatabase.java
 - Userlogin.java
 - Test Packages
 - Dependencies
 - mysql-connector-j-8.0.32.jar
 - protobuf-java-3.21.9.jar
 - Java Dependencies
 - JDK 23 (Default)
 - Project Files
 - pom.xml
 - nbactions.xml

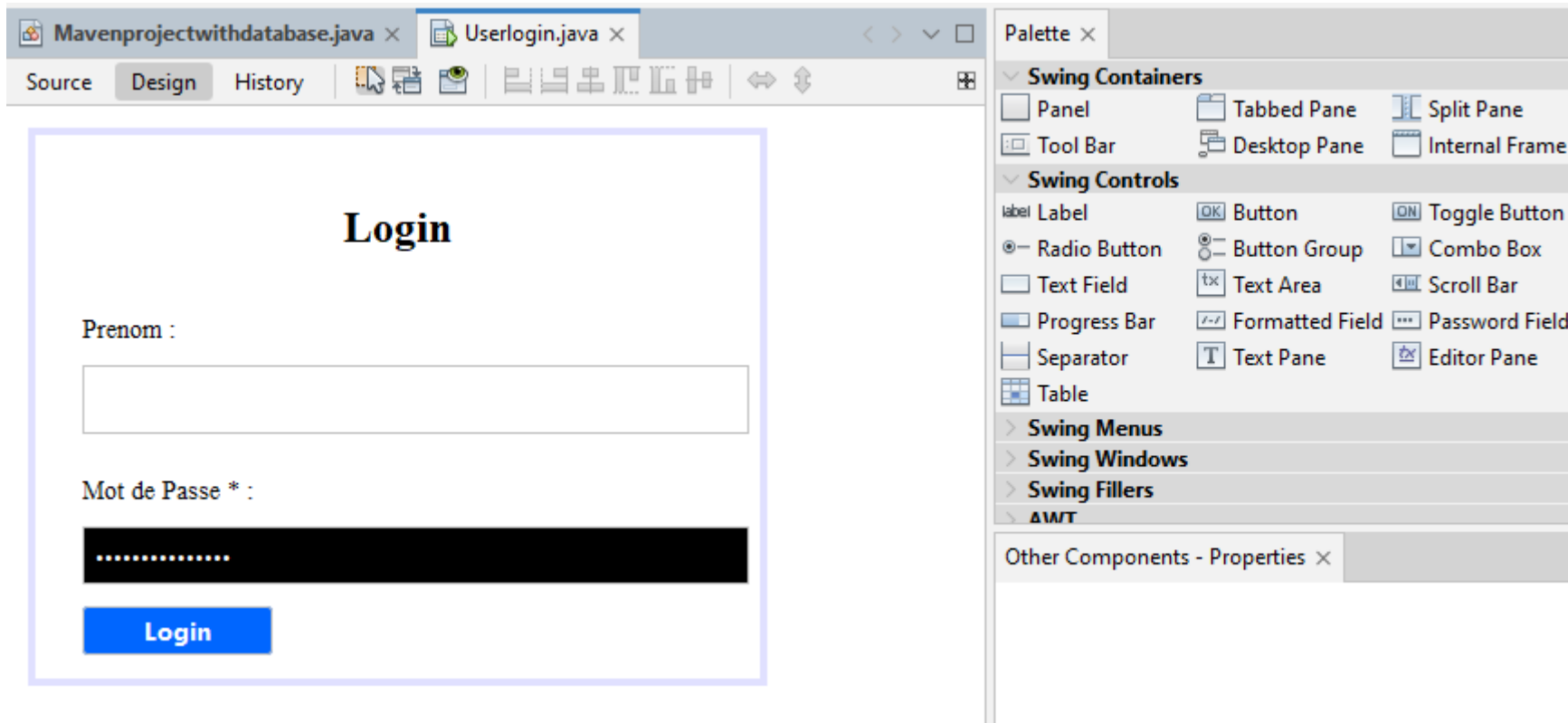
Navigator Members

- Mavenprojectwithdatabase
 - Mavenprojectwithdatabase()
 - conn() : Connection

Source History

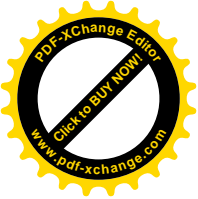
```
5
6 import java.sql.Connection;
7 import java.sql.DriverManager;
8 import java.sql.SQLException;
9 import java.util.logging.Level;
10 import java.util.logging.Logger;
11 import javax.swing.JOptionPane;
12
13 /**
14  *
15  * @author Mohamed Abdourahman
16  */
17 public class Mavenprojectwithdatabase {
18
19     public static Connection conn() {
20         try {
21             String url = "jdbc:mysql://localhost:3306/mydatabase";
22             String user = "admin";
23             String password = "Root1*";
24
25             try {
26                 Class.forName("com.mysql.cj.jdbc.Driver");
27             } catch (ClassNotFoundException ex) {
28                 Logger.getLogger(Mavenprojectwithdatabase.class.getName()).log(Level.SEVERE, null, ex);
29             }
30
31             Connection conn = DriverManager.getConnection(url, user, password);
32             return conn;
33         } catch (SQLException e) {
34             JOptionPane.showMessageDialog(null, e);
35         }
36
37         return null;
38     }
39 }
40
```

II- Classe de JFrame Userlogin





```
Mavenprojectwithdatabase.java x Userlogin.java x
Source Design History
6
7 import java.sql.Connection;
8 import java.sql.PreparedStatement;
9 import java.sql.ResultSet;
10 import java.sql.SQLException;
11 import javax.swing.JOptionPane;
12
13 /**
14  *
15  * @author Mohamed Abdourahman
16  */
17 public class Userlogin extends javax.swing.JFrame {
18
19     Connection conn;
20     PreparedStatement pst;
21     ResultSet rs;
22
23     /**
24      * Creates new form Userlogin
25      */
26     public Userlogin() {
27         initComponents();
28         jPasswordField1.setEchoChar((char)0);
29         conn = Mavenprojectwithdatabase.conn();
30     }
31
```



```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String p = jTextField1.getText();  
    char[] Mot = jPasswordField1.getPassword();  
    try{  
        String sql = "SELECT * FROM userlogin WHERE Prenom=? AND mot_de_passe=?";  
        pst = conn.prepareStatement(sql);  
        pst.setString(1, p);  
        pst.setString(2, String.valueOf(Mot));  
        rs = pst.executeQuery();  
        if(rs.next()){  
            JOptionPane.showMessageDialog(null, "Login Successful !");  
        }else{  
            JOptionPane.showMessageDialog(null, "Prenom ou Mot de passe est incorrect!");  
        }  
    } catch(SQLException e){  
        JOptionPane.showMessageDialog(null, e);  
    }  
}
```