

## TP: Accès aux bases de données avec JDBC

| Objectifs   | Type     | Durée | Outils  |
|---|----------|-------|---------|
| - Accéder et manipuler les données d'une base de données mysql à travers JDBC | Formatif | 3h00  | Eclipse |

### Exercice 1:

1- Tester et interpréter le code suivant.

```

Import java.sql.Connection ;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;

public class TestJDBC {

public static void main(String[] args) {
    // TODO Auto-generated method stub
    /*url spécifie l'utilisation de JDBC, le Driver mysql et la
    source de la base*/
    String url="jdbc:mysql://localhost/Bookdb";
    String Utilisateur="root";
    String motDepasse="";

    try{
        /*Chargement du Driver avec la méthode forName() de la
        cclasse Class*/
        Class.forName("com.mysql.cj.jdbc.Driver");

        /*Ouverture de la connexion à la base*/
        Connection con = DriverManager.getConnection(url,
        Utilisateur,
        motDepasse);

        /* Création d'un statement (requête simple)
        * avec la méthode createStatement()*/
        Statement stmt = con.createStatement();

        /*Exécution de la requête avec la méthode
        executeQuery(requête)
        * qui renvoie le résultat de l'exécution dans un
        ResultSet
        */
        ResultSet rs=stmt.executeQuery("select * from BookTable");
        /*Accès aux méta-données avec getMetaDta()
        * qui renvoie un ResultSetMetaData
        */
        ResultSetMetaData rsmd= rs.getMetaData();

        /*Nombre de colonnes sélectionnées par la requête*/
        int numcols=rsmd.getColumnCount();
    }
}

```

```

        System.out.println("Les Colonnes sélectionnées de la table
BookTable sont:");
        for(int i=1;i<=numcols;i++){
            System.out.println(rsmd.getColumnLabel(i));
        }

        System.out.println("Liste des enregistrements de la table
BookTable:");
        /*Parcours des résultats du ResultSet ligne par ligne avec
les méthodes getInt() et getString()*/
        while (rs.next()) {
            int resultat1 = rs.getInt("ISBN");
            String resultat2 = rs.getString(2);
            System.out.println("Livre:  ISBN= "+resultat1 + " Titre=
"
+resultat2);

        }

        /*fermeture des différents espaces*/
        rs.close();
        stmt.close();
        con.close();
    } //fin try
    catch(SQLException exp){

        System.out.println(exp.getMessage());
    } catch (ClassNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
}

```

## 2- Réglez votre driver mysql pour cela :

- a. copier votre driver mysql dans votre projet dans un dossier nommé lib
- b. Ajouter votre Driver (bouton droit sur votre projet->propriétés->java build path-> add jars-> sélectionner votre driver )

## 3- tester votre code. Interprétation !

## 4- Créer votre base de données en utilisant le script Bookdb.sql et tester votre code.

Interprétation !

## 5- Ajouter depuis votre code un nouvel enregistrement. (isbn=999, Titre = ALGORITHMIQUE, edition=3, auteur= KAMEL BEN SALAH.

## 6- Modifier depuis votre code le numéro d'édition du livre dont isbn=999 et la mettre égale à 2.

**Exercice2 :****A. Création de base de donnée Mysql et insertion de données :**

- 1- Créer votre base de données *entreprisedb*.
- 2- En utilisant le script *entreprisedb.sql* ci-joint créer les tables *dept* et *emp*.

**B. Configuration d'Eclipse :**

- 3- Sous Eclipse, créer un nouveau projet Java *TPEntreprise*
- 4- Réglez votre driver mysql pour cela :
  - a. copier votre driver mysql dans votre projet dans un dossier nommé lib
  - b. Ajouter votre Driver (bouton droit sur votre projet->propriétés->java build path-> add jars-> sélectionner votre driver )

**C. Connexion/deconnexion:**

**Remarque :** Dans la suite du TP toutes les méthodes à créer doivent gérer les exceptions.

- 5- Créez une classe *Entreprise* qui contient un attribut *conn* de type *Connection*, *url*, *user* et *pwd* de type chaîne de caractères.
- 6- Ajoutez un constructeur permettant de créer un objet Entreprise qui fait, pour l'instant, juste un appel à *super()* et initialise les attributs *url*, *user* et *pwd*.
- 7- Créez une méthode *main*.
- 8- Créez une méthode *connexion()* qui permet
  - a. Charger le driver JDBC( gérer Exception avec try / catch).
  - b. Faire une trace indiquant: "Driver chargé avec succès"
  - c. Se connecter la base *Entreprisedb* :
  - d. Afficher le message « la connexion à la base de données n'a pas pu être établie » en cas d'échec.
- 9- Modifier le constructeur de votre classe *Entreprise* pour permettre la connexion à la base.
- 10- Créez une méthode *deconnexion()* de déconnexion à la base qui renvoie une *SQLException* et affiche les messages d'erreur en cas d'échec sinon le message « déconnexion ok ».
- 11- Tester les 2 méthodes précédentes dans la fonction principale en créant une instance de la classe *Entreprise*.

---

**D. Interrogation de la base de données :**

**12-** Créez une méthode *interrogation()* interrogeant la base de données avec la requête suivante: «**SELECT nom FROM emp ;** » et Afficher le résultat sous forme de colonne.

**13-** Tester la méthode précédente.

**E. Manipulation des données**

**14-** Créez une méthode *createTableAffaire()* qui crée la table *affaire* avec la requête « **create table affaire (NOAFF numeric(3), NOM varchar(10), BUDGET numeric (8,2));**. Cette méthode renvoie une SQLException et retourne true si la table affaire est créée et false sinon.

**15-** Créez une méthode *dropTable(String tablename)* qui efface la table dont le nom est passé en paramètre avec la requête « **DROP TABLE table\_name;** ». Cette méthode renvoie une SQLException. Cette méthode retourne un entier qui est le nombre de lignes modifiées.

**16-** Créez une méthode *insertionDansAffaire()* qui permet d'insérer des éléments dans la table affaire dont les attributs sont passés en paramètre: NOAFF, NOM, BUDGET; avec la requête « **INSERT INTO affaire VALUES (NOAFF,NOM,BUDGET);**. Cette méthode renvoie une SQLException et sa signature est la suivante: **public void insertionDansAffaire(int no\_aff, String nom, int budget)**. Créer un objet de pstmt de type PreparedStatement pour envoyer des PreparedStatements SQL à la base de données.

**17-** Tester les 3 méthodes précédentes.

**18-** Compiler et exécuter votre application

**19-** Insérer des données dans la table affaire :

101, 'ALPHA' ,250000 ;

102, 'BETA' , 175000 ;

103, 'GAMMA' ,95000.

**20-** Compiler et exécuter votre application. Vérifiez que les données ont bien été insérées dans la table affaire.