

# TP5-JDBC : JEE

☰ Tags	
▼ status	in progress

**Par : IBTISSAM MASSA - GINFO2**

## JDBC

L'objectif de ce TP est de se familiariser avec les mécanismes d'accès aux bases de données en utilisant le JDBC,

### Exercice 1 : Création et manipulation d'une Base de données sous MySQL

---

1. Télécharger l'archive mysql-XXXwinXX.zip de MySQL server
2. Extraire l'archive dans le dossier C:\mysql.
3. Exécuter, en tant qu'administrateur, une invite de commande
4. Accéder au dossier : C:\mysql\bin
5. Initialiser le système de base de données (authentification, sécurité, . . .) avec la commande  
suivante : `mysqld.exe --initialize-insecure`

```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>d:

D:\>cd mysql

D:\mysql>dir
Volume in drive D is External disk
Volume Serial Number is 44E7-B9EA

Directory of D:\mysql

11/04/2023  02:00 PM  <DIR>          .
11/04/2023  02:00 PM  <DIR>          ..
11/04/2023  02:00 PM  <DIR>          mysql-8.1.0-winx64
               0 File(s)                0 bytes
               3 Dir(s)  297,070,227,456 bytes free

D:\mysql>cd mysql-8.1.0-winx64

D:\mysql\mysql-8.1.0-winx64>cd bin
The system cannot find the path specified.

D:\mysql\mysql-8.1.0-winx64>cd mysql-8.1.0-winx64

D:\mysql\mysql-8.1.0-winx64\mysql-8.1.0-winx64>cd bin

D:\mysql\mysql-8.1.0-winx64\mysql-8.1.0-winx64\bin>mysqld.exe --initialize-insecure

D:\mysql\mysql-8.1.0-winx64\mysql-8.1.0-winx64\bin>_

```

6. Vérifier si un nouveau sous-dossier data dans le répertoire mysql est créé.

bin	6/28/2023 9:00 AM	File folder
data	11/4/2023 2:05 PM	File folder
docs	6/28/2023 8:53 AM	File folder
include	6/28/2023 8:53 AM	File folder
lib	6/28/2023 9:00 AM	File folder
share	6/28/2023 8:53 AM	File folder
LICENSE	6/21/2023 8:52 AM	File
README	6/21/2023 8:52 AM	File

7. Lancer le serveur mysql avec la commande : C:\mysql\bin\mysqld.exe

Ne fermer pas cette fenêtre , cela arrêterait le serveur.

```
D:\mysql\mysql-8.1.0-winx64\mysql-8.1.0-winx64\bin>.\mysqld.exe
```

8. Ouvrir, en tant qu'administrateur, une nouvelle invite de commande.

9. Taper la `netstat -abn` et vérifier si le serveur tourne et noter le port 3306 d'écoute

```
[mysqld.exe]
TCP      [::]:7680          [::]:0          LISTENING
Can not obtain ownership information
TCP      [::]:33060         [::]:0          LISTENING
[mysqld.exe]
TCP      [::]:49664         [::]:0          LISTENING
```

10. Se connecter au serveur mysql avec le client mysql.exe en utilisant la commande de la  
suivante : `C:\mysql\bin\mysql.exe -u root -p` Par défaut, le mot de passe est la  
chaîne de caractères vide.

```
D:\mysql\mysql-8.1.0-win64\mysql-8.1.0-win64\bin>.\mysql.exe -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> _
```

11. Créer un nouvel utilisateur nommé « MyUser » et comme mot de passe « 1234 »  
avec la  
commande mysql : `CREATE USER 'MyUser' '@%' IDENTIFIED BY '1234';`
12. Attribuer à cet utilisateur tous les privilèges pour la base de données avec la  
commande  
suivante : `GRANT ALL PRIVILEGES ON . TO MyUser'@%' WITH GRANT OPTION;`
13. Exécuter la commande `FLUSH PRIVILEGES;` pour que ces changements prennent  
effet  
immédiatement, `FLUSH PRIVILEGES;`
14. Se déconnecter de la base avec la commande : `sql exit;`

```
mysql> CREATE USER 'MyUser'@'%' IDENTIFIED BY '1234';
Query OK, 0 rows affected (0.38 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'MyUser'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.30 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> exit;
Bye

D:\mysql\mysql-8.1.0-win64\mysql-8.1.0-win64\bin>_
```

15. Se connecter à nouveau avec le nouveau compte utilisateur MyUser.
16. Créer la base de données : biblio.
17. Créer la table : Book à cette base. Cette table pourra contenir les champs suivants:
  - book\_id (entier, identifiant),
  - Title (chaîne de caractère),
  - Author (chaîne de caractère),
  - Price (real).

```
D:\mysql\mysql-8.1.0-win64\mysql-8.1.0-win64\bin>.\mysql.exe -u MyUser -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE biblio;
Query OK, 1 row affected (0.31 sec)

mysql> USE biblio;
Database changed
mysql> CREATE TABLE Book (
  ->     book_id INT AUTO_INCREMENT PRIMARY KEY,
  ->     Title VARCHAR(255),
  ->     Author VARCHAR(255),
  ->     Price FLOAT
  -> );
Query OK, 0 rows affected (0.10 sec)
```

18. Insérer 4 books de votre choix.

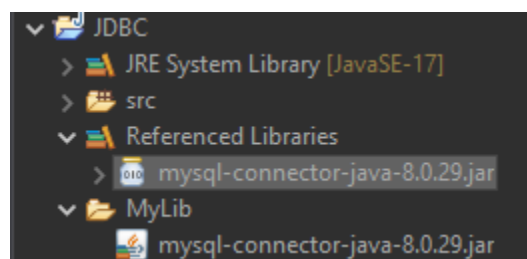
19. Vérifier que ces enregistrements sont dans la base.

```
mysql> INSERT INTO Book (Title, Author, Price) VALUES
-> ('Livre 1', 'Auteur 1', 19.99),
-> ('Livre 2', 'Auteur 2', 15.99),
-> ('Livre 3', 'Auteur 3', 24.99),
-> ('Livre 4', 'Auteur 4', 12.99);
Query OK, 4 rows affected (0.33 sec)
Records: 4  Duplicates: 0  Warnings: 0

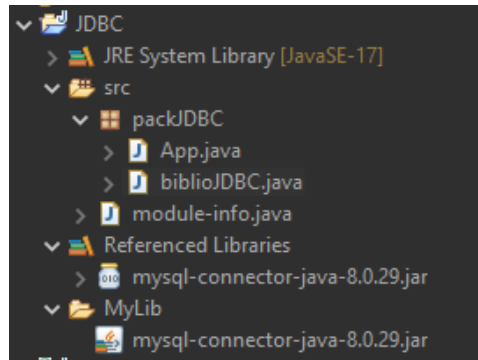
mysql> SELECT * FROM Book;
+-----+-----+-----+-----+
| book_id | Title  | Author | Price |
+-----+-----+-----+-----+
|      1 | Livre 1 | Auteur 1 | 19.99 |
|      2 | Livre 2 | Auteur 2 | 15.99 |
|      3 | Livre 3 | Auteur 3 | 24.99 |
|      4 | Livre 4 | Auteur 4 | 12.99 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

## Exercice 2 : Accès à la BD via le JDBC

1. Lancer Eclipse IDE
2. Créer un nouveau projet Java (simple projet).
3. Créer un nouveau répertoire MyLib dans votre projet Eclipse
4. Télécharger le driver JDBC pour mysql `mysql-connector-java-8.0.29.tar.gz` depuis le lien  
<https://dev.mysql.com/downloads/connector/j/>
5. Décompresser le et copie le fichier `.jar` dans le repertoire MyLib
6. Ajouter le fichier `.jar` au Build Path de votre projet



7. Créer un package `packJDBC`
8. Créer nouvelle classe avec une méthode `main`



9. Créer une méthode biblioConnect() qui permet se connecter à la BD biblio. Cette méthode affiche un message d'erreur en exception si la connexion n'est pas établie.

```

App.java  biblioJDBC.java X
1 package packJDBC;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8
9 public class biblioJDBC {
10     public Connection biblioConnect() {
11         Connection connection = null;
12         try {
13             // Charger le pilote JDBC
14             Class.forName("com.mysql.cj.jdbc.Driver");
15             // Établir la connexion
16             connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/biblio", "MyUser", "1234");
17             System.out.println("Connexion à la base de données établie.");
18         } catch (SQLException | ClassNotFoundException e) {
19             e.printStackTrace();
20             System.err.println("Erreur de connexion à la base de données : " + e.getMessage());
21         }
22         return connection;
23     }
24 }

```

10. Créer la méthode bookDisplay() qui permet d'afficher tous les enregistrements de la table book

```

public void bookDisplay(Connection connection) {
    try {
        String sql = "SELECT * FROM book";
        PreparedStatement statement = connection.prepareStatement(sql);
        ResultSet result = statement.executeQuery();

        while (result.next()) {
            int book_id = result.getInt("book_id");
            String title = result.getString("Title");
            String author = result.getString("Author");
            double price = result.getDouble("Price");
            System.out.println("ID: " + book_id + ", Titre: " + title + ", Auteur: " + author + ", Prix: " + price);
        }

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

```

11. Créer la méthode bookFind() qui permet de trouver un book

```

public void bookFind(Connection connection, int bookId) {
    try {
        String sql = "SELECT * FROM book WHERE book_id = ?";
        PreparedStatement statement = connection.prepareStatement(sql);
        statement.setInt(1, bookId);
        ResultSet result = statement.executeQuery();

        if (result.next()) {
            int book_id = result.getInt("book_id");
            String title = result.getString("Title");
            String author = result.getString("Author");
            double price = result.getDouble("Price");
            System.out.println("ID FOUND: " + book_id + ", Titre: " + title + ", Auteur: " + author + ", Prix: " + price);
        } else {
            System.out.println("Aucun livre trouvé avec l'ID " + bookId);
        }

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

```

12. Créer la méthode bookAdd() qui permet d'ajouter un nouveau book

```

public int bookAdd(Connection connection, String title, String author, double price) {
    int newBookId = -1;
    try {
        String sql = "INSERT INTO book (Title, Author, Price) VALUES (?, ?, ?)";
        PreparedStatement statement = connection.prepareStatement(sql, PreparedStatement.RETURN_GENERATED_KEYS);
        statement.setString(1, title);
        statement.setString(2, author);
        statement.setDouble(3, price);

        int rowsAffected = statement.executeUpdate();

        if (rowsAffected > 0) {
            ResultSet generatedKeys = statement.getGeneratedKeys();
            if (generatedKeys.next()) {
                newBookId = generatedKeys.getInt(1);
                System.out.println("Nouveau livre ajouté avec l'ID : " + newBookId);
            }
        } else {
            System.out.println("Échec de l'ajout du nouveau livre.");
        }

        statement.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return newBookId;
}

```

13. Créer la méthode bookUpdate() qui permet de modifier un book

```

public void bookUpdate(Connection connection, int bookId, double newPrice) {
    try {
        String sql = "UPDATE book SET Price = ? WHERE book_id = ?";
        PreparedStatement statement = connection.prepareStatement(sql);
        statement.setDouble(1, newPrice);
        statement.setInt(2, bookId);

        int rowsAffected = statement.executeUpdate();

        if (rowsAffected > 0) {
            System.out.println("Prix du livre avec l'ID " + bookId + " mis à jour avec succès.");
        } else {
            System.out.println("Échec de la mise à jour du prix du livre.");
        }

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

```

14. Créer la méthode bookDelete() qui permet de supprimer un book



```

public void bookDelete(Connection connection, int bookId) {
    try {
        String sql = "DELETE FROM book WHERE book_id = ?";
        PreparedStatement statement = connection.prepareStatement(sql);
        statement.setInt(1, bookId);

        int rowsAffected = statement.executeUpdate();

        if (rowsAffected > 0) {
            System.out.println("Livre avec l'ID " + bookId + " supprimé avec succès.");
        } else {
            System.out.println("Échec de la suppression du livre.");
        }

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

```

15. Exécuter votre classe pour :
- a. Afficher tous les enregistrements de la table book
  - b. Insérer un nouveau book
  - c. Afficher le nouveau book ajouté
  - d. Modifier le prix de ce book
  - e. Supprimer un book

```

App.java x biblioJDBC.java
1 package packJDBC;
2
3 import java.sql.Connection;
4 import java.sql.SQLException;
5
6 public class App {
7
8     public static void main(String[] args) {
9         // TODO Auto-generated method stub
10        biblioJDBC biblio = new biblioJDBC();
11        Connection connection = biblio.biblioConnect();
12        if (connection != null) {
13            //Afficher tous les enregistrements de la table
14            biblio.bookDisplay(connection);
15            //Insérer un nouveau livre
16            int BookId = biblio.bookAdd(connection, "Nouveau Livre", "Nouvel Auteur", 30);
17            if (BookId != -1) {
18                //Afficher le nouveau livre ajouté
19                biblio.bookFind(connection, BookId);
20                //Modifier le prix de ce livre
21                biblio.bookUpdate(connection, BookId, 19.99);
22                //Supprimer le livre
23                biblio.bookDelete(connection, BookId);
24            }
25
26            // Fermez la connexion à la base de données
27            try {
28                connection.close();
29            } catch (SQLException e) {
30                e.printStackTrace();
31            }
32        } else {
33            System.out.println("La connexion à la base de données a échoué.");
34        }
35    }
36
37 }

```

```

Connexion à la base de données établie.
ID: 1, Titre: Livre 1, Auteur: Auteur 1, Prix: 19.99
ID: 2, Titre: Livre 2, Auteur: Auteur 2, Prix: 15.99
ID: 3, Titre: Livre 3, Auteur: Auteur 3, Prix: 24.99
ID: 4, Titre: Livre 4, Auteur: Auteur 4, Prix: 12.99
Nouveau livre ajouté avec l'ID : 5
ID FOUND: 5, Titre: Nouveau Livre, Auteur: Nouvel Auteur, Prix: 30.0
Prix du livre avec l'ID 5 mis à jour avec succès.
Livre avec l'ID 5 supprimé avec succès.

```