

## TP 4A - Java Entreprise Edition. Introduction à la persistance de données avec JPA

Nicolas Delanoue et Sylvain Joyeau

2021

Le but de ce TP est que vous vous familiarisez avec la spécification JPA.

### Exercice 1 (Bases de données mysql)

1. Téléchargez l'archive `mysql-8.0.22-winx64.zip` de MySQL server via le lien suivant : <https://downloads.mysql.com/archives/community/>
2. Ouvrez le navigateur de fichier en administrateur afin d'extraire le contenu de l'archive dans le dossier `C:\mysql`. Vous devriez obtenir quelque chose comme sur la figure 1.

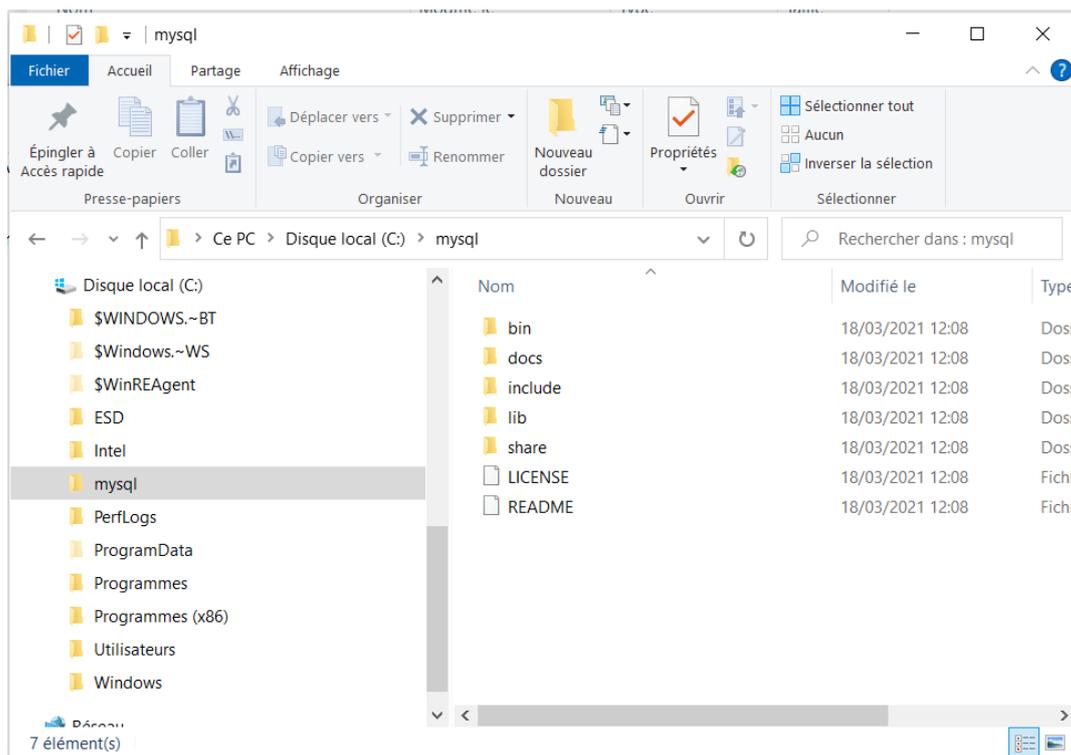


FIGURE 1 – Installation de MySQL.

3. Lancez, en tant qu'administrateur, une invite de commande afin d'initialiser le système de base de données (authentification, sécurité, ...) avec la commande suivante (comme illustré sur la figure 2).

```
C:\mysql\bin\mysqld.exe --initialize-insecure
```



```
Administrateur : Invite de commandes
C:\Windows>cd ..
C:\>cd mysql
C:\mysql>cd bin
C:\mysql\bin>mysql.exe --initialize-insecure
C:\mysql\bin>
```

FIGURE 2 – Intialisation du serveur MySQL.

Cette commande peut être longue ...

4. Observez la présence d'un nouveau sous-dossier `data` dans le répertoire `mysql`.
5. Lancez le serveur `mysql` avec la commande

```
C:\mysql\bin\mysqld.exe
```

Ne fermez pas cette fenêtre, cela arrêterait le serveur.

6. Lancez, en tant qu'administrateur, une invite de commande. Vérifiez que le serveur tourne et écoute sur le port 3306 avec la commande `netstat -abn`.
7. Connectez-vous au serveur `mysql` avec le client `mysql.exe` grace à une commande de la forme

```
C:\mysql\bin\mysql.exe -u root -p
```

Par défaut, le mot de passe est la chaîne de caractères vide.

8. Créez un nouvel utilisateur nommé `userTest` et comme mot de passe `passwordTest` avec une commande `mysql` :

```
CREATE USER 'nouveau_utilisateur'@'%' IDENTIFIED BY 'mot_de_passe';
```

9. Accordez à l'utilisateur récemment créé tous les privilèges pour la base de données avec une commande de la forme

```
GRANT ALL PRIVILEGES ON *.* TO 'nouveau_utilisateur'@'%' WITH GRANT OPTION;
```

10. Pour que ces changements prennent effet immédiatement, on exécutera la commande :

```
FLUSH PRIVILEGES;
```

11. Déconnectez-vous de la base avec la commande `sql exit;`.

12. Connectez-vous avec le nouveau compte utilisateur `userTest`.

13. Ajoutez une base de données nommée `UnionEuropeBD`.

14. Ajoutez une table nommée `Nations` à cette base. Cette table pourra contenir 3 champs :

- un identifiant `id`,
- un nom `nom`,
- une population en millier de citoyens : `population`.

15. Peuplez, toujours depuis la ligne de commande, la table avec 3 pays de votre choix.

16. Vérifiez que ces enregistrements sont dans la base avec une requête du type :

```
SELECT * FROM Nations;
```

### Exercice 2 (Java Database Connectivity)

Dans cet exercice, nous allons mettre en place un driver JDBC pour accéder à la base de données créée dans l'exercice précédent.

1. Lancez Eclipse IDE for Enterprise Java Developers.
2. Créez un simple projet Java nommé `jdbcTest` (et non pas une application web dynamique).
3. Dans votre projet Eclipse, créez un répertoire `MesLibrairies`.
4. Téléchargez le driver JDBC pour mysql nommé `mysql-connector-java-8.0.22.tar.gz` depuis l'url : <https://dev.mysql.com/downloads/connector/j/>.
5. Copiez le fichier `jar` de cette librairie dans le répertoire `MesLibrairies` de votre projet `jdbcTest`.
6. Ajoutez ce fichier `jar` au Build Path de votre projet.
7. Dans un package `jdbcPack`, créez une classe `jdbcMain` avec une méthode `main`.
8. Complétez la méthode suivante (lignes 4 et 6) et insérez-la dans votre projet :

```
1 public static void main(String[] args) throws Exception {
2     //Class.forName("com.mysql.jdbc.Driver"); //chargement du Driver
3     System.out.println( "Affichage de la table Nations");
4     String url = "_____:_____:/_____?serverTimezone=UTC";
5     Connection con=DriverManager.getConnection(url,"_____", "_____");
6     String select = "_____";
7     PreparedStatement ps = con.prepareStatement(select);
8     ResultSet rs=ps.executeQuery();
9     while(rs.next())
10    {
11        String message = rs.getString("etat") + ", ";
12        message = message + rs.getString("population") + ".";
13        System.out.println( message );
14    }
15    con.close();
16    System.out.println( "Opération de lecture terminée");
17 }
```

### Exercice 3 (Java Persistence API - en Standalone)

Dans cet exercice, nous allons créer une entité Class `Nation` qui sera mappée avec la seule table de notre base de données.

1. Lancez Eclipse IDE for Enterprise Java Developers.
2. Créez un simple projet Java nommé `jpaTest` (et non pas une application web dynamique).
3. Créez une classe Java nommée `Nation` avec les bons attributs.
4. Utilisez habilement Eclipse pour automatiquement générer les getter et setter de ces attributs.

5. Ajoutez un constructeur sans argument à cette classe.
6. Téléchargez et décompressez l'implémentation `hibernate` de la spécification JPA via <https://hibernate.org/orm/>.  
*Indications* : Il s'agit d'un fichier nommé `hibernate-release-xxx-final.zip`.
7. Dans votre projet Eclipse, créez un répertoire `MesLibrairies`.
8. Copiez tous les fichiers `jar` contenus du répertoire `hibernate-release-5.4.29.Final/lib/required/` vers le dossier `MesLibrairies`.
9. Comme dans l'exercice précédent, ajoutez aussi au dossier `MesLibrairies` le fichier `jar` contenant le driver `JDBC`.
10. Ajoutez tous ces fichiers `jar` au Build Path de votre projet.
11. Transformez votre classe `Nation` en entité avec l'aide d'annotations JPA comme `@Entity`, `@Table` et `@Id`.
12. Créez un fichier de configuration nommé `persistence.xml` dans le dossier `src/META-INF/` de votre projet. (Créez le dossier `src/META-INF/` si nécessaire).
13. Dans un package `jpaPack`, créez une classe `jpaMain` avec une méthode `main`.
14. Créez une unité de persistance nommée `UniteUnionEurope` à partir à partir du modèle suivant :

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <persistence version="2.2"
3      xmlns="http://xmlns.jcp.org/xml/ns/persistence"
4      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5      xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
6      http://xmlns.jcp.org/xml/ns/persistence/persistence_2_2.xsd">
7  <persistence-unit name="_____">
8  <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
9  <class> _____._____ </class>
10 <properties>
11 <property name="javax.persistence.jdbc.driver"
12     value="com.mysql.jdbc.Driver" />
13 <property name="javax.persistence.jdbc.url"
14     value="_____:_____:_____/_____?serverTimezone=UTC" />
15 <property name="javax.persistence.jdbc.user" value="_____" />
16 <property name="javax.persistence.jdbc.password" value="_____" />
17 </properties>
18 </persistence-unit>
19 </persistence>

```

15. Complétez la méthode suivante et insérez-la dans votre projet :

```

1  public static void main(String[] args) throws Exception {
2      EntityManagerFactory emf ;
3      emf = Persistence.createEntityManagerFactory("_____");
4      EntityManager entityManager = emf.createEntityManager();
5      Query q = entityManager.createQuery( _____ , _____ );
6      List<Nation> nations ;

```

```
7   etats = q.getResultList();
8   for (Nation n : nations) {
9       System.out.println( n.toString() );
10  }
11 }
```

16. Ajoutez au code précédent des instructions s'appuyant sur l'objet `entityManager` afin de tester les opérations de création, mise à jour et suppression d'enregistrements.
17. Si le temps le permet, lancez `wireshark` afin de scruter les échanges entre votre application et le serveur de bases de données.

#### Exercice 4 (Dao sur JPA)

1. Mettez en place le patron de conception DAO afin de proposer :
  - (a) une méthode qui liste toutes nations qui ont plus  $n$  citoyens,
  - (b) une méthode qui permet de modifier le nombre d'habitants d'une nation donnée.
2. Ajouter une table nommée `Citoyens` à votre base avec les champs `Id`, `nom`, `prenom` et une clé étrangère pour connaître le pays de naissance.
3. Développez le code JPA permettant de persister facilement des citoyens dans votre base.
4. Ajoutez une méthode à votre DAO pour récupérer tous les citoyens d'une nation donnée.