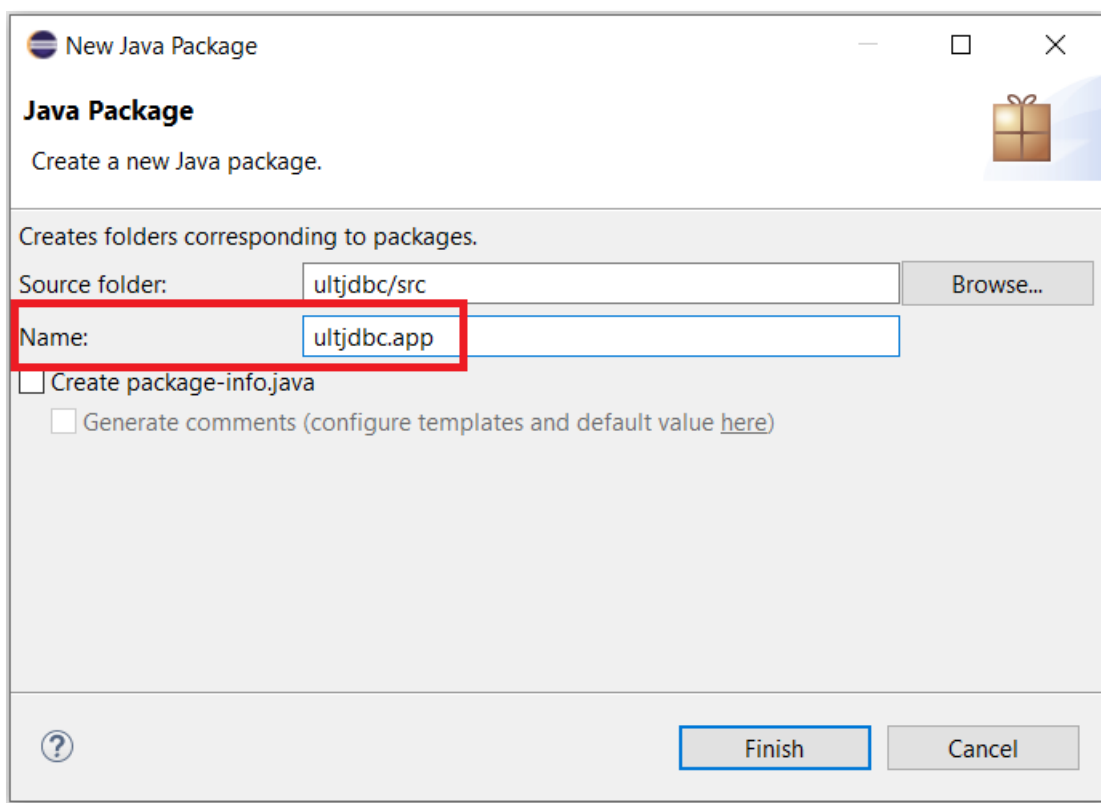


# Création d'un projet Java avec une base de données MySQL


Formateur : Dr Ing Mohamed Amine MEZGHICH

Email : amine.mezghich@gmail.com

- 1) Créer un nouveau projet ultjdbc
- 2) On va créer sous src un premier package « app »



- 3) Créer dans le package une Classe Main.java avec la fonction d'entrée public static void main(String args[])

 New Java Class

**Java Class**  
Create a new Java class.

Source folder:

Package:

☐ Enclosing type:

---

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

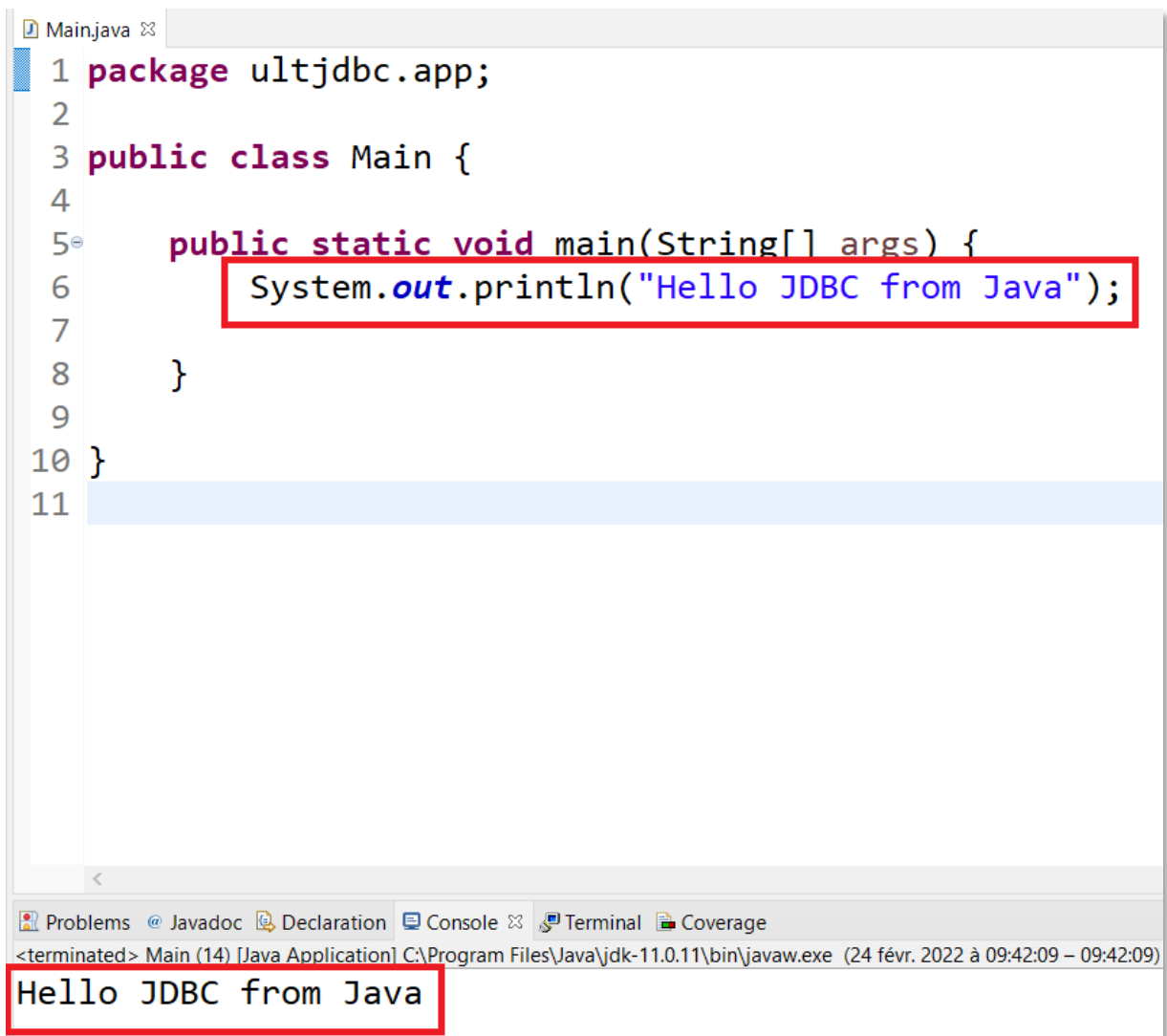
Which method stubs would you like to create?

☒ public static void main(String[] args)  
☐ Constructors from superclass  
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))  
☐ Generate comments

4)Placer le code suivant :

```
package ultjdbc.app;  
  
public class Main {  
  
    public static void main(String[] args) {  
        System.out.println("Hello JDBC from Java");  
    }  
}
```



The screenshot shows an IDE window titled 'Main.java'. The code is as follows:

```
1 package ultjdbc.app;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         System.out.println("Hello JDBC from Java");
7     }
8 }
9
10
11
```

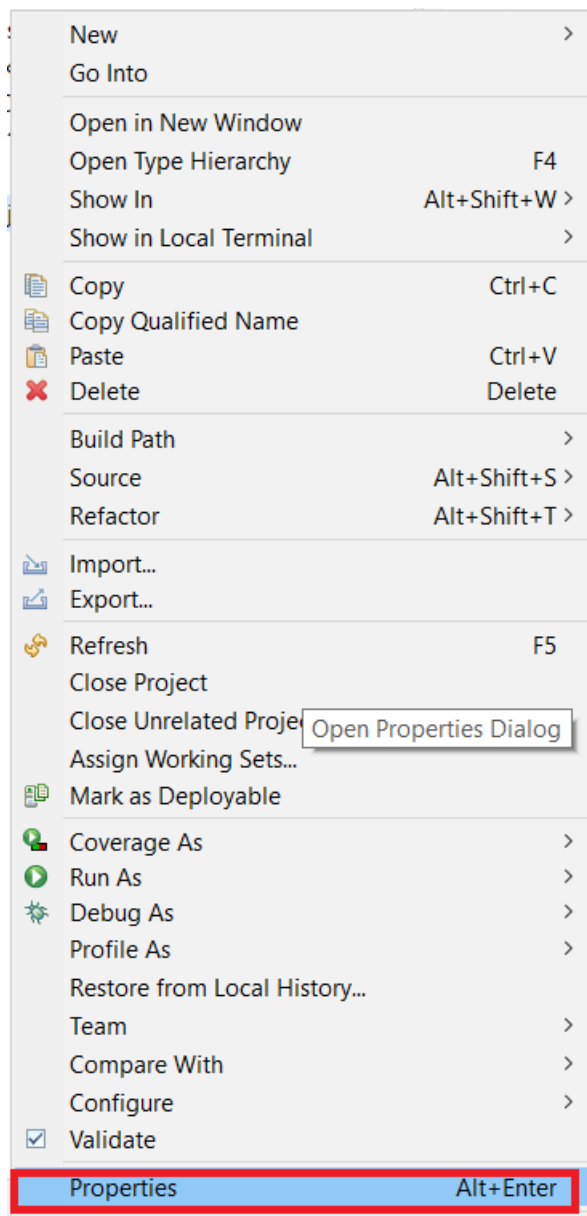
The line `System.out.println("Hello JDBC from Java");` is highlighted with a red box. Below the code editor, the 'Terminal' tab is active, showing the command prompt output:

```
<terminated> Main (14) [Java Application] C:\Program Files\Java\jdk-11.0.11\bin\javaw.exe (24 févr. 2022 à 09:42:09 - 09:42:09)
Hello JDBC from Java
```

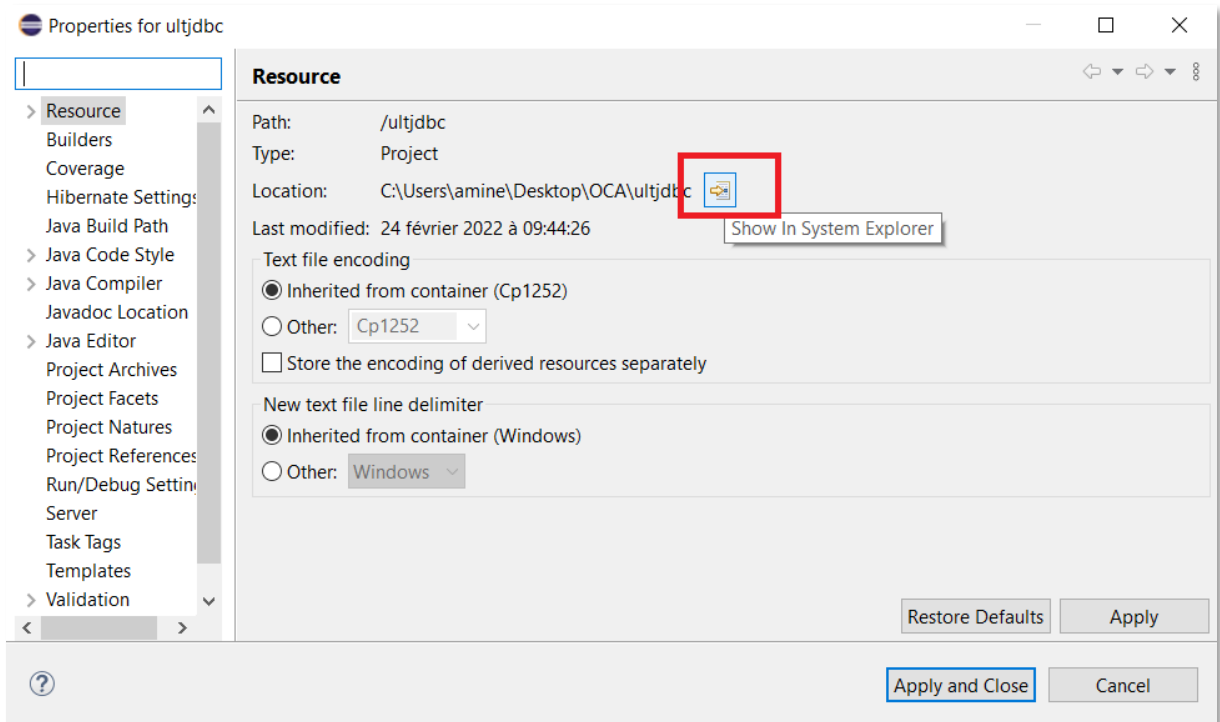
The output 'Hello JDBC from Java' is also highlighted with a red box.

5) Placer le jar « mysql-connector-java-8.0.19.jar » dans le dossier du projet, voici les étapes

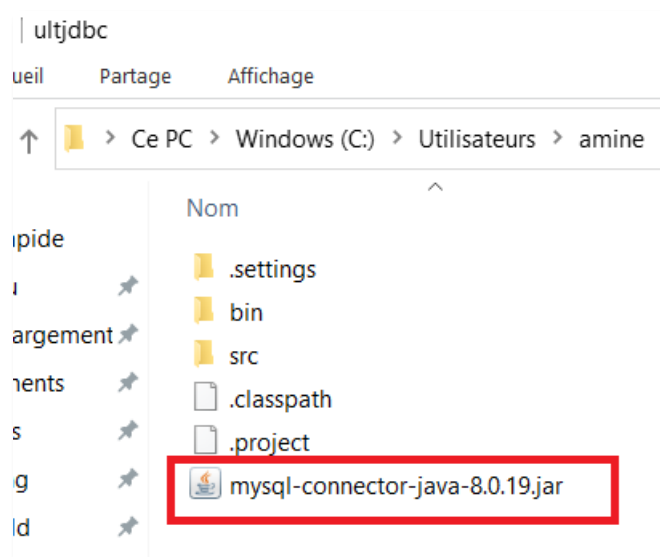
5.1) Sélectionner le projet, puis cliquer sur propriétés



## 5.2) Cliquer sur location



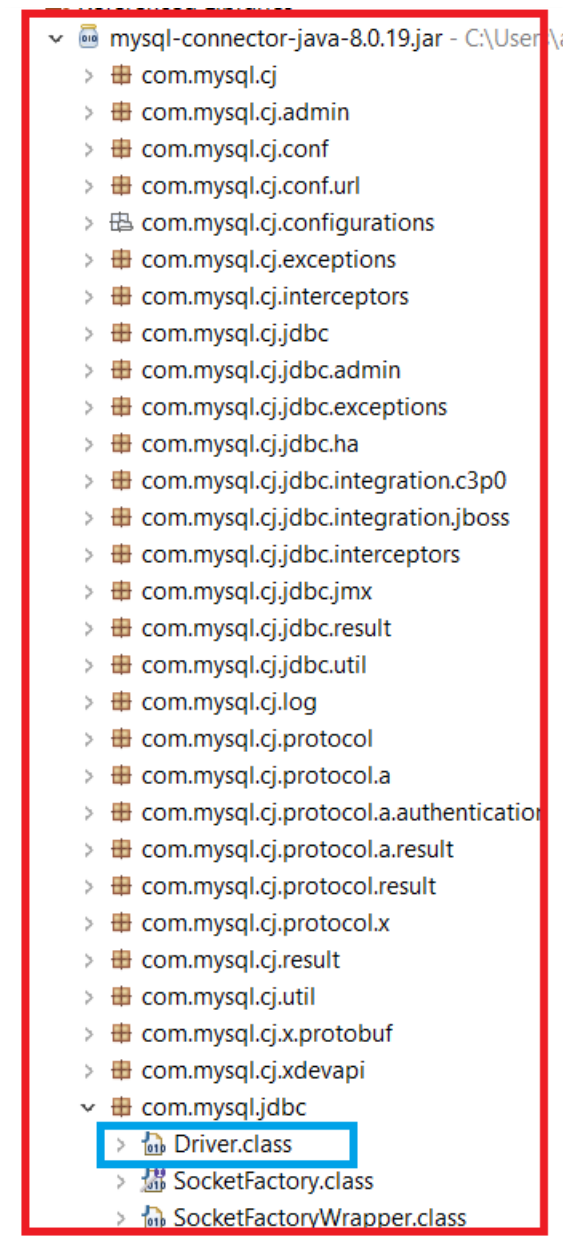
### 5.3) Placer le jar dans le projet



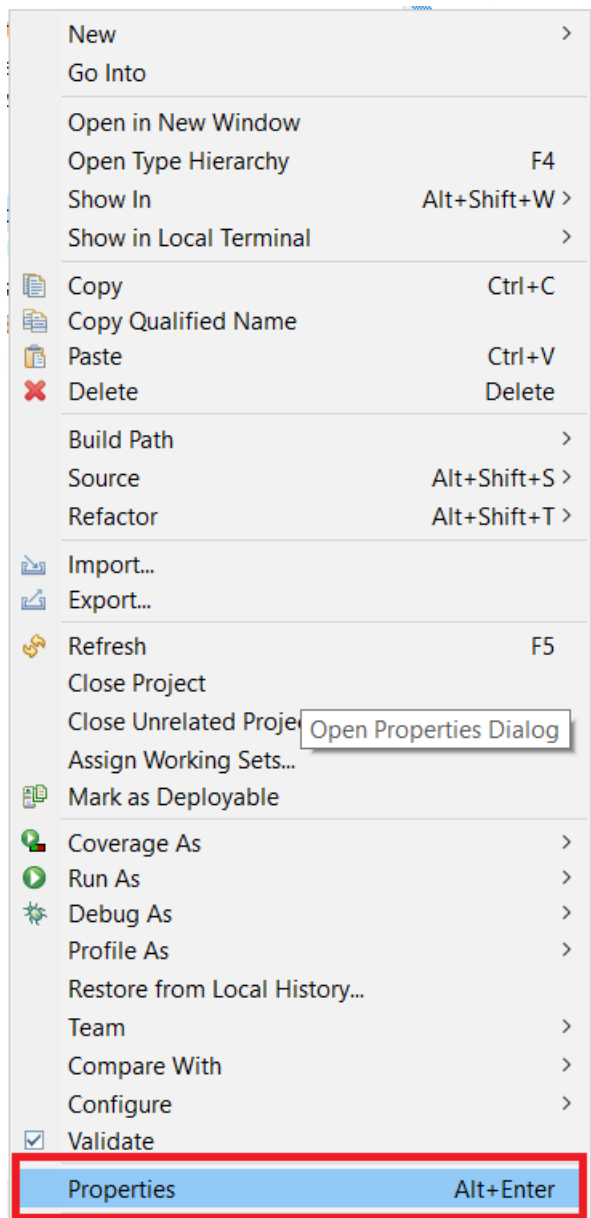
### 6) Connecter notre projet au jar

**Mysql-connector-java-8.x .jar** : est un fichier compressé qui contient plusieurs classes permettant d'assurer la communication entre les application java et la base de données type mysql.

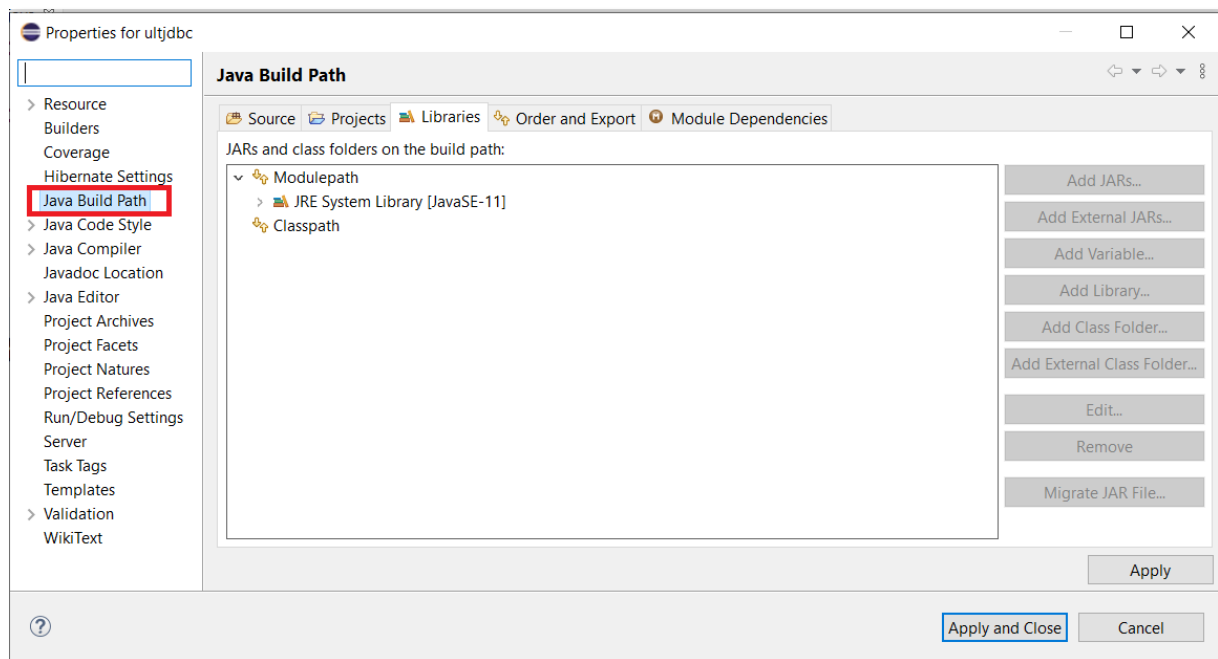
Parmi ces classes, on cite {Driver, Connection, ResultSet et Statement}



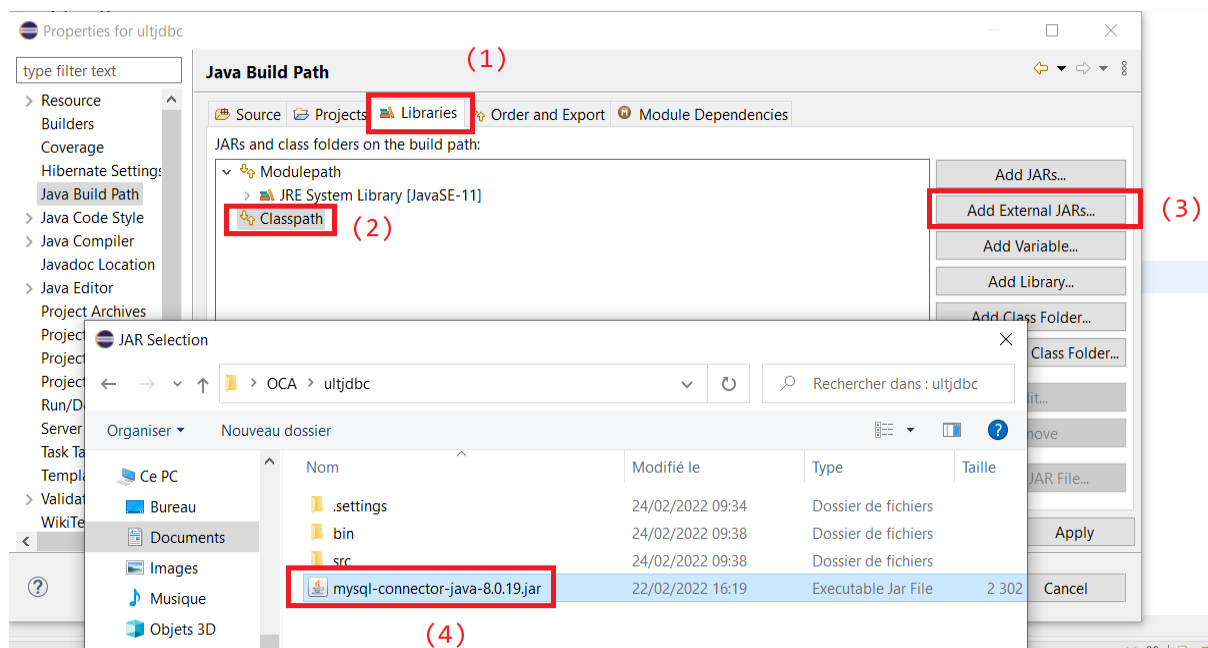
6.1) Cliquer sur properties



6.2) Choisir Java Build Path

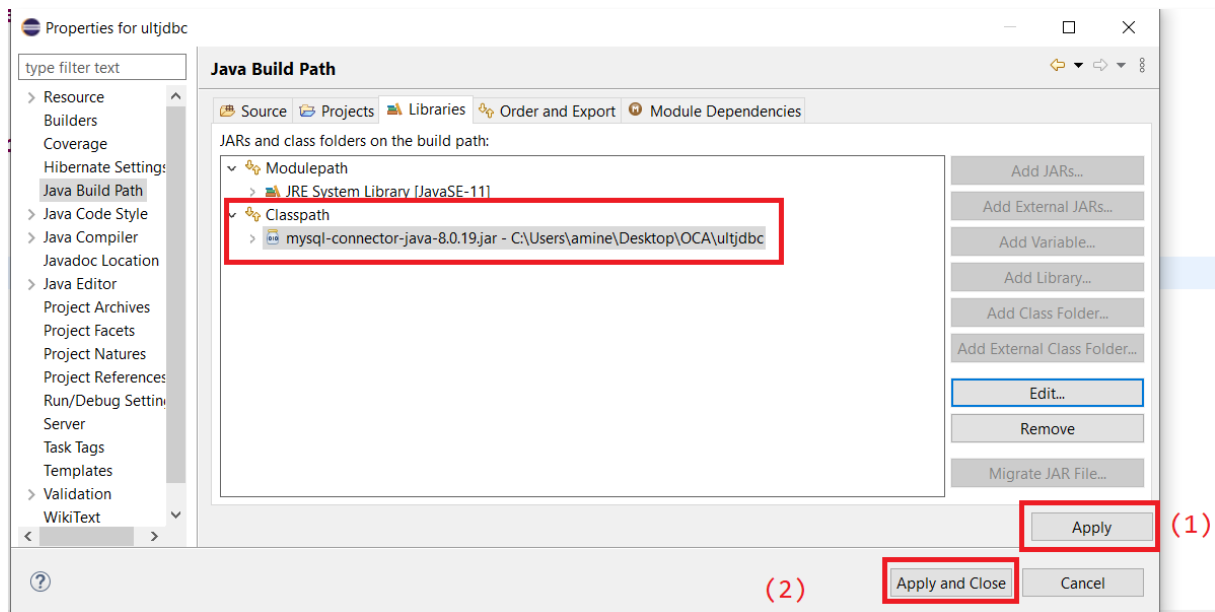


## 6.2) Ajouter le jar au class path



## 6.3) Cliquer sur Apply ensuite sur Apply and Close





6.4)Vérifier que le jar est bien ajouté comme référence dans votre projet



7)Ajouter l'import des classes de java.sql

```
import java.sql.*;
```

8)On va créer une base de données sous mysql

8.1)Installer Xamp

8.2)Lancer Xamp

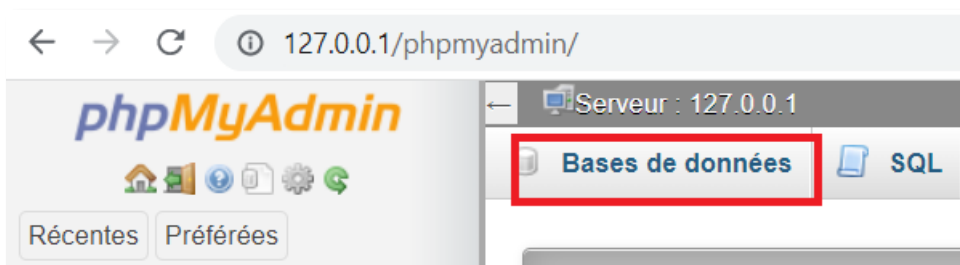
8.3)Accéder à Xamp via l'adresse 127.0.0.1



#### 8.4) Cliquer sur phpMyAdmin



#### 8.5) Créer une nouvelle base de données

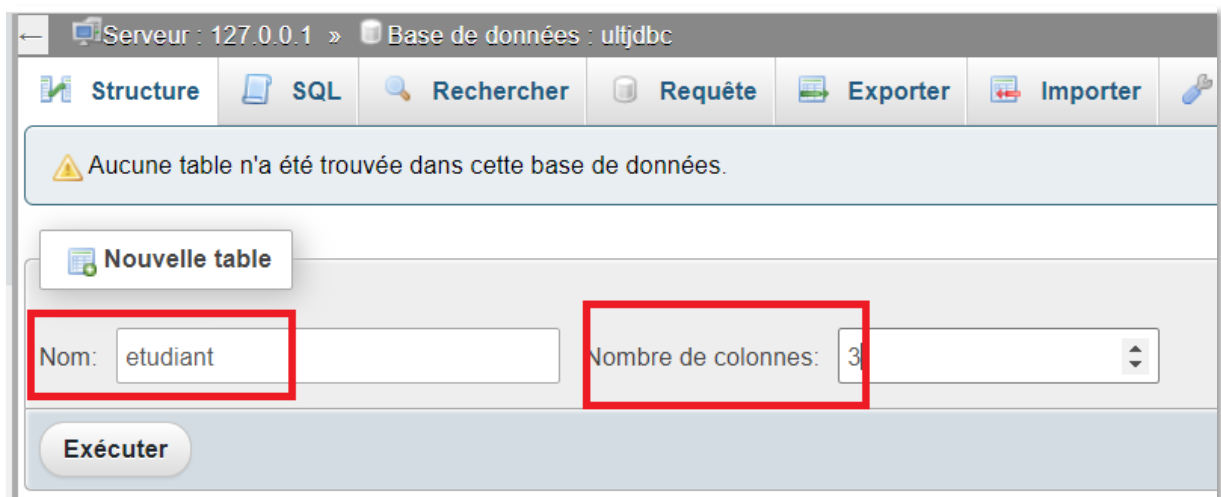


#### 8.6) Donner le nom à votre base de données



Puis cliquer sur Créer

8.7) Créer une première table etudiant(id, nom, classe)



Puis cliquer sur exécuter

8.8) Configurer les colonnes comme suit :

Seigneur : 127.0.0.1 » Base de données : ultjdbc » Table : etudiant

Nom de table: etudiant Ajouter 1 colonne(s) Exécuter

| Nom    | Type    | Taille/Valeurs | Valeur par défaut | Interclassement | Attributs | Null | Index   | A_I                                 | Com |
|--------|---------|----------------|-------------------|-----------------|-----------|------|---------|-------------------------------------|-----|
| id     | INT     | 3              | Aucun(e)          |                 |           |      | PRIMARY | <input checked="" type="checkbox"/> |     |
| nom    | VARCHAR | 50             | Aucun(e)          |                 |           |      |         |                                     |     |
| classe | VARCHAR | 50             | Aucun(e)          |                 |           |      |         |                                     |     |

Cliquer ensuite sur le bouton enregistrer  
Vous obtenez la structure suivante :

Seigneur : 127.0.0.1 » Base de données : ultjdbc » Table : etudiant

Parcourir Structure SQL Rechercher Insérer

Structure de table Vue relationnelle

| #                          | Nom    | Type        | Interclassement    | Attributs | Null | Va |
|----------------------------|--------|-------------|--------------------|-----------|------|----|
| <input type="checkbox"/> 1 | id     | int(3)      |                    |           | Non  | Au |
| <input type="checkbox"/> 2 | nom    | varchar(50) | utf8mb4_general_ci |           | Non  | Au |
| <input type="checkbox"/> 3 | classe | varchar(50) | utf8mb4_general_ci |           | Non  | Au |

↑ ☐ Tout cocher Avec la sélection : Parcourir Modifier

9)Ajouter le code suivant dans votre fonction main

```
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/ultjdbc","root","");
System.out.println(con);
```

Et tester

```
1 package ultjdbc.app;
2 import java.sql.*;
3 public class Main {
4
5     public static void main(String[] args) throws SQLException {
6         System.out.println("Hello JDBC from Java");
7         // STEP 1: on établit une connexion avec la base de données mysql
8         Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/ultjdbc", "root", "");
9         System.out.println(con);
10    }
11 }
12
13 }
14
```

Chaine de connexion  
vers votre Base de données en  
locale

ce sont  
vos paramètres  
d'accès  
(login = root  
et  
mot de passe = "

Hello JDBC from Java  
com.mysql.cj.jdbc.ConnectionImpl@10db82ae

## 10) Insertion d'un étudiant dans la base

```
Statement stmt = con.createStatement();

int result = stmt.executeUpdate("insert into etudiant(nom,classe) values('Omar', '3 ieme')");

System.out.println(result);
```

L'objet Statement permet d'insérer des objets dans la base

La méthode executeUpdate : accepte une requête SQL et fait l'insertion dans la table etudiant

```
1 package ultjdbc.app;
2 import java.sql.*;
3 public class Main {
4
5     public static void main(String[] args) throws SQLException {
6         System.out.println("Hello JDBC from Java");
7         // STEP 1: on établit une connexion avec la base de données mysql
8         Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/ultjdbc", "root", "");
9         System.out.println(con);
10
11         // STEP 2: Création d'un objet statement pour écrire les commandes SQL
12         Statement stmt = con.createStatement();
13         int result = stmt.executeUpdate("insert into etudiant(nom,classe) values('Omar', '3 ieme')");
14         System.out.println(result);
15
16     }
17
18 }
```

Exécuter Run => Vous obtenez une ligne dans la table \*



Server: 127.0.0.1 » Base de données: ultjdbc » Table: etudiant

Parcourir Structure SQL Rechercher Insérer Exporter Importer Privileges Opérations Suivi Déclencheurs

Affichage des lignes 0 - 0 (total de 1, traitement en 0,0006 seconde(s).)

SELECT \* FROM `etudiant`

Profilage [ Éditer en ligne ] [ Éditer ] [ Expliquer SQL ] [ Créer le code source PHP ] [ Actualiser ]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

Options

| id | nom  | classe |
|----|------|--------|
| 1  | Omar | 3 ieme |

**Exercice :** Lecture du nom et la classe à partir du clavier en utilisant la classe Scanner du package java.util

Correction

```
package ultjdbc.app;
import java.sql.*;
import java.util.Scanner;
public class Main {

    public static void main(String[] args) throws SQLException {
        System.out.println("Hello JDBC from Java");
        // STEP 1: on établit une connexion avec la base de données
mysql
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/ultjdbc","root","");
        System.out.println(con);

        //Lecture des paramètres à partir du clavier
        Scanner sc = new Scanner(System.in);
        System.out.println("Donner un nom");
        String nom = sc.nextLine();

        System.out.println("Donner la classe");
        String classe = sc.nextLine();

        // STEP 2: Création d'un objet statement pour écrire les commandes
SQL
        Statement stmt = con.createStatement();
        int result = stmt.executeUpdate("insert into etudiant(nom,classe)
values('"+nom+"','"+classe+"')");
        //System.out.println(result); // ici j'affiche le nombre
d'élément insérer

        System.out.println("Fin du programme");
    }
}
```

```

    }
}

```

## 11)Affichage des données à travers la classe ResultSet

```

// STEP 3: Création objet ResultSet(table en mémoire) dont le rôle est de stocker
ResultSet rs = stmt.executeQuery("select id, nom, classe from etudiant");

// STEP 4: On fait le parcours du ResultSet et on affiche son contenu

while(rs.next()) {
    |
    int id = rs.getInt("id");
    String nom = rs.getString("nom");
    String classe = rs.getString("classe");
    System.out.println(id+ " "+nom + " "+classe);
}

```

## Le code

```

package ultjdbc.app;
import java.sql.*;
import java.util.Scanner;
public class Main {

    public static void main(String[] args) throws SQLException {
        //System.out.println("Hello JDBC from Java");
        // STEP 1: on établit une connexion avec la base de données
mysql
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/ultjdbc","root","");
        //System.out.println(con);

        //Lecture des paramètres à partir du clavier
        /*
        Scanner sc = new Scanner(System.in);
        System.out.println("Donner un nom");
        String nom = sc.nextLine();

        System.out.println("Donner la classe");
        String classe = sc.nextLine();*/

        // STEP 2: Création d'un objet statement pour écrire les commandes
SQL

```

```

        Statement stmt = con.createStatement();
        //int result = stmt.executeUpdate("insert into
etudiant(nom,classe) values('"+nom+"', '"+classe+"')");
        //System.out.println(result); // ici j'affiche le nombre
d'élément insérer

        // STEP 3: Création objet ResultSet(table en mémoire) dont le
rôle est de stocker le résultat d'une requete select
        ResultSet rs = stmt.executeQuery("select id, nom, classe
from etudiant");

        // STEP 4: On fait le parcours du ResultSet et on affiche
son contenu

        while(rs.next()) {

            int id = rs.getInt("id");
            String nom = rs.getString("nom");
            String classe = rs.getString("classe");
            System.out.println(id+ " "+nom + " "+classe);
        }
        System.out.println("Fin du programme");

    }
}

```

Lien github du projet :

<https://github.com/MezghichGit/jdbcult>