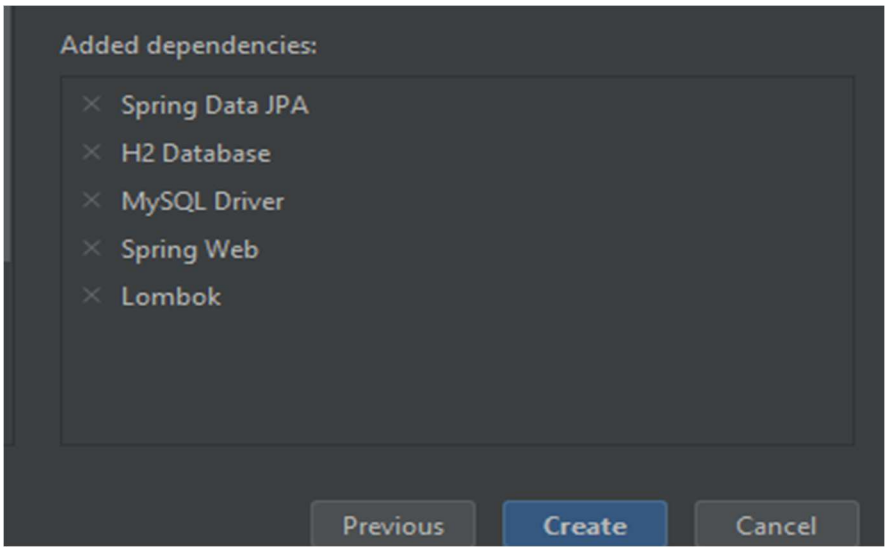
Compte rendu du tp

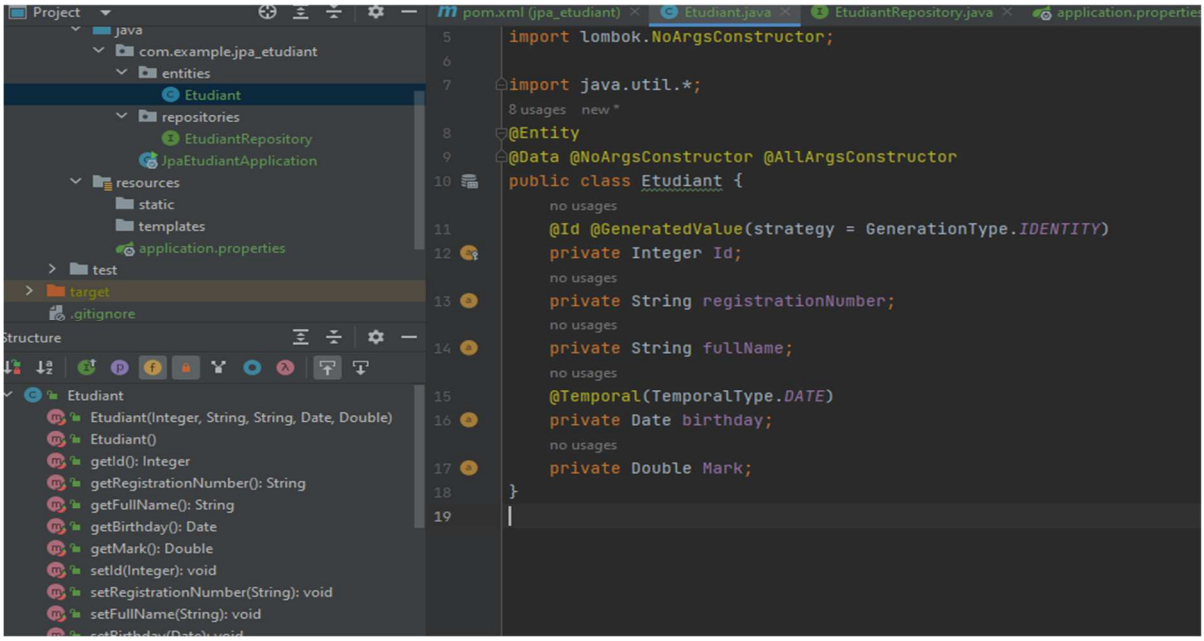
Réalisé par ELHANNIOUI CHAYMAE

En 4ème année Miage G5

1/ Tout d'abord, on a créé notre projet spring boot jpa_etudiant avec les dépandances : spring data jpa, lombok, spring web, et h2 database



2/ Après on a créé l'entité Jpa Etudiant



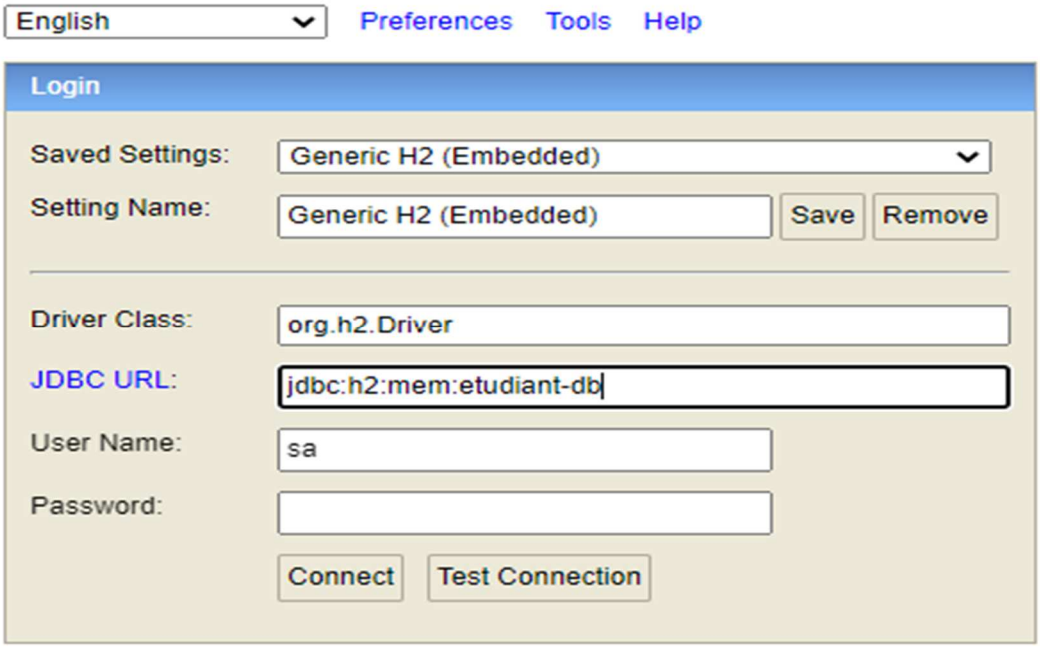
3/ Ensuite, on a configuré la data source(application properties):

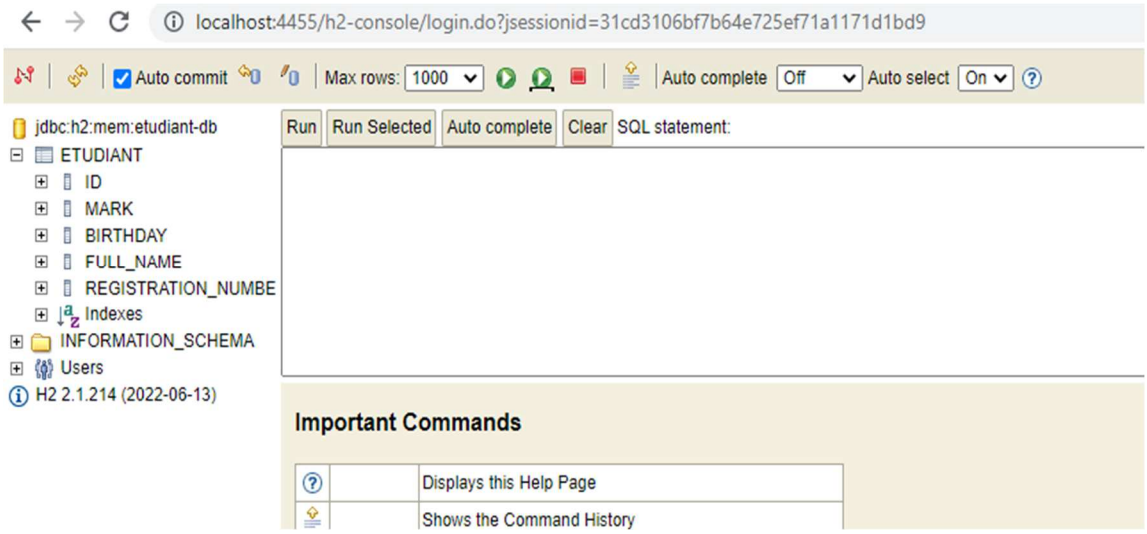
```
spring.datasource.url=jdbc:h2:mem:etudiant-db
spring.h2.console.enabled=true
server.port=4455
```

4/ Création de l'interface Etudiant Repository basé sur spring data.

```
pom.xml (jpa_etudiant) | Etudiant.java | EtudiantRepository.java | application.properties | JpaEtudiantAp
1  package com.example.jpa_etudiant.repositories;
2
3  import com.example.jpa_etudiant.entities.Etudiant;
4  import org.springframework.data.jpa.repository.JpaRepository;
5
   2 usages  new *
6  public interface EtudiantRepository extends JpaRepository<Etudiant, Integer> {
7
8
9  }
10
```

5/ Connexion à la base de données etudiant-db





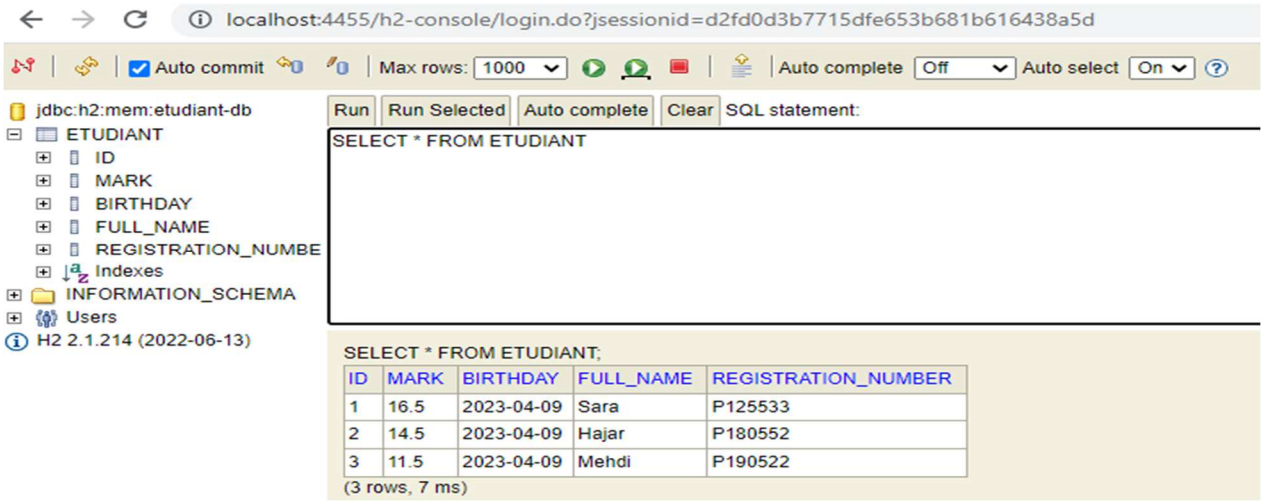
6/ Test de l'application:

L'ajout des étudiants:

```java
package com.example.jpa_etudiant;

import com.example.jpa_etudiant.entities.Etudiant;
import com.example.jpa_etudiant.repositories.EtudiantRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.util.*;

1 usage  new *
@SpringBootApplication
public class JpaEtudiantApplication implements CommandLineRunner {
    4 usages
    @Autowired
    private EtudiantRepository etudiantRepository;
    no usages  new *
    public static void main(String[] args) {
        SpringApplication.run(JpaEtudiantApplication.class, args);
    }
```

```java
new *
public void run(String... args) throws Exception {
    etudiantRepository.save(
            new Etudiant( Id: null, registrationNumber: "P125533", fullName: "Sara",new Date(), Mark: 16.50));
    etudiantRepository.save(
            new Etudiant( Id: null, registrationNumber: "P180552", fullName: "Hajar",new Date(), Mark: 14.50));
    etudiantRepository.save(
            new Etudiant( Id: null, registrationNumber: "P190522", fullName: "Mehdi",new Date(), Mark: 11.50));
```

On constate l'ajout des étudiants dans la base de données:



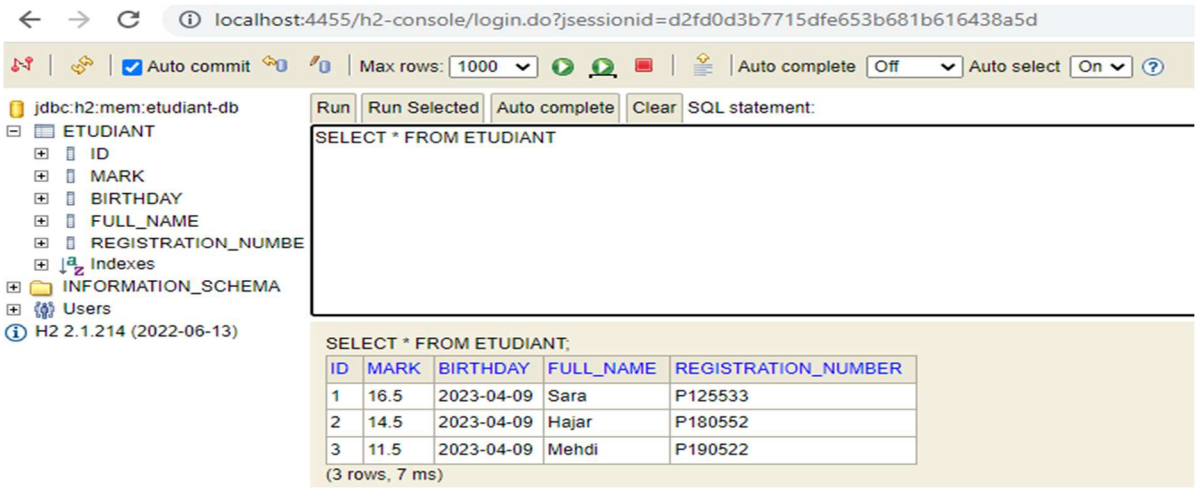
L'affichage de la liste des étudiants

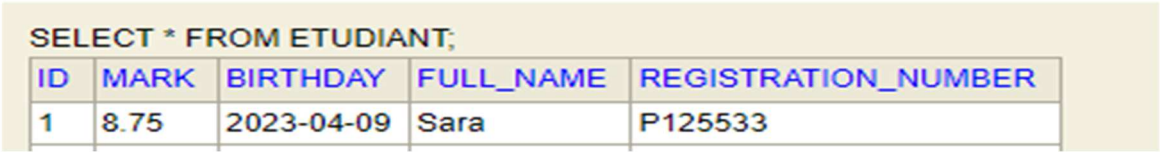
```
    List<Etudiant> etudiants= etudiantRepository.findAll();
    etudiants.forEach(e->{
        System.out.println("===============");
        System.out.println(e.getId());
        System.out.println(e.getRegistrationNumber());
        System.out.println(e.getFullName());
        System.out.println(e.getBirthday());
        System.out.println(e.getMark());

    });
    System.out.println("**********************");


}
```

La modification de la note de l'étudiant de l'id 1

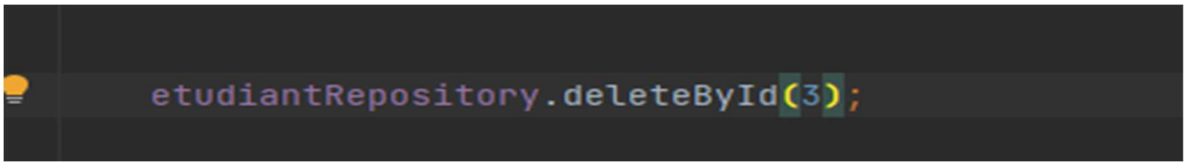
```
    Etudiant etudiant = etudiantRepository.findById(1).orElse( other: null);
    if(etudiant!=null){
        System.out.println(etudiant.getRegistrationNumber());
        System.out.println(etudiant.getFullName());
        System.out.println(etudiant.getBirthday());


    }
    etudiant.setMark(08.75);
    etudiantRepository.save(etudiant);
```

On constate que la note de l'étudiant a été bien modifiée





La suppression de l'étudiant d'id 3



L'étudiant a été supprimé automatiquement de la base de données