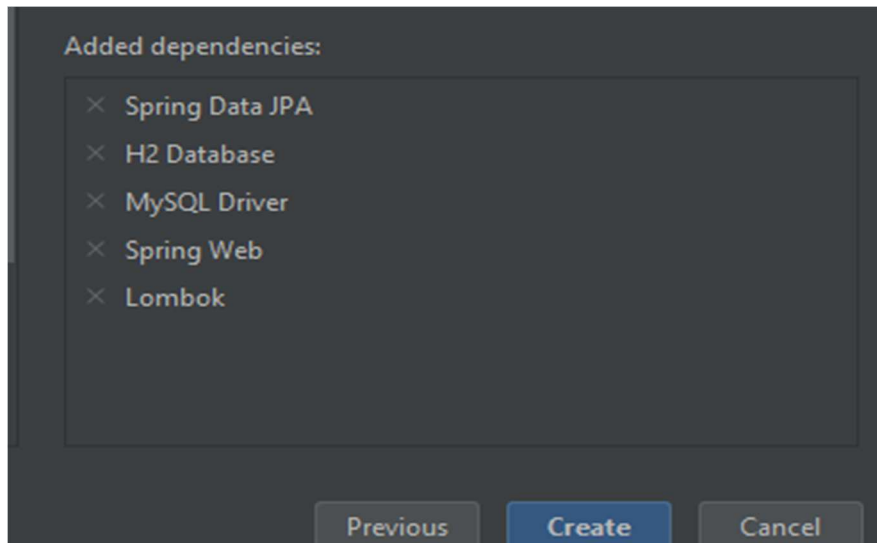


Compte rendu du tp

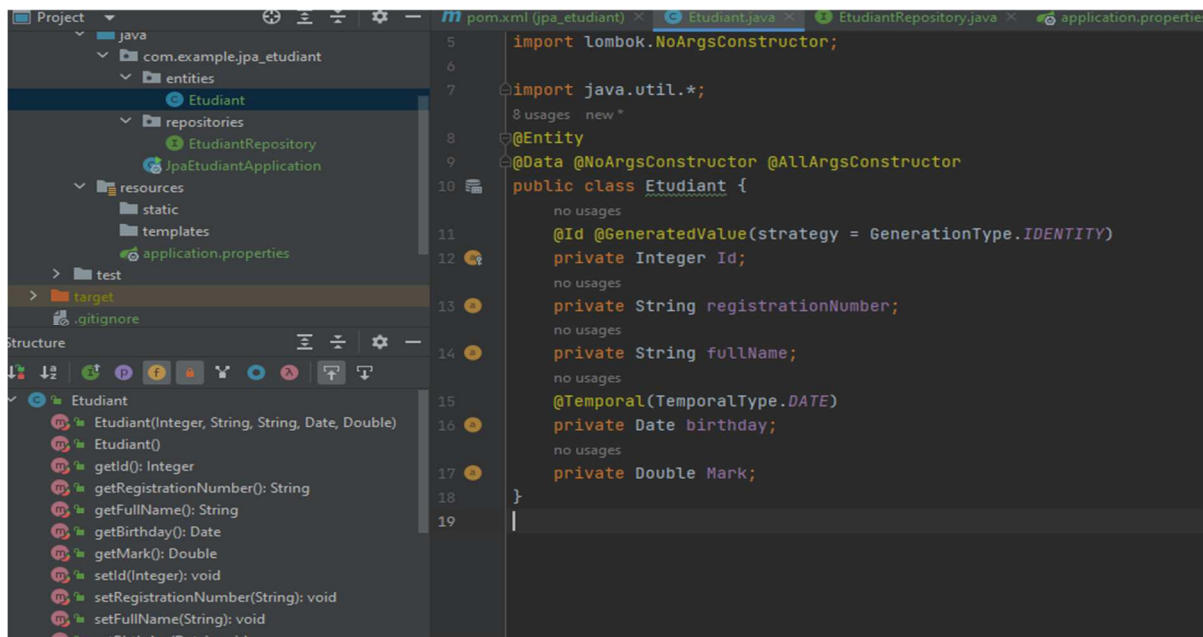
Réalisé par ELHANNIOUI CHAYMAE

En 4ème année Miage G5

1/ Tout d'abord, on a créé notre projet spring boot jpa_etudiant avec les dépendances : 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.

```
package com.example.jpa_etudiant.repositories;

import com.example.jpa_etudiant.entities.Etudiant;
import org.springframework.data.jpa.repository.JpaRepository;

public interface EtudiantRepository extends JpaRepository<Etudiant, Integer> {
}
```

5/ Connexion à la base de données etudiant-db

English Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded) ▼

Setting Name: Generic H2 (Embedded) Save Remove

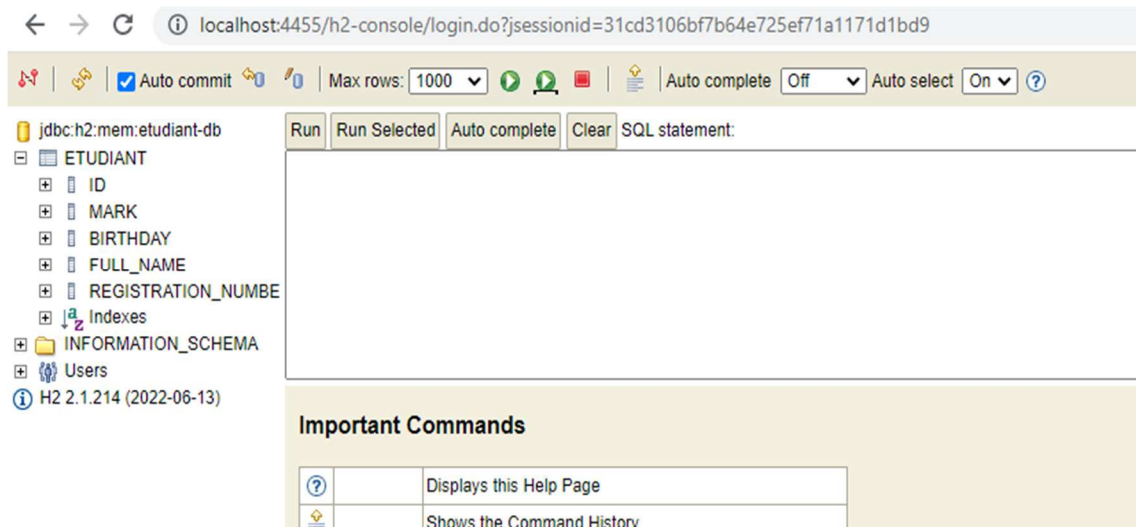
Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:etudiant-db

User Name: sa

Password:

Connect Test Connection



6/ Test de l'application:

L'ajout des étudiants:

```
package com.example.jpa_etudiant;

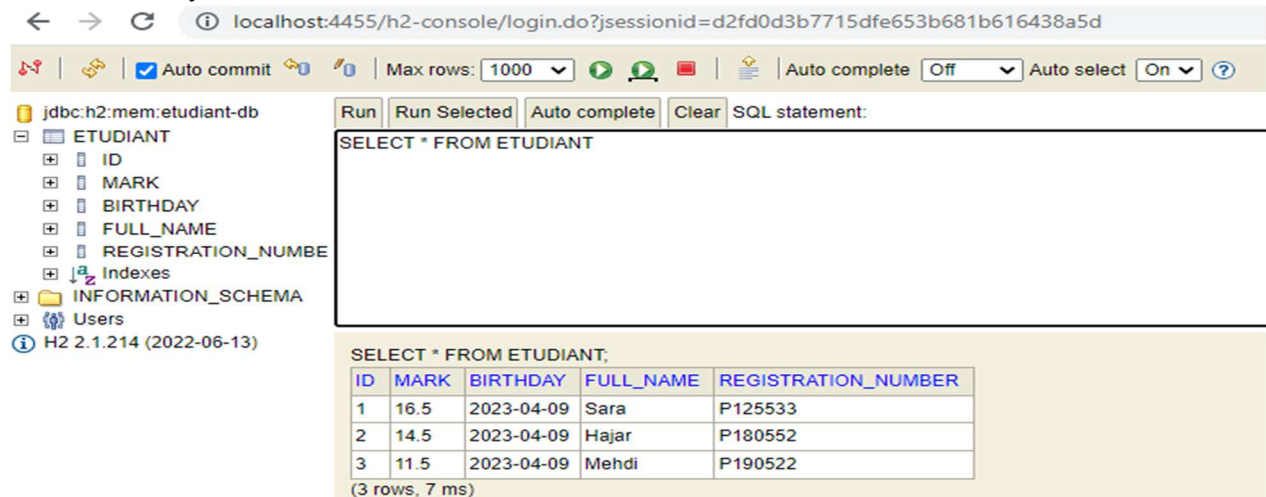
import com.example.jpa_etudiant.entities.Etudiant;
import com.example.jpa_etudiant.repositories.EtudiantRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

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);
    }
}
```

```
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:



The screenshot shows a web application interface for a database. The URL bar indicates a connection to localhost:4455/h2-console/login.do. The interface includes a sidebar with a tree view showing the database structure: jdbc:h2:mem:etudiant-db, ETUDIANT (with columns ID, MARK, BIRTHDAY, FULL_NAME, REGISTRATION_NUMBE, and Indexes), INFORMATION_SCHEMA, and Users. The main area displays the SQL statement 'SELECT * FROM ETUDIANT' and its results in a table with 3 rows and 5 columns: ID, MARK, BIRTHDAY, FULL_NAME, and REGISTRATION_NUMBER. The results are: (1, 16.5, 2023-04-09, Sara, P125533), (2, 14.5, 2023-04-09, Hajar, P180552), and (3, 11.5, 2023-04-09, Mehdi, P190522). The status bar indicates '(3 rows, 7 ms)'.

ID	MARK	BIRTHDAY	FULL_NAME	REGISTRATION_NUMBER
1	16.5	2023-04-09	Sara	P125533
2	14.5	2023-04-09	Hajar	P180552
3	11.5	2023-04-09	Mehdi	P190522

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

localhost:4455/h2-console/login.do?jsessionid=d2fd0d3b7715dfe653b681b616438a5d

Auto commit ☒ Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:mem:etudiant-db

- ETUDIANT
 - ID
 - MARK
 - BIRTHDAY
 - FULL_NAME
 - REGISTRATION_NUMBER
 - Indexes
- INFORMATION_SCHEMA
- Users
- H2 2.1.214 (2022-06-13)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM ETUDIANT

ID	MARK	BIRTHDAY	FULL_NAME	REGISTRATION_NUMBER
1	16.5	2023-04-09	Sara	P125533
2	14.5	2023-04-09	Hajar	P180552
3	11.5	2023-04-09	Mehdi	P190522

(3 rows, 7 ms)

SELECT * FROM ETUDIANT;

ID	MARK	BIRTHDAY	FULL_NAME	REGISTRATION_NUMBER
1	8.75	2023-04-09	Sara	P125533

La suppression de l'étudiant d'id 3

```
etudiantRepository.deleteById(3);
```

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

SELECT * FROM ETUDIANT;

ID	MARK	BIRTHDAY	FULL_NAME	REGISTRATION_NUMBER
1	8.75	2023-04-09	Sara	P125533
2	14.5	2023-04-09	Hajar	P180552

(2 rows, 5 ms)