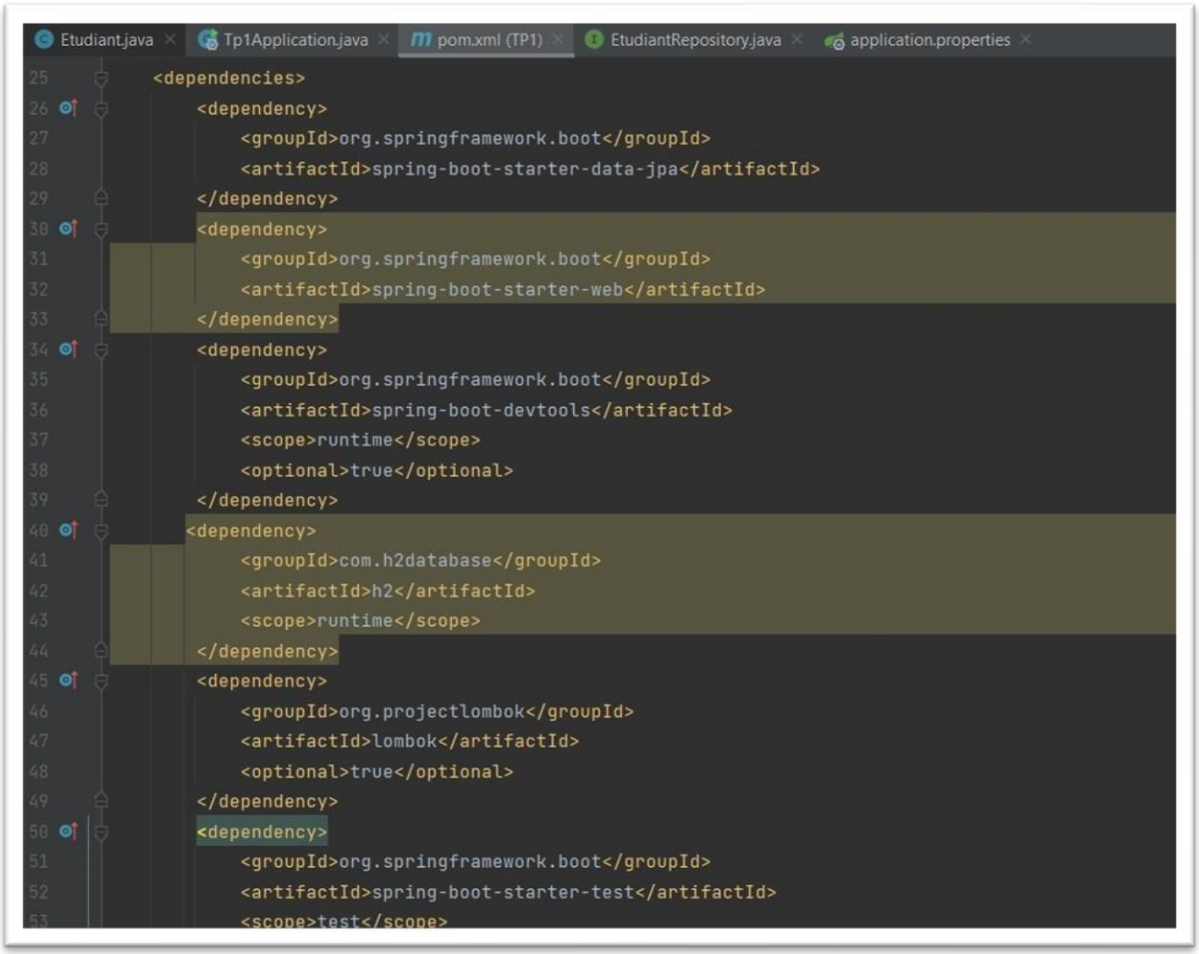
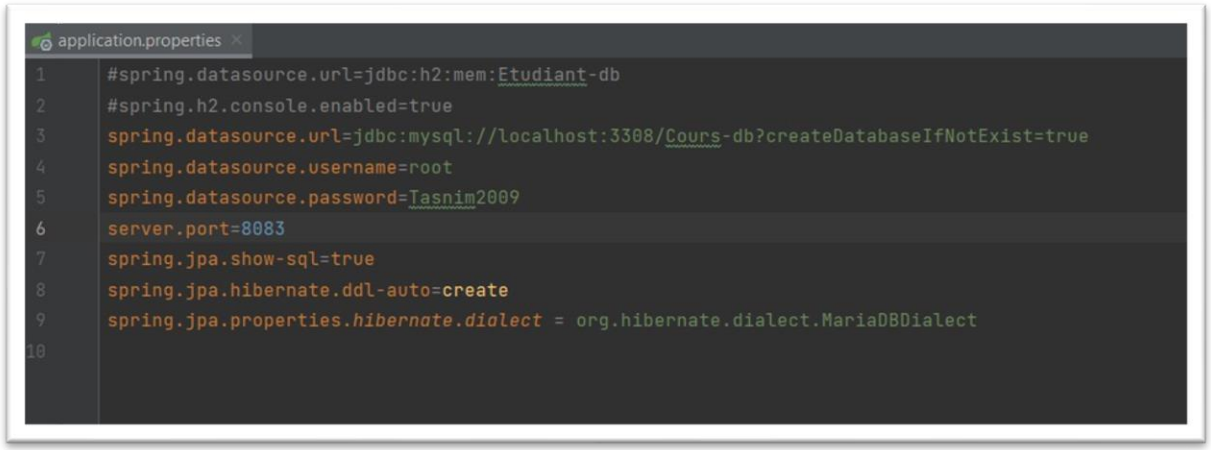
Compte Rendu :
Spring Data JPA

Réalisé par : BARAMI YASSINE




```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.29</version>
</dependency>
```


application.properties
#spring.datasource.url=jdbc:h2:mem:Etudiant-db
#spring.h2.console.enabled=true
spring.datasource.url=jdbc:mysql://localhost:3308/Cours-db?createDatabaseIfNotExist=true
spring.datasource.username=root
spring.datasource.password=Tasnim2009
server.port=8083
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=create
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MariaDBDialect

Cours.java

```java
package ma.emsi.tpspringdata.entities;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.util.Collection;

10 usages
@Data @AllArgsConstructor @NoArgsConstructor
@Entity
public class Cours {
    no usages
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    no usages
    @Column(length = 25, nullable = false, unique = false)
    private String title;
    no usages
    private String description;
    no usages
    private int timing;
    no usages
    @OneToOne
    private Professeur professeur;
    no usages
    @OneToMany(mappedBy = "cours", fetch = FetchType.EAGER)
    private Collection<Seance> seance;

```

Etudiant.java

```java
package ma.emsi.tpspringdata.entities;

import ...
7 usages
@Entity @Table(name = "EMSI_STUDENTS")
@Data @NoArgsConstructor @AllArgsConstructor @ToString
public class Etudiant {
    no usages
    @Id @GeneratedValue(strategy= GenerationType.IDENTITY)
    private Integer id;
    no usages
    @Column(name = "REGISTRATION_N",unique = true)
    private String registrationNumber;
    no usages
    @Column(name = "Name", length = 30, nullable = false)
    private String fullName;
    no usages
    @Temporal(TemporalType.DATE)
    private Date birthay;
    no usages
    private Boolean stillActive;
    no usages
    @Temporal(TemporalType.TIMESTAMP) @CreationTimestamp
    private Date lastConnection;
    no usages
    @ManyToMany(mappedBy = "etudiants", fetch = FetchType.EAGER)
 private Collection<Seance> seances=new ArrayList<>();
}
```

Professeur.java

```java
import lombok.Data;
import lombok.NoArgsConstructor;
import org.hibernate.annotations.CreationTimestamp;
import org.springframework.boot.autoconfigure.web.WebProperties;

import java.util.Date;

9 usages
@Data @AllArgsConstructor @NoArgsConstructor
@Entity
public class Professeur {
    no usages
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    no usages
    @Column(length = 30, nullable = false)
    private String fullName;
    no usages
    @Temporal(TemporalType.DATE) @CreationTimestamp
    private Date assignementDate;
    no usages
    @OneToOne(mappedBy = "professeur")
    private Cours cours;
}
```

Seance.java

```java
package ma.emsi.tpspringdata.entities;

import ...

8 usages
@Entity
@Data @AllArgsConstructor @NoArgsConstructor
public class Seance {
    no usages
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    no usages
    @Temporal(TemporalType.DATE)
    private Date date;
    no usages
    @Temporal(TemporalType.TIME)
    private Date start_time;
    no usages
    @Temporal(TemporalType.TIME)
    private Date end_time;
    no usages
    @ManyToOne
    private Cours cours;
    no usages
    @ManyToMany(fetch = FetchType.EAGER)
    @ToString.Exclude
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
        private Collection<Etudiant> etudiants = new ArrayList<>();
}
```

```java
package ma.emsi.tpspringdata.repositories;

import ma.emsi.tpspringdata.entities.Cours;
import org.springframework.data.jpa.repository.JpaRepository;

2 usages
public interface CoursRepository extends JpaRepository<Cours,Integer> {
}
```

EtudiantRepository.java

```java
package ma.emsi.tpspringdata.repositories;

import ma.emsi.tpspringdata.entities.Etudiant;
import org.springframework.data.jpa.repository.JpaRepository;

2 usages
public interface EtudiantRepository extends JpaRepository<Etudiant,Integer> {

}
```

ProfesseurRepository.java

```java
package ma.emsi.tpspringdata.repositories;


import ma.emsi.tpspringdata.entities.Professeur;
import org.springframework.data.jpa.repository.JpaRepository;

2 usages
public interface ProfesseurRepository extends JpaRepository<Professeur,Integer> {
}
```

SeanceRepository.java

```java
package ma.emsi.tpspringdata.repositories;

import ma.emsi.tpspringdata.entities.Seance;
import org.springframework.data.jpa.repository.JpaRepository;

2 usages
public interface SeanceRepository extends JpaRepository<Seance,Integer> {
}
```

TpSpringDataApplication.java

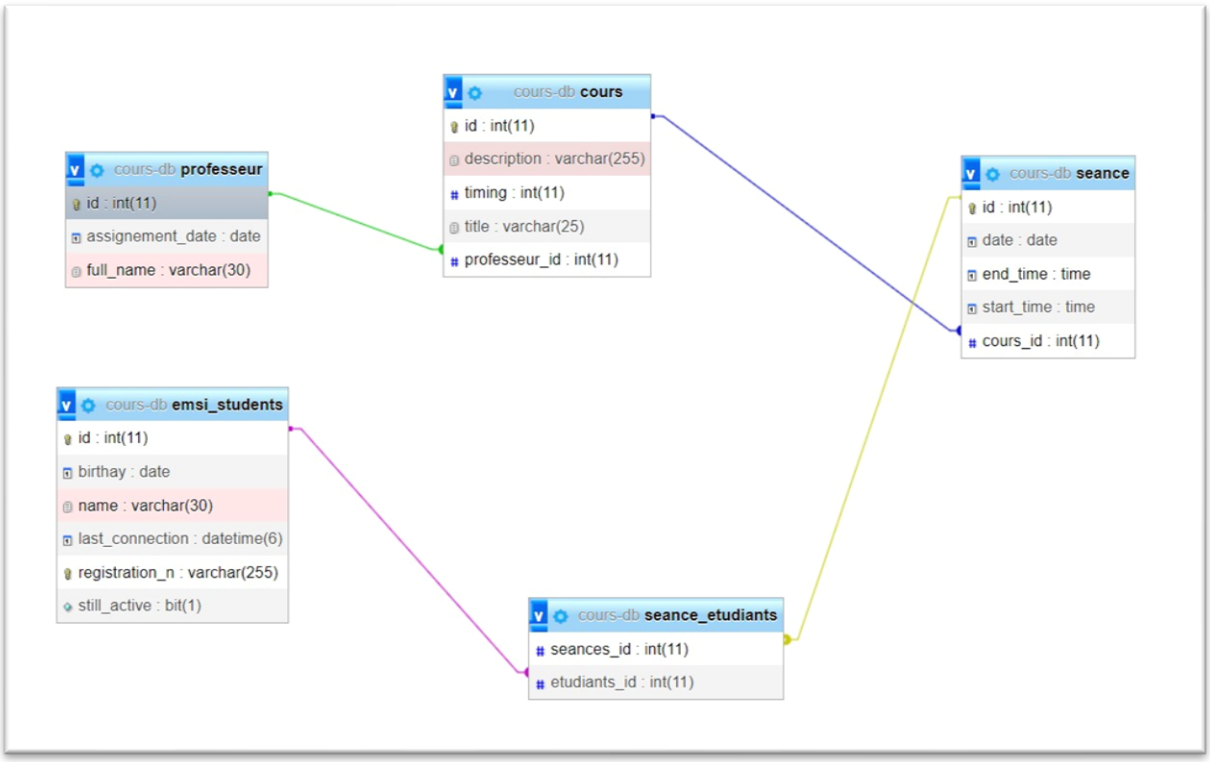
```java
package ma.emsi.tpspringdata;

import ...

1 usage
@SpringBootApplication
public class TpSpringDataApplication {

    no usages
    public static void main(String[] args) {

        SpringApplication.run(TpSpringDataApplication.class, args);
    }

    no usages
    @Bean
    CommandLineRunner start(CoursRepository coursRepository, ProfesseurRepository professeurRepository, SeanceRepository seanceRep
        return args -> {
            Stream.of( values: "Akram Lkihal", "Ahmed Zidany","Yasser Hmimou")
                    .forEach(fullName->{
            Etudiant etudiant= new Etudiant();
            etudiant.setRegistrationNumber(fullName );
            etudiant.setFullName(fullName);
            etudiant.setBirthay(new Date());
            etudiant.setLastConnection(new Date());
            etudiant.setStillActive(Math.random()>0.5?true:false);
            etudiant.setSeances(null);
            etudiantRepository.save(etudiant);
```

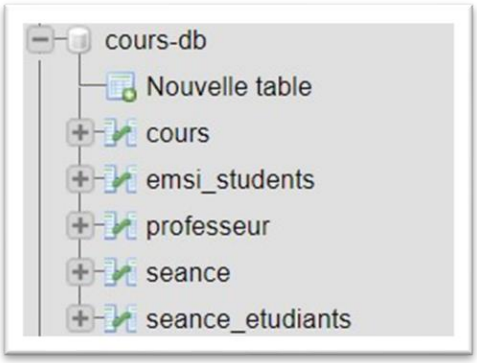
```
Stream.of("Yassine Barami", "Ilyass Mouatassim","Abdelmoughit Aitchihab")
        .forEach(fullName->{
            Professeur professeur= new Professeur();
            professeur.setFullName(fullName);
            professeur.setAssignementDate(new Date());
            professeur.setCours(null);
            professeurRepository.save(professeur);

        });

Stream.of("1", "2","3")
        .forEach(Seance->{
Seance seance= new Seance();
seance.setDate(new Date());
seance.setStart_time(new Date());
seance.setEnd_time(new Date());
seance.setCours(null);
seance.setEtudiants(null);
seanceRepository.save(seance);
        });
```

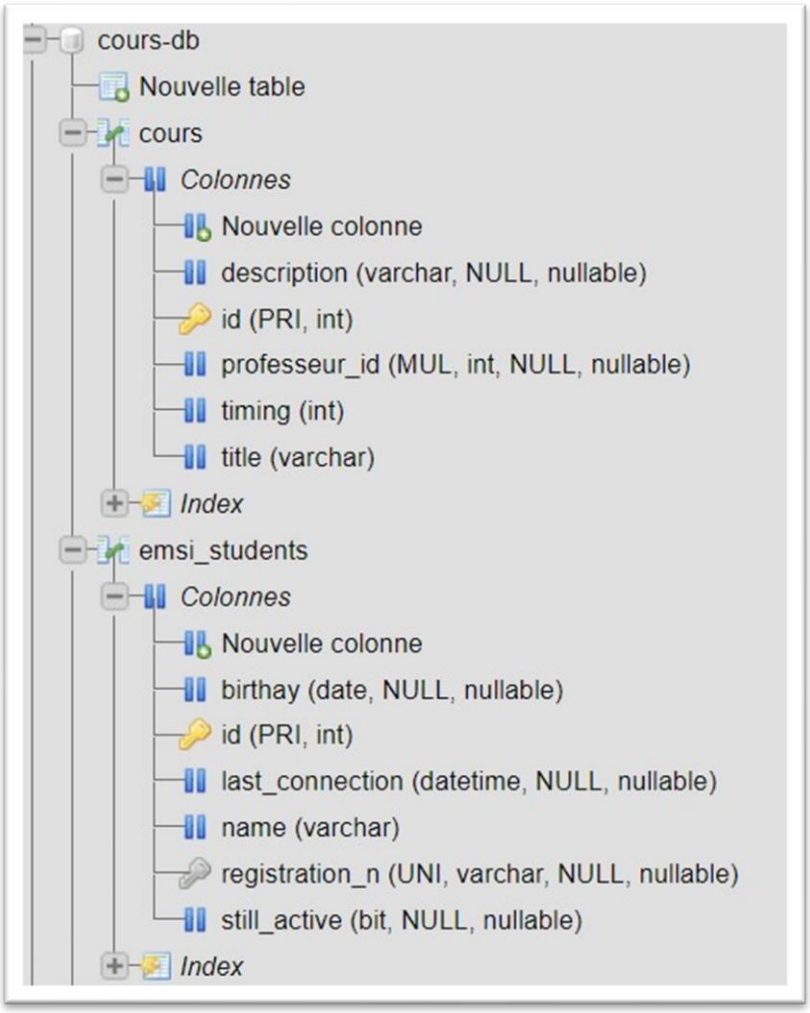
```
Stream.of( ...values: "math", "physique","informatique")
        .forEach(cour->{
            Cours cours = new Cours();
            cours.setTitle(cour);
            cours.setDescription(Math.random()>0.5?"intéréssant":"long");
            cours.setTiming(5);
            cours.setProfesseur(null);
            cours.setSeance(null);
            coursRepository.save(cours);
        });

Cours cours1= coursRepository.findById(1).orElse( other: null);
Professeur professeur1= professeurRepository.findById(1).orElse( other: null);
cours1.setProfesseur(professeur1);
coursRepository.save(cours1);

Cours cours2= coursRepository.findById(2).orElse( other: null);
Professeur professeur3= professeurRepository.findById(3).orElse( other: null);
cours2.setProfesseur(professeur3);
coursRepository.save(cours2);

Cours cours3= coursRepository.findById(3).orElse( other: null);
Professeur professeur2= professeurRepository.findById(2).orElse( other: null);
cours3.setProfesseur(professeur2);
coursRepository.save(cours3);
```

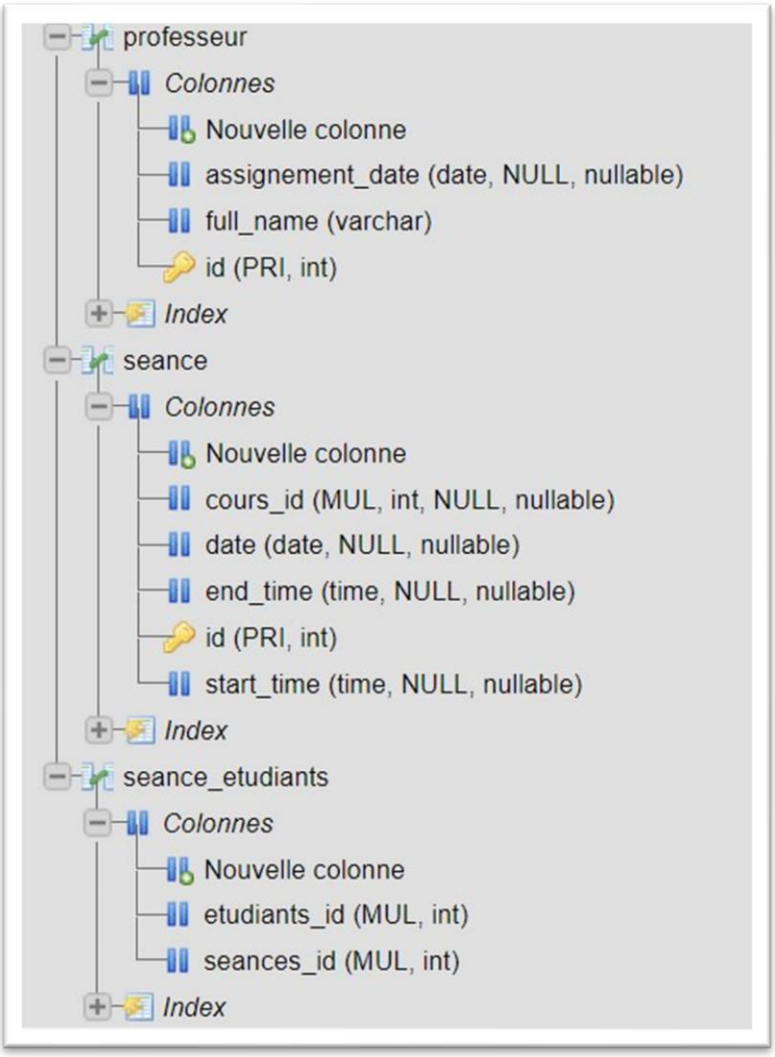
```
Seance seance1= seanceRepository.findById(1).orElse( other: null);
seance1.setCours(cours1);
seanceRepository.save(seance1);

Etudiant etudiant1= etudiantRepository.findById(1).orElse( other: null);
if(etudiant1.getSeances()!=null) {
    etudiant1.getSeances().add(seance1);
    seance1.getEtudiants().add(etudiant1);
    etudiantRepository.save(etudiant1);
    seanceRepository.save(seance1);
}

etudiant1.getSeances().forEach(s->{
    System.out.println("SEANCE=>"+s.toString());
});
```

```
Hibernate: alter table cours drop foreign key FK3vhwktsmhvc7w2qjsdbdv65b
Hibernate: alter table seance drop foreign key FKrrc1k4hpxsmh2hau4gl5wjgod
Hibernate: alter table seance_etudiants drop foreign key FKl9vlugsleaoui4y809lkcde3t
Hibernate: alter table seance_etudiants drop foreign key FKgs3nftvelfvhhrx4grl040bbi
Hibernate: drop table if exists cours
Hibernate: drop table if exists emsi_students
Hibernate: drop table if exists professeur
Hibernate: drop table if exists seance
Hibernate: drop table if exists seance_etudiants
Hibernate: create table cours (id integer not null auto_increment, description varchar(255), timing integer not null, title varchar(25) not null, professeur_id integer, primary
Hibernate: create table emsi_students (id integer not null auto_increment, birthay date, name varchar(30) not null, last_connection datetime(6), registration_n varchar(255), sti
Hibernate: create table professeur (id integer not null auto_increment, assignement_date date, full_name varchar(30) not null, primary key (id)) engine=InnoDB
Hibernate: create table seance (id integer not null auto_increment, date date, end_time time, start_time time, cours_id integer, primary key (id)) engine=InnoDB
Hibernate: create table seance_etudiants (seances_id integer not null, etudiants_id integer not null) engine=InnoDB
Hibernate: alter table emsi_students add constraint UK_a3vjtylbahy45i3mbr04y7hsu unique (registration_n)
```




cours-db
Nouvelle table
cours
emsi_students
professeur
seance
seance_etudiants


cours-db
Nouvelle table
cours
Colonnes
Nouvelle colonne
description (varchar, NULL, nullable)
id (PRI, int)
professeur_id (MUL, int, NULL, nullable)
timing (int)
title (varchar)
Index
emsi_students
Colonnes
Nouvelle colonne
birthay (date, NULL, nullable)
id (PRI, int)
last_connection (datetime, NULL, nullable)
name (varchar)
registration_n (UNI, varchar, NULL, nullable)
still_active (bit, NULL, nullable)
Index



		id	description	timing	title	professeur_id
☐	Éditer Copier Supprimer	1	intéréssant	5	math	1
☐	Éditer Copier Supprimer	2	intéréssant	5	physique	3
☐	Éditer Copier Supprimer	3	long	5	informatique	2

	id	birthay	name	last_connection	registration_n	still_active
Éditer Copier Supprimer	1	2023-04-16	Akram Lkihal	2023-04-16 12:36:41.000000	Akram Lkihal	1
Éditer Copier Supprimer	2	2023-04-16	Ahmed Zidany	2023-04-16 12:36:41.000000	Ahmed Zidany	0
Éditer Copier Supprimer	3	2023-04-16	Yasser Hmimou	2023-04-16 12:36:41.000000	Yasser Hmimou	0

	id	assignement_date	full_name
Éditer Copier Supprimer	1	2023-04-16	Yassine Barami
Éditer Copier Supprimer	2	2023-04-16	Ilyass Mouatassim
Éditer Copier Supprimer	3	2023-04-16	Abdelmoughit Aitchihab

	id	date	end_time	start_time	cours_id
Éditer Copier Supprimer	1	2023-04-16	12:36:41	12:36:41	1
Éditer Copier Supprimer	2	2023-04-16	12:36:41	12:36:41	*NULL*
Éditer Copier Supprimer	3	2023-04-16	12:36:41	12:36:41	*NULL*

seances_id	etudiants_id
1	1

```
SEANCE=>Seance(id=1, date=2023-04-16, start_time=12:36:41, end_time=12:36:41, cours=Cours(id=1, title=math, description=intéréssant, timing=5, professeur=Professeur(id=1, fullN
```