**TP 4 : JPA**

1. Les entités qu’on a utilisées :

* La classe Patient :

package com.example.TP4\_JPA.entities;  
  
  
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
import javax.persistence.\*;  
import java.util.Collection;  
import java.util.Date;  
  
@Entity  
@Data @NoArgsConstructor @AllArgsConstructor  
public class Patient {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long id;  
 private String nom;  
 @Temporal(TemporalType.*DATE*)  
 private Date datenaissance;  
 private boolean malade;  
 @OneToMany(mappedBy = "patient", fetch = FetchType.*LAZY*)  
 private Collection<Rendez\_vous> rendezVous;  
}

* la classe Médecin :

package com.example.TP4\_JPA.entities;  
  
import com.fasterxml.jackson.annotation.JsonProperty;  
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
import javax.persistence.\*;  
import java.util.Collection;  
  
@Entity  
@Data  
@NoArgsConstructor  
@AllArgsConstructor  
public class Medecin {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long id;  
 private String nom;  
 private String email;  
 private String specialite;  
 @OneToMany(mappedBy = "medecin", fetch = FetchType.*LAZY*)  
 @JsonProperty(access = JsonProperty.Access.*WRITE\_ONLY*)  
 private Collection<Rendez\_vous> rendezVous;  
  
}

* la classe Rendez-Vous:

package com.example.TP4\_JPA.entities;  
  
import com.fasterxml.jackson.annotation.JsonProperty;  
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
import javax.persistence.\*;  
import java.util.Date;  
  
@Entity  
@Data  
@NoArgsConstructor  
@AllArgsConstructor  
public class Rendez\_vous {  
 @Id  
 /\* @GeneratedValue(strategy = GenerationType.IDENTITY)  
 private Long id;\*/  
 private String id;  
 private Date date;  
 @Enumerated(EnumType.*STRING*)  
 private StatusRDV status;  
 @ManyToOne  
 @JsonProperty(access = JsonProperty.Access.*WRITE\_ONLY*)  
 private Patient patient;  
 @ManyToOne  
 private Medecin medecin;  
 @OneToOne(mappedBy = "rendezVous")  
 private Consultation consultation;  
  
}

* la classe Consultation:

package com.example.TP4\_JPA.entities;  
  
import com.fasterxml.jackson.annotation.JsonProperty;  
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
import javax.persistence.\*;  
import java.util.Date;  
  
@Entity  
@Data  
@NoArgsConstructor  
@AllArgsConstructor  
public class Consultation {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long id;  
 private Date dateConsultation;  
 private String rapport;  
 @OneToOne  
 @JsonProperty(access = JsonProperty.Access.*WRITE\_ONLY*)  
 private Rendez\_vous rendezVous;  
}

* la classe StatusRDV:

package com.example.TP4\_JPA.entities;  
  
public enum StatusRDV {  
 *PENDING*,  
 *CANCELED*,  
 *DONE*}

1. Les repositories:

* Pour Patient:

package com.example.TP4\_JPA.repositories;  
  
import com.example.TP4\_JPA.entities.Patient;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface PatienRepository extends JpaRepository<Patient,Long> {  
 Patient findByNom(String name); // on suppose que c'est un nom unique  
}

* Pour Medecin:

package com.example.TP4\_JPA.repositories;  
  
import com.example.TP4\_JPA.entities.Medecin;  
import com.example.TP4\_JPA.entities.Patient;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface MedecinRepository extends JpaRepository<Medecin, Long> {  
 Medecin findByNom(String name); // on suppose que c'est un nom unique  
}

* Pour Rendez-Vous:

package com.example.TP4\_JPA.repositories;  
  
import com.example.TP4\_JPA.entities.Rendez\_vous;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface RendezVousRepository extends JpaRepository<Rendez\_vous, Long> {  
}

* Pour Consultation:

package com.example.TP4\_JPA.repositories;  
  
import com.example.TP4\_JPA.entities.Consultation;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface ConsultationRepository extends JpaRepository<Consultation,Long> {  
}

1. Les services:

* Interface:

package com.example.TP4\_JPA.service;  
  
import com.example.TP4\_JPA.entities.Consultation;  
import com.example.TP4\_JPA.entities.Medecin;  
import com.example.TP4\_JPA.entities.Patient;  
import com.example.TP4\_JPA.entities.Rendez\_vous;  
  
public interface IHospitalService {  
 Patient savePatient(Patient patient);  
 Medecin saveMedecin(Medecin medecin);  
 Rendez\_vous saveRDV(Rendez\_vous rendezVous);  
 Consultation saveConsultation(Consultation consultation);  
}

* Classe :

package com.example.TP4\_JPA.service;  
  
import com.example.TP4\_JPA.entities.Consultation;  
import com.example.TP4\_JPA.entities.Medecin;  
import com.example.TP4\_JPA.entities.Patient;  
import com.example.TP4\_JPA.entities.Rendez\_vous;  
import com.example.TP4\_JPA.repositories.ConsultationRepository;  
import com.example.TP4\_JPA.repositories.MedecinRepository;  
import com.example.TP4\_JPA.repositories.PatienRepository;  
import com.example.TP4\_JPA.repositories.RendezVousRepository;  
import lombok.AllArgsConstructor;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import javax.transaction.Transactional;  
import java.util.UUID;  
  
@Service  
@Transactional @AllArgsConstructor  
public class HospitalServiceImpl implements IHospitalService{  
 private PatienRepository patientRepository;  
 private MedecinRepository medecinRepository;  
 private RendezVousRepository rendezVousRepository;  
 private ConsultationRepository consultationRepository;  
  
 @Override  
 public Patient savePatient(Patient patient) {  
 return patientRepository.save(patient);  
 }  
  
 @Override  
 public Medecin saveMedecin(Medecin medecin) {  
 return medecinRepository.save(medecin);  
 }  
  
 @Override  
 public Rendez\_vous saveRDV(Rendez\_vous rendezVous) {  
 rendezVous.setId(UUID.*randomUUID*().toString());  
 return rendezVousRepository.save(rendezVous);  
 }  
  
 @Override  
 public Consultation saveConsultation(Consultation consultation) {  
 return consultationRepository.save(consultation);  
 }  
}

1. Service Web :

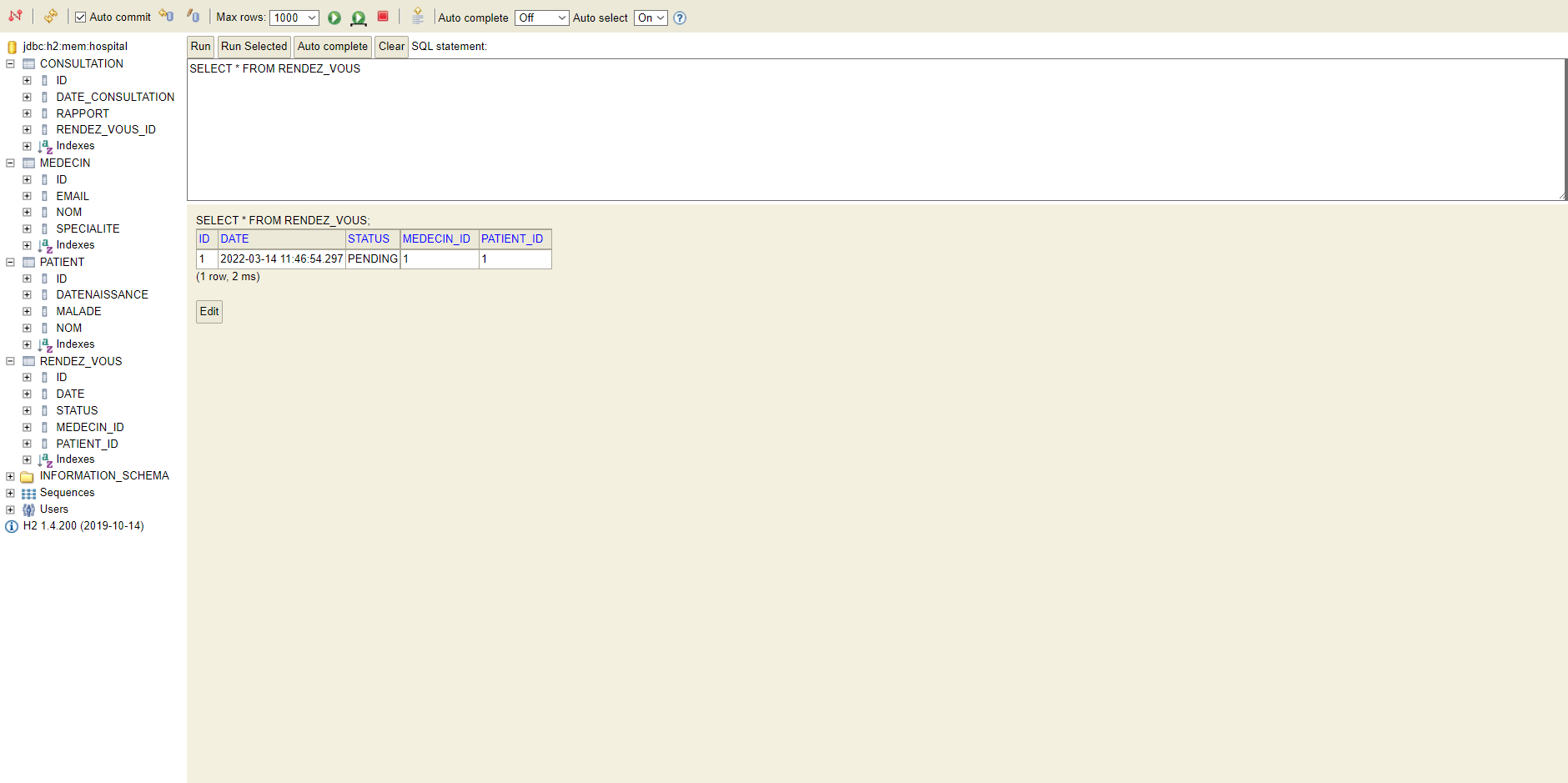
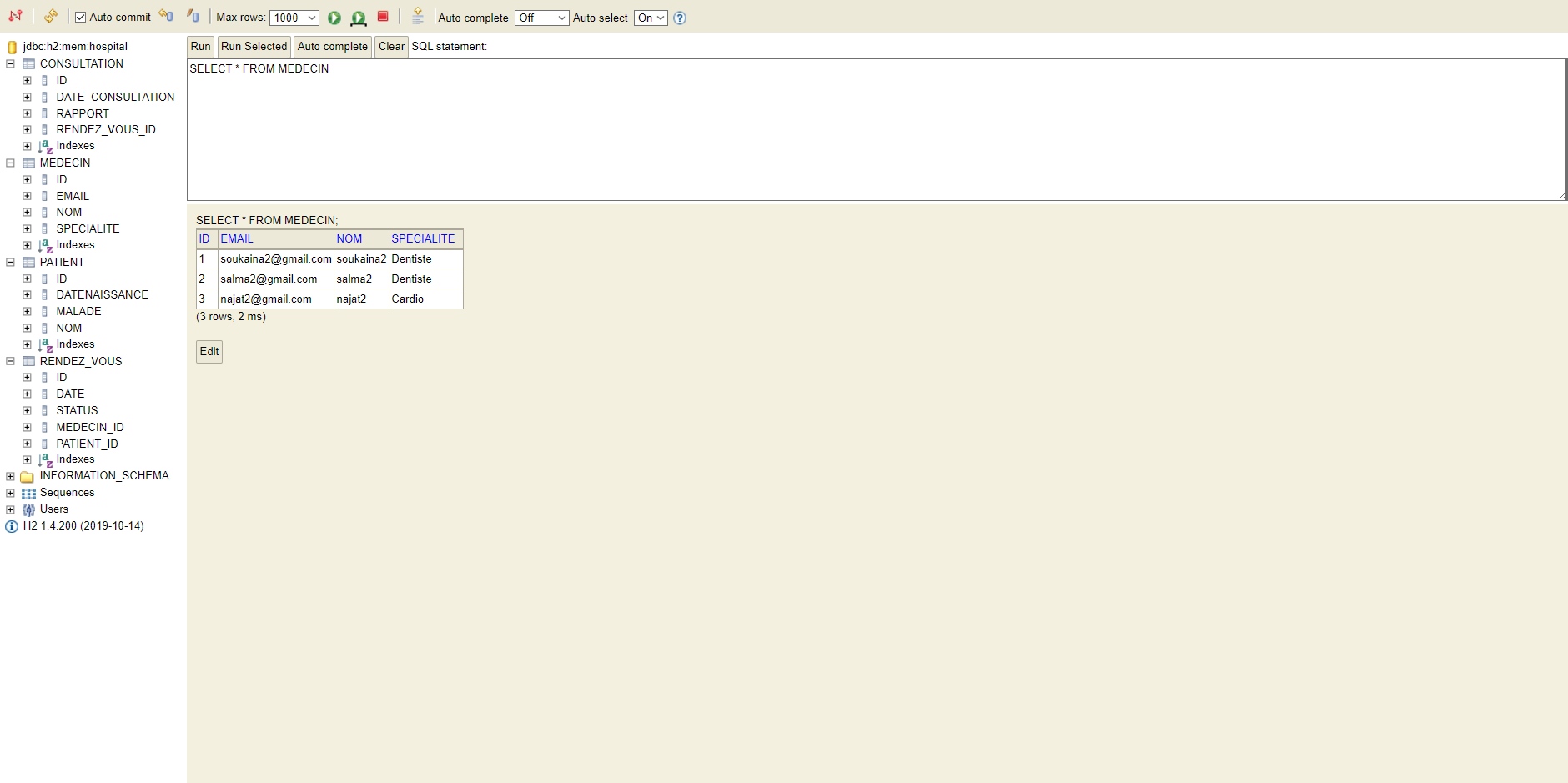
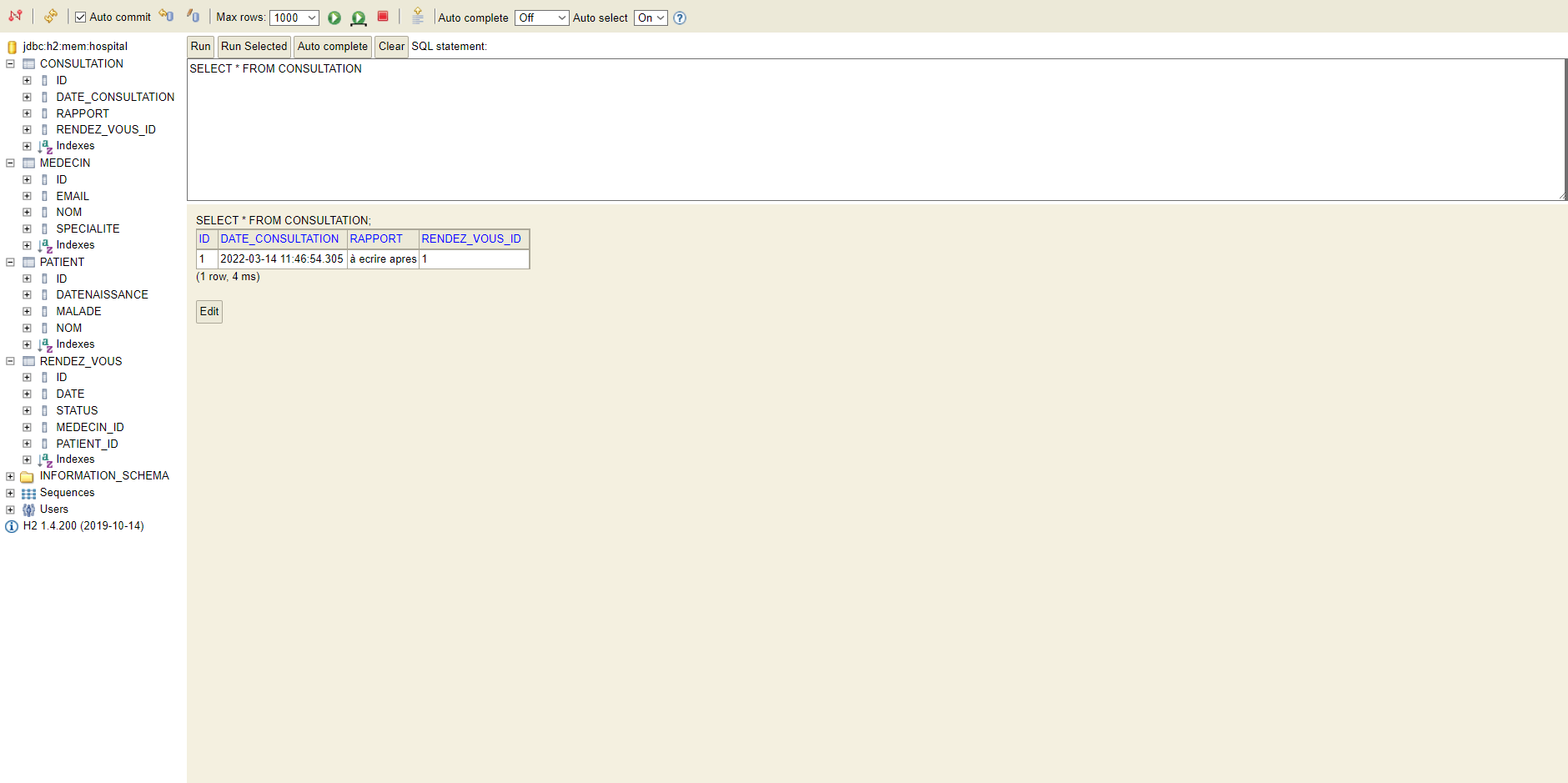
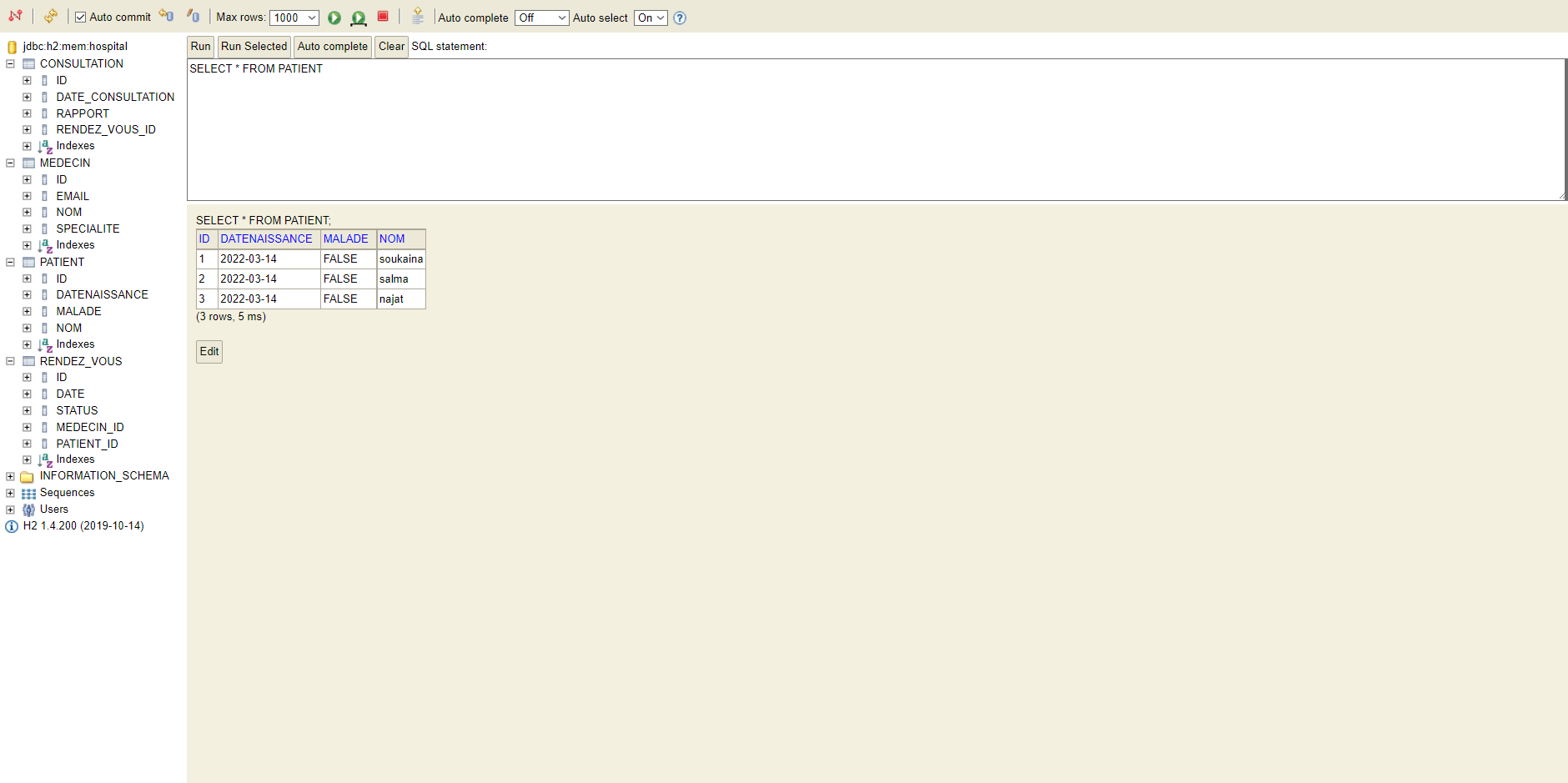
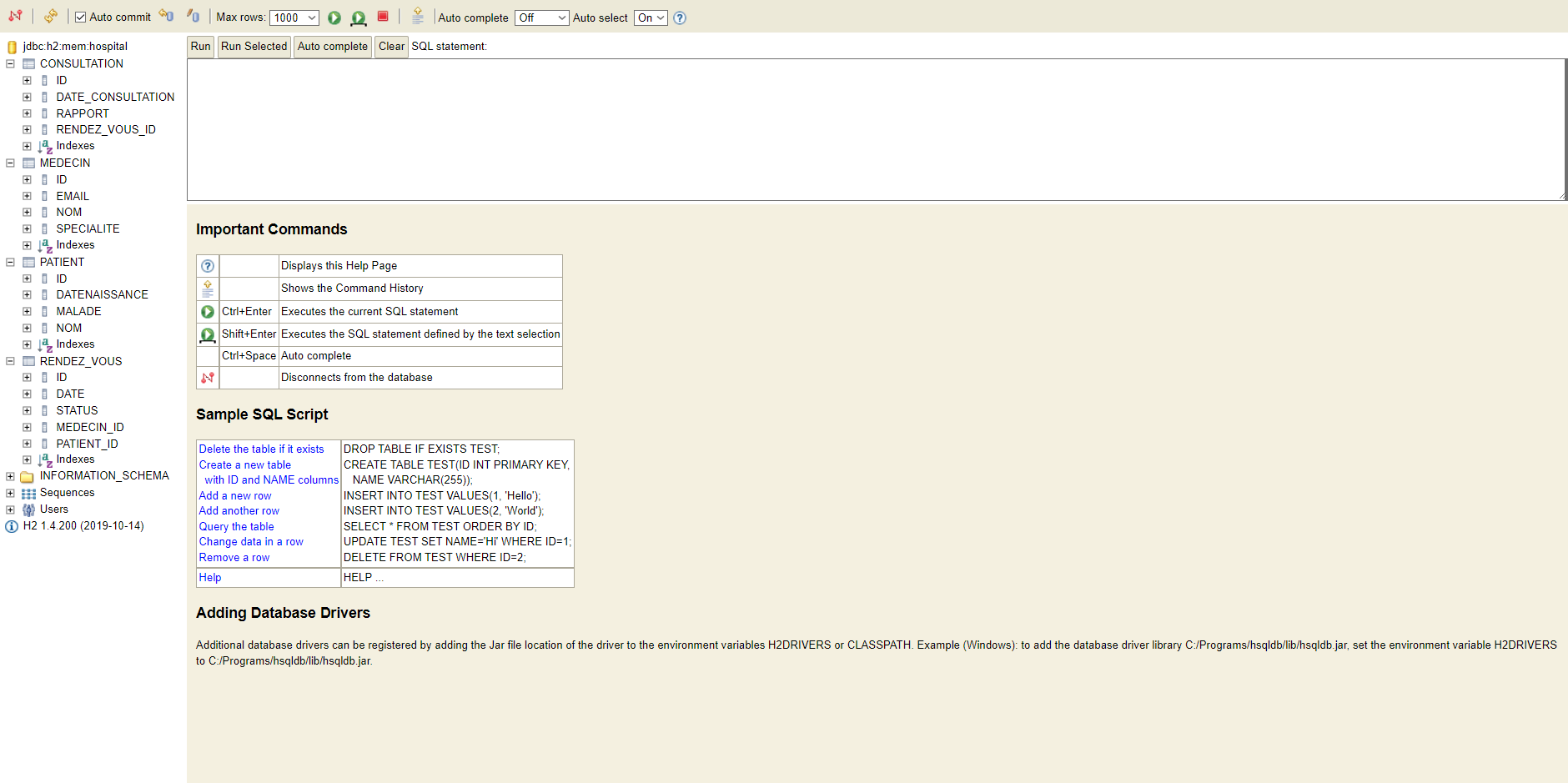
package com.example.TP4\_JPA.web;  
  
import com.example.TP4\_JPA.entities.Patient;  
import com.example.TP4\_JPA.repositories.PatienRepository;  
import com.sun.xml.bind.annotation.XmlIsSet;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
import java.util.List;  
  
@RestController  
public class PatientRestController {  
 @Autowired  
 private PatienRepository patienRepository;  
 @GetMapping("/patient")  
 public List<Patient> patientsList(){  
 return patienRepository.findAll();  
 }  
}

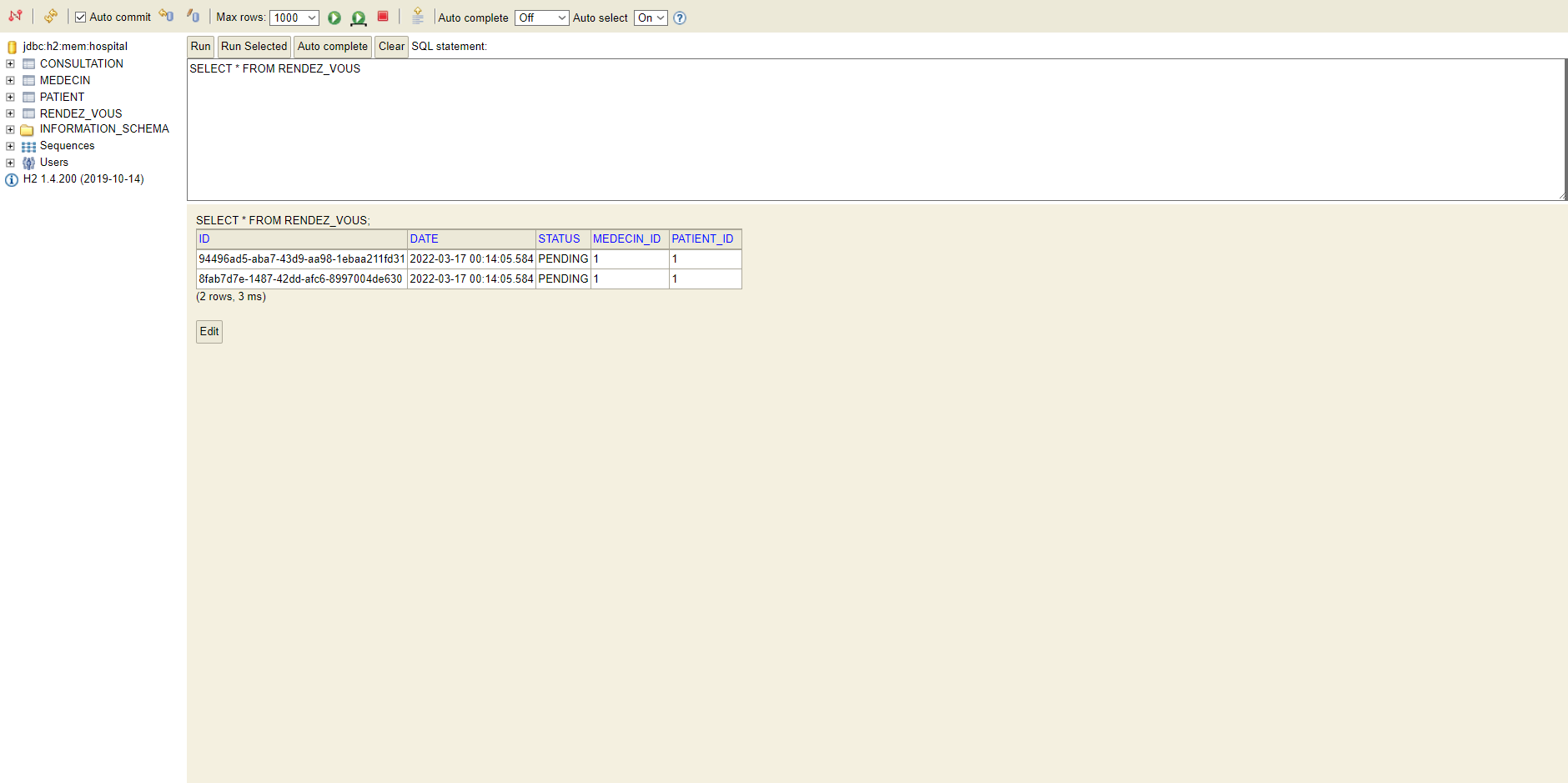
1. Affichage :

* Le Main :

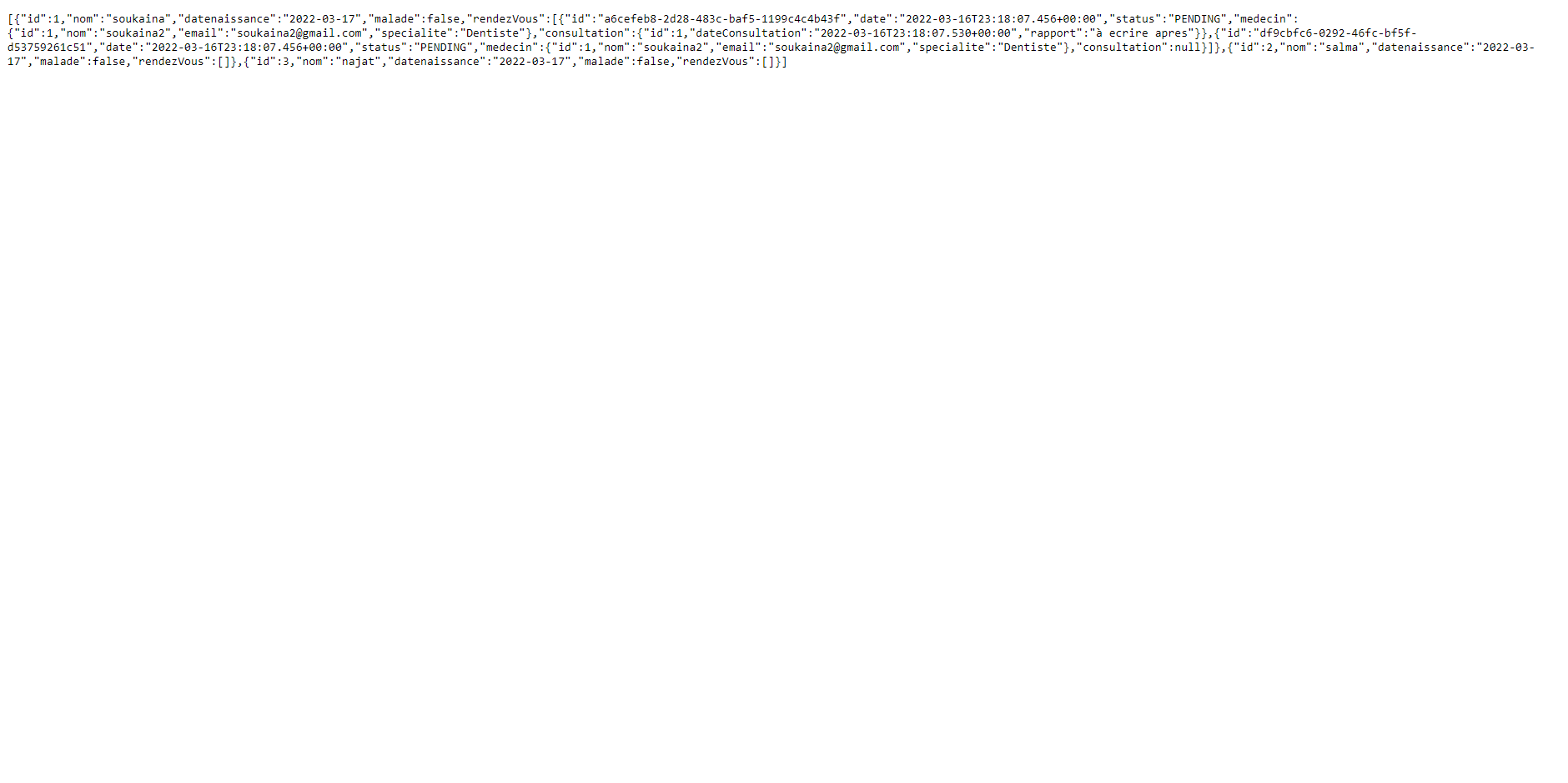
package com.example.TP4\_JPA;  
  
import com.example.TP4\_JPA.entities.\*;  
import com.example.TP4\_JPA.repositories.ConsultationRepository;  
import com.example.TP4\_JPA.repositories.MedecinRepository;  
import com.example.TP4\_JPA.repositories.PatienRepository;  
import com.example.TP4\_JPA.repositories.RendezVousRepository;  
import com.example.TP4\_JPA.service.IHospitalService;  
import org.springframework.boot.CommandLineRunner;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.annotation.Bean;  
  
import java.util.Date;  
import java.util.stream.Stream;  
  
@SpringBootApplication  
public class Tp4JpaApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(Tp4JpaApplication.class, args);  
 }  
  
 @Bean // pour démarrer cette méthode au démarage  
 CommandLineRunner start(IHospitalService hospitalService, PatienRepository patienRepository, MedecinRepository medecinRepository, RendezVousRepository rendezVousRepository){  
 return args ->{  
 Stream.*of*("soukaina","salma","najat").forEach(name-> {  
 hospitalService.savePatient(new Patient(null, name, new Date(), false, null));  
 });  
 /\*  
 Stream.of("soukaina","salma","najat").forEach(name-> {  
 Patient patient=new Patient();  
 patient.setNom(name);  
 patient.setDateNaissance(new Date());  
 patient.setMalade(false);  
 patientRepository.save(patient);  
 });  
 \*/  
  
 Stream.*of*("soukaina2","salma2","najat2").forEach(name-> {  
 Medecin medecin=new Medecin();  
 medecin.setNom(name);  
 medecin.setEmail(name+"@gmail.com");  
 medecin.setSpecialite(Math.*random*()>0.5?"Cardio":"Dentiste");  
 hospitalService.saveMedecin(medecin);  
 });  
  
 Patient patient=patienRepository.findById(1L).orElse(null);  
 Patient patient1=patienRepository.findByNom("soukaina");  
  
 Medecin medecin = medecinRepository.findByNom("soukaina2");  
  
 Rendez\_vous rendezVous = new Rendez\_vous();  
 rendezVous.setDate(new Date());  
 rendezVous.setStatus(StatusRDV.*PENDING*);  
 rendezVous.setMedecin(medecin);  
 rendezVous.setPatient(patient);  
 hospitalService.saveRDV(rendezVous);  
 Rendez\_vous savedRDV = hospitalService.saveRDV(rendezVous);  
 System.*out*.println(savedRDV.getId());  
  
 //Rendez\_vous rendezVous2 = rendezVousRepository.findById(1L).orElse(null);  
 Rendez\_vous rendezVous2 = rendezVousRepository.findAll().get(0);  
 Consultation consultation = new Consultation();  
 consultation.setDateConsultation(new Date());  
 consultation.setRendezVous(rendezVous2);  
 consultation.setRapport("à ecrire apres");  
 hospitalService.saveConsultation(consultation);  
 };  
 }  
  
}

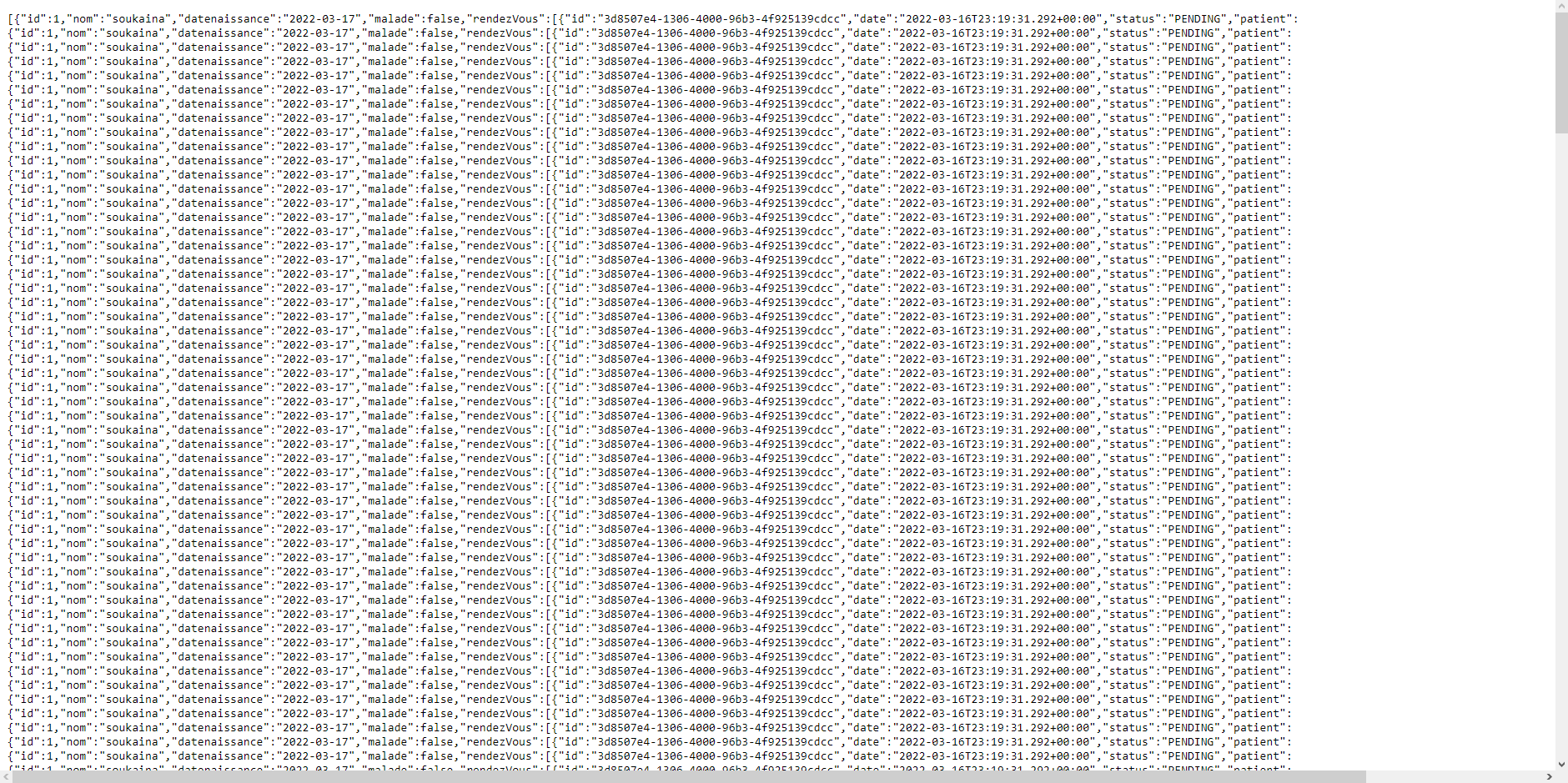
* Exécution :





* Affichage en Web :





My GitHub Repository is : https://github.com/SoukainaElkm/Soukaina-ELKAMOUNI.JEE