Ecole Normale Supérieure de l'Enseignement Technique Mohammedia

Université Hassan II de Casablanca

ENSET

ا لمدرسة العليا لأساتذة التعليم التقني المحمدية
جامعة الحسن الثاني بالدار البيضاء

DÉPARTEMENT MATHÉMATIQUES ET INFORMATIQUE

Compte Rendu

Filière :

« Génie du Logiciel et des Systèmes Informatiques Distribués »

GLSID

Activité Pratique N°2 - JPA, Hibernate et Spring Data

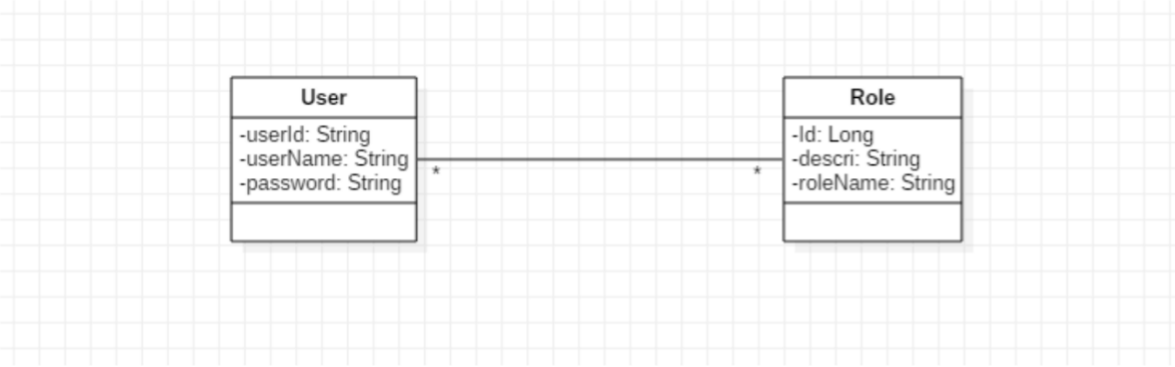
le 20/3/2022

Réalisé par :
Hassan ELMAKHLOUFI

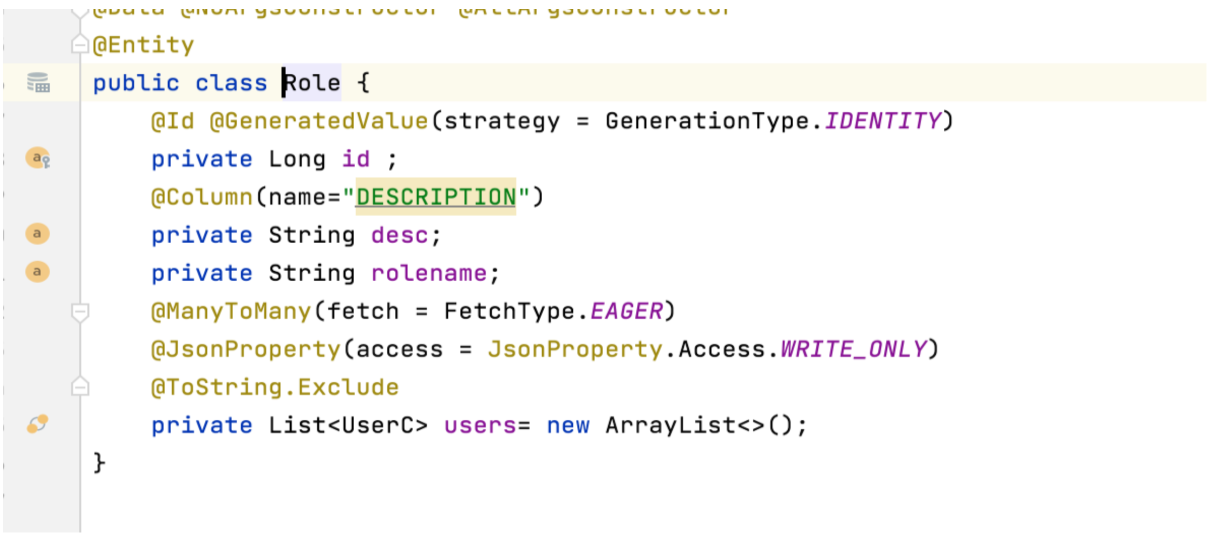
Encadré par :
M. Mohamed YOUSFI

Année Universitaire : 2021-2022

Dans ce Tp je vais implémenter ce diagramme de classe qui contient deux classe associer par une association many to many en utilisant Spring boot (spring data) :

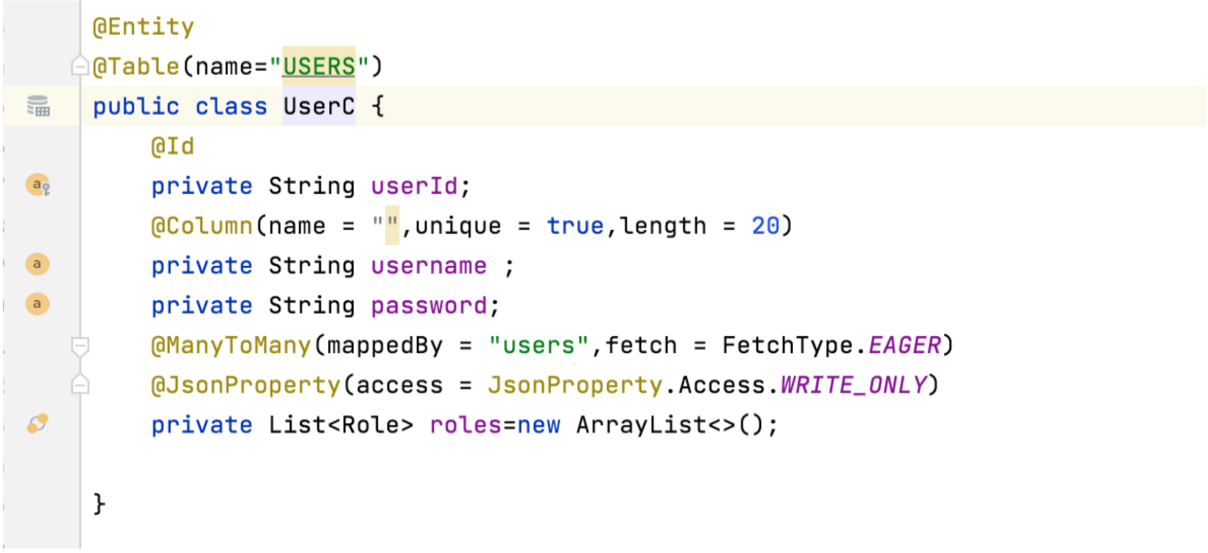


Entitie Role :



```
@Data @NoArgsConstructor @AllArgsConstructor
@Entity
public class Role {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id ;
    @Column(name="DESCRIPTION")
    private String desc;
    private String rolename;
    @ManyToMany(fetch = FetchType.EAGER)
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    @ToString.Exclude
    private List<UserC> users= new ArrayList<>();
}
```

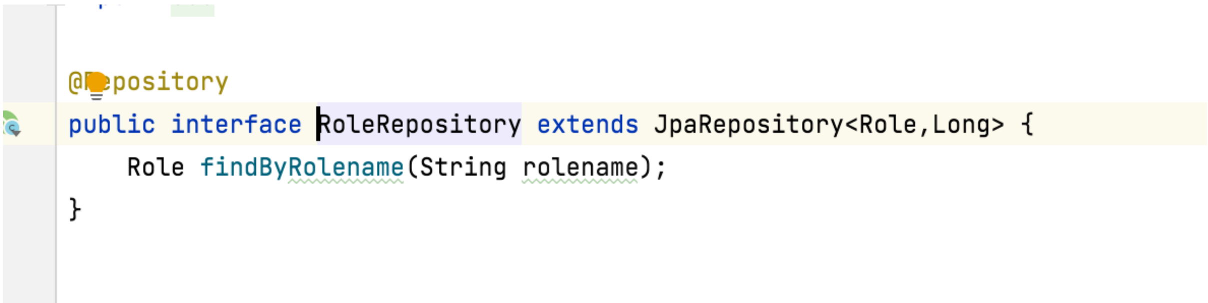
Entitie User :



```java
@Entity
@Table(name="USERS")
public class UserC {
    @Id
    private String userId;
    @Column(name = "",unique = true,length = 20)
    private String username ;
    private String password;
    @ManyToMany(mappedBy = "users",fetch = FetchType.EAGER)
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    private List<Role> roles=new ArrayList<>();

}
```

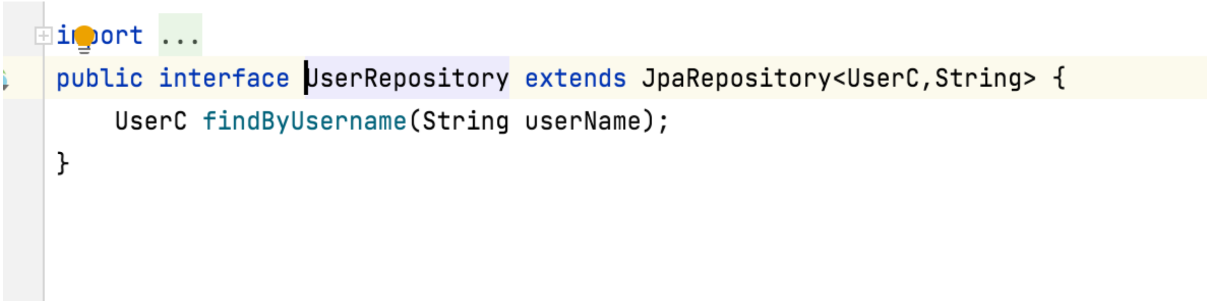
Repositories:

Repository Role:



```java
@Repository
public interface RoleRepository extends JpaRepository<Role,Long> {
    Role findByRolename(String rolename);
}
```

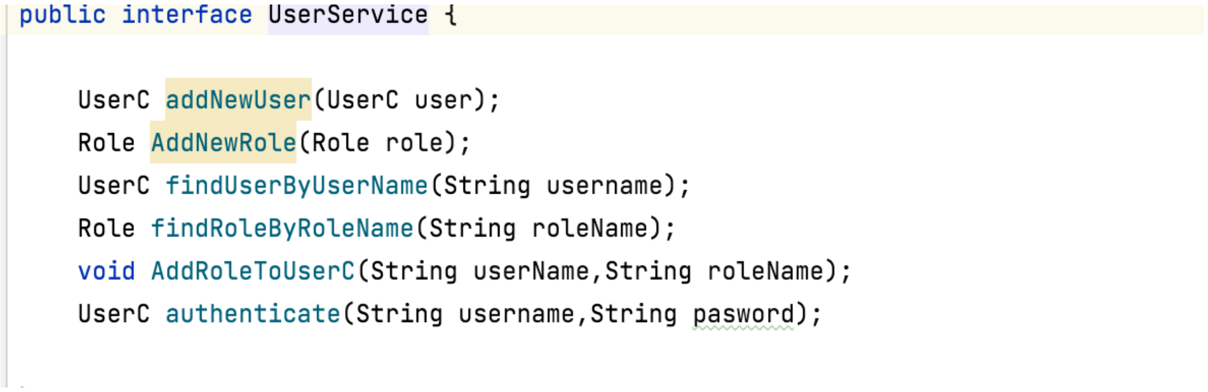
Repository User:



```java
import ...
public interface UserRepository extends JpaRepository<UserC,String> {
    UserC findByUsername(String userName);
}
```

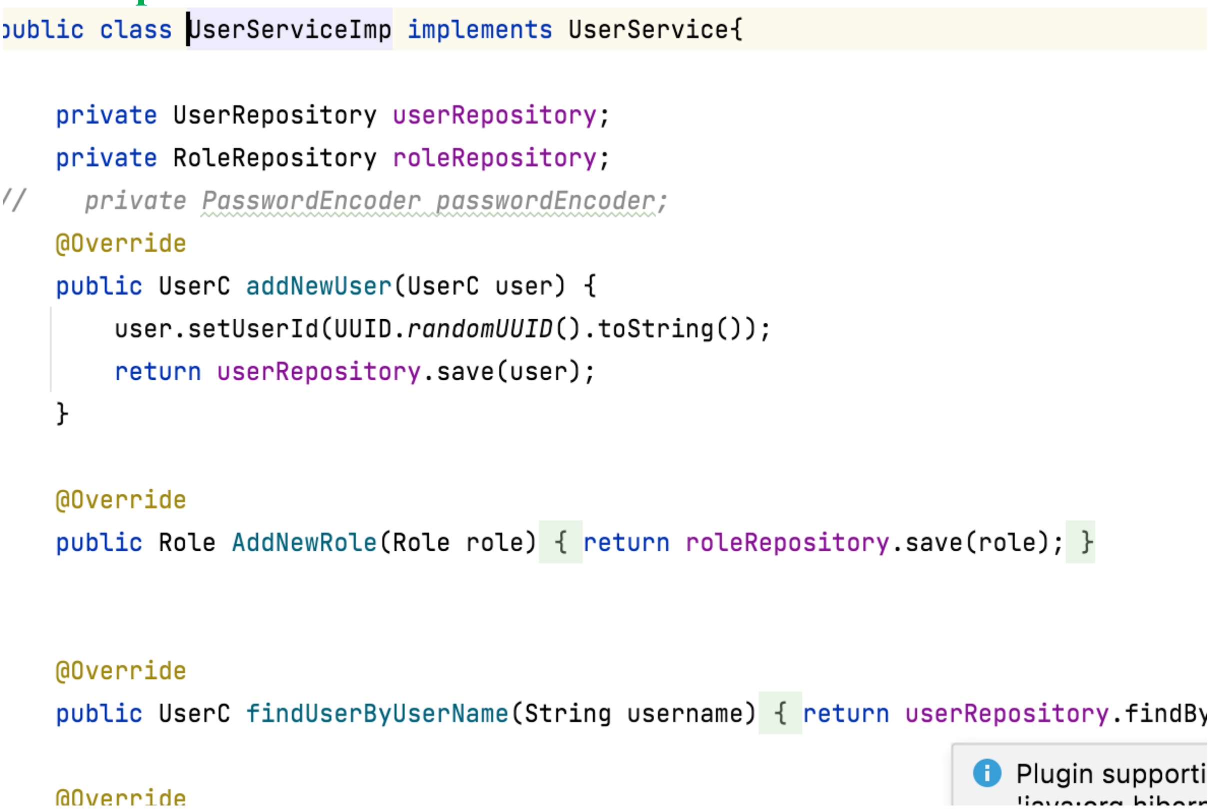
Métier:

Interface métier:



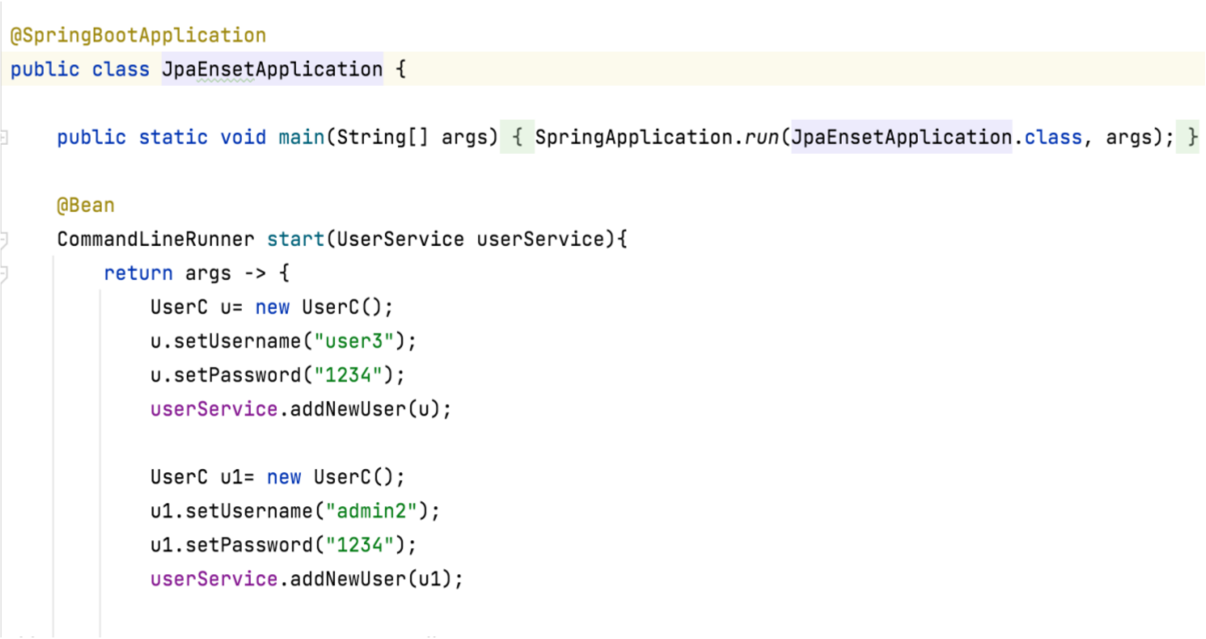
```
public interface UserService {

    UserC addNewUser(UserC user);
    Role AddNewRole(Role role);
    UserC findUserByUserName(String username);
    Role findRoleByRoleName(String roleName);
    void AddRoleToUserC(String userName,String roleName);
    UserC authenticate(String username,String pasword);
```

Une implementation de l’interface Métier :



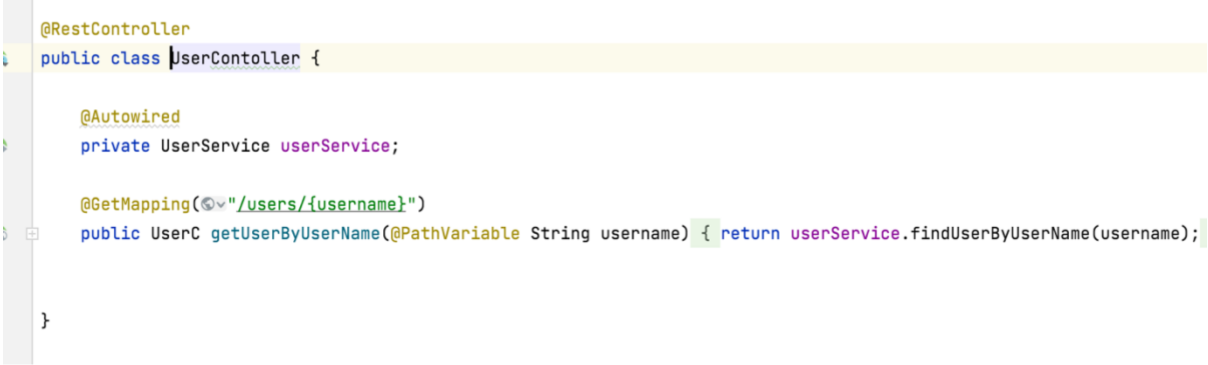
```
public class UserServiceImp implements UserService{

    private UserRepository userRepository;
    private RoleRepository roleRepository;
//    private PasswordEncoder passwordEncoder;
    @Override
    public UserC addNewUser(UserC user) {
        user.setUserId(UUID.randomUUID().toString());
        return userRepository.save(user);
    }

    @Override
    public Role AddNewRole(Role role) { return roleRepository.save(role); }


    @Override
    public UserC findUserByUserName(String username) { return userRepository.findBy

    @Override
```

Classe Application:



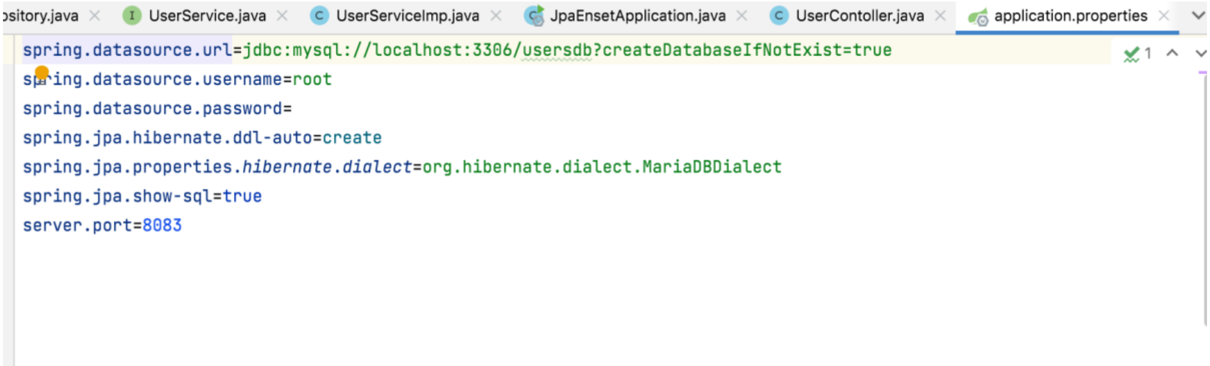
```java
@SpringBootApplication
public class JpaEnsetApplication {

    public static void main(String[] args) { SpringApplication.run(JpaEnsetApplication.class, args); }

    @Bean
    CommandLineRunner start(UserService userService){
        return args -> {
            UserC u= new UserC();
            u.setUsername("user3");
            u.setPassword("1234");
            userService.addNewUser(u);

            UserC u1= new UserC();
            u1.setUsername("admin2");
            u1.setPassword("1234");
            userService.addNewUser(u1);
```

Rest controller:



```java
@RestController
public class UserContoller {

    @Autowired
    private UserService userService;

    @GetMapping("/users/{username}")
    public UserC getUserByUserName(@PathVariable String username) { return userService.findUserByUserName(username);


}
```

Fichier de Application.properties:



```properties
spring.datasource.url=jdbc:mysql://localhost:3306/usersdb?createDatabaseIfNotExist=true
spring.datasource.username=root
spring.datasource.password=
spring.jpa.hibernate.ddl-auto=create
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MariaDBDialect
spring.jpa.show-sql=true
server.port=8083
```

Base de donnnees mysql:

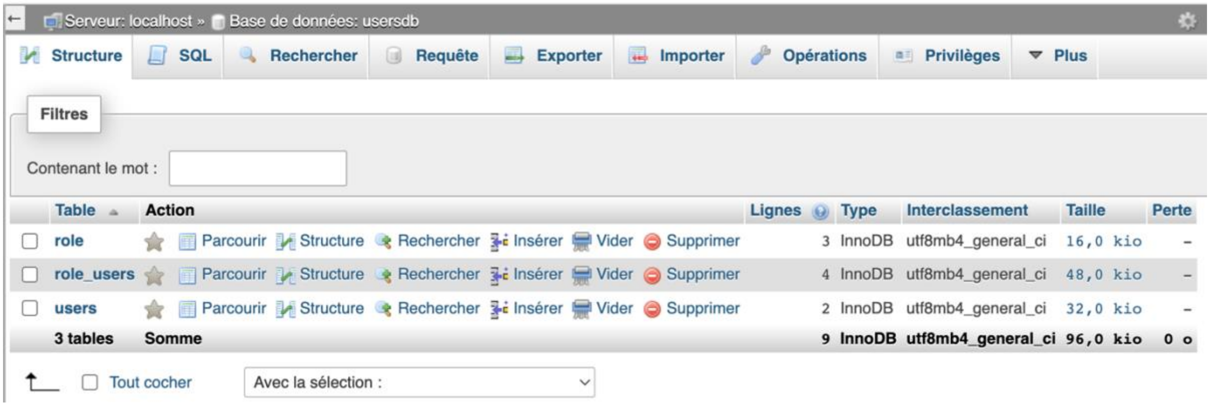
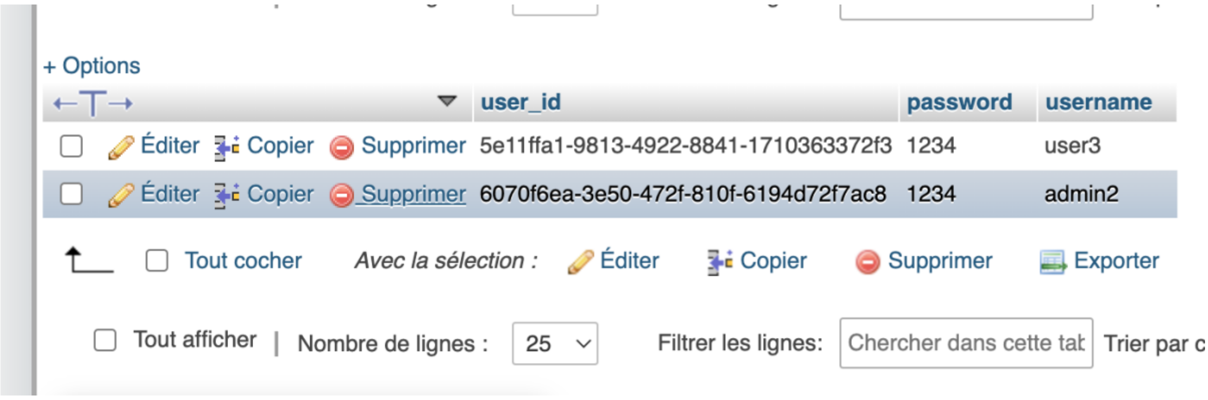


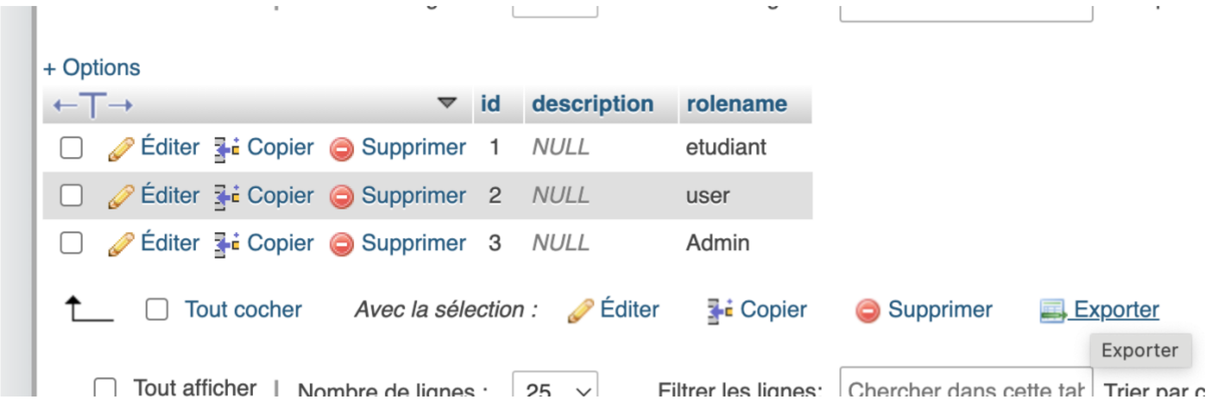
Table	Action	Lignes	Type	Interclassement	Taille	Perte
role	Parcourir Structure Rechercher Insérer Vider Supprimer	3	InnoDB	utf8mb4_general_ci	16,0 kio	–
role_users	Parcourir Structure Rechercher Insérer Vider Supprimer	4	InnoDB	utf8mb4_general_ci	48,0 kio	–
users	Parcourir Structure Rechercher Insérer Vider Supprimer	2	InnoDB	utf8mb4_general_ci	32,0 kio	–
3 tables	Somme	9	InnoDB	utf8mb4_general_ci	96,0 kio	0 o

Table Utilisateur.



←T→	user_id	password	username
□ Éditer Copier Supprimer	5e11ffa1-9813-4922-8841-1710363372f3	1234	user3
□ Éditer Copier Supprimer	6070f6ea-3e50-472f-810f-6194d72f7ac8	1234	admin2

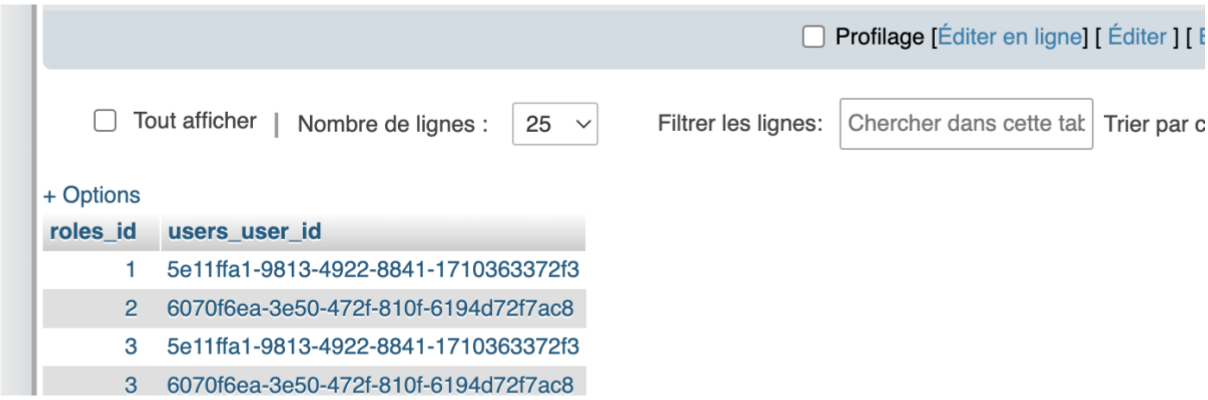
Table Role:



	id	description	rolename
□ Éditer Copier Supprimer	1	*NULL*	etudiant
□ Éditer Copier Supprimer	2	*NULL*	user
□ Éditer Copier Supprimer	3	*NULL*	Admin

□ Tout cocher
Avec la sélection :
Éditer
Copier
Supprimer
Exporter
Exporter
□ Tout afficher | Nombre de lignes : 25
Filtrer les lignes: Chercher dans cette tat
Trier par c

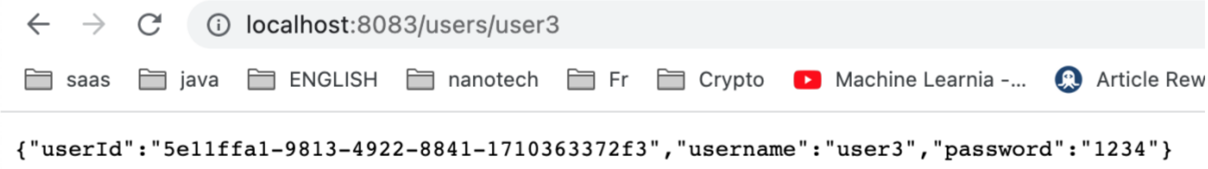
Table Role User:



□ Profilage [Éditer en ligne] [Éditer] [E
□ Tout afficher | Nombre de lignes : 25
Filtrer les lignes: Chercher dans cette tat
Trier par c
+ Options

roles_id	users_user_id
1	5e11ffa1-9813-4922-8841-1710363372f3
2	6070f6ea-3e50-472f-810f-6194d72f7ac8
3	5e11ffa1-9813-4922-8841-1710363372f3
3	6070f6ea-3e50-472f-810f-6194d72f7ac8

Retrouver sous format Json l'utilisateur qui a comme nom d'utilisateur user3 .



```
{"userId":"5e11ffa1-9813-4922-8841-1710363372f3","username":"user3","password":"1234"}
```