Mapping des associations et d'héritage

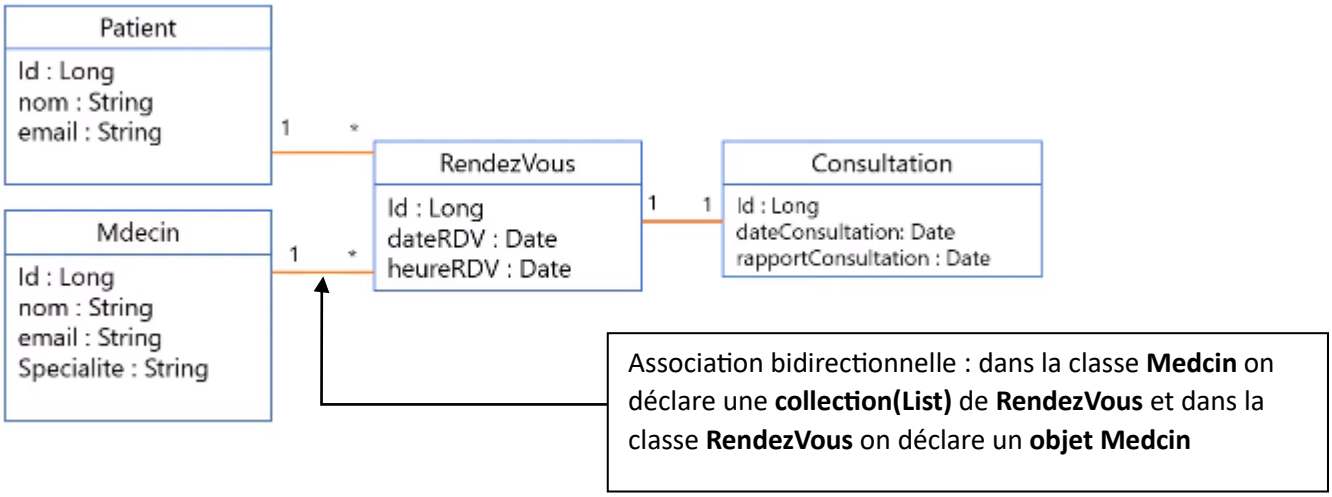
JPA Hibernate Spring Data

➢ **A savoir :**

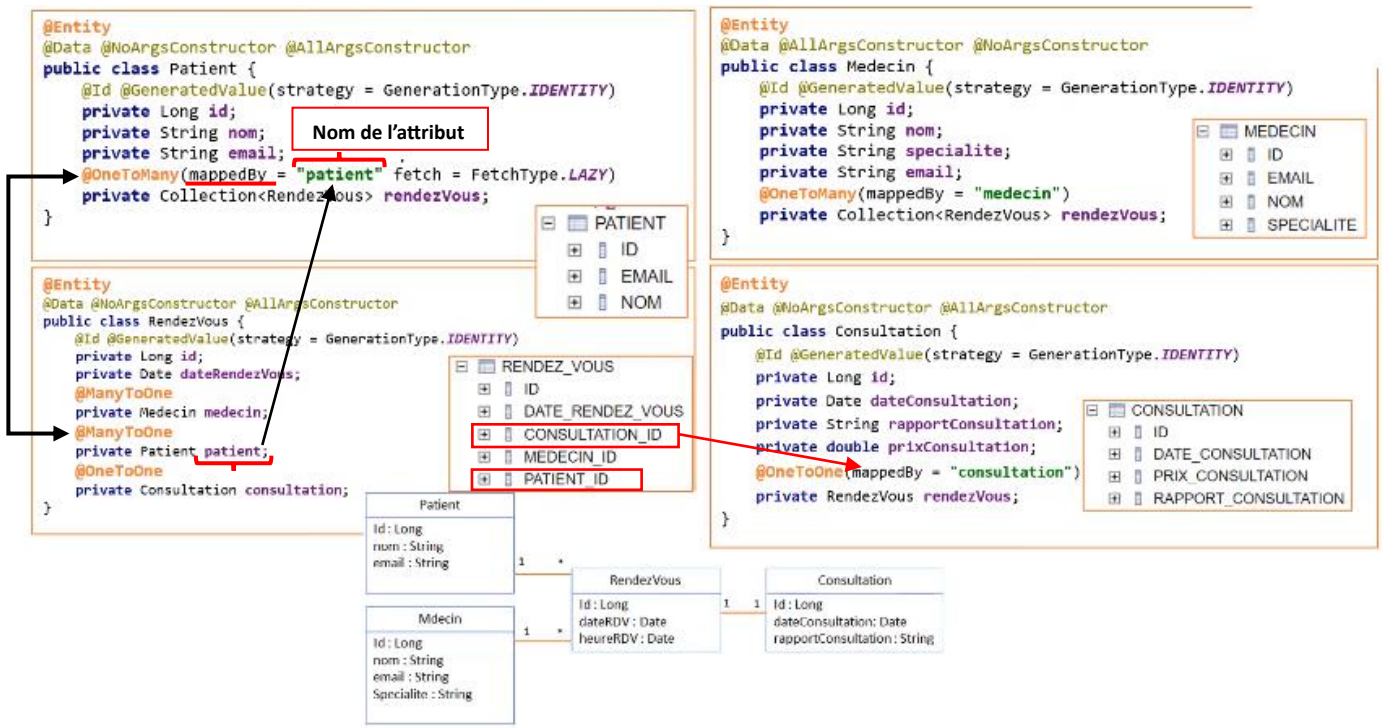
– Accès aux données est un aspect technique alors il faut utiliser in Framework (IOC)

Association: OneToMany, ManyToOne, OneToOne

➢ **Exemple :**



Implémentation :



➢ On utilise **mappedBy** pour indiquer que c'est la même association

➢ La clé étrangère va être dans la table qui ne contient pas la propriété **mappedBy**

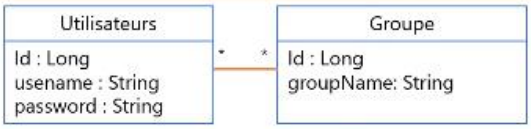
- **mappedBy <-> OneToMany**

Attribute fetch:

- **LAZY:** quand l'objet Patient est chargé dans la RAM, il ne va pas ramener la collection de rendezVous.
- **EAGER:** A chaque fois l'objet Patient est chargé dans la RAM il va ramener aussi la collection de ces rendezVous.

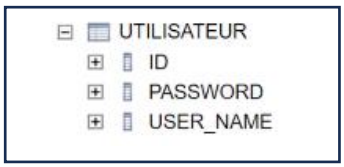
Association: ManyToMany

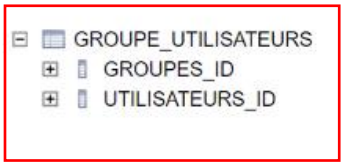
- On suppose que l'on souhaite de créer une application qui permet de gérer des Utilisateurs appartenant à des groupes. Chaque Groupe peut contenir plusieurs utilisateurs.

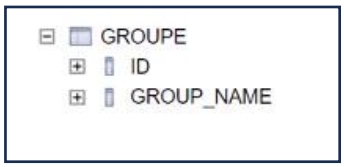



```java
@Entity
@Data @NoArgsConstructor @AllArgsConstructor
public class Utilisateur {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @Column(unique = true)
    private String userName;
    private String password;
    @ManyToMany(mappedBy = "utilisateurs",fetch = FetchType.EAGER)
    private Collection<Groupe> groupes=new ArrayList<>();
}
```

```java
@Entity
@Data @AllArgsConstructor @NoArgsConstructor
public class Groupe {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @Column(unique = true)
    private String groupName;
    @ManyToMany(fetch = FetchType.EAGER)
    private Collection<Utilisateur> utilisateurs=new ArrayList<>();
}
```



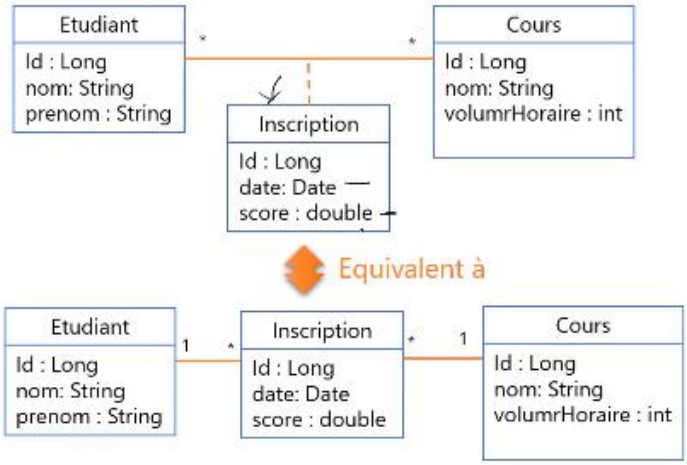




Cas de ManyToMany avec Classe d'association

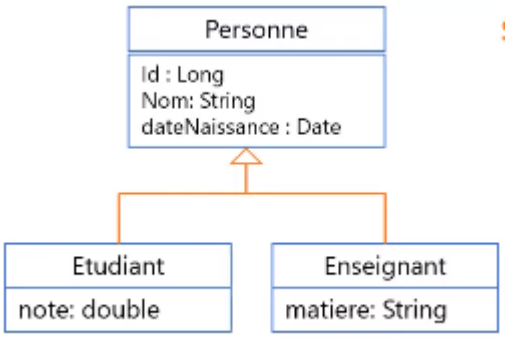
Exemple :

- Un Etudiant peut s'inscrire dans plusieurs Cours à une date donnée avec un score obtenu. Un cours concerne plusieurs Inscriptions. (Plusieurs à Plusieurs avec Classe d'association Inscription)
- Equivalent à : Un Etudiant peut effectuer Plusieurs Inscription. Chaque Inscription Concerne un Cours
- Deux Associations : Un à Plusieurs + Plusieurs à 1



Mapping de l'héritage : **Au niveau de la BD**

➢ Exemple



Il existe 3 stratégies de mapping de l'héritage :

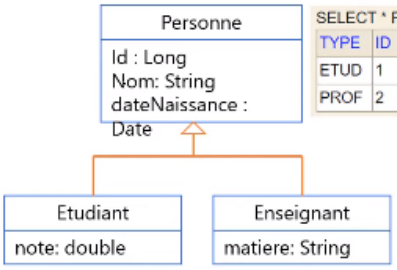
- **SINGLE_TABLE** : Une table qui rassemble tous

SINGLE_TABLE : PERSONNES

ID	TYPE	NOM	DATE_NAISSANCE	NOTE	MAIETRE
1	ET	N1	d1	15	NULL
2	PROF	N2	d2	NULL	MATH

Discriminator Column: used in **SINGLE_TABLE** inheritance because we need a column to identify the type of the record.

Mapping de l'héritage : Stratégie SINGLE_TABLE



SELECT * FROM PERSONNE;

TYPE	ID	DATE_NAISSANCE	NOM	MATIERE	NOTE
ETUD	1	2020-09-03 20:05:28.358	Hassan	*null*	14.0
PROF	2	2020-09-03 20:05:28.362	Mohamed	MATH	*null*

```
@Entity
@Data @NoArgsConstructor @AllArgsConstruc
@DiscriminatorValue("ETUD")
public class Etudiant extends Personne {
    private double note;
}
```

```
@Entity
@Data @NoArgsConstructor @AllArgsConstructor
@DiscriminatorValue("PROF")
public class Enseignant extends Personne {
    private String matiere;
}
```

```
@Entity
@Data @NoArgsConstructor @AllArgsConstructor
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
@DiscriminatorColumn(name = "TYPE",length = 4)
public abstract class Personne {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nom;
    private Date dateNaissance;
}
```

```
@Autowired
private PersonneRepository personneRepository;
```

```
Etudiant etudiant=new Etudiant();
etudiant.setNom("Hassan");
etudiant.setDateNaissance(new Date());
etudiant.setNote(14);
personneRepository.save(etudiant);
```

```
Enseignant enseignant=new Enseignant();
enseignant.setNom("Mohamed");
enseignant.setDateNaissance(new Date());
enseignant.setMatiere("MATH");
personneRepository.save(enseignant);
```

```
public interface PersonneRepository extends JpaRepository<Personne,Long> {
}
```

- **TABLE_PER_CLASSE** : Une table pour chaque classe concrète

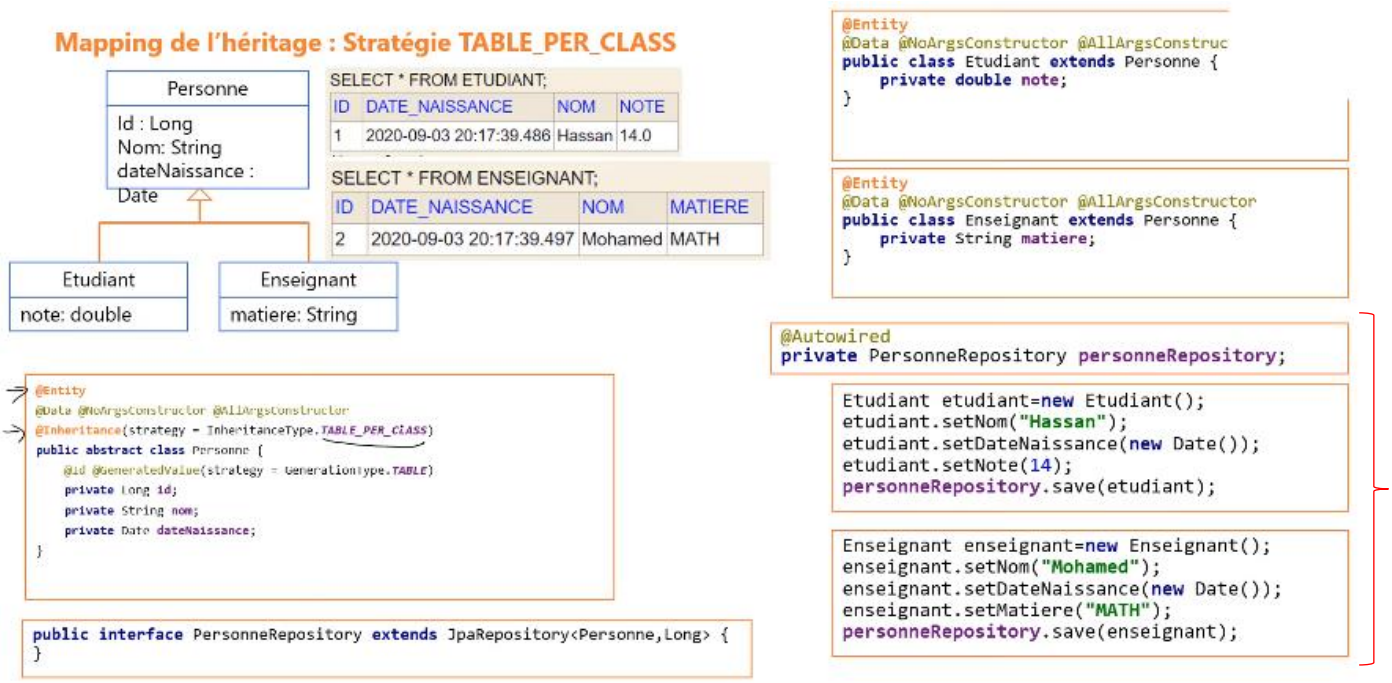
TABLE_PER_CLASS

ETUDIANTS

ID	NOM	DATE_NAISSANCE	NOTE
1	N1	d1	15

ENSEIGNANTS

ID	NOM	DATE_NAISSANCE	MATIERE
2	N2	d2	MATH



– **JOINED_TABLE** : Une table pour la classe parente et une table pour chaque classe fille