Héritage –TABLE\_PER\_CLASS

# Démonstration 12 du module 2

|  |
| --- |
| L’objectif de cette démonstration :   * Déclaration d’une hiérarchie d’héritage avec TABLE\_PER\_CLASS |

Déroulement

# Contexte

* Continuer l’application précédente
* Dans cette itération, nous allons uniquement changer la configuration d’héritage
* Nous allons utiliser TABLE\_PER\_CLASS

# Modification de l’entité Employe :

* Mettre en commentaire la gestion d’héritage en SINGLE\_TABLE
* Aouter l’héritage en TABLE\_PER\_CLASS
* Modifier la stratégie de génération de la clef primaire en TABLE

…

@Entity

@Table(name = "EMPLOYEE")

@Inheritance(strategy = InheritanceType.***TABLE\_PER\_CLASS***)

//@Inheritance(strategy = InheritanceType.SINGLE\_TABLE)

//@DiscriminatorColumn(name = "DISCR")

//@DiscriminatorValue(value = "Emp")

**public** **class** Employe {

// Attributs

// Clef primaire -- gestion par IDENTITY

//@GeneratedValue(strategy = GenerationType.IDENTITY)

// Clef primaire -- gestion par TABLE

@GeneratedValue(strategy = GenerationType.***TABLE***)

@Id

@Column(name = "EMPLOYEE\_ID")

**private** Integer id;

…

}

# Modification des entités Formateur et ChargeRelationsEcoleEntreprises :

* Mettre en commentaire la colonne discriminante

@Entity

@Table(name = "TRAINER")

//@DiscriminatorValue(value = "F")

**public** **class** Formateur **extends** Employe {

* Nous avions laissé l’annotation pour nommer explicitement les tables de chaque classe
  + Dans le cas de SINGLE\_TABLE ; seule l’annotation @Table sur Employe a été utilisée
  + Dans le cas de TABLE\_PER\_CLASS ; chaque classe aussi sa table correspondante

# Tests

* La classe TestHeritage n’a pas besoin d’être modifiée
* Tous les tests doivent être verts
* Au niveau base de données :
  + Création de 4 tables
  + La table pour l’entité Employee

create table employee (

address\_id integer unique,

civility\_id varchar(3),

employee\_id integer not null,

cell\_number varchar(12),

home\_phone\_number varchar(12),

last\_name varchar(90) not null,

employee\_registration varchar(100) not null unique,

first\_name varchar(150) not null,

email varchar(255) not null unique,

primary key (employee\_id))

* + Création d’une table de séquence pour gérer la clef primaire

create table hibernate\_sequences (

next\_val bigint,

sequence\_name varchar(255) not null,

primary key (sequence\_name))

* + La table pour l’entité ChargeRelationsEcoleEntreprises

create table school\_business\_officer (

address\_id integer unique,

civility\_id varchar(3),

employee\_id integer not null,

cell\_number varchar(12),

home\_phone\_number varchar(12),

office\_phone\_number varchar(12),

last\_name varchar(90) not null,

employee\_registration varchar(100) not null unique,

first\_name varchar(150) not null,

email varchar(255) not null unique,

primary key (employee\_id))

* + Et la table pour l’entité Formateur

create table trainer (

address\_id integer unique,

civility\_id varchar(3),

employee\_id integer not null,

cell\_number varchar(12),

home\_phone\_number varchar(12),

last\_name varchar(90) not null,

employee\_registration varchar(100) not null unique,

computer\_science\_course varchar(150),

first\_name varchar(150) not null,

email varchar(255) not null unique,

primary key (employee\_id))

* La clef employee\_id est présente dans les 3 tables
* Les attributs de l’entité Employe sont dupliqués dans les tables des entités Formateur et ChargeRelationsEcoleEntreprises
* Dans les entités filles, il y a une colonne supplémentaire correspondante à l’attribut spécifique de chaque fille : filiere et numeroBureau
* Quand il y a insertion d’un employé, d’un formateur ou d’un chargé de relations Ecole-Entreprises, il y a
  + Un SELECT pour la clef primaire dans la table des séquences
  + Et un UPDATE pour la prochaine valeur

select tbl.next\_val from hibernate\_sequences tbl where tbl.sequence\_name=? for update

update hibernate\_sequences set next\_val=? where next\_val=? and sequence\_name=?

* Les employés enregistrés sont :

[

Employe(

id=1,

civilite=null,

nom=NICOLAS,

prenom=Cédric,

email=cnicolas@campus-eni.fr,

immatriculation=ENI\_ECOLE\_10100,

numDom=02XXXXXXXX,

numPortable=06XXXXXXXX,

adresse=null),

Formateur(

super=Employe(

id=2,

civilite=null,

nom=BAILLE,

prenom=Anne-Lise,

email=abaille@campus-eni.fr,

immatriculation=ENI\_ECOLE\_12398,

numDom=02XXXXXXXX,

numPortable=null,

adresse=null),

filiere=Développement),

ChargeRelationsEcoleEntreprises(

super=Employe(

id=3,

civilite=null,

nom=PAGEOT,

prenom=Emma,

email=epageot@campus-eni.fr,

immatriculation=ENI\_ECOLE\_13398,

numDom=02XXXXXXXX,

numPortable=06XXXXXXXX,

adresse=null),

numeroBureau=04XXXXXXXX)

]

* Les INSERT correspondant sont :

insert into employee (address\_id,civility\_id,email,employee\_registration,last\_name,home\_phone\_number,cell\_number,first\_name,employee\_id) values (?,?,?,?,?,?,?,?,?,?)

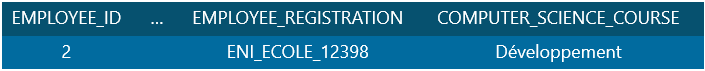
insert into trainer (address\_id,civility\_id,email, employee\_registration,last\_name,home\_phone\_number,cell\_number,first\_name,computer\_science\_course,employee\_id) values (?,?,?,?,?,?,?,?,?,?,?)

insert into school\_business\_officer (address\_id,civility\_id,email,employee\_registration,last\_name,home\_phone\_number,cell\_number,first\_name,office\_phone\_number,employee\_id) values (?,?,?,?,?,?,?,?,?,?,?)

* + 3 insertions sur chaque table correspondante à l’entité
    - Table de l’entité Employe



* + - Table de l’entité Formateur



* + - Table de l’entité ChargeRelationsEcoleEntreprises

