

Mathissart Nicolas

# HIBERNATE

- produit → 1 catégorie
- 1 catégorie → 0,n produits

\* Ligne = détail de la commande

## Création des classes métier

### 1) Comment est déclarée la table assurant la persistance d'un objet ?

Pour déclarer une table assurant la persistance d'un objet, il suffit d'ajouter la ligne de code ci-dessous qui permet de définir le nom de la table.

Il faut mettre la ligne de code au dessus de la classe :

[|h Déclaration d'une classe de mapping](#)

```
@Table(name="Categorie") // C'est cette ligne qui permet de définir la table
public class Categorie {
    @Id
    @Column(name="id")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private int id;
```

### 2) Comment est déclaré le mapping entre un membre de la classe et un champ de la table relationnelle ?

Afin de déclarer un mapping entre un membre de la classe et un champ de la table relationnelle, il suffit d'ajouter une ligne de code qui permet de définir le nom de notre champ et de définir son type. voici la ligne de code qu'il vous faut :

[|h Champ](#)

```
@Column(name="libelle") //On indique que notre champ s'appellera Libelle
private String libelle; // On indique que notre champ sera un String
```

```
@Column(name="monChamp")
```

```
private monType monChamp
```

### 3) Comment est déclarée la clé primaire de la table ?

Pour déclarer une clé primaire sur Hibernate, il suffit de rajouter la balise "@Id" au-dessus de la colonne concerné. Par exemple :

## |h Champ

```
@Id // C'est cette balise qui permet de définir la clé primaire.  
@Column(name="id")  
private int id;
```

### 4) Quelles sont les possibilités de déclaration des clés primaires ?

Sur hibernate, il y à plusieurs méthodes pour déclarer les clés primaires.

- Afin d'auto incrémenter à partir de la base donnée une clé primaire, il suffit d'utiliser cette ligne de code ;

## |h Auto

```
@GeneratedValue(strategy=generationType.AUTO)
```

-

## |h Identity

```
@GeneratedValue(strategy=GenerationType.IDENTITY)
```

- Afin d'auto incrémenter à partir de Hibernate une clé primaire, il suffit d'utiliser cette ligne de code ;

## |h Sequence

```
@GeneratedValue(strategy=GenerationType.SEQUENCE)
```

-

## |h Table

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

### 5) tableau montrant la correspondance de type (entier, chaine, etc.) entre les propriétés d'une classe et les champs d'une table

Voici un tableau recapitulant, la correspondance de type entre les propriétés d'une classe et les champs d'une table :

Type Java	Type Sql
String	Varchar
int	Integer
float	Float

## 6) lien bidirectionnel entre deux classes



## 7) À quoi correspond la méthode persist() ?

La méthode persist sur Hibernate permet d'ajouter un élément dans la base de données :

### |h Méthode Persist

```

Categorie aCategorie=new Categorie("Presse"); // On instancie un produit, donc
l'objet que l'on insérera dans la base de donnée
session.persist(aCategorie); // Une fois l'objet instancier, on l'insère à la
BDD

```

## 8) À quoi correspond la méthode commit() ?

La méthode commit permet de mettre à jour la base de donnée avec les objets persistants. Autrement dit, la méthode commit permet d'insérer dans la base de donnée les objets qui sont valide.

Code de la méthode commit :

### |h Méthode Commit

```

Transaction trans = session.beginTransaction();

trans.commit();

```

## 8) Comment ont été traduits les liens objet entre le membre categorie et produits entre ces classes dans les tables de la base ?

Lors de l'insertion d'un produit et d'une catégorie dans la base de donnée, nous avons indiqué au produit qu'il fera partit de la catégorie "aCategorie" que nous avons instancier au dessus ( Catégorie : Presse ).

### |h Produit

```

Produit aProduit=new Produit("Programmez!", 3.0f, aCategorie) //On ne passe
pas l'id de la catégorie mais directement la catégorie concerné;

```

Lors de l'insertion dans la base de donnée, l'iDCategorie (clé étrangère) de notre produit à donc prit comme valeur la clé primaire de la catégorie "aCategorie".

## 9) Quelles requêtes SQL ont été créées par Hibernate pour réaliser la persistance ?

Afin de réaliser la persistance, Hibernate à mit en places les requêtes SQL, Insert, Update et Delete.

From:

<http://slamwiki2.kobject.net/> - **Broken SlamWiki 2.0**

Permanent link:

<http://slamwiki2.kobject.net/slam4/orm/etudiants/nicolas?rev=1354632064>

Last update: **2019/08/31 14:39**

