

# TP-TD6 - Comparaison de frameworks de mapping relationnel/objet

## Création des classes métier

### Produits et catégories

1. La table assurant la persistance d'un objet est déclaré en héritant de la classe Kobject :

```
public class KProduit extends KObject
```

2. Le mapping entre un membre de la classe et un champ de la table relationnelle est déclaré dans le constructeur en tant que membre :

```
hasMany(KLigne.class);  
belongsTo(KCategorie.class);
```

3. La clé primaire de la table est déclaré avec un keyFields.

```
keyFields="id";
```

4.

Propriété d'une classe	Champs d'une table	Type
int	int	entier
string	varchar	chaîne

5. Le lien bidirectionnel entre deux classes se paramètre par :

### Programme de test

1. La méthode kstart() correspond au démarrage de l'application java qui permettra l'ajout dans la base de donnée.

2. Les liens objet entre le membre **catégorie** et **produits** entre ces classes dans les tables de la base ont été traduit par le fait que l'id de la table **catégorie** s'est mis en clé étrangère de la table **produits**.

3. Les requêtes SQL qui ont été créées par KObject pour réaliser la persistance sont :

- Liste à puce

4. Si l'insertion de la catégorie échoue alors

## Chargement d'un objet

### Programme de chargement d'une catégorie

1. Ce que charge exactement KObject lors du chargement d'un Objet est

2. Les instances liées à un objet chargé pour les liens belongsTo et hasMany sont chargées grâce à

3. Le chargement paresseux de KObject consiste à

## Chargement de listes d'objets

### Projection

1. Résultat obtenu :

### Sélection

1. "NB" requêtes SQL sont exécutées par KObject.

2. On peut l'interpréter par

Remplacer le lien belongsTo sur la classe Produit par : `belongsTo(KCategorie.class).setLazy(true);`

1. "NB" requêtes SQL sont maintenant exécutées par KObject.

2. On peut l'interpréter par

Exécutez à nouveau le programme.

1. On peut interpréter les requêtes SQL exécutées par KObject par

### Sélection avec distinct et projection

1. Ce programme

## Gestion des commandes

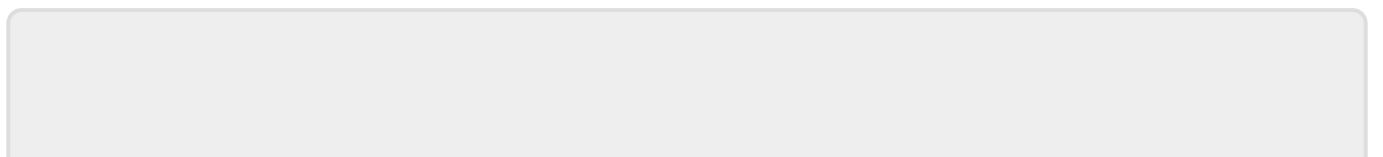
1. L'appel des méthodes permettant de mettre en œuvre la contrainte d'intégrité multiple est justifié par

### Création de commandes

1. Code commenté de CreateCommandeGood.java

2. L'exécution a effectué les ajouts dans la base de données :

## Test Web



From:

<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:

<http://slamwiki2.kobject.net/slam4/orm/etudiants/clement?rev=1354977492>

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

