Produits et catégories

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

La table assurant la persistance d'un objet est déclarée s'il hérite de KObject –> public class KCategorie extends KObject.

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

code Java

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

"hasMany(KLigne.class)" un produit correspondant à une ligne, il y a un produit par ligne "belongsTo(KCategorie.class)" les produits correspondent à une catégorie, il y a des produits dans une catégorie.

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

code Java

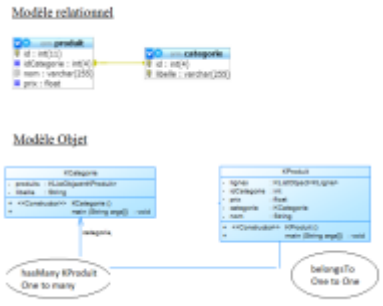
```
keyFields="id";
```

keyFields="id"; , la clé primaire est déclaré en keyFields

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

int	int	entier
string	varchar	chaine

5) Montrez à l'aide d'un schéma (par ex. deux classes liées au dessus de deux tables liées) comment se paramètre le lien bidirectionnel entre deux classes (en spécifiant les éléments à fournir dans le constructeur)



Programme de test

Analysez le code du programme et répondez aux questions en vous aidant au besoin de la documentation :

1) À quoi correspond la méthode kstart() ?

code Java

```
public static void main(String[] args) {
        Ko.useCache=true;
        KCache.loadAllCache();
        try {
            Ko.kstart();
```

La méthode kstart() correspond au démarrage du test de l'ajout d'un produit dans la base de données

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

Les liens objets entre “catégorie” et “produits” entre ces classes dans les tables de la base de données ont été traduit par le simple fait que l'id de la table “catégorie” s'est mis en clé étrangère de la table “produits”.

3) Quelles requêtes SQL ont été créées par KObject pour réaliser la persistance ?

Afin de réaliser la persistance, KObject a utilisé les requêtes SQL d'insertion suivantes :
(visibles dans la console lors du test)
INSERT INTO categorie(libelle) VALUES('Presse')
INSERT INTO produit(idCategorie,prix,nom) VALUES('19','3.0','Programmez!')

4) Que se passe t-il si l'insertion de la catégorie échoue ?

En cas d'échec de l'insertion de la catégorie, il est impossible par la suite d'insérer des produits car un produits appartient à une catégorie et ces deux objets sont liés par l'id de la catégorie.

Chargement d'un objet

Observation du chargement d'un objet, par l'intermédiaire de sa clé primaire.

1) Précisez ce que charge exactement KObject lors du chargement d'un Objet

Lors du chargement d'un Objet, KObject va charger les catégories mais aussi les objets de ces catégories
SELECT * FROM categorie WHERE categorie.id='1'
KOBJECT : KCategorie.loadFromDb → {id=1}
KOBJECT : KConstraintHasMany.load → class metier.KProduit
SELECT * FROM produit WHERE idCategorie='1'
KOBJECT : KProduit.loadFromDb → {id=52}
[…]
KOBJECT : KProduit.loadFromDb → {id=182}

2) Précisez comment sont chargées les instances liées à un objet chargé pour les liens belongsTo et hasMany

belongsTo:
Pour charger les instances liées à un objet en Many to One, par exemple un produit, KObject va dans un premier temps récupérer le produit à l'aide d'un select. Ensuite KObject va récupérer la catégorie correspondante.
Console:
SQL : KDataBase.sendQuery → SELECT * FROM produit WHERE produit.id='52'
SQL : KDataBase.sendQuery → SELECT * FROM Ligne WHERE idProduit='52'
SQL : KDataBase.sendQuery → SELECT * FROM categorie WHERE id='1'
hasMany:
Pour le cas d'un One To Many KObject va effectuer deux requêtes SQL. Dans un premier temps la catégorie sera chargée ensuite tous les produits appartenant à cette catégorie seront chargés.
Console:
SQL : KDataBase.sendQuery → SELECT * FROM categorie WHERE categorie.id='1'
SQL : KDataBase.sendQuery → SELECT * FROM produit WHERE idCategorie='1'

3) En quoi consiste le chargement paresseux de KObject ?

Le chargement dit “paresseux” de KObject consiste a ne pas charger chaque objets d'un objet précis à moins de lui en donner l'instruction.
Exemple : Il n'affichera pas tous les produits d'une catégorie s'il l'utilisateur ne le demande pas.

Un peu comme Quentin .. 

Chargement de listes d'objets

Interrogation de données avec KObject :

Projection

1) Interprétez et expliquez le résultat obtenu

Voici le résultat obtenu :

code java

```
    System.out.println(categories.showWithMask("{libelle}:\n{produits}\n"));
```

Explication : On affiche à l'écran les libellés et les produits des catégories affichés par ordre alphabétique.

Sélection

1) Combien de requêtes SQL sont exécutées par KObject ?

2) Comment l'interprétez vous ?

Remplacer le lien belongsTo sur la classe Produit par :

|h code Java

```
belongsTo(KCategorie.class).setLazy(true);
```

1) Combien de requêtes SQL sont maintenant exécutées par KObject ?

2) Comment l'interprétez vous ?

A partir du programme :

1) Interprétez les requêtes SQL exécutées par KObject

Sélection avec distinct et projection

A partir du code et de son exécution :

1) Expliquer ce que fait le programme

Gestion des commandes

Commande

Ligne

1) Justifiez l'appel des méthodes permettant de mettre en oeuvre la contrainte d'intégrité multiple

Création de commandes

1) Analysez puis commentez chaque ligne (dans le code) de ce programme

2) Vérifier que l'exécution a effectué les ajouts dans la base de données

Test Web

...
Ajouter toutes les classes (servlet) et méthodes nécessaires pour éviter d'avoir à effectuer un quelconque traitement dans les JSP.

Document réalisé par : — *maxime*