Produits et catégories

Afficher le concepteur pour visualiser les tables, et les relations : Pour chaque table, notez les contraintes d'intégrité :

```
d'entité (clé primaire)
référentielle (relations)
```

Produit :

```
id (primary key)
idCategorie (foreign key references categorie.id)
```

Ligne :

```
numero (primary key)
idCommande (foreign key references commande.id)
```

Commande :

```
id (primary key)
```

Catégorie :

```
id (primary key)
```

A partir de l'observation de cette première implémentation et en utilisant à bon escient la documentation, répondez aux questions suivantes :

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

@Entity

@Table(name = "Categorie")

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

@Column(name = "libelle")

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

@Id

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

@Id

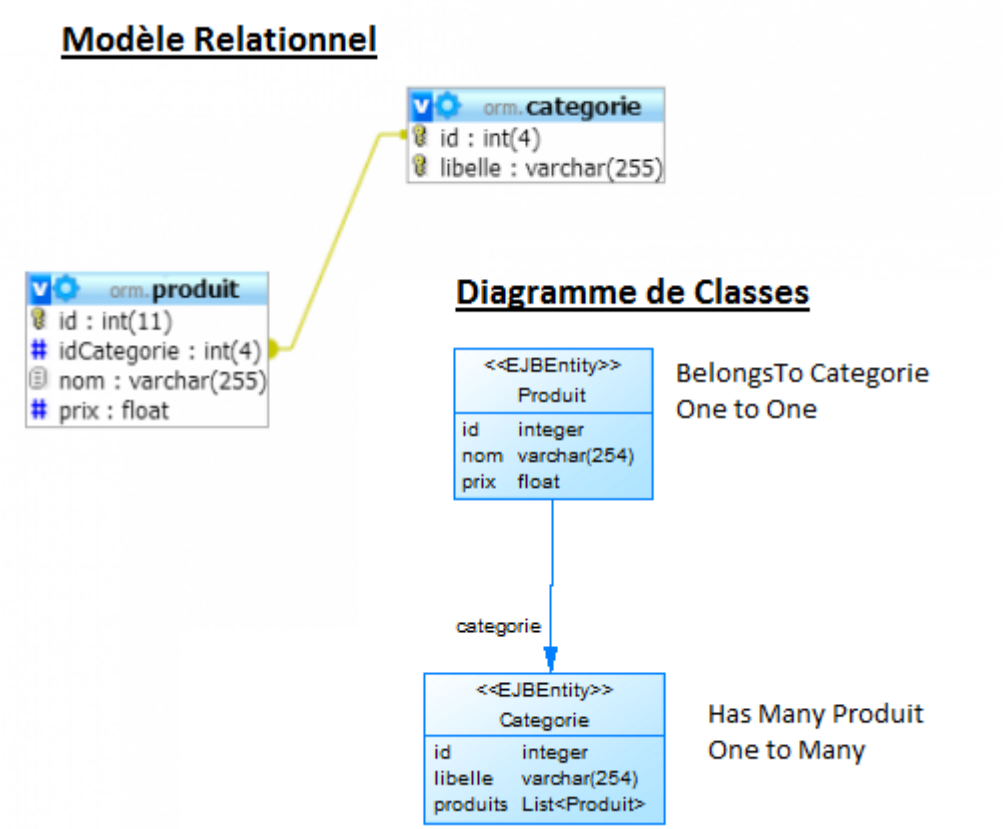
@GeneratedValue

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

Propriété de la Classe	**Champ de la table**	**Type**
int	int	Entier
String	varchar	Chaine
Date	timestamp	Date
float	float	flottant

6)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 les annotations)

Modèle Relationnel :



Programme de test

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

À quoi correspond la méthode persist() ?

persist() est la méthode qui permet d'ajouter à la sessions les objets qui doivent être synchronisés avec la base de données.

À quoi correspond la méthode commit () ?

La méthode commit() permet de transmettre les objets à synchroniser à la base de données et de les y ajouter.

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

Ces liens sont représentés par une clé étrangère dans la table produits. Pour le produit "Programmez" la clé étrangère est idCategorie est correspond à l'id de la catégorie "Presse".

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

Hibernate:

```
insert
into
    Categorie
    (libelle)
values
```

```
    (?)
```

Hibernate:

```
insert
into
    Produit
    (idCategorie, nom, prix, id)
values
    (?, ?, ?, ?)
```

Pourquoi comportent t-elles des points d'interrogation ?

Les points d'interrogation représentent les paramètres JDBC liés

Chargement d'un objet

A partir de ses 2 programmes et de leur exécution :

Précisez ce que charge exactement Hibernate lors du chargement d'un Objet :

Il charge toutes ses propriétés. (Toutes les colonnes de la table)

Précisez comment sont chargés les instances liées à un objet chargé pour les liens oneToMany et manyToOne :

1 - Pour **manyToOne**, Produit par exemple, lors du chargement d'un produit, on charge sa catégorie. Cette catégorie possède une liste de produits, automatiquement chargés.

2 - Pour **oneToMany**, Catégorie, par exemple, on charge d'abord ses propriétés, libelle, id etc... On ne chargera la liste des produits qui lui sont propre que lorsque l'on cherchera à acceder à un élèment de cette liste.

En quoi consiste le chargement paresseux d'Hibernate et la qualification **lazy** (rechercher dans l'aide) :

Le chargement paresseux correspond à la réponse 2 de la question précédente. On ne charge pas directement tous les elements de la liste de la catégorie.

Chargement de listes d'objets

A partir de ce programme :

Interprétez la forme de la requête passée à la méthode createQuery, pourquoi n'est-elle pas complète ?

Il semble que createQuery permette de charger l'ensemble des élèments d'une table. Il suffit donc de lui indiquer from which table.

Renseignez-vous sur HQL dans la documentation