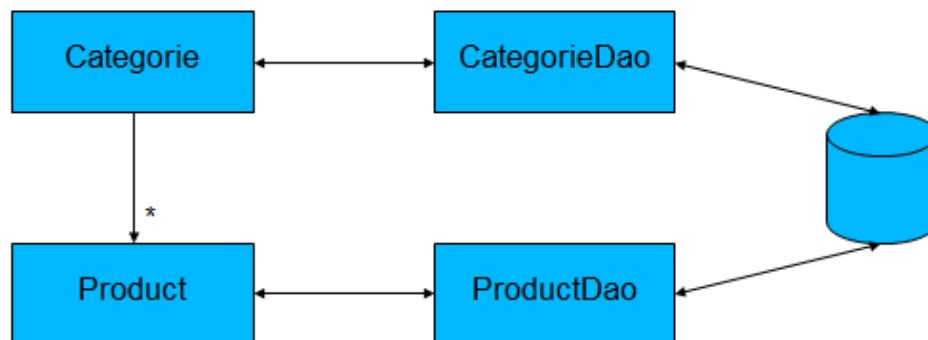


P2W2 Persistentie – Opgave DAO2

Pak het gezippte IntelliJ-project uit en bestudeer de code die al klaar is. Het gaat over producten die tot een bepaalde categorie behoren. Dus tussen **Categorie** en **Product** is er een één op veel relatie.



Werk best in deze volgorde:

1. De klasse **Product** is al klaar. Graag je aandacht voor volgende aspecten:
 - **id** is de primary key, die door de database zelf wordt aangemaakt (autonummering). **categorieId** is de foreign key naar het gerelateerde **categorie**-record.
 - 3 constructors, waarvan 2 public; daar wordt de **id** voorlopig op -1 gezet. Eén van de constructors heeft package-toegang.
 - De methoden **equals** en **hashCode** zijn uiteraard enkel gebaseerd op het attribuut **id**.
2. De klasse **Categorie** is al klaar.
 - **id** is de primary key; wordt door de database aangemaakt.
 - Elk **categorie**-object bevat een **HashSet** met gerelateerde producten.
 - Verder dezelfde opmerkingen als bij **Product**.
3. De klasse **HsqlDao** is al klaar. Ze bevat allerlei algemene functionaliteiten zoals het laden van de driver, het leggen van een connectie, het aanmaken van een statement en afsluiten. De andere DAO-klassen erven hiervan over. Merk op: de methoden hebben **protected** access.
4. De klasse **ProductDao** erft over van **HsqlDao**:
 - **retrieveByCategorie**: Zoek alle producten die gerelateerd zijn aan een bepaalde categorie en retourneer ze in de vorm van een Set. Begin altijd met het aanmaken van een nieuwe connection en een nieuw statement (via de superklasse). Gebruik: **statement.executeQuery** voor het ophalen van de records.
 - **delete**: Verwijder het product met de meegegeven **id** en geef feedback in de vorm van een boolean. Gebruik **statement.executeUpdate** voor het verwijderen van een record.

5. De klasse **CategorieDao** erft over van **HsqlDao**:

- Merk op dat **ProductDao** via de constructor binnenkomt en beschikbaar is als attribuut.
- **create**: Het toevoegen van het nieuwe categorie-record is al in orde. Jij moet nu alle producten van deze categorie overlopen:
 - gebruik **setCategorieId** om de categoriële id in te stellen
 - gebruik **getId** om te weten of het een nieuw product is (-1) of een product dat zich reeds in de database bevindt. Afhankelijk hiervan doe je een create of een update in de database.
- **update**: verwijder de producten die wel in de database zitten, maar niet in de set. Voeg vervolgens de nieuwe producten toe.
- **delete**: het verwijderen van het categorie-record is al in orde. Jij moet nu nog alle gerelateerde producten verwijderen.

6. Run nu de klasse **TestDao2** en controleer de output:

voor toevoegen aan de database:

```
categorie[-1, Fruit]
  product[ 0, Bananen, €1,69, cat: -1]
  product[ 1, Appelen, €2,29, cat: -1]
```

na ophalen uit de database:

```
categorie[0, Fruit]
  product[ 0, Bananen, €1,69, cat: 0]
  product[ 1, Appelen, €2,29, cat: 0]
```

na koppelen nieuw product (peren):

```
categorie[0, Fruit]
  product[ 0, Bananen, €1,69, cat: 0]
  product[ 1, Appelen, €2,29, cat: 0]
  product[ 2, Peren, €2,99, cat: 0]
```

na verwijderen product (peren):

```
categorie[0, Fruit]
  product[ 0, Bananen, €1,69, cat: 0]
  product[ 1, Appelen, €2,29, cat: 0]
```

na verwijderen categorie (fruit):

```
null
```

op zoek naar de bananen:

```
null
```