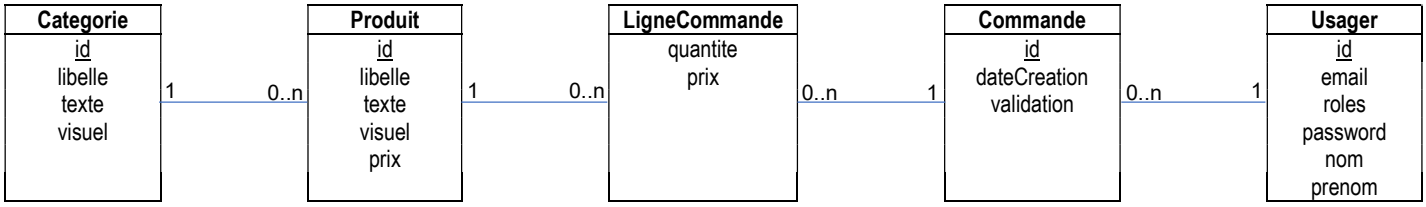
R4.01 - TP 05 – Usager, Commande
1 séance

5.0. Travail à réaliser – Vue d'ensemble

Il faut maintenant donner la possibilité à un usager enregistré de transformer un panier en commande.
Pour cela, il faut compléter notre « modèle » en y rajoutant les entités **`Usager`**, **`Commande`** et **`LigneCommande`**.
Le diagramme de classe des entités devra être le suivant :



Pour l'instant, nous n'avons pas mis en place la « sécurité » de notre application, c'est-à-dire l'authentification de l'utilisateur *via* le composant « sécurité » de Symfony. En attendant cette étape, nous allons, dans un premier temps :

- Donner la possibilité de créer un nouvel **`Usager`**
- Mettre en place la validation du panier, validation qui consistera à :
  - Créer une commande pour l'usager d'identifiant égal à 1 *(c'est temporaire, en attendant le TP6)*
  - Créer autant de lignes de commande qu'il y a d'articles dans le panier

Par la suite nous modifierons ce code lorsque nous aurons mis en place l'authentification pour que ce soit l'utilisateur authentifié qui passe commande. La modification sera très rapide à réaliser !

5.1. Entité Usager : Grimpons sur l'Echaffaudage

L'entité Usager est un peu particulière car c'est l'entité qui sera utilisée pour mettre en place l'authentification par la suite. Elle devra avoir des propriétés particulières. Pour cela :

- Taper la commande suivante pour installer un package nécessaire : **`composer require doctrine/dbal:^3.8`**
- Créer une entité nommée **`Usager`** par la commande : **`php bin/console make:user Usager`**
  - Préciser que c'est une entité persistante « Doctrine »
  - Préciser que la propriété **`email`** de cette entité sera celle utilisée plus tard pour l'authentification
  - Préciser que le mot de passe sera encrypté
- L'entité Usager a été créée avec les propriétés nécessaires à la « sécurité » de symfony (**`email`**, **`password`**, **`roles`**).
- On peut maintenant la compléter par la commande habituelle : **`php bin/console make:entity Usager`**
  - Rajouter les propriétés **`nom`** et **`prenom`**
- Répercuter ces changements sur la BD par les commandes habituelles :
  - **`php bin/console doctrine:migration:diff`**
  - **`php bin/console doctrine:migration:migrate`**

Pour pouvoir rapidement disposer d'un formulaire permettant de créer un nouvel utilisateur, on va se servir de la fonctionnalité CRUD de symfony :

- Créer l'interface CRUD pour l'entité **`Usager`** par la commande : **`php bin/console make:crud`**

Il ne reste plus qu'à adapter le code ainsi créé aux besoins de notre maquette :

- Dans le contrôleur **`src/Controller/UsagerController.php`** :
  - Ne conserver que les méthodes :
    - **`index`** (affichage de la page d'accueil de l'usager)
    - **`new`** (formulaire d'inscription d'un nouvel usager)
  - Modifier les annotations `#route` pour y ajouter la locale (*cf* TPs précédents)
  - Modifier la méthode **`index`** pour qu'elle transmette à son *template* l'entité **`Usager`** dont l'identifiant est égal à 1 *(c'est temporaire, en attendant le TP6)*
  - Modifier la méthode **`new`** pour qu'elle encrypte le mot de passe, à l'aide du service **`UserPasswordHasherInterface`**, avant de faire persister le nouvel Usager :

```php
if ($form->isSubmitted() && $form->isValid()) {
  // Encoder le mot de passe qui est en clair pour l'instant
  $hashedPassword = $passwordHasher->hashPassword($usager, $usager->getPassword());
  $usager->setPassword($hashedPassword);
  // Définir le rôle de l'usager qui va être créé
  $usager->setRoles(["ROLE_CLIENT"]);
  //…
}
```

  - Modifier la méthode **`new`** pour qu'à la fin elle redirige vers la route **`app_usager_index`**

- Dans le répertoire **`templates/usager`** :
  - Ne conserver que les templates `index.html.twig`, `new.html.twig` et `_form.html.twig`
  - Modifier le template `index.html.twig` pour qu'il affiche le `nom` et `prenom` de `l'Usager` qui lui sera transmis (s'il existe).
  - Supprimer les liens de navigation vers les pages du CRUD dans les templates `index.html.twig` et `new.html.twig`
- Dans le formulaire `src/Form/UsagerType.php` :
  - Modifier la méthode `buildForm` pour que seuls les champs `email`, `password`, `nom` et `prenom` figurent dans le formulaire qui sera soumis à l'utilisateur

Modifier enfin la barre de navigation pour proposer un menu déroulant Usager avec deux entrées « inscription » et « accueil » qui pointeront respectivement vers les routes `app_usager_new` et `app_usager_index`

5.2. Entités Commande et LigneCommande

- Créer ces entités à l'aide de la commande habituelle **`php bin/console make:entity`**
- L'entité **`Commande`** possède une relation **`ManyToOne`** vers l'entité **`Usager`**
- L'entité **`LigneCommande`** possède :
  - Une relation **`ManyToOne`** vers l'entité **`Commande`**
  - Une relation **`ManyToOne`** vers l'entité **`Produit`**
- Répercuter ces changements sur la BD par les commandes habituelles :
  - **`php bin/console doctrine:migration:diff`**
  - **`php bin/console doctrine:migration:migrate`**

5.3. Transformer un Panier en Commande

- Dans votre service **`src/Service/PanierService.php`** :
  - Ajouter une nouvelle méthode **`panierToCommande`** qui reçoit en paramètre une entité de type **`Usager`** et qui crée, pour cet usager, une commande (et ses lignes de commande) à partir du contenu du panier (s'il n'est pas vide).
  - Le contenu du panier devra être supprimé à l'issue de ce traitement.
  - Cette méthode renverra en résultat l'entité Commande qui aura été créée

```php
public function panierToCommande(Usager $usager) : ?Commande {
    // à compléter
}
```

- Créer une route **`app_panier_commander`** et une méthode **`commander`** dans le contrôleur **`PanierController`** :
  - Cette action devra bien sûr utiliser la méthode **`panierToCommande`** créée précédemment
  - Elle utilisera (*temporairement*) l'usager d'identifiant égal à 1 comme propriétaire de la commande
  - Elle se terminera par l'affichage d'un *template* **`commande.html.twig`** qui indiquera à l'utilisateur son prénom, son nom, le numéro et la date de la commande qu'il vient de passer