

DUBECQ John-Paul  
LE COAT Frédéric

# Mini-projet de développement client/serveur

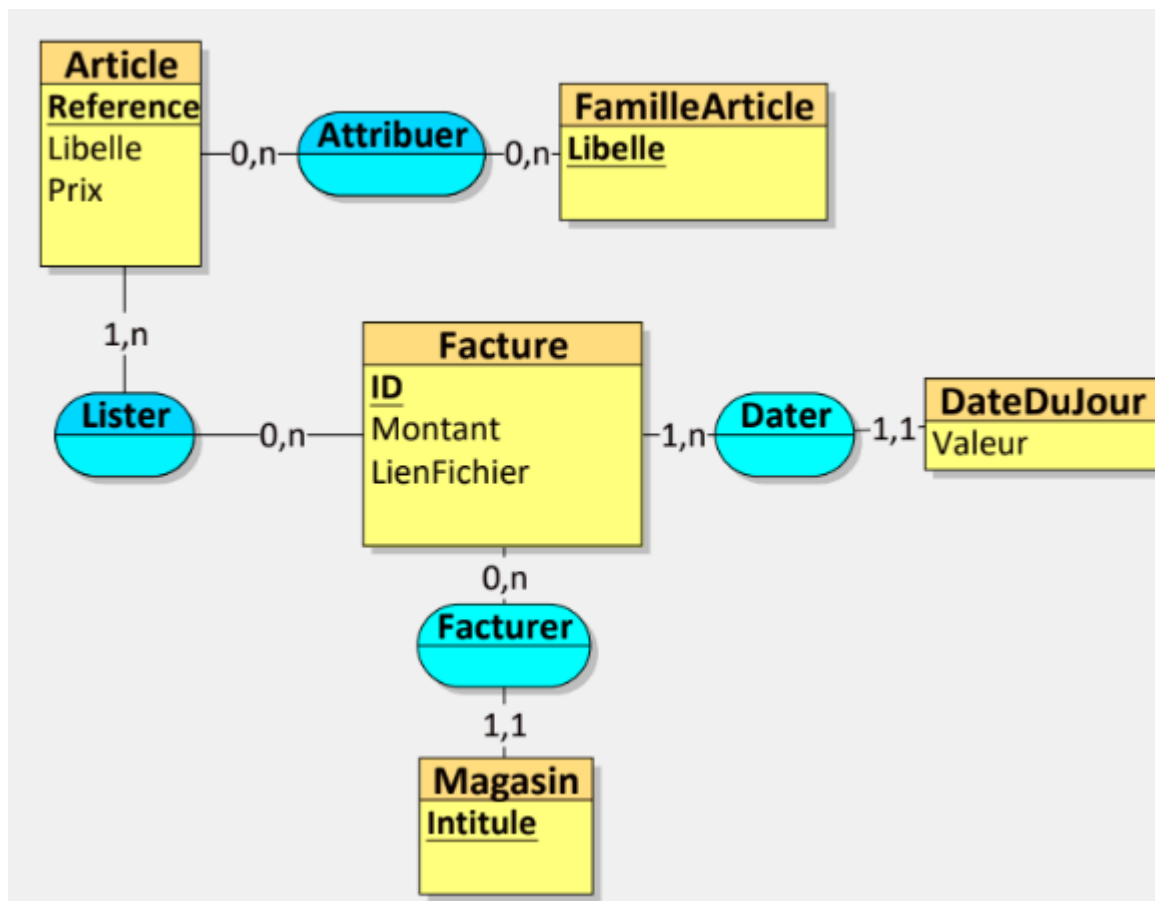
---

# Organisation du projet :

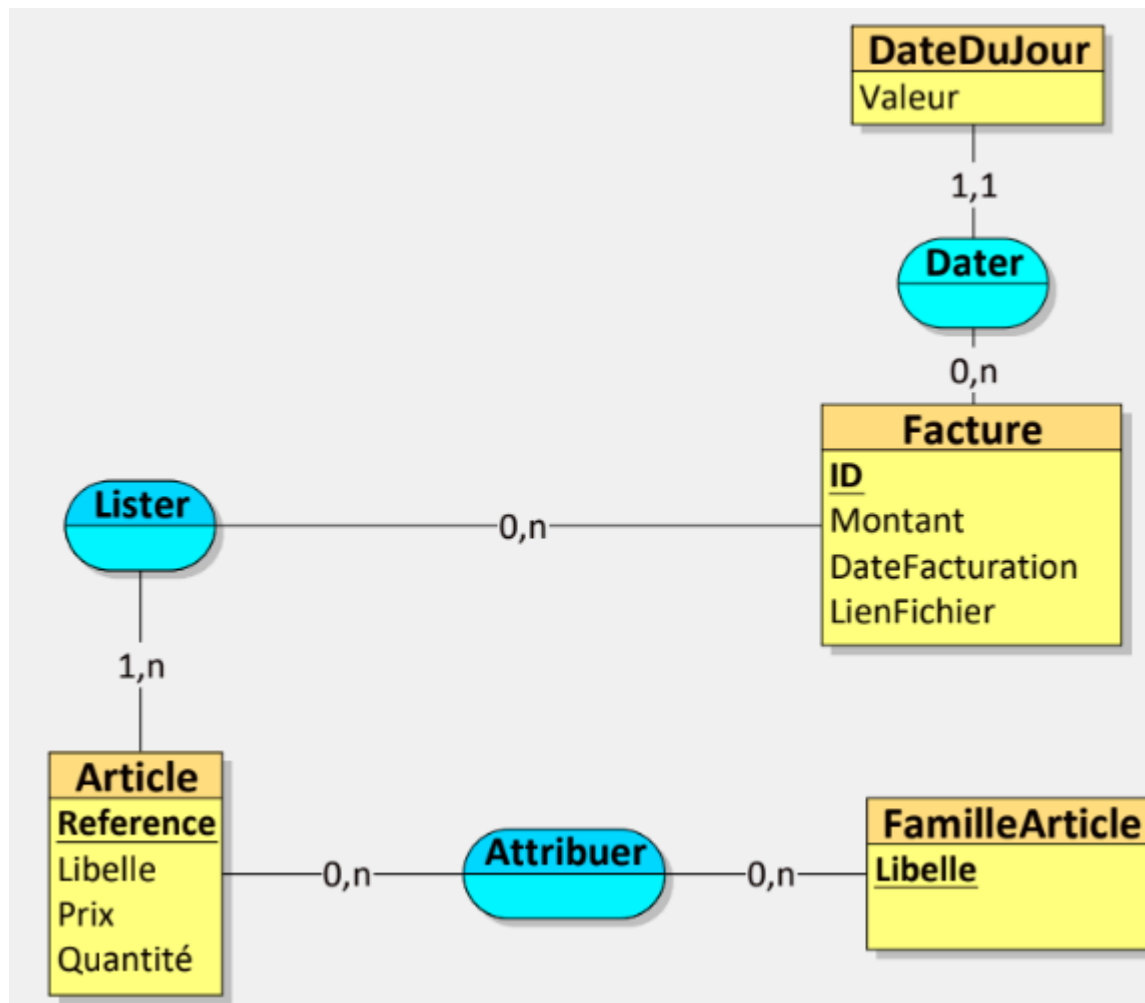
Pour l'organisation du projet au niveau de la propagation, nous avons utilisé GitHub et son système de branches pour avancer chacun de son côté sur les fonctionnalités et même au-delà de ça, nous avons utilisé GitHub afin d'organiser notre liste de tâches à réaliser.

Concernant l'organisation de nos bases de données, nous avons décidé deux organisations différentes pour les deux bases, car elles n'ont pas les mêmes besoins. Dans notre cas, le magasin n'a pas d'utilité à stocker la liste des magasins.

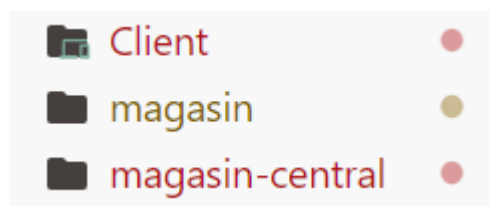
Voici le MCD du serveur du magasin central:



et ensuite, le MCD au niveau du magasin :



Au niveau de l'organisation du code, on a décidé de séparer les trois parties dans 3 dossiers différents :



# Lancement du projet :

Pour démarrer le projet, il faut ouvrir 3 terminaux et aller dans :

/Client

/magasin

/magasin-central

pour le terminal de /client, faire :

```
mvn clean package;java -jar .\target\client-1.0-SNAPSHOT.jar
```

pour /magasin :

```
mvn clean package;java -jar .\target\heptathlon-1.0-SNAPSHOT.jar
```

et pour /magasin-central :

```
mvn clean package;java -jar .\target\magasin-central-1.0-SNAPSHOT.jar
```

# Difficultés rencontrées :

Suite à notre décision de faire de la clean architecture, on a perdu beaucoup de temps, car c'était une première pour nous, donc il fallait faire beaucoup de recherche afin de réaliser ce projet en respectant cette clean architecture. Sauf qu'au final, avec le retard accumulé, on a dû arrêter toute cette clean architecture pour se focaliser uniquement sur les fonctionnalités demandées que sont les connexions RMI et les connexions aux dB.

# Prise de recul :

John-Paul : C'était une première pour moi d'avoir une application avec laquelle il fallait lancer les parties de manières séparé (en dehors des API dans les projets web). Donc c'était intéressant de voir comment c'était possible d'envoyer des informations via ce "canal" et de les utiliser de l'autre côté.

Frédéric : C'était une expérience intéressante qui nous a permis d'apprendre à manipuler les communications entre les clients et les serveurs. Malheureusement, nous avons rencontré des problèmes au niveau de la mise en place de RMI. Problèmes que nous avons réussi à résoudre mais qui nous ont fait perdre beaucoup de temps. L'architecture logicielle que nous avons utilisée était très fonctionnelle mais c'était la première qu'on la mettait en place. On a donc eu une phase d'apprentissage très élevée.

