

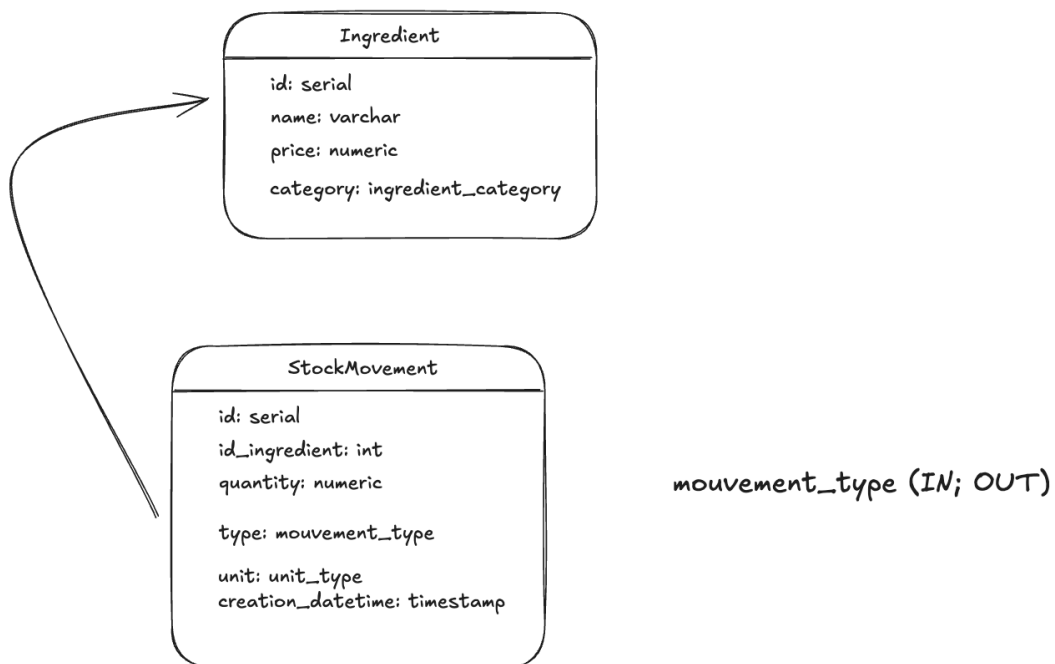
Exercice TD4 : Java & PostgreSQL (JDBC) - Gestion de stocks

Dans un contexte réel, il est essentiel pour un restaurant de pouvoir gérer les stocks de ses ingrédients. En particulier, il faut acheter les ingrédients pour augmenter le niveau de stock et à chaque vente réalisée, le stock diminue en fonction des quantités nécessaires et du nombre de plats vendus.

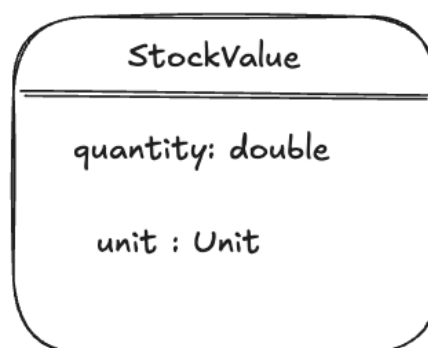
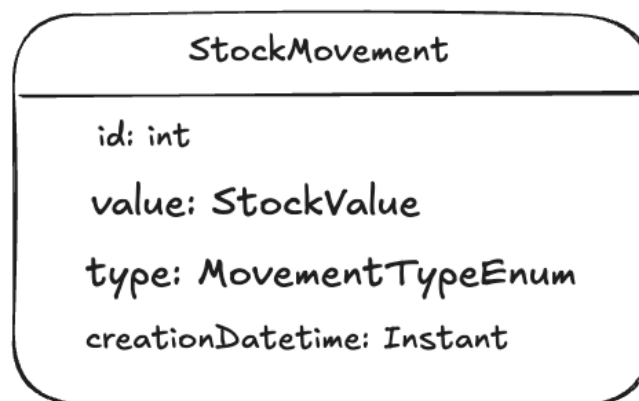
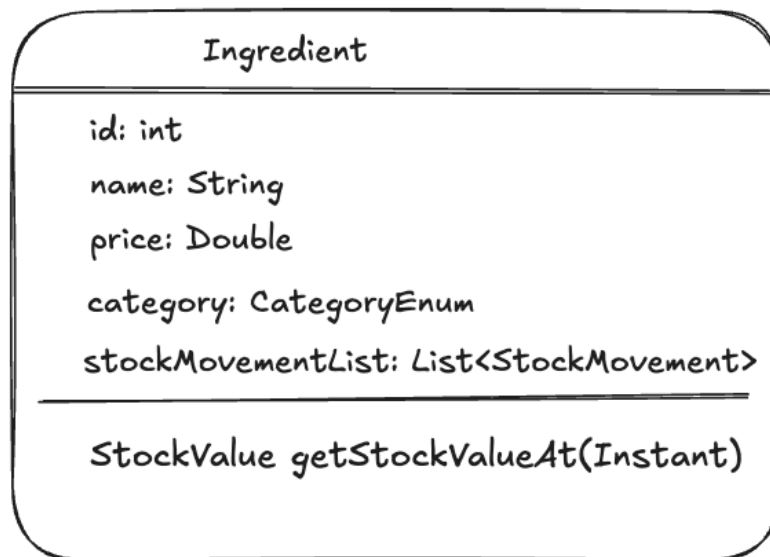
Dans cette étape, nous allons essentiellement considérer le mouvement des stocks pour le moment. Un mouvement de stock ou **StockMovement** est caractérisé par :

- Un identifiant unique;
- L'ingrédient correspondant au mouvement;
- La quantité du mouvement ainsi que l'unité correspondant;
- La date et heure du mouvement.

Voici les changements correspondants à l'ajout des mouvements des stocks au niveau de la base de données :



Au niveau applicatif, voici la nouvelle structure des classes attendues au niveau de Java :



- 1) Effectuez les modifications nécessaires au niveau de la base de données pour pouvoir gérer les stocks et ajoutez les données suivantes.

StockMovement

id	id_ingredient	quantity	type	unit	creation_datetime
1	1	5.0	IN	KG	2024-01-05 08:00
2	1	0.2	OUT	KG	2024-01-06 12:00
3	2	4.0	IN	KG	2024-01-05 08:00
4	2	0.15	OUT	KG	2024-01-06 12:00
5	3	10.0	IN	KG	2024-01-04 09:00
6	3	1.0	OUT	KG	2024-01-06 13:00
7	4	3.0	IN	KG	2024-01-05 10:00
8	4	0.3	OUT	KG	2024-01-06 14:00
9	5	2.5	IN	KG	2024-01-05 10:00
10	5	0.2	OUT	KG	2024-01-06 14:00

2) Effectuez les modifications nécessaires au niveau de la classe `DataRetriever` en suivant les consignes intermédiaires suivantes. Notez que nous ne considérons pas encore la conversion des unités, d'où toutes les unités sont en KG.

a) Créer la méthode `Ingredient saveIngredient(Ingredient toSave)` ;

b) Lorsqu'on veut ajouter un mouvement à un `Ingredient`, il faut lui attribuer à travers l'attribut `stockMovementList` et lors du sauvegarde (effectuée par la méthode `saveIngredient`). En particulier, il faut utiliser l'identifiant unique du mouvement de stock pour différencier la création et la mise à jour : si un identifiant est existant et qu'on essaie d'effectuer une insertion, alors il faut en rien faire (`on conflict do nothing` durant l'insertion) sinon, on ajoute le nouveau mouvement. Aussi, il n'est pas possible de supprimer ou d'enlever les mouvements de stock.

3) Utilisez les données de tests suivants pour les calculs du niveau de stock de chaque ingrédient.

Voici le niveau de stock initial à considérer :

Ingrédient	Stock
Laitue	5.0
Tomate	4.0
Poulet	10.0
Chocolat	3.0
Beurre	2.5

Vérifiez les résultats retournés par la méthode `getStockValueAt(Instant t)` pour tous les ingrédients en utilisant `t = 2024-01-06 12:00`, en ajoutant les mouvements de stocks :

id	id_ingredient	quantity	type	unit	creation_datetime
6	1	0.2	OUT	KG	2024-01-06 12:00
7	2	0.15	OUT	KG	2024-01-06 12:00
8	3	1.0	OUT	KG	2024-01-06 12:00
9	4	0.3	OUT	KG	2024-01-06 12:00
10	5	0.2	OUT	KG	2024-01-06 12:00

Le niveau de stock de chaque ingrédient (donc `Stock::getQuantity`) est :

Ingrédient	Calcul	Stock
Laitue	5.0 – 0.2	4.8
Tomate	4.0 – 0.15	3.85
Poulet	10.0 – 1.0	9.0
Chocolat	3.0 – 0.3	2.7
Beurre	2.5 – 0.2	2.3