

# **Rapport BDD**

BALLOUK Hussein  
ANDRIANANTENAINA Manoa

## Diagramme Entités/Associations

- le créateur de recettes est un client particulier, donc nous avons créé une table nouvelle "Créateur de recettes" qui est liée à la table "Client" mais également à la table "Recette" via une table "Point" qui représente les points Cook. Cette dernière table sert principalement à éviter une relation ternaire entre les tables "Client", "Créateur de recettes" et "Recette".

## Organisation de la base de données

- gestion des comptes et des mots de passe : lors de l'inscription d'une personne, nous stockons dans une base de données les informations renseignées, et particulièrement les identifiants et les mots de passe afin de gérer une éventuelle reconnexion. Cependant, nous ne protégeons pas (hash) les données sensibles telles que les mots de passe car il ne s'agit pas réellement d'une application entièrement fonctionnelle. Nous aurions pu faire une méthode si nous avions plus de temps.
- pour offrir la possibilité d'accéder à chaque ingrédient de la liste d'ingrédients, il faut écrire la liste en séparant les ingrédients par des points-virgules, comme selon la forme suivante : "farine;oeufs;sel;levure...".
- nous avons créé un backup de la base de données pour ne pas avoir à rentrer les données à chaque fois.
- après relecture du sujet, nous traitons la gestion des prix des recettes comme suit : le prix d'une recette est augmenté de 2 cooks quand elle dépasse 10 commandes (10 non inclus), et non chaque 10 commandes. De même pour 50 recettes commandées.
- à chaque connexion de l'admin lors du lancement de l'application, on met à jour la database.
- les adresses **doivent** être de la forme suivante : voie, code postal, ville

## Développements supplémentaires

- lors d'une inscription, nous offrons la possibilité de créer un compte Client ou Créateur de Recettes. Il faut alors renseigner un identifiant Client, et dans le cas d'un compte Créateur de Recettes, il faudra également renseigner un identifiant CdR.
- il est également possible de créer un compte Créateur de Recettes si on possède déjà un compte Client. On vérifiera alors que l'adresse mail du compte Client existe bien dans notre base de données Clients.

- lors de la création d'une nouvelle recette, le Créateur de Recettes a la possibilité d'utiliser des produits déjà existants, mais il peut aussi renseigner de nouveaux produits. Alors, à l'aide d'une description des fournisseurs disponibles, le Créateur de Recettes peut choisir quel sera le fournisseur.  
Aussi, si c'est un CdR qui renseigne le nouveau produit, le premier stock d'un nouveau produit sera de 50 fois la quantité demandée dans la recette, le stock minimum sera de 30 fois la quantité demandée et le stock maximum sera de 80 fois la quantité demandée.  
Si il s'agit d'un admin qui renseigne le nouveau produit, il doit indiquer le stock minimum du nouveau produit et nous calculons les stock actuel et stock maximum comme suit :  $\text{stock max} = \text{stock min} * 4$  ;  $\text{stock actuel} = (\text{stock max} + \text{stock min}) / 2$
- il y a deux possibilités de supprimer un compte : si un Client ou CdR le décide, on supprime dans l'ordre : les recettes s'il s'agit d'un Créateur de Recette, son compte CdR et son compte Client.  
Si c'est un admin qui supprimer le compte d'un CdR, il peut choisir de le laisser en tant que Client et supprimer son rang de CdR.

### Détails de l'application

- Lors du lancement de l'application, il faut entrer l'identifiant et le mot de passe qui ont été créés sur MySQL.
- pour vérifier les conditions suivantes :  
*“Le prix de vente de la recette augmente de 2 cook si (la commande en cours comprise) le nombre de commande dépasse 10” et “Le prix de vente de la recette augmente de 5 cook si (la commande en cours comprise) le nombre de commande dépasse 50 et dans ce cas la rémunération du CdR passe à 4 cook”.*  
 Nous avons mis à disposition les recettes R4 et R1 qui sont respectivement à 48 et 10 commandes
- pour vérifier le réapprovisionnement des produits, nous avons mis les stocks des produits lardon, sucre, lait, oeuf, gruyère râpé et basilic en dessous du stock minimum. Ainsi, la quantité commandée de ces produits doit être respectivement de : 2 759 g, 95 000 g et 32 L. Les stocks actuels des produits oeuf, gruyère râpé, basilic doivent être égaux aux stocks maximums.  
 Nous avons choisi dimanche comme étant le jour de réapprovisionnement des produits (fin de semaine) : ce jour est modifiable dans le code pour la revue de code de mardi si nécessaire.

Voici le code Reapprovisionnement.xml avant de lancer l'application, puis après l'avoir lancée, les stocks de gruyère râpé, oeuf et basilic sont mis à jour : la lecture du fichier .xml s'est bien déroulée.

```
<commande>
  <fournisseur>
    <idF>F200</idF>
    <produit>
      <nomP>oeuf</nomP>
      <quantiteCommander>400</quantiteCommander>
    </produit>

    <produit>
      <nomP>gruyère râpé</nomP>
      <quantiteCommander>8200</quantiteCommander>
    </produit>
  </fournisseur>

  <fournisseur>
    <idF>F1600</idF>
    <produit>
      <nomP>basilic</nomP>
      <quantiteCommander>6500</quantiteCommander>
    </produit>
  </fournisseur>
</commande>
```

Pour le réapprovisionnement de la semaine, le fichier XML sorti par C# (Reapprovisionnement.xml est écrasé) indique bien les produits lardon, sucre et lait à réapprovisionner.

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <commande>
3   <fournisseur>
4     <idF>F1100</idF>
5     <produit>
6       <nomP>lait</nomP>
7       <quantiteCommander>32</quantiteCommander>
8       <unite>L</unite>
9     </produit>
10  </fournisseur>
11
12  <fournisseur>
13    <idF>F200</idF>
14    <produit>
15      <nomP>lardon</nomP>
16      <quantiteCommander>2759</quantiteCommander>
17      <unite>g</unite>
18    </produit>
19
20    <produit>
21      <nomP>sucré</nomP>
22      <quantiteCommander>95000</quantiteCommander>
23      <unite>g</unite>
24    </produit>
25  </fournisseur>
26 </commande>
27
```

Dans le fichier bin/Debug, on retrouve les fichiers Reapprovisionnement\_Debut\_BackUp.xml (fichier avant de lancer l'application) et Reapprovisionnement\_Resultat\_BackUp.xml (fichier après avoir lancé l'application).

Avant de lancer l'application, on a ces stocks pour les produits en question :

	nomP	categorieP	unite	stockActuel	stockMin	stockMax	idF
▶	basilic	légume	g	500	1200	7000	F1600
	gruyère râpé	fromage	g	1800	2000	10000	F200
	lait	boisson	L	8	10	40	F1100
	lardon	aliment	g	241	400	3000	F200
	oeuf	aliment	unite	100	150	500	F200
	sucre	sucre	g	5000	10000	100000	F200
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Après avoir lu le fichier, voici ce qu'on obtient :

	nomP	categorieP	unite	stockActuel	stockMin	stockMax	idF
▶	basilic	légume	g	7000	1200	7000	F1600
	gruyère râpé	fromage	g	10000	2000	10000	F200
	oeuf	aliment	unite	500	150	500	F200
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Voici les stocks qui doivent être enregistrés dans le nouveau fichier .xml (vérifié) :

	nomP	categorieP	unite	stockActuel	stockMin	stockMax	idF
▶	lait	boisson	L	8	10	40	F1100
	lardon	aliment	g	241	400	3000	F200
	sucre	sucre	g	5000	10000	100000	F200
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- Pour se connecter en tant qu'admin, il faut renseigner les id et mdp de MySQL sur la page connexion