



Projet BDR

“Just Brew It”

Gestionnaire de recettes de brassage interactif

Auteurs :

Thomas Germano
Timothée Van Hove

1	Introduction.....	3
2	Phase 1 : Cahier des charges	3
2.1	Description des fonctionnalités	3
2.1.1	Gestion des recettes	3
2.1.2	Création de recette	3
2.1.3	Brassage grâce aux recettes	3
2.2	Description des données	3
2.2.1	Utilisateur	3
2.2.2	Bière	3
2.2.3	Recette	4
2.2.4	Ingrédient	4
2.2.5	Étape.....	4
2.3	Étapes de fabrication (catégories)	5
3	Phase 2 : Modélisation conceptuelle	6
4	Phase 3 modèle relationnel	7
4.1	Choix concernant le modèle.....	8
5	Phase 4 : Requêtes, vues et insertion des données	9
5.1	Vues.....	9
5.2	Requêtes.....	10
5.3	Insertion des données.....	10
6	Phase 5 : Création de l'application client	11
6.1	Lancer l'infrastructure	11
6.2	Structure de l'application web	11
7	Conclusion	12
8	Bilans personnels	12
8.1	Timothée Van Hove	12
8.2	Thomas Germano	12

1 Introduction

Nous souhaitons créer une application web interactive qui permet aux utilisateurs de brasser de la bière en suivant les recettes en temps réel. La difficulté du brassage de la bière réside dans le fait qu'il faille avoir en tête une dizaine d'étapes successives tout en surveillant la température de brassage durant plusieurs heures. Chaque recette de bière peut être très différente et peut varier énormément de l'une à l'autre.

2 Phase 1 : Cahier des charges

2.1 Description des fonctionnalités

2.1.1 Gestion des recettes

Chaque utilisateur peut créer ses propres recettes sur l'application. En sélectionnant une recette, l'utilisateur peut placer automatiquement les ingrédients dans un panier pour les commander directement chez lui. La gestion du stock et de la disponibilité des ingrédients dans le shop n'est pas gérée par l'application.

2.1.2 Création de recette

L'utilisateur peut ajouter les ingrédients nécessaires à la recette. C'est lui qui crée les étapes de brassage en spécifiant les durées, températures et techniques de brassage.

2.1.3 Brassage grâce aux recettes

Le brassage de la bière est effectué en une dizaine d'étapes successives. Lorsqu'une recette est sélectionnée l'utilisateur peut lancer un chronomètre pour garder la notion du temps lors du brassage. Lors de chaque étape, l'application indique à l'utilisateur la marche à suivre sous forme de texte, ainsi que les ingrédients nécessaires. A la fin de chaque étape le chronomètre s'arrête. L'utilisateur peut alors passer à la prochaine étape.

2.2 Description des données

2.2.1 Utilisateur

Un utilisateur possède :

- Un nom, prénom
- Une adresse, et un titre (M. Mme)
- Une adresse mail
- Un mot de passe
- Des recettes de brassage

2.2.2 Bière

Chaque bière possède :

- Un nom, un ID
- Couleur : Il s'agit de la couleur de la bière exprimée en EBC
- Amertume : Il s'agit de l'amertume de la bière exprimée en IBU
- Alcool : La quantité d'alcool dans la bière exprimé degré alcoolique

2.2.3 Recette

Chaque recette possède un nom. Une recette est basée au minimum sur les ingrédients suivants, avec leurs attributs :

- Houblon
 - Type (Amérisant / Aromatique)
 - Taux d'acide alpha (%) (taux haut et taux bas)
 - Houblon de substitution
- Levure
 - Type de bière associé
 - Type de fermentation
 - Température de fermentation max
 - Température de fermentation min
- Malt
 - EBC (min-max)
 - Type spécifique
 - Type de céréale (Orge/Blé/Seigle/épeautre/maïs)

Des ingrédients supplémentaires peuvent être ajoutés. Dans la recette est aussi présente la quantité de bière voulue. La quantité des ingrédients varie en fonction de la quantité de bière voulue. Une recette contient plusieurs étapes. Chaque étape de fabrication est très variée en fonction du type de bière souhaité. Une recette produit une bière.

2.2.4 Ingrédients

Un ingrédient est très abstrait, car il peut avoir n'importe quel attribut en fonction de sa nature. En revanche, tous les ingrédients possèdent au minimum :

- Un nom
- Une catégorie
- Une quantité
- Une provenance (et sous-provenance)
- Description (texte)
- Unité de quantité
- Prix/unité

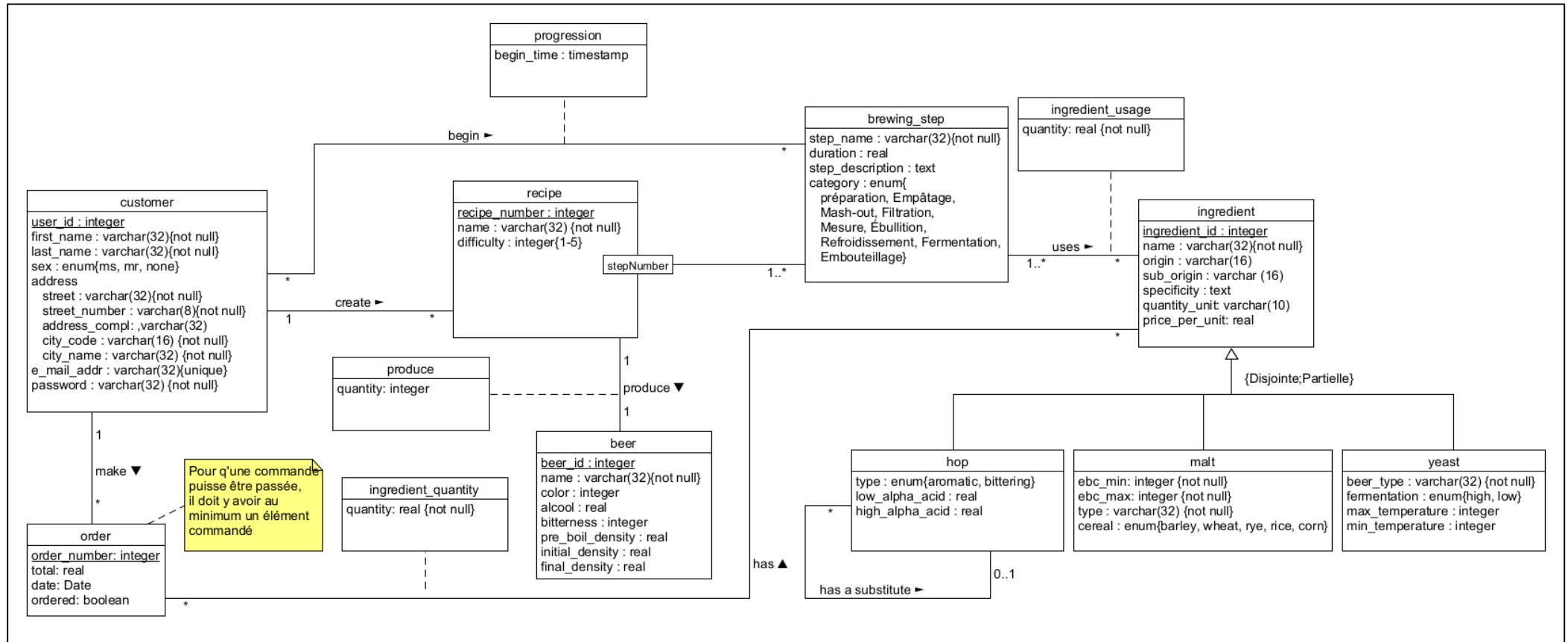
2.2.5 Étape

Une étape est identifiée par un numéro. Elle fait partie d'une catégorie (voir chapitre suivant). Elle peut contenir des ingrédients et contient une durée. Une étape peut être chronométrée ou non. Finalement, une étape peut contenir des notes spécifiques aux techniques de brassages à respecter (paliers de température, techniques de refroidissement, etc...).

2.3 Étapes de fabrication (catégories)

1. La préparation : Il s'agit de préparer tout le matériel nécessaire, désinfecter l'environnement de travail, et peser la quantité de chaque ingrédient nécessaire.
2. L'empâtage : Il faut verser le malt dans de l'eau chauffée à une certaine température, puis brasser le tout à plusieurs paliers de température très précis pendant environ 1 heure. Le but est de dissoudre dans l'eau tous les sucres contenus dans le malt.
3. Mash-out : Augmentation de la température de la maische à 75°C afin de stopper la saccharification, en détruisant les enzymes contenues dans le malt.
4. Filtration & rinçage des drêches : La séparation du malt dont nous avons extrait les sucres avec une grande passoire ou un tissu maillé. Le but étant de ne récupérer que le liquide (moût).
5. Mesure de la densité : Dès la fin de la filtration des drêches, il faut mesurer la densité du moût pour savoir s'il faut corriger la densité, mais aussi pour connaître la teneur finale en alcool de la bière.
6. Ebullition du moût : Une fois le liquide récupéré, il faut le faire bouillir entre 40 et 60 minutes, tout en brassant et surveillant la température.
7. Refroidissement et oxygénation du moût : Le but est de refroidir très rapidement le moût à environs 20°C dans un environnement stérile, de le transvaser dans une cuve de fermentation et d'y ajouter les levures qui transformeront les sucres et l'oxygène en alcool et CO₂.
8. Fermentation : Une fois le moût dans la cuve de fermentation, il faut laisser les levures transformer les sucres pendant une durée qui varie entre 3 et 5 jours.
9. Garde : Il s'agit d'une fermentation secondaire. Cette étape optionnelle nécessite de refroidir la bière entre 0 et 5°C, ce qui permet de la raffiner en décantant les levures, et permet de la clarifier en laissant les particules se déposer dans le fond de la cuve de fermentation.
10. Mise en bouteille : Il s'agit de la dernière étape de la fabrication, ajouter du sucre dans la bière, puis mettre le tout en bouteille. La bière sera prête entre 3 semaines et 1 mois après la mise en bouteille.

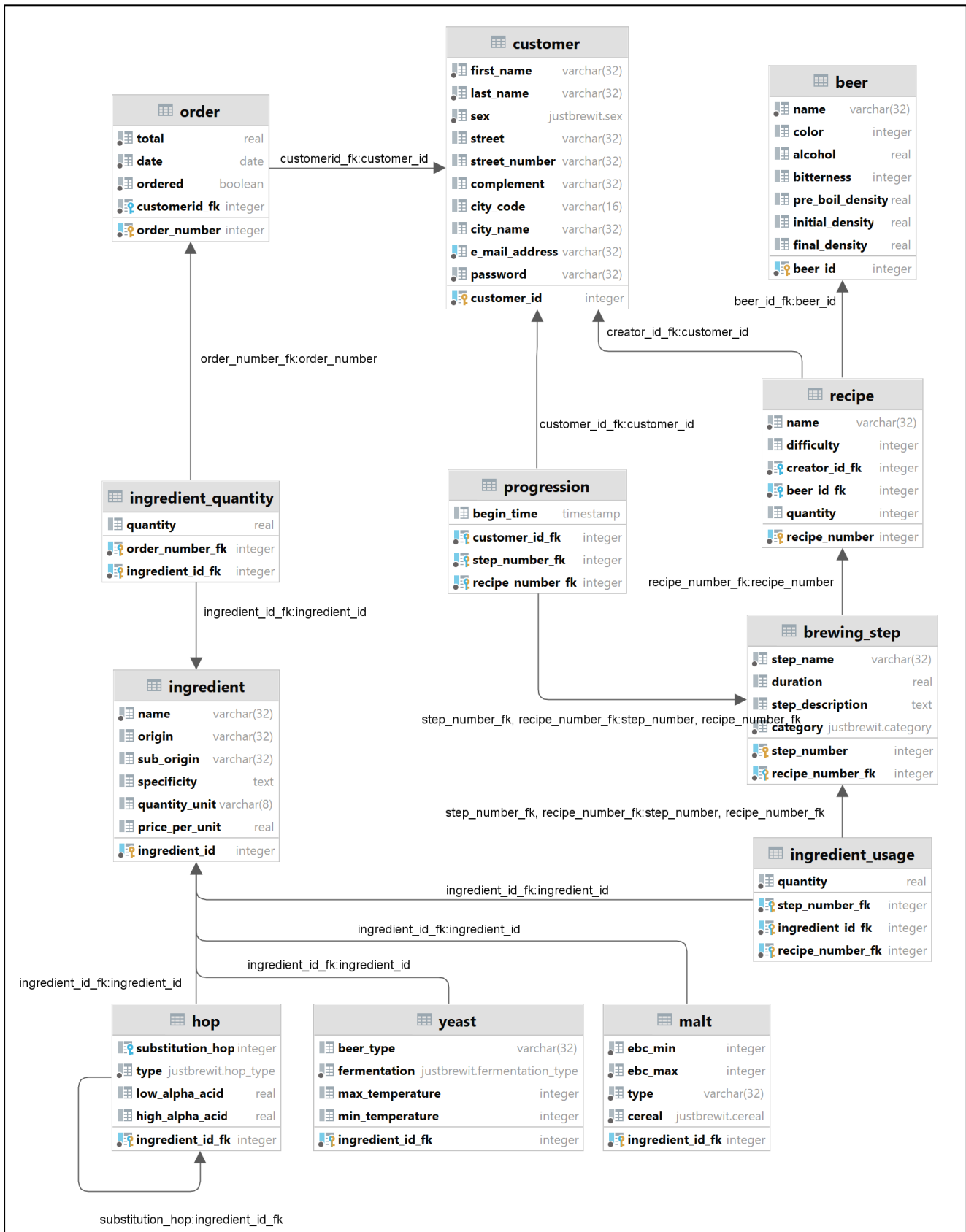
3 Phase 2 : Modélisation conceptuelle



Durant cette Phase, nous avons été amenés à concevoir le schéma conceptuel de notre base de données (voir ci-dessus). Ce schéma UML reprend représente la description des données effectuée à la phase précédente, mais ajoute aussi toutes les dépendances existantes entre chaque entité. Nous nous sommes concentrés sur une modélisation simple de notre idée d'application.

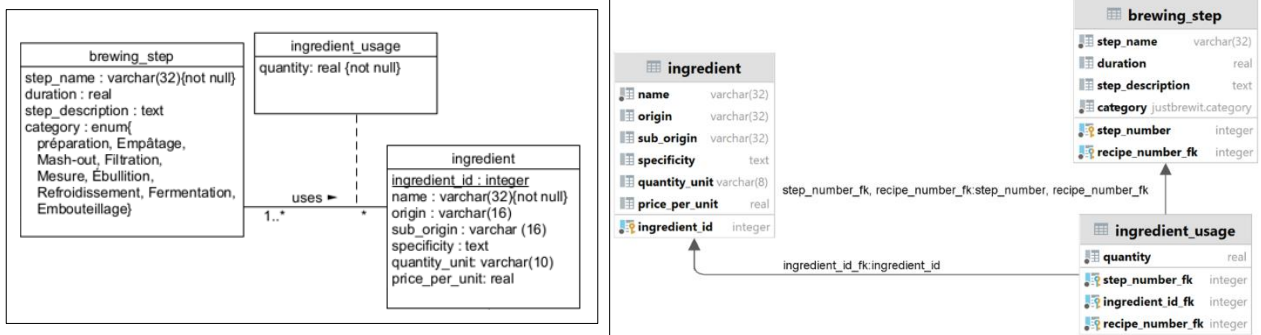
Durant cette étape de modélisation

4 Phase 3 modèle relationnel

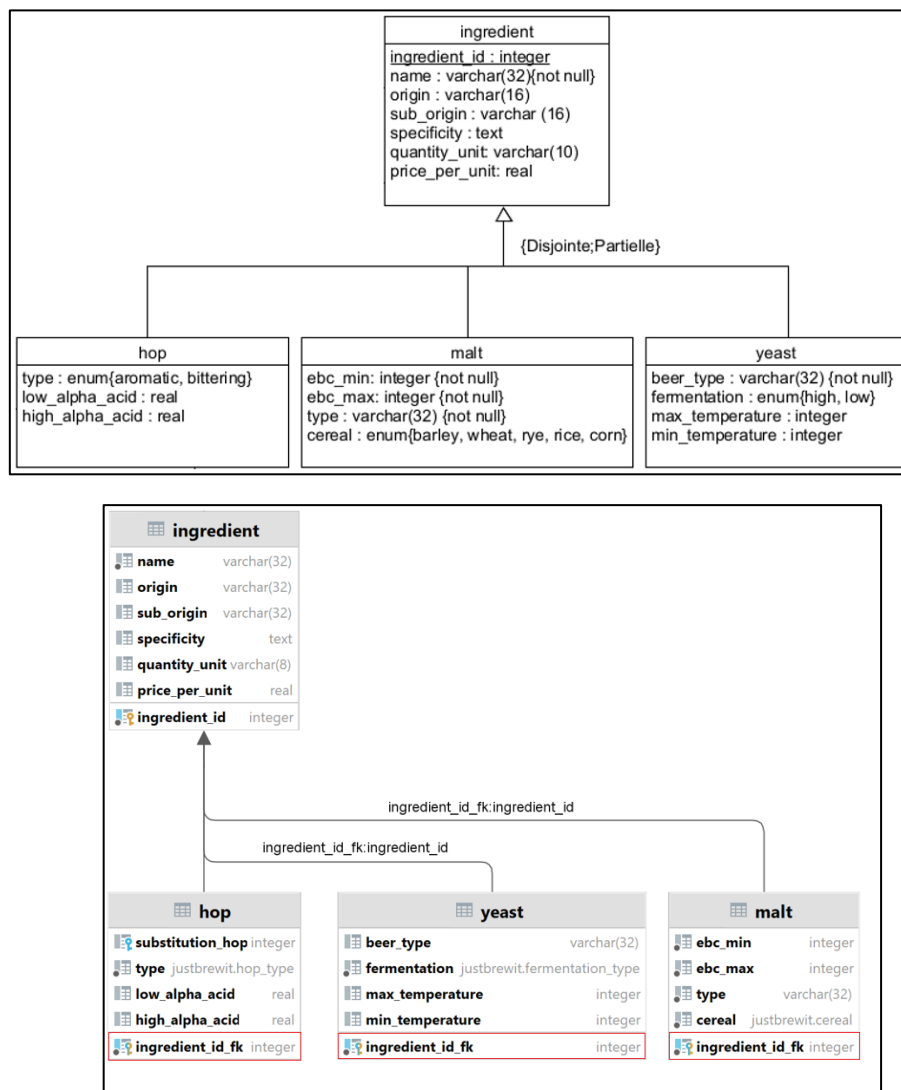


4.1 Choix concernant le modèle

Afin de réaliser ce modèle, nous nous sommes basés sur les techniques de passage d'un modèle conceptuel vers un modèle relationnel vues en cours. Dans notre modèle conceptuel, plusieurs associations ont des multiplicités many-to-many, ce qui nous a conduit à créer des tables intermédiaires. Par exemple, l'association entre les entités « Ingredient » et « brewing_step » nous a conduit à créer la table « ingredient_usage ».



Pour gérer l'héritage, nous avons choisi de stocker dans chacune des entités héritant, la clé étrangère « ingredient_id_fk » référençant l'entité parent.



5 Phase 4 : Requêtes, vues et insertion des données

Une fois que notre modèle est créé, nous avons créé toutes les requêtes, vues et triggers qui nous seront utiles à l'application web. Notre idée est de mettre à disposition toutes les fonctions lui étant nécessaire. Cela évite de devoir créer des requêtes custom, ou compliquées dans le code de notre application

5.1 Vues

Pour simplifier la création des fonctions, nous avons décidé de créer des vues spécifiques aux données que notre application doit traiter.

🔍 yeastfromrecipes	
📄 recipe_number	integer
📄 step_name	varchar(32)
📄 name	varchar(32)
📄 quantity	real
📄 quantity_unit	varchar(8)
📄 specificity	text
📄 origin	varchar(32)
📄 ingredient_id	integer
📄 beer_type	varchar(32)
📄 fermentation	justbrewit.fermentation_type
📄 min_temperature	integer
📄 max_temperature	integer

🔍 maltsfromrecipes	
📄 recipe_number	integer
📄 step_name	varchar(32)
📄 name	varchar(32)
📄 quantity	real
📄 quantity_unit	varchar(8)
📄 specificity	text
📄 origin	varchar(32)
📄 ingredient_id	integer
📄 ebc_min	integer
📄 ebc_max	integer
📄 type	varchar(32)
📄 cereal	justbrewit.cereal

🔍 hopsfromrecipes	
📄 recipe_number	integer
📄 step_name	varchar(32)
📄 name	varchar(32)
📄 quantity	real
📄 quantity_unit	varchar(8)
📄 specificity	text
📄 origin	varchar(32)
📄 ingredient_id	integer
📄 type	justbrewit.hop_type
📄 low_alpha_acid	real
📄 high_alpha_acid	real

🔍 miscingredientsfromrecipes	
📄 recipe_number	integer
📄 step_name	varchar(32)
📄 name	varchar(32)
📄 quantity	real
📄 quantity_unit	varchar(8)
📄 price_per_unit	real
📄 specificity	text
📄 origin	varchar(32)
📄 ingredient_id	integer

🔍 ingredientsfromrecipes	
📄 recipe_number	integer
📄 step_name	varchar(32)
📄 name	varchar(32)
📄 quantity	real
📄 quantity_unit	varchar(8)
📄 price_per_unit	real
📄 specificity	text
📄 origin	varchar(32)
📄 ingredient_id	integer

🔍 ordersfromcustomers	
📄 customer_id	integer
📄 order_number	integer
📄 ordered	boolean
📄 name	varchar(32)
📄 quantity	real
📄 quantity_unit	varchar(8)
📄 price_per_unit	real

🔍 stepsfromrecipe	
📄 recipe_number	integer
📄 step_number	integer
📄 duration	real
📄 step_name	varchar(32)
📄 step_description	text
📄 category	justbrewit.category

🔍 beersfromrecipes	
📄 recipe_number	integer
📄 name	varchar(32)
📄 color	integer
📄 alcohol	real
📄 bitterness	integer

🔍 recipesfromcustomers	
📄 customer_id	integer
📄 recipe_number	integer

Nous avons donc créé 5 vues permettant de chercher des ingrédients spécifiques. Par exemple, la vue `miscIngredientsFromRecipe` affiche uniquement les ingrédients qui ne sont ni des houblons, ni des malts, ni des levures. La vue `ingredientsFromRecipe` affiche tous les ingrédients des recettes.

5.2 Requêtes

Nous avons créé une trentaine de fonctions, toutes répondant au besoin de lecture ou d'écriture de notre application dans la base de données. Par exemple, nous avons une fonction permettant de récupérer le temps restant à une étape en cours :

```
CREATE OR REPLACE FUNCTION getRemainingTime
(
    recipeId integer,
    stepId integer,
    customerId integer
)
RETURNS real
language plpgsql
AS
$$
DECLARE
    time real;
    timeStamp timestamp;
    dur real;
BEGIN
    SELECT begin_time into timeStamp
    FROM progression
    WHERE recipe_number_fk = recipeId
    AND step_number_fk = stepId
    AND customer_id_fk = customerId;
    SELECT duration into dur
    FROM brewing_step
    WHERE recipe_number_fk = recipeId
    AND step_number = stepId;
    time = (EXTRACT(EPOCH FROM (now() - timeStamp)) / 60);
    RETURN dur - time;
END;
$$;
```

5.3 Insertion des données

Pour l'insertion des données, nous avons ajouté à la base de données une trentaine de malts et houblons, une dizaine de levures, 3 utilisateurs, 4 bières et 4 recettes de bière avec 9 étapes chacune.

6 Phase 5 : Création de l'application client

Notre application web est déployée sur un serveur PHP et une base de données Postgres, tous deux dockerisés. Le fichier docker-compose contient la configuration du déploiement des images de ces deux services.

6.1 Lancer l'infrastructure

Il suffit d'exécuter la commande `docker-compose up --build` dans le terminal pour lancer les containers de l'infrastructure. Une fois cela fait, il est possible d'accéder à l'application web à l'adresse localhost dans n'importe quel browser.

Lorsque l'image docker est créée, la SGBD utilise le fichier `init.sql` pour s'initialiser. Ce fichier contient le schéma, ainsi que toutes les vues, fonctions, triggers et données d'initialisation.

6.2 Structure de l'application web

Depuis la page d'accueil (`index.php`), il faut se connecter avec un utilisateur et un mot de passe valide pour pouvoir accéder aux autres pages. Pour tester l'application, il est possible de se connecter avec l'utilisateur « admin » et le mot de passe « admin ».

Depuis la page « recipes », il est possible de consulter ou ajouter des nouvelles recettes. En cliquant sur une recette, vous vous dirigez vers la page « recipe » où sont affichées toutes les informations de la recette : Bière obtenue, ingrédients et étapes de brassage. En bas de la page, le bouton « Commencer le brassage » vous dirige vers la page « brewingStep » où il est possible de consulter toutes les étapes du brassage.

Pour ajouter une recette, il suffit de remplir le formulaire en cliquant sur le bouton « ajouter une recette », depuis la page « recipes ».

7 Conclusion

Notre application web est déployée sur un serveur PHP et une base de données, tous deux Dockerisés. Nous pouvons accéder à notre application, la gestion des utilisateurs et des sessions est fonctionnelle, nous pouvons créer des recettes, consulter les recettes existantes et brasser une bière en suivant toutes les étapes contenues dans les recettes. Certaines fonctionnalités n'ont malheureusement pas pu être déployées. Il s'agit de la gestion du web shop, ainsi que de la gestion des paniers et commandes des utilisateurs. Nous avons préféré nous focaliser sur les fonctionnalités les plus importantes.

Nous sommes très satisfaits de notre application et estimons que les objectifs de ce projet sont atteints.

8 Bilans personnels

Ce projet nous a permis de mettre en pratique toute la théorie vue en cours, au fur et à mesure des étapes. La phase finale du projet, à savoir la création de l'application web a clairement été la plus compliquée. Nous n'avons pas réussi à implémenter toutes les fonctionnalités imaginées au début du projet, mais notre base de données contient les fonctions nécessaires à ces fonctionnalités, si un jour nous voudrions continuer ce projet. Étant un groupe de deux personnes, le panel de compétences est moins étendu. Nous avons énormément appris durant les différentes phases de ce travail, notamment la dernière avec la mise en place de l'application web.

8.1 Timothée Van Hove

Je n'avais jamais codé en SQL et PHP de ma vie avant ce cours. C'est une belle opportunité de mettre en pratique un projet concret de A à Z. Le temps était clairement notre plus grand challenge, car les 3 semaines suivant les vacances d'hiver ont été particulièrement chargées de gros projets et tests. J'ai appris le PHP en une semaine, mais suis très fier du résultat. Je remercie Mme. Nastaran et M. Meier pour nous avoir guidé tout au long du projet.

8.2 Thomas Germano

Durant mon CFC, j'avais déjà un peu utilisé des bases de données, mais jamais dans un site web comme dans le cas de ce projet. Concernant le PHP/Javascript, la seule fois que j'y avais touché de ma vie était pour développer une extension WordPress. Tout ça pour dire que tous ces langages étaient nouveaux pour moi et qu'il m'a fallu les apprendre sur le tas pour le projet. Sur le tas car j'ai été un peu pris de court par la charge de travail que j'aurais dû mieux répartir dans le mois dernier. Mais ce fût tout de même une expérience intéressante et qui m'a un peu réconcilié avec le web, ayant juste fait jusqu'ici un site web statique en HTML/CSS. Le résultat final tient la route mais bénéficierait d'un peu plus de travail que nous n'avons malheureusement pas pu fournir. Tous mes remerciements à l'enseignant et l'assistant pour le suivi tout le long du projet.