

* **Jean-François Lhomme**
* **Alexis Hoyez**
* **Dimitri Libessart**
* **Valentin Naessens**

**Projet**

**Base**

**De**

**Données**

**CONTEXTE :**

**Utilisation de la base de données afin de pouvoir établir des commandes**

**Onguent fabriquer à partir des potions sans diluant ni chauffer**

**Vente de potion en pots /fioles/tube -> prix diffèrent**

**Un a plusieurs fournisseurs par ingrédient**

**Fraicheur en jour et température**

**Efficacité dépendant de la fraicheur sur les recettes**

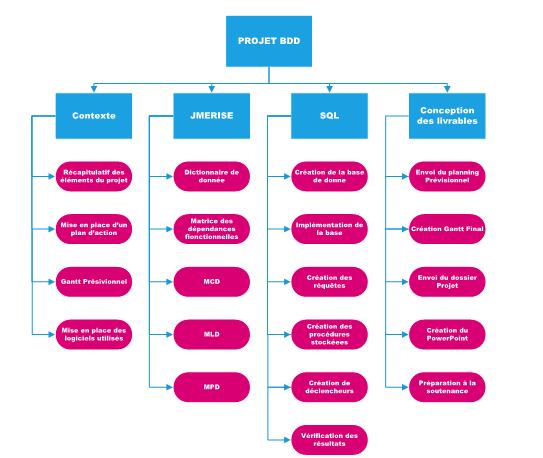
**Suppression des types d’ingrédient ainsi que des types de magie**

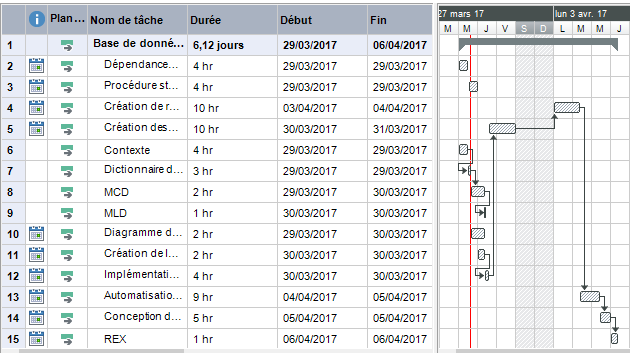
**Possibilité de commande d’un même ingrédient a différente fraicheur**

1. **Dictionnaire de données**
2. **Diagramme des flux**
3. **Matrice de dépendance fonctionnel**
4. **Mcd**
5. **Mld**
6. **Mpd**
7. **Insertion donnée**
8. **Requête**
9. **Procédure stockes**

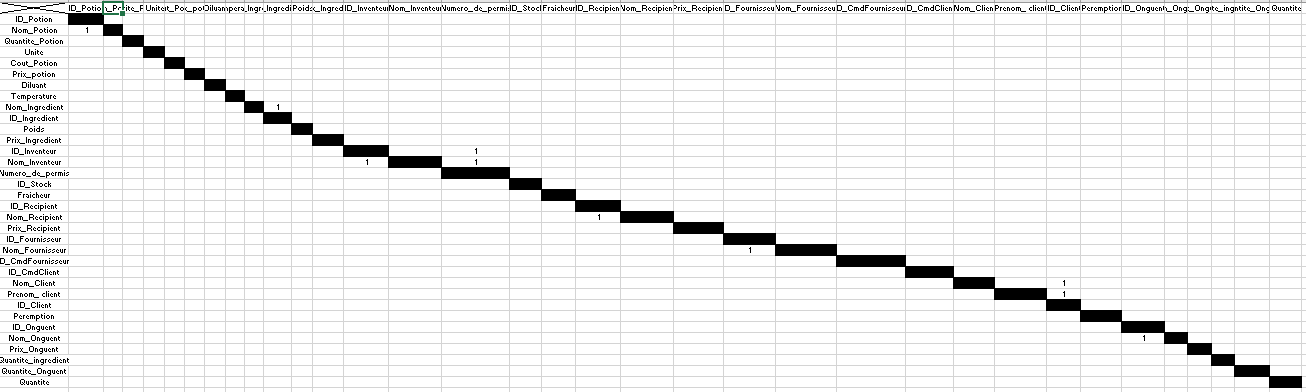
**Pour réaliser tout cela nous allons devoir garder des éléments de la base de données : les tables potion, ingrédient, inventeur et stock**

**En modifiant la base de données pour ajouter certains champs en enlever d’autres ainsi que créer d’autres tables.**

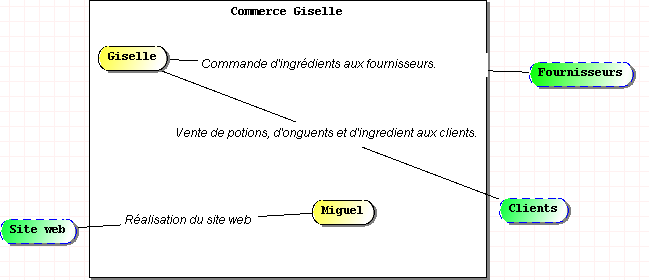
**Pour répondre au mieux aux besoins nous élaborons un plan d’action prévisionnel dans lequel nous retrouvons les taches important que nous avons trouver **

****

**La matrice de dépendance fonctionnel nous connaissons les parties indispensables à la réalisation**

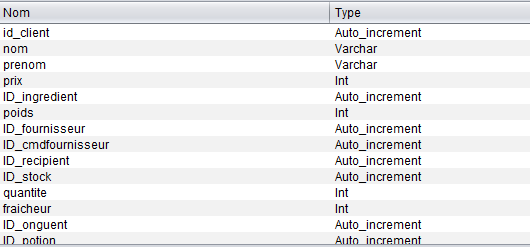
****

**Le diagramme des flux permet la visualisation des interactions entre les diffèrent acteurs : les fournisseurs, les clients, Giselle et Miguel**

****

**JMerise :**

**Avant de pouvoir établir un mcd il nous faut bien évidemment établir un dictionnaire de données pour une fois réalisé importer dans une entité le nom et le style de données quelle posséderas permettant de définir les valeurs dans les futurs champs**

****

**Nous utilisons quatre types de données :**

* **Int**

**Qui sont des entiers signés positivement ou négativement, cette valeur étant comprise pour 32 bits entre -2 147 483 648 à 2 147 483 647.**

* **Auto-incrémente**

**Permet de données la valeur sans avoir besoin de la saisir ces mêmes valeurs nous serviront de clé primaire pour communiquer entre nos tables**

* **Varchar**

**Pour ne pas allouer trop de mémoire à des éléments n’en ayant pas forcément besoin et de limiter la taille maximale utilisable. Contrairement au char qui laisse la place en mémoire pour l’ensemble des caractères possible**

**Le varchar alloue donc de la mémoire dynamiquement**

* **Bool**

**Valeur booléenne qui permet de donner deux états : vrai ou faux, ou encore 0 ou 1. Cela permet dans nos cas de dire si une recette est validée ou non et de savoir si elle attend d’être accepter ou même refuser**

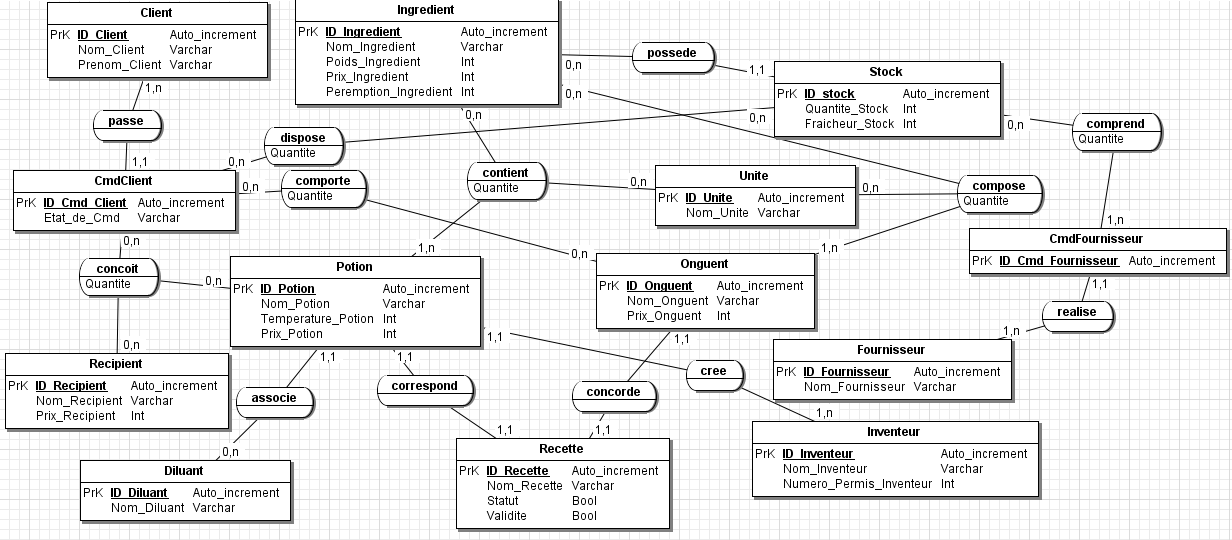
**Nous commençons donc à réaliser le MCD a l’aide de notre dictionnaire de données**

**Création des différentes tables, et remplissage des tables avec leurs noms de champs et le type de la valeur puis vient l’étape des associations entre les tables**

**Avec les cardinalités adéquate.**

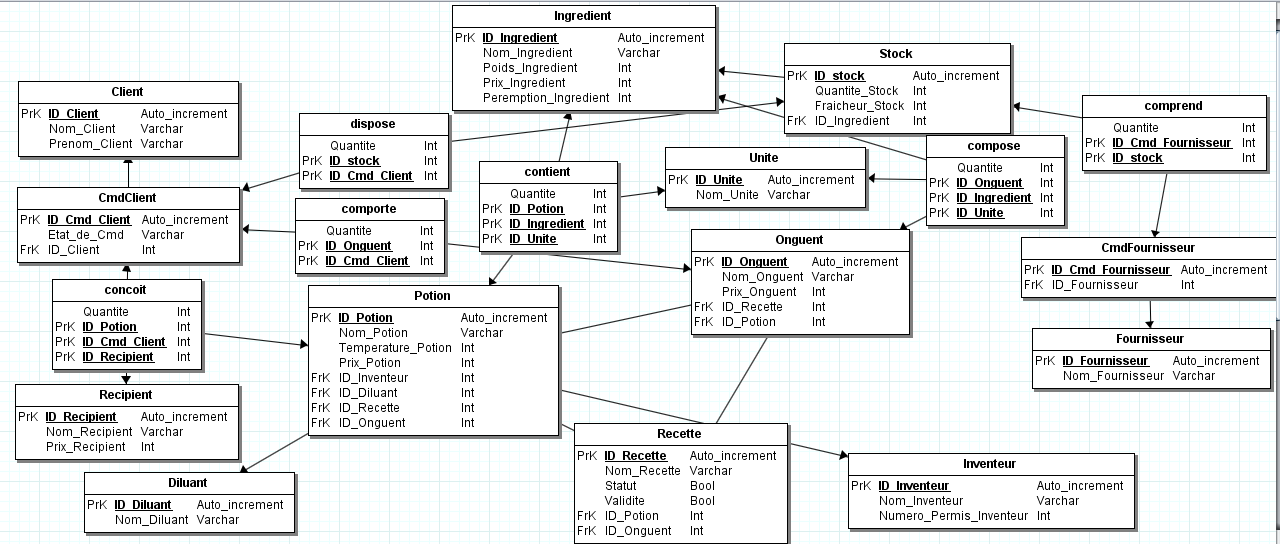
**Nous utilisons 3 relations ternaires dans notre mcd relation ente trois table différentes**

**Voici le schéma de notre mcd :**

****

**La table centrale de notre MCD est la table potion c’est elle qui possède le plus d’association.**

**Grace aux logiciel JMerise qui nous permet de transformer notre MCD en MLD**

****

**Ce MPD issue du MCD précédemment montré nous permet de connaitre l’ensemble des éléments de chaque table cela nous permettra d’établir un ordre lors de la création du SQL car les dépendances des relations entre deux tables fonctionnent dans un sens ce qui explique pourquoi nous ne pouvons pas créer nos tables de l’ordre que nous le désirons la table CmdFournisseur par exemple a besoin que la table Fournisseur existe pour récupérer des informations de celle si**

**-pour notre choix de système de gestion de base de données relationnel. Nous avons choisi d’utiliser MySQL car il est rapide, il tourne sous divers systèmes d’exploitation (Windows, Linux, Unix ...).**

**Il est aussi très simple d’utilisation. Toute fois sous Windows il faut une licence mais sur les autres systèmes, hors but commercial, celle-ci est gratuite et il permet d’utiliser plusieurs langages C, PERL, C++, Java, Python et PHP.**

**On peut gérer ses accès à différents utilisateurs et les stocker sur un serveur. Son interface est pratique.**

**SQL :**

**Les tables Client, Fournisseur, Inventeur, Recette, Diluant et Recipient doivent être créé et remplie avant les autres.**

**Puis nous pouvons créer les tables CmdClient, CmdFournisseur.**

**La table Stock.**

**La table Ingredient.**

**La table Potion, Onguent.**

**Pour terminer par la table Unite**

**Ensuite cela nous permet de faire nos tables intermédiaire du au association n,n**

**Les 1,1 ne disposant que de clé étrangère**

**Nous séparons notre SQL en plusieurs partie dans des documents textes séparer une première partie étant celle de la création de la base entière des différentes tables, champs**

**Une deuxième partie étant seulement afin de remplir les tables de différents enregistrements**

**Et un dernier script qui crée et remplit la table en même temps en cas de suppression de la table gagné du temps en faisant les deux parties précédentes**

**Nous utilisons des procédures stockées qui sont :**

* **Affichage du stock des ingrédients et mise à jour à la réception de nouveaux ingrédients**

**DELIMITER |**

|  |
| --- |
| **CREATE PROCEDURE reception\_ingredient(in ID\_Stock\_add int, in nbr\_add int)** |
|  |

|  |
| --- |
| **BEGIN** |
|  |

|  |
| --- |
| **update stock** |

|  |
| --- |
| **set Quantite\_Stock = Quantite\_Stock + nbr\_add** |
|  |

|  |
| --- |
| **where ID\_Stock = ID\_Stock\_add;** |
|  |

|  |
| --- |
| **select stock.ID\_Stock, ingredient.Nom\_Ingredient,stock.Quantite\_Stock** |
|  |

|  |
| --- |
| **from Stock,ingredient** |
|  |

|  |
| --- |
| **where stock.ID\_Stock = ingredient.ID\_Ingredient;** |
|  |

|  |
| --- |
| **END|** |
|  |

**Delimiter ;**

* **Affichage des commandes d’un client**

|  |
| --- |
| **DELIMITER |** |
|  |

|  |
| --- |
| **CREATE PROCEDURE cmd\_client(in ID\_Cmd int)** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **BEGIN** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **SELECT DISTINCT CmdClient.ID\_Cmd\_Client,onguent.Nom\_Onguent, comporte.Quantite,Potion.Nom\_Potion, concoit.Quantite,Ingredient.Nom\_Ingredient,dispose.Quantite** |
|  |

|  |
| --- |
| **FROM CmdClient, comporte, onguent, concoit, potion, dispose, Stock,Ingredient** |
|  |

|  |
| --- |
| **WHERE CmdClient.ID\_Cmd\_Client = ID\_Cmd** |
|  |

|  |
| --- |
| **AND comporte.ID\_Cmd\_Client =ID\_Cmd** |
|  |

|  |
| --- |
| **AND comporte.ID\_Onguent = onguent.ID\_onguent** |
|  |

|  |
| --- |
| **AND concoit.ID\_Cmd\_Client = ID\_Cmd\_Client** |
|  |

|  |
| --- |
| **AND concoit.ID\_Potion = Potion.ID\_Potion** |
|  |

|  |
| --- |
| **AND dispose.ID\_Cmd\_Client = ID\_Cmd\_Client** |
|  |

|  |
| --- |
| **AND dispose.ID\_stock = Stock.ID\_stock** |
|  |

|  |
| --- |
| **AND Ingredient.ID\_Ingredient = Stock.ID\_Ingredient;** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **END|** |
|  |

**Delimiter ;**

* **Suppression d’un aliment arriver à péremption**

|  |
| --- |
| **DELIMITER |** |
|  |

|  |
| --- |
| **CREATE PROCEDURE peremption\_ingredients()** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **BEGIN** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **DELETE stock FROM stock JOIN ingredient ON stock.ID\_Ingredient = ingredient.ID\_Ingredient** |
|  |

|  |
| --- |
| **WHERE ingredient.Peremption\_Ingredient = stock.Fraicheur\_Stock;** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **END|** |
|  |

**Delimiter ;**

* **Mise à jour du stock une fois une commande envoyée**

|  |
| --- |
| **DELIMITER |** |
|  |

|  |
| --- |
| **CREATE PROCEDURE envoi\_commande(in ID\_Cmd int)** |
|  |

|  |
| --- |
| **BEGIN** |
|  |

|  |
| --- |
| **update stock** |
|  |

|  |
| --- |
| **inner join dispose on dispose.ID\_stock = Stock.id\_stock** |
|  |

|  |
| --- |
| **set Stock.Quantite\_Stock = Stock.Quantite\_Stock - dispose.Quantite** |
|  |

|  |
| --- |
| **where dispose.ID\_Cmd\_Client = ID\_Cmd** |
|  |

|  |
| --- |
| **and Stock.ID\_stock = dispose.ID\_Stock;** |
|  |

|  |
| --- |
| **update CmdClient** |
|  |

|  |
| --- |
| **set CmdClient.Etat\_Cmd = 'Envoye'** |
|  |

|  |
| --- |
| **where CmdClient.ID\_Cmd\_Client = ID\_Cmd;** |
|  |

|  |
| --- |
| **END|** |
|  |

**Delimiter ;**

**Les requêtes utilisées dans la base de données sont :**

* **La liste des potions /onguents utilisant un ingredient**

**SELECT Nom\_Potion, Nom\_ingredient, Nom\_Onguent**

**FROM potion, compose, ingredient, contient, onguent**

**WHERE potion.ID\_Potion = contient.ID\_Potion**

**AND contient.ID\_Ingredient = ingredient.ID\_Ingredient**

**AND onguent.ID\_Onguent = compose.ID\_Onguent**

**AND compose.ID\_Ingredient = Ingredient.ID\_Ingredient**

**AND Nom\_Ingredient = ‘Nom\_De\_L\_Ingredient’ ;**

* **Le nombre moyens d’ingredient utiliser par potion**

**SELECT AVG(cnt.nb\_ing)**

**FROM (SELECT COUNT(contient.ID\_Ingredient) cnt**

**FROM contient GROUP BY contient.ID\_Potion ) nb\_ing ;**

* **La recherche des ingrédients utilisant un diluant précis :**

**SELECT Nom\_potion,Nom\_diluant**

**FROM (Potion INNER JOIN Diluant ON Diluant.ID\_Diluant = Potion.ID\_Diluant)**

**WHERE Diluant.Nom\_Diluant = ‘Nom\_du\_Diluant’ ;**

* **Le classement des potions par température**

**SELECT Nom\_Potion, Temperature\_Potion**

**FROM potion**

**ORDER BY Temperature\_Potion**

* **La comparaison du cout des ingrédients et du prix de vente de la potion**

**SELECT Nom\_Potion, Prix\_Potion, SUM(prix\_ingredient)\*quantite AS prix\_total\_ingredient, Prix\_Potion-SUM(prix\_ingredient)\*quantite AS Resultat**

**FROM potion, contient, ingredient**

**WHERE potion.ID\_Potion = contient.ID\_Potion**

**AND contient.ID\_Ingredient=ingredient.ID\_Ingredient**

**GROUP BY Nom\_Potion**

**ORDER BY Resultat**

**Pour sauvegarder une table dans un script nous utilisons la commande suivante :**

**mysqldump -u root -h localhost -p nom\_de\_la\_base\_de\_donnees nom\_du\_champ > localisation\_du\_fichier.sql**

**Pour utiliser un script stocké voici la commande devant être exécuté :**

**mysqldump -u root -h localhost -p nom\_de\_la\_base\_de\_donnees nom\_du\_champ < localisation\_du\_fichier**

**Avant de sauvegarder les commandes non envoyer nous devons réaliser une vue qui n’affichera que les enregistrements étant différent de livree**

**CREATE VIEW `commande\_client` AS SELECT ID\_Cmd\_Client,Etat\_de\_Cmd FROM cmdclient**

**WHERE Etat\_de\_Cmd != 'Livree';**

**Bien que la requête de création de recette ne fonctionne pas nous avons essayé de la faire au maximum :**

|  |
| --- |
| **DELIMITER |** |
|  |

|  |
| --- |
| **CREATE PROCEDURE ajout\_recette(** |
|  |

|  |
| --- |
| **IN NomInventeur VARCHAR(255),** |
|  |

|  |
| --- |
| **IN NumeroPermis INT(11),** |
|  |

|  |
| --- |
| **IN Exisant BOOLEAN,** |
|  |

|  |
| --- |
| **IN NomRecette VARCHAR(255),** |
|  |

|  |
| --- |
| **IN NomDiluant VARCHAR(255),** |
|  |

|  |
| --- |
| **IN PrixPotion INT(11),** |
|  |

|  |
| --- |
| **IN TemperaturePotion INT(11),** |
|  |

|  |
| --- |
| **IN NomIngredient1 VARCHAR(255),** |
|  |

|  |
| --- |
| **IN NomUnite1 VARCHAR(255),** |
|  |

|  |
| --- |
| **IN Quantite1 INT(11),** |
|  |

|  |
| --- |
| **IN durée1 INT(11),** |
|  |

|  |
| --- |
| **IN NomIngredient2 VARCHAR(255),** |
|  |

|  |
| --- |
| **IN NomUnite2 VARCHAR(255),** |
|  |

|  |
| --- |
| **IN Quantite2 INT(11),** |
|  |

|  |
| --- |
| **IN durée2 INT(11),** |
|  |

|  |
| --- |
| **IN NomIngredient3 VARCHAR(255),** |
|  |

|  |
| --- |
| **IN NomUnite3 VARCHAR(255),** |
|  |

|  |
| --- |
| **IN Quantite3 INT(11),** |
|  |

|  |
| --- |
| **IN durée3 INT(11))** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **BEGIN** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **SET @ninv = NomInventeur;** |
|  |

|  |
| --- |
| **SET @permis = NumeroPermis;** |
|  |

|  |
| --- |
| **SET @exi = Exisant;** |
|  |

|  |
| --- |
| **SET @nrec = NomRecette;** |
|  |

|  |
| --- |
| **SET @ndil = NomDiluant;** |
|  |

|  |
| --- |
| **SET @prix = PrixPotion;** |
|  |

|  |
| --- |
| **SET @Ptemp = TemperaturePotion;** |
|  |

|  |
| --- |
| **SET @ning1 = NomIngredient1;** |
|  |

|  |
| --- |
| **SET @quant = Quantite1;** |
|  |

|  |
| --- |
| **SET @duree = durée1;** |
|  |

|  |
| --- |
| **SET @unit1 = NomUnite1;** |
|  |

|  |
| --- |
| **SET @ning2 = NomIngredient2;** |
|  |

|  |
| --- |
| **SET @quant = Quantite2;** |
|  |

|  |
| --- |
| **SET @duree = durée2;** |
|  |

|  |
| --- |
| **SET @unit2 = NomUnite2;** |
|  |

|  |
| --- |
| **SET @ning3 = NomIngredient3;** |
|  |

|  |
| --- |
| **SET @quant = Quantite3;** |
|  |

|  |
| --- |
| **SET @duree = durée3;** |
|  |

|  |
| --- |
| **SET @unit3 = NomUnite3;** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **INSERT INTO `Inventeur` (`Nom\_Inventeur`, `Numero\_Permis\_Inventeur`)** |
|  |

|  |
| --- |
| **VALUES (@ninv, @permis)** |
|  |

|  |
| --- |
| **WHERE @exi = '1';** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **INSERT INTO `recette` (Nom\_recette)** |
|  |

|  |
| --- |
| **VALUES (@nrec);** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **SET @dil = ( SELECT ID\_Diluant** |
|  |

|  |
| --- |
| **FROM diluant** |
|  |

|  |
| --- |
| **WHERE @ndil = Nom\_Diluant** |
|  |

|  |
| --- |
| **);;** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **SET @inv = (SELECT ID\_Inventeur** |
|  |

|  |
| --- |
| **From inventeur** |
|  |

|  |
| --- |
| **Where @ninv = Nom\_Inventeur** |
|  |

|  |
| --- |
| **);;** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **INSERT INTO `Potion` (`Nom\_Potion`, `Prix\_Potion`,`ID\_Diluant`,`Temperature\_Potion`,`ID\_Inventeur`)** |
|  |

|  |
| --- |
| **VALUES (@nrec, @prix, @dil, @Ptemp, @inv );** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **SET @pot = (SELECT ID\_Potion** |
|  |

|  |
| --- |
| **FROM potion** |
|  |

|  |
| --- |
| **WHERE @nrec = Nom\_Potion);;** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **SET @unit = (SELECT ID\_Unite** |
|  |

|  |
| --- |
| **FROM unite** |
|  |

|  |
| --- |
| **WHERE @unit1 = Nom\_Unite);;** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **SET @ing = (SELECT ID\_Ingredient** |
|  |

|  |
| --- |
| **FROM Ingredient** |
|  |

|  |
| --- |
| **WHERE Nom\_Ingredient = @ning1 );;** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **INSERT INTO `contient` (`Duree`, `ID\_Ingredient`, `ID\_Potion`, `ID\_Unite`, `Quantite`)** |
|  |

|  |
| --- |
| **VALUES (@duree, @ing, @pot, @unit, @quant);** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **INSERT INTO `compose` (`ID\_Ingredient`, `ID\_Onguent`, `Quantite`, `ID\_Unite`)** |
|  |

|  |
| --- |
| **VALUES (@ing, @pot, @quant, @unit);** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **SET @unit = (SELECT ID\_Ingredient** |
|  |

|  |
| --- |
| **FROM Ingredient** |
|  |

|  |
| --- |
| **WHERE @unit2 = Nom\_Unite);;** |
|  |

|  |
| --- |
| **SET @ing = (SELECT ID\_Unite** |
|  |

|  |
| --- |
| **FROM unite** |
|  |

|  |
| --- |
| **WHERE @ning2 = Nom\_Ingredient);;** |
|  |

|  |
| --- |
| **INSERT INTO `contient` (`Duree`, `ID\_Ingredient`, `ID\_Potion`, `ID\_Unite`, `Quantite`)** |
|  |

|  |
| --- |
| **VALUES (@duree, @ing, @pot, @unit, @quant);** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **INSERT INTO `compose` (`ID\_Ingredient`, `ID\_Onguent`, `Quantite`, `ID\_Unite`)** |
|  |

|  |
| --- |
| **VALUES (@ing, @pot, @quant, @unit);** |
|  |

|  |
| --- |
|  |

|  |
| --- |
| **SET @unit = (SELECT ID\_Unite** |
|  |

|  |
| --- |
| **FROM unite** |
|  |

|  |
| --- |
| **WHERE @unit3 = Nom\_Unite);;** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **SET @ing = (SELECT ID\_Ingredient** |
|  |

|  |
| --- |
| **FROM Ingredient** |
|  |

|  |
| --- |
| **WHERE @ning3 = Nom\_Ingredient);;** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **INSERT INTO `contient` (`Duree`, `ID\_Ingredient`, `ID\_Potion`, `ID\_Unite`, `Quantite`)** |
|  |

|  |
| --- |
| **VALUES (@duree, @ing, @pot, @unit, @quant);** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **INSERT INTO `compose` (`ID\_Ingredient`, `ID\_Onguent`, `Quantite`, `ID\_Unite`)** |
|  |

|  |
| --- |
| **VALUES (@ing, @pot, @quant, @unit);** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **END |** |
|  |

**DELIMITER ;**

**La requête donnant les couples onguents/potions utilisant les mêmes ingrédients :**

**CREATE VIEW Vu\_Potion43**

**AS SELECT DISTINCT contient.ID\_Ingredient**

**FROM contient**

**WHERE ID\_Potion=43;**

**CREATE VIEW Vu\_Onguent43**

**AS SELECT DISTINCT compose.ID\_Ingredient**

**FROM compose**

**WHERE ID\_Onguent=43;**

**--------------------------------------------------------------------------**

**DELIMITER |**

**CREATE PROCEDURE test(in nbr\_couple int)**

**BEGIN**

**set @tblname = "Vu\_Onguent";**

**set @tblname1 = "Vu\_Potion";**

**set @nbr = 0;**

**set @potion = nbr\_couple;**

**premierewhile: WHILE @nbr <> 42 DO**

**set @nbr = @nbr + 1 ;**

**set @sql = CONCAT('UPDATE Couple set Couple.ID\_Onguent=',@nbr,' where exists (SELECT DISTINCT ID\_Ingredient FROM ',@tblname1,@potion,' WHERE ID\_Ingredient In ( SELECT ID\_Ingredient FROM ', @tblname,@nbr,'));');**

**PREPARE stmt FROM @sql ;**

**EXECUTE stmt ;**

**DEALLOCATE PREPARE stmt ;**

**END WHILE premierewhile;**

**update Couple**

**set COuple.Id\_Potion=nbr\_couple;**

**SELECT DISTINCT Potion.Nom\_Potion, Compose.Id\_Ingredient, Onguent.Nom\_Onguent**

**FROM Couple INNER JOIN Compose ON Couple.ID\_Onguent = Compose.ID\_Onguent INNER JOIN Onguent ON compose.ID\_Onguent = Onguent.ID\_Onguent INNER JOIN contient ON Couple.ID\_Potion=Contient.ID\_Potion INNER JOIN Potion ON Potion.ID\_Potion = contient.Id\_Potion;**

**END|**

**Delimiter ;**

**Validation de l’ajout d’une recette :**

**DELIMITER |**

**CREATE PROCEDURE ValidatingRecette**

**BEGIN**

**DELETE FROM recette**

**WHERE Statut = '1'**

**AND etat = '1'**

**END |**

**Suppression d’une recette non retenue**

**DELIMITER |**

**CREATE PROCEDURE Suppression\_recette**

**BEGIN**

**DELETE FROM potion INNER JOIN recette ON potion.ID\_Potion = recette.ID\_Potion**

**WHERE recette.Statue = '0' ;**

**END |**

**DELIMITER ;**

**Durant le projet nous avons rencontré différents problèmes :**

* **Des problèmes quant à l’utilisation du logiciel MYSQL**
* **Différentes erreurs de programmation**
* **L’appel à des objets en oubliant de définir la liaison des tables**
* **Création de la requête d’ajout de recette erreur dépassant les compétences du groupe**

**Les perceptives d’amélioration pour ce projet :**

* **Utilisation d’un trigger toutes les jours qui permettrait de changer la fraicheur de tout le stock en augmentant toutes les fraicheurs de 1**
* **Utilisation de différent trigger pour supprimer et accepter les recettes**
* **Lier les fournisseurs a des ingrédients**

**Retour d’expérience du groupe :**

**Valentin :**

**Avant le projet la base de données ne m’intéresser pas du tout mais après une pratique intensive du début à la fin dans un projet de base de données cela m’a convaincu que ce domaine était beaucoup plus intéressant que je ne le penser. De plus j’ai travaillé avec un groupe dont je ne connaissais pas beaucoup les membres avant ce projet comme ça cela m’a permis de m’ouvrir un peu plus à ma classe.**

**Alexis :**

**Le projet a été très agréable, car il m’a donné goût à la base de données. Il m’a permis d’appliquer mes connaissances en base de données apprises lors des prosits mais aussi de développer mes connaissances et d’apprendre de nouvelles fonctions du langage SQL. Le groupe a été très productif et les tâches ont bien étaient repartis ce qui a permis une bonne progression sans avoir de pressions.**

**Dimitri :**

**Pour ce projet en base de données, j’ai trouvé celui-ci intéressant car il perpétué le prosit en gardant certains éléments donc il m’a permis de m’y intéresser. Concernant le groupe de projet, j’ai pu découvrir un nouveau groupe car je n’avais travaillé avec ceux-ci et j’ai pu faire plus amples connaissances. Le groupe a été bien organisé dans l’ensemble nous avons touchés à tous les points du projet pour répondre au mieux au problème demandé mais les erreurs et problèmes de logiciels ont étaient nombreux.**

**J-f :**

**Le projet a été très intéressent et ma permit de développer les points n’ayant pas étant compris lors des prosits les taches on était repartit entre les membres du groupe de manière plutôt efficace pour cette première expérience en tant que chef de projet à au début était difficile à mettre en place du au différent taches supplémentaire ajouter mais l’ensemble du groupe c’est vite mis en place**