Les relations du MLD :

**Potions** (ID\_Potion, Nom\_Potion#, Prix\_Potion)

**Onguents** (ID\_Onguent, Nom\_ Onguent#, Prix\_ Onguent)

**Ingredients** (ID\_Ingredient, Nom\_Ingredient#, Prix\_Ingredient, Fraicheur, Seuil)

**Recipients** (ID\_Recipient, Nom\_Recipient#, Prix\_Recipient)

**Recette\_Potions** (ID\_Potion#, Nom\_Inventeur#, Nom\_Potion, Ingredient\_Recette, Prix\_Ingredient\_Recette, Seuil, Quantite, Diluant, Prix\_Diluant, Temperature, Prix\_Total\_Produit)

**Recettes\_Onguent** : (ID\_Onguent#, Nom\_Inventeur#, Nom\_ Onguent, Ingredient\_Recette, Prix\_Ingredient\_Recette, Seuil, Quantite, Diluant, Prix\_Diluant, Temperature, Prix\_Total\_Produit)

**Nouvelles\_Recettes** (ID\_Recette, Nom\_Recette, Type\_Recette, Ingredient\_Recette, Prix\_Ingredient\_Recette, Seuil, Quantite, Diluant, Prix\_Diluant, Temperature, Prix\_Total\_Produit, Nom\_Inventeur#)

**Stock\_Ingredients** (ID\_Ingredient#, ID\_Fournisseur#, Nom\_Ingredient, Prix\_Ingredient, Quantite\_Stock, Fraicheur, Seuil)

**Stock\_Recipients** (ID\_Recipient#, Nom\_ Recipient, Prix\_ Recipient, Nombre\_En\_Stock)

**Inventeurs** (Nom\_Inventeur, Nom\_Recette#)

**Commandes** (ID\_Commande, Date\_Commande, Produit\_Commande, Recipient\_Commande, Nb\_Exemplares\_Commande, Prix\_Produit, Etat\_Commande, N\_Client#)

**Client** (N\_Client, Nom\_Client, Prenom\_Client, Adresse\_Client, Telephone\_Client)

**Fournisseur** (ID\_Fournisseur, Nom\_Fournisseur, Telephone\_Fournisseur)

Avant de s’atteler à la réalisation des tables, nous nous sommes d’abord penchés sur le script de création de la base de données en elle-même (disponible ci-dessous). Puis, à l’aide des fonctions CREATE et ALTER, nous avons respectivement retranscrit chaque relation du MLD en une table cohérente avec chacune son identifiant et ses attributs, puis, une fois ceci fait, nous avons attribué aux données correspondantes les clés primaires et étrangères des relations, avec les options ON UPDATE CASCADE et ON DELETE CASCADE, afin de pouvoir maintenir l’intégrité fonctionnelle de toutes les tables liées lorsque l’une d’entre elles voit une de ses clés étrangères modifiées.

Implémentation sous MySQL (phpmyadmin) :

-- Script de création de la Base de Données (PENSER A ENLEVER LES FONCTIONS D'APPEL DES FONCTIONS PRESENTES EN FIN DE REQUETES)

-- Mot de passe du root : exia

DROP DATABASE IF EXISTS ProjetBDD;

CREATE DATABASE ProjetBDD CHARACTER SET utf8;

USE ProjetBDD;

DROP TABLE IF EXISTS `Potions`;

DROP TABLE IF EXISTS `Onguents`;

DROP TABLE IF EXISTS `Ingredients`;

DROP TABLE IF EXISTS `Recipients`;

DROP TABLE IF EXISTS `Inventeurs`;

DROP TABLE IF EXISTS `Clients`;

DROP TABLE IF EXISTS `Fournisseurs`;

DROP TABLE IF EXISTS `Nouvelles\_Recettes`;

DROP TABLE IF EXISTS `Commandes`;

DROP TABLE IF EXISTS `Recettes\_Potions`;

DROP TABLE IF EXISTS `Recettes\_Onguents`;

DROP TABLE IF EXISTS `Stock\_Ingredients`;

DROP TABLE IF EXISTS `Stock\_Recipients`;

CREATE TABLE Potions

(

ID\_Potion INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

Nom\_Potion VARCHAR(100),

Prix\_Potion INT NOT NULL DEFAULT '0'

);

CREATE TABLE Onguents

(

ID\_Onguent INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

Nom\_Onguent VARCHAR(100),

Prix\_Onguent INT NOT NULL DEFAULT '0'

);

CREATE TABLE Ingredients

(

ID\_Ingredient INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

Nom\_Ingredient VARCHAR(100),

Prix\_Ingredient INT NOT NULL DEFAULT '0'

);

CREATE TABLE Recipients

(

ID\_Recipient INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

Nom\_Recipient VARCHAR(100),

Prix\_Recipient INT NOT NULL DEFAULT '0'

);

CREATE TABLE Inventeurs

(

Nom\_Inventeur VARCHAR(100) PRIMARY KEY,

Nom\_Recette VARCHAR(100)

);

CREATE TABLE Clients

(

N\_Client INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

Nom\_Client VARCHAR(100),

Prenom\_Client VARCHAR(100),

Adresse\_Client VARCHAR(100),

Telephone\_Client INT(10)

);

CREATE TABLE Fournisseurs

(

ID\_Fournisseur INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

Nom\_Fournisseur VARCHAR(100),

Telephone\_Fournisseur INT(10)

);

CREATE TABLE Nouvelles\_Recettes

(

ID\_Recette INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

Nom\_Recette VARCHAR(100),

Type\_Recette VARCHAR(10),

Ingredient\_Recette VARCHAR(100),

Prix\_Ingredient\_Recette INT NOT NULL DEFAULT '0',

Seuil INT,

Quantite\_Recette INT,

Diluant VARCHAR(100),

Prix\_Diluant INT NOT NULL DEFAULT '0',

Temperature INT,

Prix\_Total\_Produit INT NOT NULL DEFAULT '0',

Nom\_Inventeur VARCHAR(100)

);

CREATE TABLE Commandes

(ID\_Commande INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

Date\_Commande TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

Produit\_Commande VARCHAR (100),

Recipient\_Commande VARCHAR (100),

Nb\_Exemplaires\_Commande INT,

Prix\_Produit INT NOT NULL DEFAULT '0',

Etat\_Commande VARCHAR (20),

N\_Client INT NOT NULL

);

CREATE TABLE Recettes\_Potions

(

ID\_Potion INT NOT NULL,

Nom\_Inventeur VARCHAR (100),

Nom\_Potion VARCHAR (100),

Ingredient\_Recette VARCHAR (100),

Prix\_Ingredient\_Recette INT NOT NULL DEFAULT '0',

Seuil INT,

Quantite INT,

Diluant VARCHAR (100),

Prix\_Diluant INT NOT NULL DEFAULT '0',

Temperature INT,

Prix\_Total\_Produit INT NOT NULL DEFAULT '0'

);

CREATE TABLE Recettes\_Onguents

(

ID\_Onguent INT NOT NULL,

Nom\_Inventeur VARCHAR (100),

Nom\_Onguent VARCHAR (100),

Ingredient\_Recette VARCHAR (100),

Prix\_Ingredient\_Recette INT NOT NULL DEFAULT '0',

Seuil INT,

Quantite INT,

Prix\_Total\_Produit INT NOT NULL DEFAULT '0'

);

CREATE TABLE Stock\_Ingredients

(

ID\_Ingredient INT NOT NULL,

ID\_Fournisseur INT NOT NULL,

Nom\_Ingredient VARCHAR (100),

Prix\_Ingredient INT NOT NULL DEFAULT '0',

Quantite\_Stock INT,

Fraicheur INT

);

CREATE TABLE Stock\_Recipients

(

ID\_Recipient INT NOT NULL,

Nom\_Recipient VARCHAR (100),

Prix\_Recipient INT NOT NULL DEFAULT '0',

Nombre\_En\_Stock INT

);

ALTER TABLE Potions

ADD FOREIGN KEY (Nom\_Potion) REFERENCES Recettes\_Potions(Nom\_Potion)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE Onguents

ADD FOREIGN KEY (Nom\_Onguent) REFERENCES Recettes\_Onguents(Nom\_Onguent)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE Ingredients

ADD FOREIGN KEY (Nom\_Ingredient) REFERENCES Stock\_Ingredients(Nom\_Ingredient)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE Recipients

ADD FOREIGN KEY (Nom\_Recipient) REFERENCES Stock\_Recipients(Nom\_Recipient)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE Inventeurs

ADD FOREIGN KEY (Nom\_Recette) REFERENCES Nouvelles\_Recettes(Nom\_Recette)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE Nouvelles\_Recettes

ADD FOREIGN KEY (Nom\_Inventeur) REFERENCES Inventeurs(Nom\_Inventeur)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE Commandes

ADD FOREIGN KEY (N\_Client) REFERENCES Clients(N\_Client)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE Recettes\_Potions

ADD FOREIGN KEY (ID\_Potion) REFERENCES Potions(ID\_Potion)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE Recettes\_Potions

ADD FOREIGN KEY (Nom\_Inventeur) REFERENCES Inventeurs(Nom\_Inventeur)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE Recettes\_Potions

ADD CONSTRAINT Pk\_Recettes\_Potions PRIMARY KEY (ID\_Potion, Nom\_Inventeur);

ALTER TABLE Recettes\_Onguents

ADD FOREIGN KEY (ID\_Onguent) REFERENCES Onguents(ID\_Onguent)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE Recettes\_Onguents

ADD FOREIGN KEY (Nom\_Inventeur) REFERENCES Inventeurs(Nom\_Inventeur)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE Recettes\_Onguents

ADD CONSTRAINT Pk\_Recettes\_Onguents PRIMARY KEY (ID\_Onguent, Nom\_Inventeur);

ALTER TABLE Stock\_Ingredients

ADD FOREIGN KEY (ID\_Ingredient) REFERENCES Ingredients(ID\_Ingredient)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE Stock\_Ingredients

ADD FOREIGN KEY (ID\_Fournisseur) REFERENCES Fournisseurs(ID\_Fournisseur)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE Stock\_Ingredients

ADD CONSTRAINT Pk\_Stock\_Ingredients PRIMARY KEY (ID\_Ingredient, ID\_Fournisseur);

ALTER TABLE Stock\_Recipients

ADD FOREIGN KEY (ID\_Recipient) REFERENCES Recipients(ID\_Recipient)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE Stock\_Recipients

ADD CONSTRAINT Pk\_ID\_Recipient PRIMARY KEY (ID\_Recipient);