

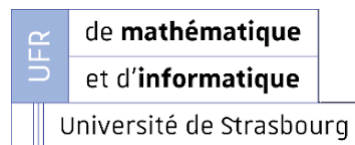
RENDU DE PROJET

« Recettes de cuisines »

BDD2

UFR Math-Info - Université de Strasbourg

3^{ième} année de Licence Informatique



Date de rendu : 16 novembre 2021

Par

Matthieu FREITAG

<https://github.com/Zapharaos/ProjetBD2>

Index

1	Introduction	1
2	Modèle entité-association	2
2.1	Illustration	2
2.2	Implémentation et choix	3
3	Contraintes d'intégrité	9
3.1	Contraintes statiques des tables principales	9
3.2	Contraintes statiques des tables de liaisons	11
3.3	Contraintes dynamiques	14
4	Modèle logique relationnel	16
4.1	Illustration	16
4.2	Implémentation	17
5	Conclusion	18
	Bibliographie	19

Introduction

On se propose de réaliser dans ce projet l'implémentation de la base de données d'un site pour gérer des recettes de cuisine dans le cadre de l'UE "Base de donnée 2".

Pour simplifier cette étude, le sujet, disponible en annexe, nous guide de manière grossière et cite les informations que l'on souhaiterait sauvegarder. De ce fait, les interprétations seront multiples.

Nous sont également décrit, avec précision, les différentes requêtes, contraintes d'intégrité, procédures et fonctions à implémenter.

Ce présent rapport ne concerne que la première partie du sujet, à savoir la modélisation de la base de données, les contraintes d'intégrités et les scripts SQL (création, suppression, insertion).

La seconde partie, quant à elle, correspond à la réalisation des requêtes, contraintes d'intégrité, procédures et fonctions. Elle sera réalisée par après et ne figurera donc pas dans ce rapport.

2.1 Illustration

2.2 Implémentation et choix

Utilisateur

La table "USERS" contient l'ensemble des informations sur les utilisateurs. Les champs "nom", "prénom" et "adresse" était uniquement demandé pour la deuxième partie, mais j'ai préféré les implémenter en amont.

- **idUsers** : CLE PRIMAIRE, elle permet d'identifier chaque utilisateur avec un numéro.
- **login** : pseudo, il permet d'identifier un utilisateur par un surnom.
- **email** : adresse e-mail, elle permet de joindre un utilisateur.
- **pwd** : mot de passe, il permet de prouver l'identité d'un utilisateur.
- **name** : prénom, il permet d'identifier un utilisateur.
- **lastName** : nom de famille, il permet d'identifier un utilisateur.
- **address** : lieu de résidence, il permet de localiser un utilisateur.

Ingredient

La table "INGREDIENT" contient l'ensemble des ingrédients que peut posséder une recette.

- **idIngredient** : CLE PRIMAIRE, elle permet d'identifier chaque ingrédient avec un numéro unique.
- **nameIngredient** : nom, il permet d'identifier un ingrédient.
- **unity** : unité, elle permet d'attribuer une unité à ses quantités.

Dans le cas d'une unité, on indiquera le pluriel, par exemple : "oeuf(s)". Comme demandé dans le sujet, la quantité d'un ingrédient peut être indiquée soit en unité, soit en gramme. De ce fait, les recettes possédant des quantités comme "3 cuillères à soupe" sont biaisées.

Afin de palier à ce problème, on pourrait ajouter une table "EQUIVALENCE", reliée aux ingrédients nécessitant d'effectuer des conversions, et qui donnerait par exemple l'équivalent d'une cuillère à café de sucre en gramme.

Ainsi, par soucis de simplicité et afin de respecter la consigne, j'ai effectué manuellement la conversion de la quantité. Pour cela, je me suis servi d'un site (disponible dans bibliographie). De plus, j'admet une densité de 1 pour chaque liquides et poudres.

Recette

La table "RECIPE" contient l'ensemble des recettes.

- **idRecipe** : CLE PRIMAIRE, elle permet d'identifier chaque recette avec un numéro unique.
- **nameRecipe** : nom, il permet d'identifier une recette.
- **author** : auteur, il permet d'indiquer l'auteur.
- **descRecipe** : description, elle apporte des détails à une recette.
- **difficulty** : difficulté de la réalisation de la recette.
- **price** : coût de la réalisation de la recette
- **nbPers** : nombre de personnes, pour les quantités d'ingrédients.

Planning

La table "PLANNING" contient l'ensemble des plannings.

- **idPlanning** : CLE PRIMAIRE, elle permet d'identifier chaque planning avec un numéro unique.
- **namePlanning** : nom, il permet d'identifier un planning.
- **descPlanning** : description, elle apporte des détails à un planning.
- **startPlanning** : date du premier repas dans le planning
- **endPlanning** : date du dernier repas dans le planning

Il existe deux tables qui font le lien entre un planning et un utilisateur. Les deux sont des listes de plannings, mais une seule permet d'archiver. Pourquoi ne pas avoir simplement deux tables de plannings, une d'archive et une actuelle ?

- quand un planning doit être archivé, il suffit de supprimer la liaison dans la liste des plannings actuels et de l'insérer dans les archives, ce qui évite de devoir copier-coller des plannings à chaque fois.

De plus, avec cette implémentation, plusieurs utilisateurs peuvent se partager un planning.

Liste des courses

La table "SHOPPING" contient l'ensemble des listes de courses d'après un stock et un planning.

- **idShopping** : CLE PRIMAIRE, elle permet d'identifier chaque liste avec un numéro unique.
- **nameShopping** : nom, il permet d'identifier une liste des courses.
- **descShopping** : description, elle apporte des détails à une liste.
- **startShopping** : date de début d'une liste des courses
- **endShopping** : date de fin d'une liste des courses

Il existe deux tables qui font le lien entre une liste des courses et un utilisateur. Les deux sont des listes de listes des courses, mais une seule permet d'archiver. Pourquoi ne pas avoir simplement deux tables de listes des courses, une d'archive et une actuelle ?

- quand une liste des courses doit être archivée, il suffit de supprimer la liaison dans la liste des listes des courses actuelles et de l'insérer dans les archives, ce qui évite de devoir copier-coller des listes de courses à chaque fois.

De plus, il existe une table qui permet de relier un ou plusieurs plannings à une liste de courses. Ainsi, les dates d'une liste des courses correspondent à la date de début la plus petite et à la date de fin la plus grande parmi les plannings composant cette liste.

Régime alimentaire

La table "DIET" contient l'ensemble des régimes alimentaires que peut posséder un ingrédient.

- **idDiet** : CLE PRIMAIRE, elle permet d'identifier chaque régime avec un numéro unique.
- **nameDiet** : nom, il permet d'identifier un régime alimentaire.

La table "INGREDIENT" est reliée à celle ci. Pourquoi ne pas avoir ajouté directement ces champs dans la table "INGREDIENT" ?

- il sera plus simple d'ajouter des régimes par la suite.
- si par la suite on souhaite relier cette table à la table "RECIPE".

Je n'ai pas relié cette table à la table "RECIPE" car ce n'était pas explicitement demandé dans le sujet. En effet, on souhaite seulement savoir si elle est conforme à un régime et j'estime qu'une fonction sera suffisante.

Par défaut, j'estime qu'il existe cinq régimes alimentaires : le végétarisme, le pesco-végétarisme, le véganisme, le sans-gluten et le sans-lactose. Il en existe bien plus, mais par soucis de simplicité, je n'ai considéré que ces cinq là lors de l'insertion des données. De plus, pas tout sont présent dans les ingrédients que j'ai sélectionné.

Categorie

La table "CATEGORY" contient l'ensemble des catégories que peut posséder un ingrédient.

- idCategory : CLE PRIMAIRE, elle permet d'identifier chaque catégorie avec un numéro unique.
- nameCategory : nom, il permet d'identifier une catégorie.

Il existe une table qui fait le lien avec celle-ci et la table "INGREDIENT", mais aussi une clé étrangère dans la table "RECIPE". Pourquoi ne pas avoir ajouté un champ "category" dans ces tables ?

- il sera plus simple d'ajouter des catégories par la suite.
- pour vérifier l'unicité d'une catégorie.
- pour faire un lien entre la catégorie d'un ingrédient dans une recette donnée et tous les autres ingrédients possédant cette même catégorie.
- j'estime qu'un ingrédient peut posséder plusieurs catégories, comme le camembert : c'est à la fois un produit laitier, un fromage, un fromage pâte molle et un fromage crôte fleurie. Ainsi, en fonction d'une recette il pourrait avoir une utilisation/catégorie bien différente !

Qualité diététique

La table "QUALITY" contient l'ensemble des qualités diététique que peut posséder un ingrédient ou une recette.

- **idQuality** : CLE PRIMAIRE, elle permet d'identifier chaque qualité avec un numéro unique.
- **nameQuality** : nom, il permet d'identifier une qualité.

Il existe deux tables qui font respectivement le lien entre celle-ci et "INGREDIENT", respectivement "RECIPE". Pourquoi ne pas avoir ajouté directement ces champs dans les tables "INGREDIENT" et "RECIPE" ?

- il sera plus simple d'ajouter des qualités par la suite.
- pour vérifier l'unicité d'une qualité diététique.

Par défaut, j'estime qu'il existe quatre qualités diététique : les calories, les protides, les glucides et les lipides. Il en existe bien plus, mais par soucis de simplicité, je n'ai considéré que ces quatres là lors de l'insertion des données.

Illustration

La table "MEDIA" contient l'ensemble des illustrations que possède une recette.

- **idMedia** : CLE PRIMAIRE, elle permet d'identifier chaque illustration avec un numéro unique.
- **nameMedia** : nom, il permet d'identifier une illustration.
- **descMedia** : description, elle apporte des détails à une illustration.
- **media** : url, lien vers l'illustration
- **idRecipe** : CLE ETRANGERE, elle permet de relier un media à une recette.

Etape

La table "STEP" contient l'ensemble des étapes que possède une recette.

- **idStep** : CLE PRIMAIRE, elle permet d'identifier chaque étape avec un numéro unique
- **weight** : poids, il permet d'ordonner les étapes d'une même recette.
- **nameStep** : nom, il permet d'identifier une étape.
- **descStep** : description, elle apporte des détails à une étape.
- **idRecipe** : CLE ETRANGERE, elle permet de relier une étape à une recette.

Durée

La table "DURATION" contient l'ensemble des types de durée que peut posséder une étape ou une recette.

- **idDuration** : CLE PRIMAIRE, elle permet d'identifier chaque type de durée avec un numéro unique
- **nameDuration** : nom, il permet d'identifier un type de durée.

Il existe deux tables qui font respectivement le lien entre celle-ci et "STEP", respectivement "RECIPE". Pourquoi ne pas avoir ajouté directement ces champs dans les tables "STEP" et "RECIPE" ?

- il sera plus simple d'ajouter des types de durée par la suite.
- pour vérifier l'unicité d'un type de durée.

Par défaut, j'estime qu'il existe quatre types de durée : le temps de préparation, le temps de cuisson, le temps de repos et le temps total.

Contraintes d'intégrité

3.1 Contraintes statiques des tables principales

Utilisateur

- idUsers : CLE PRIMAIRE.
- login : unique, non vide, maximum 32 caractères.
- email : unique, non vide, maximum 128 caractères, de la forme "%@%.%".
- pwd : non vide, maximum 64 caractères (chiffré).
- name : maximum 32 caractères.
- lastName : maximum 32 caractères.
- address : maximum 128 caractères.

Ingredient

- idIngredient : CLE PRIMAIRE.
- nameIngredient : unique, non vide, maximum 64 caractères.
- unity : non vide, maximum 64 caractères.

Recette

- idRecipe : CLE PRIMAIRE.
- nameRecipe : non vide, maximum 128 caractères.
- author : non vide, maximum 64 caractères.
- difficulty : non vide, maximum 32 caractères, de la forme : 'Tres facile', 'Facile', 'Standard', 'Difficile'.
- price : non nul, un flottant entre 1 et 5.
- nbPers : non nul, strictement supérieur à 0.

Planning

- idPlanning : CLE PRIMAIRE.
- namePlanning : non vide, maximum 32 caractères.
- Pas de contraintes statiques sur les dates, elles seront insérées et vérifiées dynamiquement lors de l'ajout d'une recette au planning.

Liste des courses

- idShopping : CLE PRIMAIRE.
- nameShopping : non vide, maximum 32 caractères.
- Pas de contraintes statiques sur les dates, elles seront insérées et vérifiées dynamiquement lors de l'ajout d'un planning à la liste des courses.

Regime alimentaire

- idDiet : CLE PRIMAIRE.
- nameDiet : unique, non vide, maximum 32 caractères.

Categorie

- idCategory : CLE PRIMAIRE.
- nameCategory : unique, non vide, maximum 64 caractères.

Qualité diététique

- idQuality : CLE PRIMAIRE.
- nameQuality : unique, non vide, maximum 32 caractères.

Illustration

- idMedia : CLE PRIMAIRE.
- nameMedia : non vide, maximum 32 caractères.
- media : non vide, maximum 256 caractères.
- idRecipe : CLE ETRANGERE.

Etape

- idStep : CLE PRIMAIRE.
- weigh : non nul, strictement supérieur à 0.
- nameStep : maximum 32 caractères.
- idRecipe : CLE ETRANGERE.

Durée

- idDuration : CLE PRIMAIRE.
- nameDuration : unique, non vide, maximum 32 caractères.

3.2 Contraintes statiques des tables de liaisons

Ingrédients d'un utilisateur

- idUsers, idIngredient : CLE PRIMAIRE.
- idUsers : CLE ETRANGERE.
- idIngredient : CLE ETRANGERE.
- qtyAvailable : non nul, supérieur ou égal à 0.

Recettes d'un utilisateur

- idUsers, idRecipe : CLE PRIMAIRE.
- idUsers : CLE ETRANGERE.
- idRecipe : CLE ETRANGERE.

Catégories d'un ingrédient

- idIngredient, idCategory : CLE PRIMAIRE.
- idIngredient : CLE ETRANGERE.
- idCategory : CLE ETRANGERE.

Ingrédients d'une recette

- idRecipe, idIngredient : CLE PRIMAIRE.
- idRecipe : CLE ETRANGERE.
- idIngredient : CLE ETRANGERE.
- quantity : non nul, supérieur ou égal à 0.
- idCategory : CLE ETRANGERE.

Durée d'une étape

- idStep, idDuration : CLE PRIMAIRE.
- idStep : CLE ETRANGERE.
- idDuration : CLE ETRANGERE.
- durStep : non nul, supérieur ou égal à 0.

Durée d'une recette

- idRecipe, idDuration : CLE PRIMAIRE.
- idRecipe : CLE ETRANGERE.
- idDuration : CLE ETRANGERE.
- durRecipe : non nul, supérieur ou égal à 0.

Qualités diététiques d'un ingrédient

- idIngredient, idQuality : CLE PRIMAIRE.
- idIngredient : CLE ETRANGERE.
- idQuality : CLE ETRANGERE.
- qtyQuality : non nul, supérieur ou égal à 0.

Qualités diététiques d'une recette

- idRecipe, idQuality : CLE PRIMAIRE.
- idRecipe : CLE ETRANGERE.
- idQuality : CLE ETRANGERE.
- qtyQuality : non nul, supérieur ou égal à 0.

Recettes d'un planning

- idPlanning, idRecipe : CLE PRIMAIRE.
- idPlanning : CLE ETRANGERE.
- idRecipe : CLE ETRANGERE.
- dateMeal : non nul, supérieur à la date du jour.
- nbPersMeal : non nul, strictement supérieur à 0.

Plannings d'un utilisateur

- idUsers, idPlanning : CLE PRIMAIRE.
- idUsers : CLE ETRANGERE.
- idPlanning : CLE ETRANGERE.

Plannings archivés d'un utilisateur

- idUsers, idPlanning : CLE PRIMAIRE.
- idUsers : CLE ETRANGERE.
- idPlanning : CLE ETRANGERE.

Ingrédients d'une liste des courses

- idShopping, idIngredient : CLE PRIMAIRE.
- idShopping : CLE ETRANGERE.
- idIngredient : CLE ETRANGERE.
- qtyShopping : non nul, supérieur ou égal à 0.

Plannings d'une liste des courses

- idShopping, idPlanning : CLE PRIMAIRE.
- idShopping : CLE ETRANGERE.
- idPlanning : CLE ETRANGERE.

Listes des courses d'un utilisateur

- idUsers, idShopping : CLE PRIMAIRE.
- idUsers : CLE ETRANGERE.
- idShopping : CLE ETRANGERE.

Listes des courses archivées d'un utilisateur

- idUsers, idShopping : CLE PRIMAIRE.
- idUsers : CLE ETRANGERE.
- idShopping : CLE ETRANGERE.

Régimes alimentaires d'un ingrédient

- idIngredient, idDiet : CLE PRIMAIRE.
- idIngredient : CLE ETRANGERE.
- idDiet : CLE ETRANGERE.

3.3 Contraintes dynamiques

Je ne mentionne pas les contraintes dynamiques précisée dans la deuxième partie du sujet.

Utilisateur

- name : première lettre en majuscule.
- lastName : nom de famille en majuscule.

Etape

- **nameStep** : si le champ n'est pas renseigné, on initialise le nom à 'Etape POIDS', POIDS étant la valeur du champ weight dans la table.

Recettes d'un planning

- **dateMeal** : la date du repas ne peut pas être inférieure à la date d'ajout (j'ajoute une marge de cinq minutes).

Ainsi, à chaque opération sur une recette d'un planning, on modifie la date du planning correspondant et des listes de courses contenant ce planning.

Planning

- **startPlanning** : date de repas, la plus petite.
- **endPlanning** : date de repas, la plus grande.

Ces dates dépendent de chaque repas inséré dans le planning. Sachant qu'un repas est inséré uniquement si sa date est supérieure à la date à laquelle il est ajouté, je ne pense pas avoir besoin d'effectuer des vérifications ici.

Liste des courses

- **startShopping** : date de planning, la plus petite.
- **endShopping** : date de planning, la plus grande.

Ces dates dépendent de chaque planning inséré dans la liste des courses. Ainsi, pour des raisons évidentes, je ne pense pas non plus avoir besoin d'effectuer des vérifications. De plus, un utilisateur pourrait avoir besoin de générer une liste de courses après le début ou la fin d'un planning.

Modèle logique relationnel

4

4.1 Illustration

4.2 Implémentation

Tables principales :

- USERS (idUsers, login, email, pwd, name, lastName, adress)
- INGREDIENT (idIngredient, nameIngredient, unity)
- RECIPE (idRecipe, nameR, author, descR, difficulty, price, nbPers)
- PLANNING (idPlanning, nameP, descP, startP, endP)
- SHOPPING (idShopping, nameS, descS, startS, endS)
- DIET (idDiet, nameDiet)
- CATEGORY (idCategory, nameCategory)
- QUALITY (idQuality, nameQuality)
- MEDIA (idMedia, nameMedia, descMedia, media, #idRecipe)
- STEP (idStep, weight, nameStep, descStep, #idRecipe)
- DURATION (idDuration, nameDuration)

Tables de liaisons :

- STOCK (#idUsers, #idIngredient, qtyAvailable)
- USERS_RECIPE (#idUsers, #idRecipe)
- USERS_PLANNING (#idUsers, #idPlanning)
- USERS_OLD_PLANNING (#idUsers, #idPlanning)
- USERS_SHOPPING (#idUsers, #idShopping)
- USERS_OLD_SHOPPING (#idUsers, #idShopping)
- SHOPPING_PLANNING (#idShopping, #idPlanning)
- SHOPPING_INGREDIENT (#idShopping, #idIngredient, qtyShop)
- PLANNING_RECIPE (#idPlanning, #idRecipe, dateMeal)
- INGREDIENT_DIET (#idIngredient, #idDiet)
- INGREDIENT_CATEGORY (#idIngredient, #idCategory)
- INGREDIENT_QUALITY (#idIngredient, #idQuality, qtyQuality)
- RECIPE_QUALITY (#idRecipe, #idQuality, qtyQuality)
- RECIPE_INGREDIENT (#idRecipe, #idIngre, quantity, #idCategory)
- STEP_DURATION (#idStep, #idDuration, duration)
- RECIPE_DURATION (#idRecipe, #idDuration, duration)

Conclusion

Afin d'apporter plus de contexte, je pense que relier ce type de projet à l'UE "Programmation Web" serait une bonne idée. En effet, le sujet étant assez libre, les interprétations sont nombreuses et certaines zones restent floues.

J'imagine qu'implémenter un site en complément nous permettant d'insérer/-modifier/supprimer des données nous apporterait une vision différente du projet. Ainsi, nous aurions deux projets à relier, ce qui ôterait ces zones floues et permettrait plus d'interprétations.

De plus on pourrait imaginer un planning hebdomadaires de suggestion, en fonction des tendances, d'un régime, etc...

Bibliographie

- [1] LANUTRITION. *Qualité diététique du miel (par exemple)*. Date non mentionnée. URL : <https://sante.journaldesfemmes.fr/calories/miel/aliment-31008>.
- [2] LE JOURNAL DES FEMMES SANTÉ. *Qualité diététique du miel (par exemple)*. Date non mentionnée. URL : <https://www.lanutrition.fr/miel>.
- [3] NON MENTIONNÉ. *Recette : Béchamel rapide et facile*. Date non mentionnée. URL : https://www.marmiton.org/recettes/recette_bechamel-rapide-et-facile_14764.aspx.
- [4] NON MENTIONNÉ. *Recette : Pain d'épices*. Date non mentionnée. URL : https://www.marmiton.org/recettes/recette_pain-d-epices_11087.aspx.
- [5] OCÉANNE ALISSON PRZYMIRSKI. *Comment peser ses ingrédients sans balance ?* 17/04/2020. URL : <https://cuisine.journaldesfemmes.fr/astuces-termes-et-tournemains/1195515-comment-peser-ses-ingredients-sans-balance/>.
- [6] ROMY_15714486. *Recette : Camembert rôti au miel*. Date non mentionnée. URL : https://www.marmiton.org/recettes/recette_camembert-roti-au-miel_45038.aspx.