

UNIVERSITÉ DE STRASBOURG

LICENCE INFORMATIQUE

BASES DE DONNÉES 2

Rapport du projet: Recette de cuisine

Auteur:
Godwin AMEGAH

Professeur:
Mr. Gabriel FREY

November 28, 2021

The logo of the University of Strasbourg, featuring two blue curved segments that form a stylized 'S' shape, positioned behind the text.

UNIVERSITÉ DE STRASBOURG

Contents

1	La modélisation	2
1.1	Le modèle entité association	2
1.1.1	La table Utilisateurs	2
1.1.2	La table Ingredient	2
1.1.3	La table Unites	4
1.1.4	La table Categorie	4
1.1.5	La table Regimes	4
1.1.6	La table Recette	5
1.1.7	La table Etape	5
1.1.8	La table phaseEtape	6
1.1.9	La table medias	6
1.1.10	La table types	6
1.2	Les contraintes d'intégrité	6
1.2.1	La table Utilisateurs	6
1.2.2	La table Ingredients	7
1.2.3	La table Unites	7
1.2.4	La table Categories	7
1.2.5	La table Regimes	7
1.2.6	La table Recettes	7
1.2.7	La table Etapes	7
1.2.8	La table phasesEtapes	8
1.2.9	La table medias	8
1.2.10	La table types	8
1.3	Les relations	8
1.3.1	Liens entre la table Recette et Utilisateurs	8
1.3.2	Liens entre la table Recettes et Etapes	8
1.3.3	Liens entre la table Etapes et PhasesEtapes	8
1.3.4	Liens entre la table Recettes et Medias	8
1.3.5	Liens entre la table Medias et Types	8
1.3.6	Liens entre la table Recettes et Ingrédients	9
1.3.7	Liens entre la table Ingrédients et Categories	9
1.3.8	Liens entre la table Ingrédients et Unites	9
1.3.9	Liens entre la table Ingrédients et Utilisateurs	9
1.3.10	Liens entre la table Regimes et Utilisateurs	9
1.3.11	Liens entre la table Regimes et Ingrédients	10
1.4	Le modèle logique relationnel	10

2	L'implémentation de la base données	11
2.1	Création des tables	11
2.2	Suppression des tables	11
2.3	Insertion de données	11

Introduction générale

Dans le cadre de l'UE de base de données la troisième année de licence d'informatique à l'Université de Strasbourg, il nous a été demandé de mettre en place une base de données pour gérer des recettes de cuisine. En d'autres termes, il s'agit de concevoir une base de données receillant toutes les informations (les recettes, les ingrédients, les utilisateurs inscrits, etc) relatives à un site de recette de cuisine.

Dans ce présent rapport, est détaillé la modélisation, la conception et l'implémentation de la base de donnée.

D'autre par, le travail réaliser respecte au mieux les besoins fonctionnels indiqués dans l'énoncé. Elle pourra donc évoluer en cas d'autres besoins exprimés.

Bonne lecture !

Chapter 1

La modélisation

1.1 Le modèle entité association

Cette partie présente les différentes tables ainsi que les relations qui relient ces dernières. Après une analyse des différentes informations présente dans l'énoncé, voici le modèle entité association qui en découle. Pour des questions de lisibilités, vous pouvez aussi vous référer au dossier:

`images/RecetteDeCuisineJMerise.png`.

Figure 1.1 shows MCD.

1.1.1 La table Utilisateurs

Elle recueillera l'ensembles des utilisateurs inscrits sur notre site web. Un utilisateur est caractérisé par:

idUt Une clé primaire, nombre permettant d'identifier de façon unique un utilisateur.

login nom d'utilisateur permettant de se connecter à la base de données. Il est propre à chaque users donc unique.

email une adresse mail sur lequel on pourra le joindre. Elle est unique.

nom contenant son nom patrimonial. Il peut être renseigner par l'utilisateur ou non.

prenom contenant son prénom. Peut lui aussi être vide.

motDePass le mot de pass de l'utilisateur. Elle sera necessaire pour se logger. Cette information étant sensible, sera crypée avant stockage en interne à l'aide d'un hash du type **sha1** ou autre variante.

adresse contiendra l'adresse postale de l'utilisateur.

1.1.2 La table Ingredient

Elle sera utile pour stocker l'ensemble des ingrédients disponibles. Un ingrédient est caractérisé par:

idIngredient clé primaire, numéro permettant d'indentifier de façon unique un ingrédient.

nomIngredient le nom de l'ingrédient.

calories le nombre de calories qu'il contient.

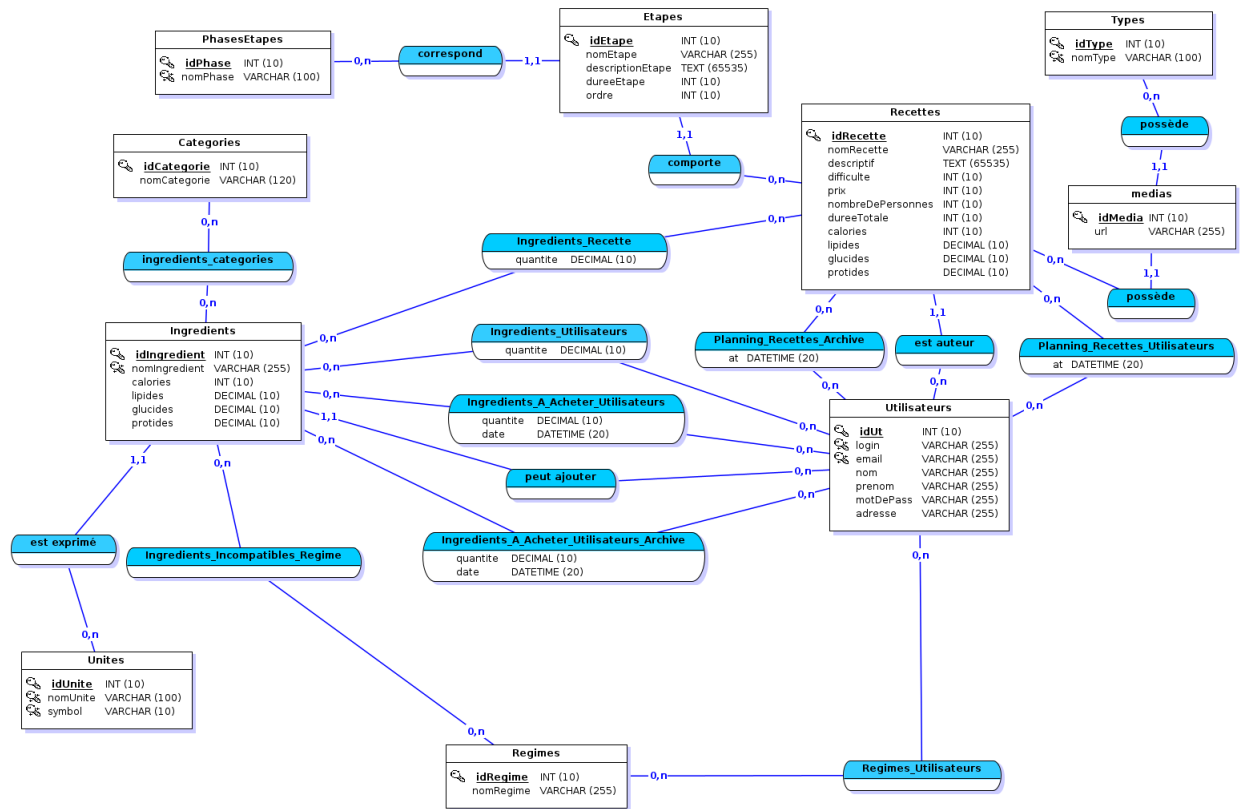


Figure 1.1: Modèle entité association.

lipides nombre décimal représentant la quantité de lipides.

glucides nombre décimal représentant la quantité de glucides.

protides nombre décimal représentant la quantité de protides.

Quand est-il de l'unité ?

En effet, pour pouvoir exprimer la quantité d'un ingrédient, on aura besoin d'une unité de mesure. Cette unité dépendra du type d'ingrédient. On pourra par exemple dire: *4 oeufs* ou *250g de farine* etc. L'unité fera l'objet d'une table.

Pourquoi donc ne pas se contenter d'un champ supplémentaire ?

Dans un premier temps, dans la spécification du sujet, il est indiqué que la quantité pourra être indiqué soit en unité, soit en gramme. Je trouve cela trop restreint en faisant une comparaison avec le monde réel. Dans une tel situation, il sera impossible d'utiliser par exemple le *Litre* et se serait dommage.

En second lieu, cela faciliterait l'ajout d'une nouvelle unité et donc une mise à jour des éléments présents sur notre site sans modifier à chaque fois le code *HTML*. On pourra par exemple via un formulaire *HTML* récupérer dynamiquement les unités présentes dans la BDD et donc les mettre à disposition de l'utilisateur.

1.1.3 La table Unites

Elle contiendra l'ensemble des unités disponibles. Une unité est caractérisé par:

idUnite numéro permettant d'identifier une unité.

nomUnite le nom de l'unité. Elle est unique.

symbole le symbole associé à cette unité. par exemple *g pour gramme, kg pour kilogramme etc..*
Elle peut être NULL mais dans le cas contraire sera unique.

1.1.4 La table Categorie

Elle contiendra l'ensemble des Categorie ou type d'ingrédient disponibles. Par exemple: *Poisson*. On pourra remplacer dans certaines recettes à base de poisson *la lotte* par de *la saumonette*. Elle est caractérisé par:

idCategorie numéro permettant d'identifier un type d'ingrédient

nomCategorie Le nom de la categorie.

1.1.5 La table Regimes

Elle stock l'ensemble des régimes possibles pour un individu. Elle sra caractérisé par:

idRegime numéro permettant d'indentifier un régime particulier

nomRegime le nom du régime. par exemple: *Végétarien, Halal* etc.

1.1.6 La table Recette

Elle contiendra l'ensemble des recettes disponibles. Elle est caractérisé par:

idRecette numéro permettant d'identifier une recette en particulier.

nomRecette le nom de la recette

descriptif La description de la recette.

difficulte une évaluation du type: *Très facile, Facile, Intermédiaire, Difficile, Très difficile.*

prix entier allant de 1 à 5 estimant le prix.

- 1: Gratuit
- 2: Bon marché
- 3: Prix moyen
- 4: élevé
- 5: Très élevé

nombreDePersonne entier représentant le nombre de personne pour lequel la recette est destinée.
Ce nombre peut être ajusté au niveau du site web entrainant ainsi l'ajustement d'autres paramètres comme la quantité des ingrédients etc.

dureeTotal un entier représentant la durée total exprimé en minute pour la réalisation de la recette.

calories entier représentant le nombre de calories fournis par la recette.

lipides nombre décimal indiquant la quantité global de lipides contenu dans la recette.

glucides nombre décimal indiquant la quantité global de glucides

protides nombre indiquant la quantité de global protides.

1.1.7 La table Etape

Elle contiendra l'ensemble des étapes à suivre pour la réalisation de la recette. Elle est caractérisé par:

idEtape numéro identifiant une Etape

nomEtape nom de l'étape. par exemple: *la sauce !*

descriptionEtape une description détaillé de l'étape. par exemple: *"Mélanger la crème à la moutarde. Ajoutez le sel et le poivre"*

dureeEtape la durée de l'étape exprimé en minute.

ordre entier d'indiquant le rang (ou la priorité) dans la suite des étape de réalisation d'une recette.

1.1.8 La table phaseEtape

Elle présente l'ensemble des phases relatives à aux différentes étapes. par exemple la phase de *Cuisson, Préparation, Repos etc.* Elle est caractérisé par:

idPhase numéro identifiant une phase.

nomPhase le nom de la phase.

1.1.9 La table medias

Elle stock l'ensemble des médias disponibles pour les recettes. Par exemple: *photos, images, vidéos, audios etc.* Elle est caractérisé par:

idMedia numéro identifiant un média.

url lien permettant d'accéder au média.

1.1.10 La table types

Elle présente les différents types de média disponibles. Par exemple: *vidéo spot publicitaire, vidéo storytelling, vidéo tutoriel etc..* elle est caractérisé par:

idType numéro identifiant le type de média

nomType le nom du type de média.

1.2 Les contraintes d'intégrité

Pour garantir la cohérence du modèle précédant, les données doivent respecter à tout instant un certain nombre de contraintes.

Les identifiants des tables sont des clés primaires et par définition doivent être **unique** et **non NULL**.

1.2.1 La table Utilisateurs

login est propre à chaque utilisateur, donc unique

email doit être unique, doit respecter le format d'une adresse valide c'est-à-dire de la forme `[au_moins_un_caractère]@[au_moins_un_caractère]`. Ce traitement pourra se faire dans un langage dynamique à l'instar de *php* grâce aux *rége*x.

motDePass doit être **non vide**.

nom doit être **non vide**.

prenom doit être **no vide**.

1.2.2 La table Ingredients

nomIngredient doit être unique et non NULL

calories doit être positive ou null ≥ 0 .

lipides doit être positive ou null ≥ 0 .

glucides doit être positive ou null ≥ 0 .

protides doit être positive ou null ≥ 0 .

1.2.3 La table Unites

nomUnite doit être non NULL et unique.

symbole doit être unique mais peut être NULL. C'est le cas des ingrédients quantifiables par unité. Par exemple: *oeufs*

1.2.4 La table Categories

nomCategorie doit être unique, non NULL.

1.2.5 La table Regimes

nomRegime doit être unique, non NULL.

1.2.6 La table Recettes

nomRecette doit être non NULL.

difficulte doit appartenir à l'ensemble $\{Très facile, Facile, Intermédiaire, Difficile, Très difficile\}$.

prix doit être compris entre 1 et 5.

nombreDePersonne doit être positive ou null ≥ 0 .

dureeTotale doit être positive ou null ≥ 0 .

calories doit être positive ou null ≥ 0 .

lipides doit être positive ou null ≥ 0 .

glucides doit être positive ou null ≥ 0 .

protides doit être positive ou null ≥ 0 .

1.2.7 La table Etapes

nomEtape doit être non NULL.

dureeEtape doit être positive ou null ≥ 0 .

ordre doit être positive ou null ≥ 0 .

1.2.8 La table `phasesEtapes`

`nomPhase` doit être unique, non NULL.

1.2.9 La table `medias`

`url` doit être non NULL.

1.2.10 La table `types`

`nomType` doit être unique, non NULL.

1.3 Les relations

Le modèle précédent présente un ensemble étiqueté d'entités chacune jouant un rôle particulier. Il s'agit des relations/liasons indiquant les liens entre différentes tables. Les exemples suivantes expliquent la sémantique des liens entre les différentes tables.

1.3.1 Liens entre la table `Recette` et `Utilisateurs`

Un utilisateur inscrit dans la base de données peut créer et donc être l'auteur de 0 à `n` recettes et une recette est éditée par 1 et 1 seul utilisateur.

Un utilisateur peut établir un planning contenant 0 à `n` recettes et une recette peut faire l'objet de 0 à `n` planning. Il faut donc par la suite créer une table `Planning_Recettes_Utilisateurs` qui établira la correspondance utilisateur (`idUt`) et la recette (`idRecette`) prévu dans son planning.

Comme il s'agit d'un planning, cette table devra contenir en plus un champ (`at`) de type `datetime` indiquant la date prévue pour cette recette. Une des contraintes que devra respecter ce champ est que sa valeur doit être comprise entre l'instant `t` d'établissement du planning et les `n` prochains mois (*`n` fixé par l'admin. Pour simplifier on supposera que `n` vaut 1*).

Si cette date est dépassée, le planning sera archivé c'est-à-dire conservé dans une autre table `Planning_Recettes_Archive`.

1.3.2 Liens entre la table `Recettes` et `Etapes`

Une recette peut comporter 0 (*Si elle vient juste d'être créée*) à `n` étapes. Par contre une étape est relative à une recette et donc est utilisée dans 1 et 1 seule recette.

1.3.3 Liens entre la table `Etapes` et `PhasesEtapes`

A une étape on peut faire correspondre 1 et 1 seule phase (par exemple: le fait de couper les légumes correspondre à la phase de préparation) et une phase peut comporter 0 à `n` étapes différentes.

1.3.4 Liens entre la table `Recettes` et `Medias`

Une recette peut posséder 0 à `n` médias et un média référence à 1 et 1 seule recette.

1.3.5 Liens entre la table `Medias` et `Types`

Un média est d'1 et 1 seul type donné et un type de média peut regrouper 0 à `n` médias différents.

1.3.6 Liens entre la table Recettes et Ingrédients

Une recette est composée de 0 (*si elle vient juste d'être créée*) à n ingrédients et une recette peut être utilisée dans 0 à n recette.s. Il faut donc par la suite créer une table **Ingredients_Recettes** qui établira la correspondance ingrédient (*idIngrédient*) et la recette (*idRecette*) dans laquelle elle est utilisée.

Dans cette table, Il faudrait disposer aussi d'un champ (**quantite**) de type décimal indiquant la quantité de cet ingrédient nécessaire pour la recette.

1.3.7 Liens entre la table Ingrédients et Categories

Un ingrédient peut appartenir à 0 à n catégorie.s et une catégorie regroupe 0 à n ingrédients. Il faut donc par la suite créer une table **Ingredients_Categories** qui établira la correspondance ingrédient (*idIngrédient*) et la catégorie (*idCategorie*) dans laquelle elle appartient.

1.3.8 Liens entre la table Ingrédients et Unites

Un ingrédient est quantifié/mesuré à l'aide d'une 1 et 1 seule unité et une unité est utilisée pour mesurer 0 à n ingrédient.s. Il faut donc par la suite créer un champ dans la table **Ingredients** qui fera référence à *idUnite* de la table **Unites**.

1.3.9 Liens entre la table Ingrédients et Utilisateurs

Un utilisateur peut disposer de 0 à n ingrédients et un ingrédient peut être possédé par 0 ou n utilisateur.s. Il faut donc par la suite créer une table **Ingredients_Utilisateurs** qui établira la correspondance ingrédient (*idIngrédient*) et son possesseur (*idUt*). Dans cette table, Il faudrait disposer aussi d'un champ (**quantite**) de type décimal indiquant la quantité de cet ingrédient que possède l'utilisateur.

Un utilisateur pourra comparer la liste des ingrédients dont il dispose avec la liste des ingrédients nécessaires pour réaliser les recettes sur son planning. Dans ce cas il peut ajouter 0 à n ingrédient.s et un ingrédient est ajouté par 1 et 1 seul utilisateur.

Il faut donc par la suite créer un champ dans la table **Ingredients** qui fera référence à *idUt* de la table **Utilisateurs**.

En fonction des ingrédients qui lui manque, l'utilisateur pourra établir une liste d'achat. Un utilisateur peut donc acheter 0 à n ingrédient.s et un ingrédient peut faire l'objet d'un achat par 0 à n utilisateur.s. Il faut donc par la suite créer la table **Ingredients_A_Acheter_Utilisateurs** qui établira la correspondance ingrédient (*idIngrédient*) et l'utilisateur (*idUt*) qui l'achète.

Dans cette table, Il faudrait disposer aussi d'un champ (**quantite**) de type décimal indiquant la quantité de cet ingrédient que va acheter l'utilisateur ainsi qu'un champ (**date**) de type **datetime** indiquant la date d'achat.

Une des **contraintes** (*statique*) que devra respecter le champ *date* est que sa valeur doit être comprise entre l'instant t d'établissement de la liste d'achat et les n prochains mois (*n fixé par l'admin. Pour simplifier on supposera que n vaut 1*).

L'autre **contrainte** (*dynamique*) sera tel que, si cette date est dépassée, la liste sera archivée c'est-à-dire conservée dans une autre table **Ingredients_- A_Acheter_Utilisateurs_Archive**.

1.3.10 Liens entre la table Regimes et Utilisateurs

Un régime peut être suivi par 0 à n utilisateur.s et un utilisateur peut suivre 0 à n régime.s. Il faut donc par la suite créer une table **Regimes_Utilisateurs** qui établira la correspondance régime (*idRegime*) et l'utilisateur (*idRecette*) qui le suit.

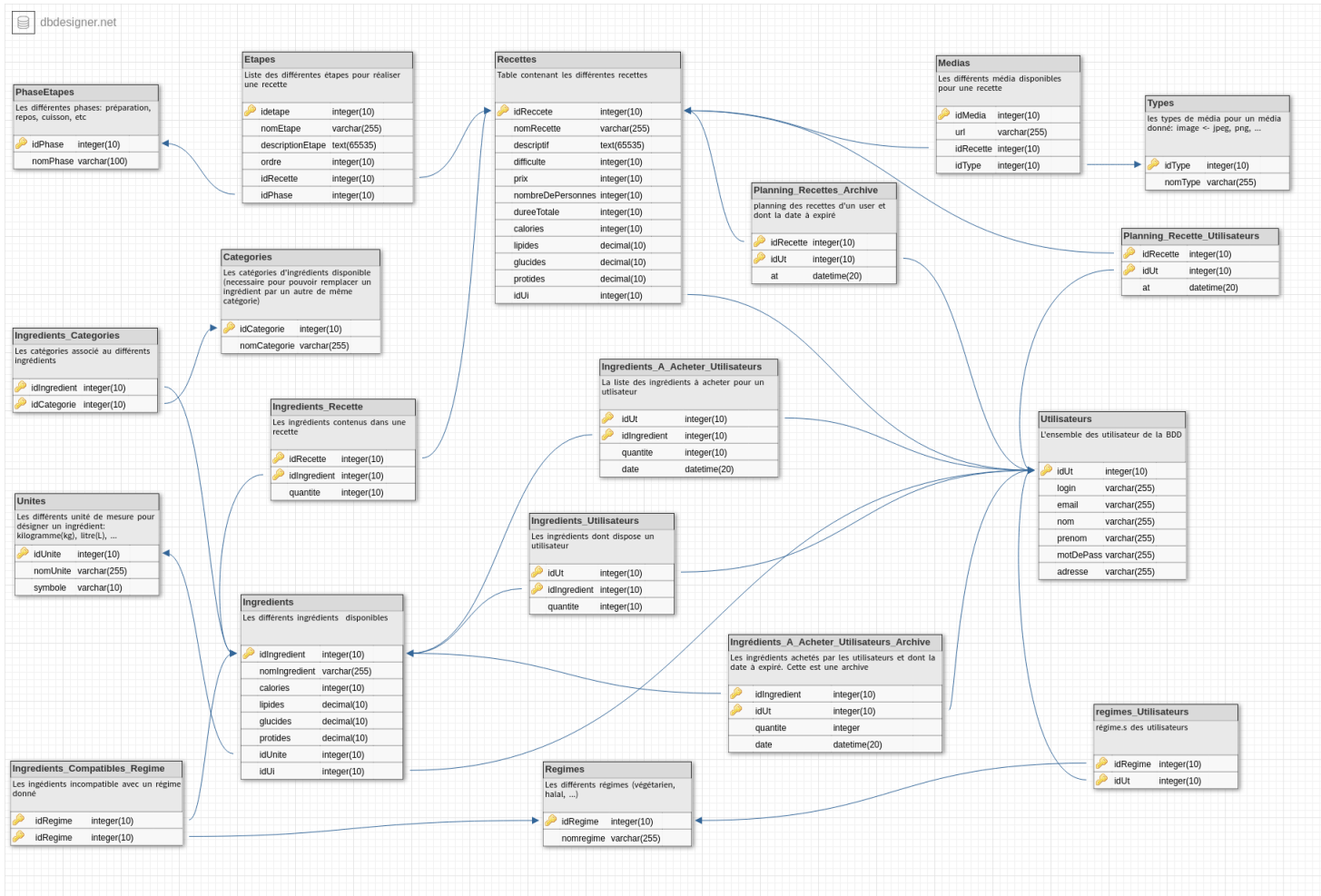


Figure 1.2: Modèle logique relationnel.

1.3.11 Liens entre la table Regimes et Ingrédients

Un ingrédient peut être incompatible à 0 à n régime.s et un régime peut exclure 0 à n ingrédient.s. Il faut donc par la suite créer une table **Ingredients_Incompatibles_Regimes** qui établira la correspondance régime (*idRegime*) et l'ingrédient (*idIngredient*) exclus.

1.4 Le modèle logique relationnel

Cette partie présente les différentes tables ainsi que l'ensemble des relations déduites du *Modèle entité association*. voici donc le **modèle relationnel** correspondant. Pour des questions de lisibilités, vous pouvez aussi vous référer au dossier *images/index.png*.

Figure 1.2 shows MLR.

Chapter 2

L'implémentation de la base données

2.1 Création des tables

confert le repertoire `/requetes-pl-sql/RDC_create_table.sql`

2.2 Suppression des tables

confert le repertoire `/requetes-pl-sql/RDC_drop_table.sql`

2.3 Insertion de données

confert le repertoire `/requetes-pl-sql/RDC_insert.sql`

Bibliography

- [1] Marmiton. "*Marmiton : 70000 recettes de cuisine ! Recettes commentées et notées pour toutes les cuisines.*" RECETTE. Dernière modification Mai 2020. (visité le 5 Novembre 2021). URL: <https://www.marmiton.org/>
- [2] Les differents types de régimes.(visité le 7 Novembre 2021). URL: <https://www.domyos.fr/nutrition-les-differents-types-de-regime>
- [3] Tableau de calories.(visité le 7 Novembre 2021). URL: <https://www.tableaudecalories.net/aliment/viande>
- [4] Generation de mot de pass. (visité le 13 Novembre 2021) URL: <https://www.motdepasse.xyz/>