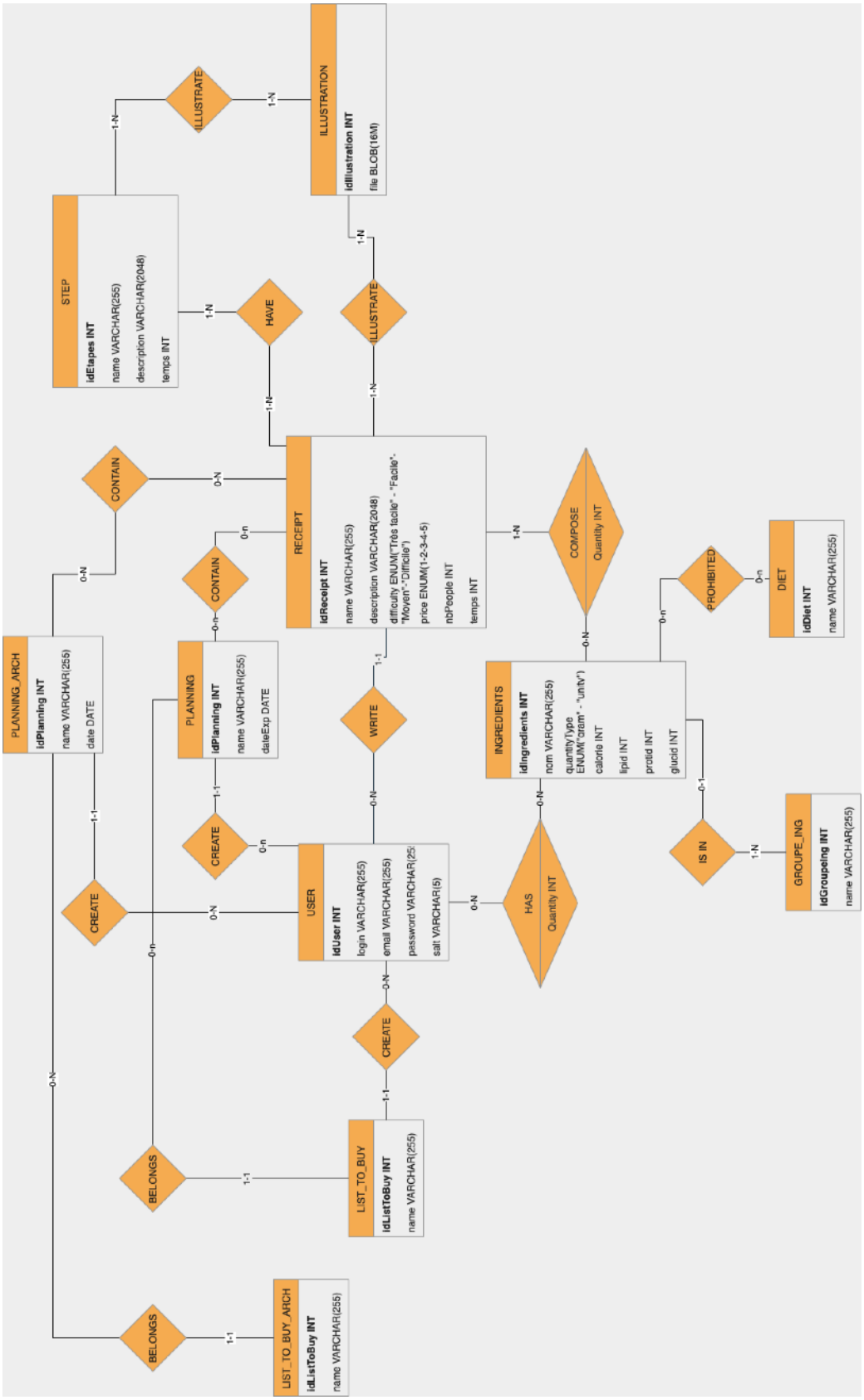
Nathanaël Derousseaux—Lebert

Projet base de données : recettes de cuisine

Partie n°1

I - Modèle entité association

En lisant le sujet, on peut dessiner un modèle entité association tel que celui là :



Une version svg du modèle est disponible dans le dossier modele_ea du rendu.

II - Modèle relationnel

A partir du modèle Entités/Association, on peut écrire le modèle relationnel :

Dans un cas réel, il pourrait être pertinent d'ajouter à chaque enregistrement de chaque table un champ date_creation et date_modification pour rendre l'interface plus intuitive. De la même manière il manque beaucoup de champs à la table USER pour correspondre à un cas réel (date de vérification de l'adresse email, liste des précédents mot de passes, date de dernière connection etc...) cependant, pour des raison de lisibilité, ces champs seront ignorés.

Tables métier

Voici ici la liste des tables qui représentent des objets concrets.

```
USER(idUser:int, login:varchar(255), email:varchar(255), password:varchar(255),
salt:varchar(5))
RECEIPT(idReceipt:int, name:varchar(255), description:varchar(2048),
difficulty:varchar(30), price:int, nbPeople:int, temps:int, #idUser:int)
STEP(idStep:int, name:varchar(255), description:varchar(255), temps:int)
ILLUSTRATION(idIllustration:int, file:blob)
INGREDIENT(idIngredient:int, name:varchar(255), quantityType:varchar(50),
calorie:int, lipid:int, portid:int, glucid:int, #idGroupe:int)
DIET(idDiet:int, name:varchar(255))
GROUPE_ING(idGroupeIng:int, name:varchar(255))
PLANNING(idPlanning:int, name:varchar(255), dateExp:date, #idUser:int)
LIST_TO_BUY(idListToBuy:int, name:varchar(255), #idPlanning:int, #idUser:int)
PLANNING_ARCH(idPlanning:int, name:varchar(255), dateExp:date, #idUser:int)
LIST_TO_BUY_ARCH(idListToBuy:int, name:varchar(255), #idPlanning:int,
#idUser:int)
```

Les attributs temps dans RECEIPT et STEP représentent le temps d'exécution en minutes.

La table DIET contiendra les différents type de régime alimentaire : végétarien, sans gluten, etc...

La table GROUPE_ING permet de ranger les ingrédients dans des groupes, dans une recette ingrédient pourra être substitué aux autres ingrédients du même groupe.

La table PLANNING représente le planning de recette que souhaite faire l'utilisateur. Le planning à une date d'expiration, à partir de laquelle il serra archivé.

La table LIST_TO_BUY contient un enregistrement si l'utilisateur souhaite créer une liste d'ingrédients à acheter à partir d'un planning de recettes. Il est inutile de stocker spécifiquement les ingrédients dans cette table, puisque l'on connait déjà les ingrédients que possède l'utilisateur (dans la table d'association USER_INGREDIENT). Il sera plus pertinent de créer cette liste à la volé, à chaque fois que l'utilisateur voudra la consulter. Cela évitera la duplication de données.

Les tables PLANNING_ARCH et LIST_TO_BUY_ARCH, sont des archives des plannings et des listes d'achats qui ont dépassées leur date d'expiration.

Tables relationnelles

On liste ici les tables d'association entre deux tables métier, lorsque la cardinalité est de type N;N.

```
RECEIPT_STEP(#idReceipt:int, #idStep:int)
RECEIPT_ILLUSTRATION(#idReceipt:int, #idIllustration:int)
STEP_ILLUSTRATION(#idStep:int, #idIllustration:int)
RECEIPT_INGREDIENT(#idReceipt:int, #idIngredient:int)
USER_INGREDIENT(#idUser:int, #idIngredient:int quantity:int)
INGREDIENT_DIET(#idIngredient:int, #idDiet:int)
PLANNING_RECEIPT(#idPlanning:int, #idReceipt:int)
PLANNING_ARCH_RECEIPT(#idPlanning:int, #idReceipt:int)
```

La table RECEIPT_STEP permet d'associer les étapes aux recettes. La cardinalité N;N me semble pertinente, car il n'est impossible qu'une étape serve à plusieurs recette. De la même manière pour STEP_ILLUSTRATION et RECEIPT_ILLUSTRATION : une même illustration peut servir à plusieurs endroits.

La table USER_INGREDIENT liste les ingrédients que possède l'utilisateur, et leur associe une quantité.

La table INGREDIENT_DIET liste les ingrédients interdit pour un régime spécifique.

La table PLANNING_RECEIPT liste les recettes associés à un planning de recettes.

III - Contraintes d'intégrités

La grande majorités des contraintes d'intégrités sont indiqués implicitement sur le modèle entités/associations précédant. En effet les clés primaires sont indiqués, les clés étrangères sont implicitement déclarés dans les associations et les cardinalités sont clairement indiqués. Cependant il reste quelques contraintes qu'il faut expliciter.

Contraintes statiques

- Dans la table USER, le login et l'email devront être unique. De plus, aucun des attributs de cette table ne pourra être nul. L'email de l'utilisateur doit respecter le format classique d'un email.
- Dans la table RECEIPT, l'attribut name ne pourra pas être nul. Afin de simuler le type ENUM dans Oracle, il faudra créer une contrainte statique qui vérifiera que la valeur de l'attribut difficulty est bien contenu dans la liste ["Très facile", "Facile", "Moyen", "Difficile"]. On fera de même avec l'attribut price, qui devra être compris entre 1 et 5.
- Dans la table INGREDIENT, les attributs name et quantityType ne pourront pas être nul. Afin de simuler le type ENUM dans Oracle, il faudra créer une contrainte statiquel qui vérifiera que la valeur de l'attribut quantityType soit contenu dans la liste ["gram","unity"].
- Dans la table STEP, l'attribut description ne pourra pas être nul.
- Dans la table ILLUSTRATION, l'attribut file ne pourra pas être nul.
- Dans la table DIET, l'attribut name ne pourra pas être nul.
- Dans la table PLANNING (et PLANNING_ARCH), l'idUser ne peut pas être nul.
- Dans la table LIST_TO_BUY (et LIST_TO_BUY_ARCH), les idUser et idPlanning ne peuvent être nul

Contraintes dynamiques

- Dans la table USER,L'attribut SALT devra prendre une valeur aléatoire.
- Dans la table PLANNING, il sera impossible de créer un planning de recette dont la date d'expiration est inférieure à la date d'aujourd'hui.
- Dans la table PLANNING_CHECK, il sera impossible de créer un enregistrement dont la date d'expiration est supérieure à aujourd'hui.
- Dans la table RECEIPT, l'attribut temps doit être au moins égal à la somme du temps de ses étapes.

IV - Implémentation

Le script de création des tables, d'insertion de données de test, et celui de suppression des tables sont disponible dans le dossier implementation du rend.