

# PROJET

## Objectifs

Ce projet vise à mettre en œuvre les méthodes et concepts vus en BD1 et BD2 pour réaliser une petite base de données et quelques fonctionnalités sur celle-ci. En théorie le processus général à suivre ressemble à ceci :

### SCHÉMA LOGIQUES

1. Récolte des informations réelles permettant de concevoir le schéma de très haut niveau
2. Détermination des Dfs  
Domaines, CIs, droits, ...
3. Réalisation du schéma de très haut niveau (y compris schéma externe)
4. Passage de ce schéma de très haut niveau à un schéma de haut niveau (schéma externe → vues)
5. Normalisation/décomposition des relations <sup>a</sup>

### SCHÉMA PHYSIQUE

6. Quel est le volume de données à traiter ? → faut-il définir des index ?  
Analyse des “types” de requêtes qui vont être mises en œuvre sur la base  
Accès fréquents à quels attributs ? → index primaires/secondaires
7. Quelle utilisation de la base ?  
Consultation ou modification des données ? Combien d'utilisateurs ?  
Modification des données (intensive) → gestion de la concurrence
8. APPLIS

Dans le cadre de ce projet nous allons le simplifier, et seuls les points suivants seront à traiter :

- Vous partirez d'un schéma de très haut niveau (E-A) déjà construit.
- Il vous faudra tout de même définir les domaines / CIs <sup>b</sup> / Droits
- Vous réaliserez le passage au relationnel
- Vous définirez les index nécessaires
- Vous implanterez cette base
- Vous réaliserez un ensemble de fonctionnalités sur cette base

<sup>a</sup>. en général en partie dénormalisé par la suite pour des questions de performance (e.g. trop de tables ⇒ beaucoup de jointures potentielles)

<sup>b</sup>. Attention, certaines CIs seront plus ou moins explicitement précisées dans le sujet ... mais pas toutes !



## Livrables & organisation

Ce travail est à réaliser en binôme. Deux séances de TP sont affectées à la réalisation du projet (XX/XX/2015, XX/XX/2015). **Chaque membre du binôme** déposera sur moodle au plus tard le XX/XX/2015 12h00, sous la forme d'une archive intitulée Nom1-Nom2.tgz l'ensemble des éléments suivants :

- Le rapport *rapport\_Nom1-Nom2.pdf* (une quinzaine de pages) qui explique votre démarche, vos choix etc. Le squelette de ce rapport -à utiliser obligatoirement- est disponible sur moodle sous l'archive rapport.tgz.
- 2 fichiers .sql :
  - *creation.sql* cré la base/les fonctions/droits/index... et qui la remplit.
  - *suppression.sql* qui supprime tout
- Le remplissage de vos tables doit être fait avec les **données fournies**. Ces données accompagnées de leurs domaines sont disponibles sur moodle (donnees.dat). Attention, l'évaluation de votre projet sera réalisé avec ces données.
- Un fichier *appli.pgc* qui contient votre "appli"
- Un fichier *script-compile* permettant de compiler votre projet

→ Le contenu de votre base est à réaliser dans la base grtti qui vous est affectée pour le projet.

## 1 La base de données

Nous souhaitons gérer une base de données de recettes de cuisine. Soit le schéma E-A Figure 1. Pour simplifier, les vues ont été décrites à part Figure 2. Une partie de la sémantique associée aux schéma est la suivante :

### Schéma principal

#### 1. Entités

- une PERSONNE possède un identifiant unique (idPers), un nom et un téléphone. Elle possède également un attribut type\_lien qui peut prendre les valeurs : Famille, Ami, Collègue. Une personne peut apparaître dans la base sans avoir encore été invitée à manger.
- une RECETTE possède un identifiant unique (idRec), un titre, le temps en minutes nécessaire à la réaliser et à la cuisson (ou refroidissement), la note qui lui a été attribuée (entre 0 et 10), le détail de la recette (description), et s'il s'agit d'un Amuse-gueule, d'une Entrée, d'un Plat ou d'un Dessert (type).
- un INGRÉDIENT possède un nom (supposé unique) et un prix en euros. L'attribut unité précise s'il s'agit d'un prix au kilo, au 100g au litre, ou à l'unité.
- un RANGEMENT possède un identifiant unique, un nom, une description qui explique où est rangée la recette, et un type dont la valeur est : Manuscrit, Livre, Url ou Magazine. S'il s'agit d'un Magazine ou Url, le champ date est renseigné (respectivement date de sortie du magazine et date de dernière consultation du site). Plusieurs rangements peuvent avoir le même nom, sauf s'ils sont de type différent.

#### 2. Associations

- association réaliserPour(RECETTE - PERSONNE) : une RECETTE peut avoir été réalisée pour une ou plusieurs PERSONNE(S), dans ce cas on précise la date et *éventuellement* la nature de l'évènement dont la valeur est : Anniversaire, Fête, Noël, Nouvel-An. Si l'évènement est Noël ou Nouvel-An, le champ date doit se remplir automatiquement.
- association contenir(RECETTE - INGRÉDIENT) : une RECETTE est composée d'INGRÉDIENT(S), et on précise la quantité nécessaire pour chacun.
- association ranger(RECETTE - RANGEMENT) : une RECETTE est rangée dans un certain RANGEMENT.

## Schéma externes

1. la vue LISTEEVALUATIONRECETTE permet d'afficher pour chaque recette (titre, type, temps de préparation et de cuisson) un attribut *appréciation* calculé en fonction de la note à cette recette : note < 5 BOF, note ≥ 5 et ≤ 7 BON, note > 7 EXCELLENT.
2. la vue LISTEINVITER permet d'afficher la liste des personnes (nom, type\_lien, téléphone) avec un attribut aInviter qui prend la valeur VRAI si la personne n'a pas encore été invitée, FAUX sinon et la date de la dernière invitation est précisée le cas échéant, ainsi que le nombre de fois où la personne a été invitée.

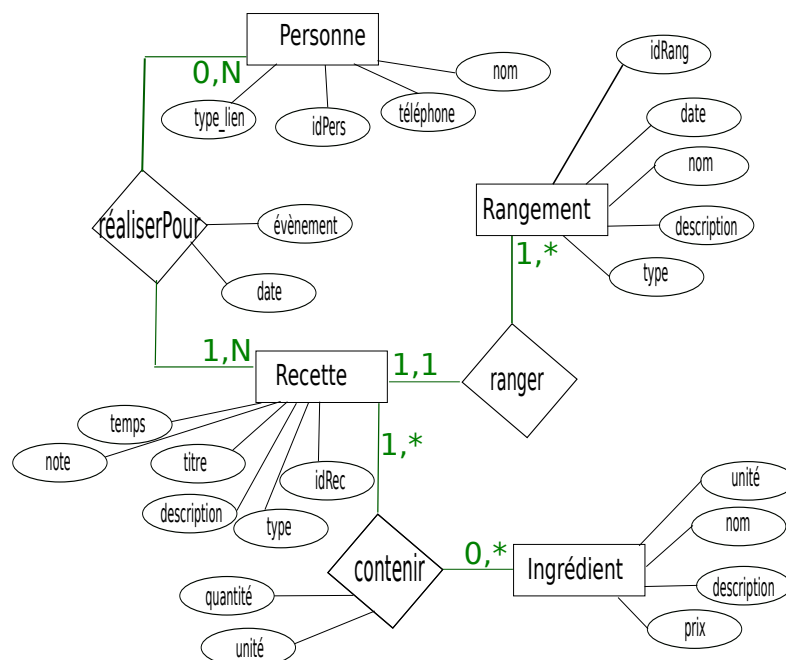


FIGURE 1 – Schéma E-A

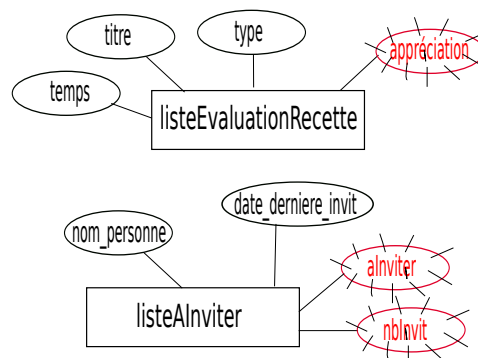


FIGURE 2 – Schéma externes

**Droits** Les droits suivants devront être définis sur cette base :

- grtt42 aura droit de consultation et de mis à jour sur l'ensemble des relations de la base
- choisissez un binôme ami qui aura seulement le droit de consultation sur la base

**Informations relatives aux types de requêtes potentielles sur la base (en plus de l'appli)**

- On veut pouvoir rechercher des recettes selon la première lettre de leur titre
- La recherche d'une personne est souvent effectuée en se basant sur le type de lien

## 2 Les fonctionnalités

L'application finale a pour but de gérer :

1. l'insertion/la suppression d'une recette dans la base. L'ajout ou la suppression s'entend avec les ingrédients **contenus** dans la recette :
  - Pour la suppression l'ingrédient ne doit pas être supprimé de la base mais juste de la recette.
  - Pour l'insertion, si l'ingrédient n'est pas présent dans la base il faut l'ajouter
2. un affichage de la liste des ingrédients (nom, quantité, unité) d'une recette (fournir son identifiant)
3. un affichage de la liste des recettes évaluées (cf. vue listeEvaluationRecette)  
Facultatif : si les temps sont supérieurs à 60mn, les convertir en heure(s) et minute(s).
4. un affichage des invités/non invités (cf. vue listeAInviter)
5. un affichage de la liste des recettes (titre, type) réalisées pour une personne donnée (fournir son nom)
6. une recherche multi-critères (type, prix max, temps préparation max, nom personne)
7. une recherche de recette(s) par liste d'ingrédients (1) tous les ingrédients doivent/ne doivent pas s'y trouver pour sélectionner la recette (2) au moins un ingrédient doit/ne doit pas s'y trouver pour sélectionner la recette

Elle prendra la forme d'un menu comme présenté Figure 3.

FIGURE 3 – Maquette générale de l'application

→ Un exemple d'exécution de l'appli est disponible sur moodle.



### Rappel

- Pour vous connecter à la base de données 'grtti', tapez dans un terminal :  
`psql -h asi-pg.insa-rouen.fr -U grtti grtti`

Vous remplacerez le 'i' du nom d'utilisateur et de la base 'grtti' par le numéro de binôme qui vous aura été attribué par l'enseignant.

Le mot de passe est de manière absolument pas sécurisée le nom d'utilisateur.

- Quelques commandes :

```
\h aide mémoire des commandes
\q pour terminer la connexion à la base
q pour sortir de l'affichage d'une table
```

En particulier la commande `\i nomfichier.sql` permet d'exécuter l'ensemble des requêtes présentes dans le fichier (mettre le fichier dans le répertoire d'où vous vous connectez à la base, ou bien spécifiez le chemin d'accès au fichier).

- **En complément du cours**, la documentation suivante peut vous aider : <http://docs.postgresqlfr.org/9.1/>