TP2 – Les modèles

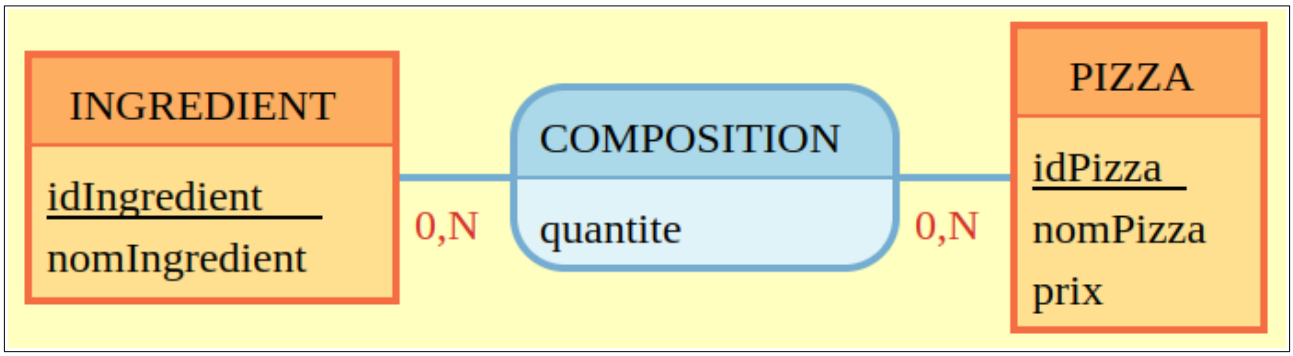
Introduction, Contexte et Objectifs

Django permet de créer à la main les modèles (équivalent des classes). Ensuite, par le mécanisme des migrations, les tables de données sont automatiquement ajoutées à la base de données.

Nous avons donc besoin du schéma relationnel de l'application applipizza. Pour le moment nous nous contenterons d'un MCD simple, avec deux entités INGREDIENT et PIZZA, et une association COMPOSITION. Un ingrédient peut entrer dans la composition de plusieurs pizzas, et une pizza est composée d'un certain nombre d'ingrédients.

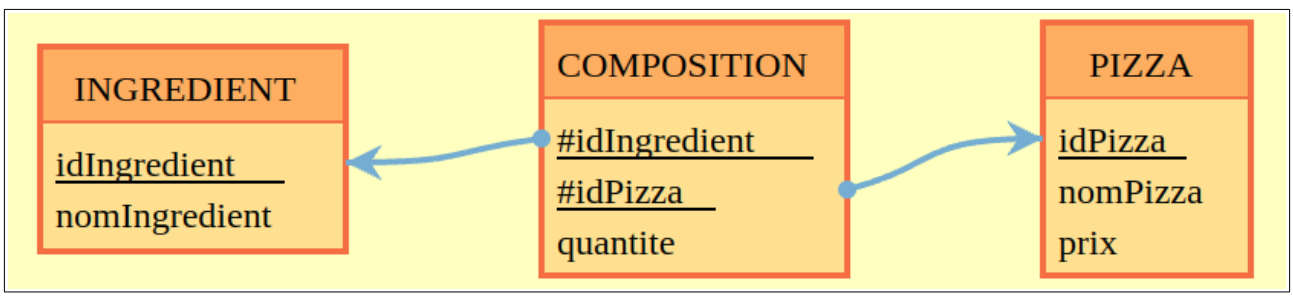
Plus tard, nous modifierons le modèle de Pizza pour lui donner un attribut supplémentaire (une image). Ce sera un complément intéressant.

MCD



Le schéma relationnel donne en théorie les trois tables suivantes, avec deux clés étrangères. La table COMPOSITION aurait pour clé primaire le couple (idIngredient, idPizza).

SR

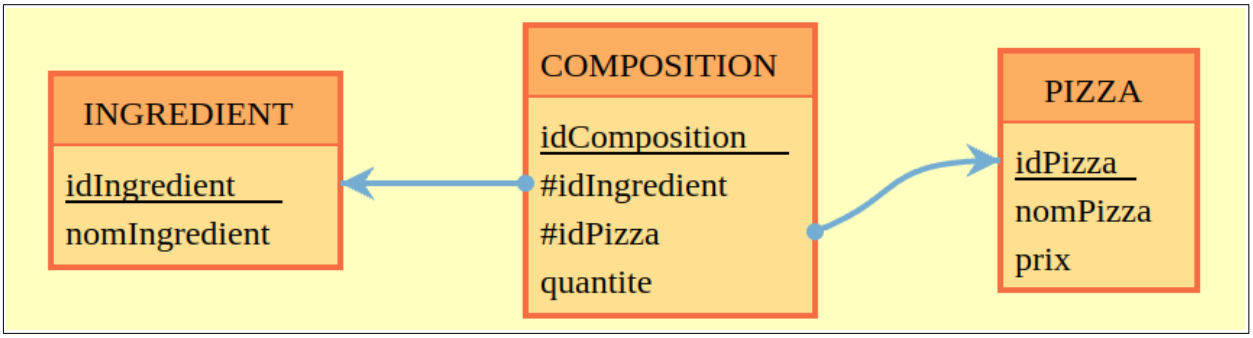


Avec la base de données sqlite locale, l'adaptation de COMPOSITION va faire apparaître :

- une clé primaire idComposition,
- une contrainte d'unicité pour le couple de clés étrangères (idIngredient, idPizza).

Ce n'est pas vraiment différent. Le schéma pour notre projet sera donc plutôt le suivant, qui est équivalent au précédent :

SR



Les modèles avec Django

Une fois que le schéma relationnel est établi, nous allons confier à django le soin de créer les tables de données, en lui indiquant seulement quelles classes nous souhaitons construire.

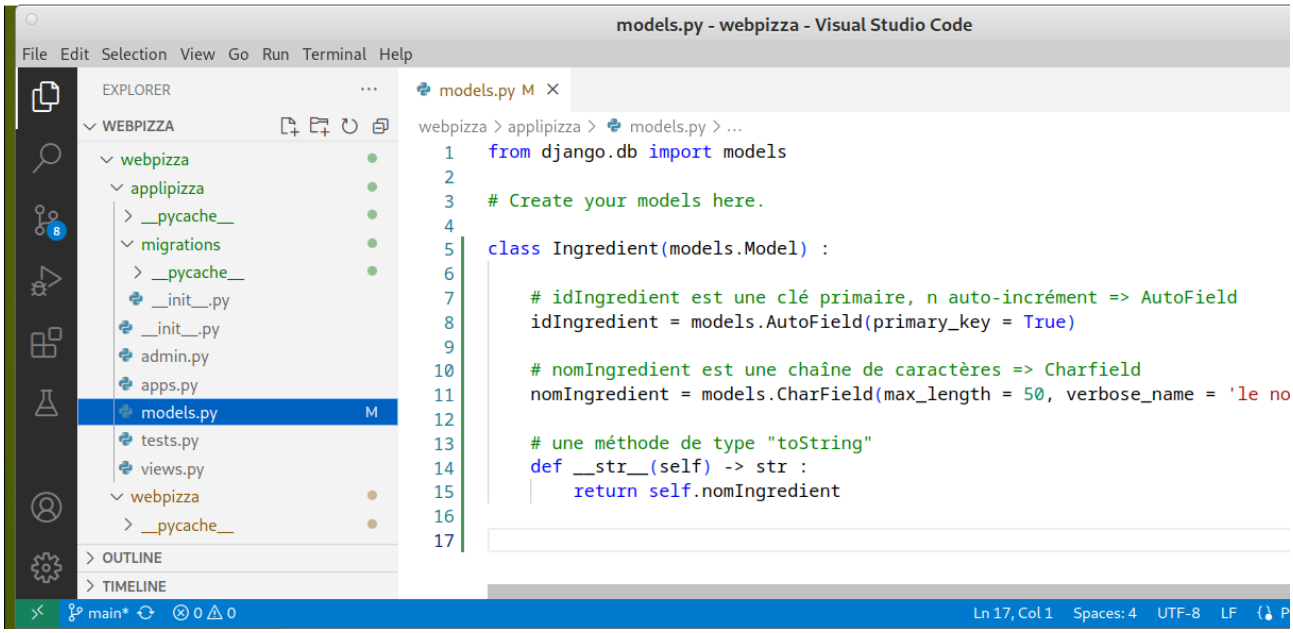
Ceci permet une cohérence forte qui pouvait être mise en défaut dans le système classique PHP / MySQL pur, puisque la création de tables n'était pas automatisée à partir des classes PHP construites.

Notre première étape est donc de construire les classes python, et ceci se fait dans le fichier models.py du projet.

1. Ajout des modèles dans models.py

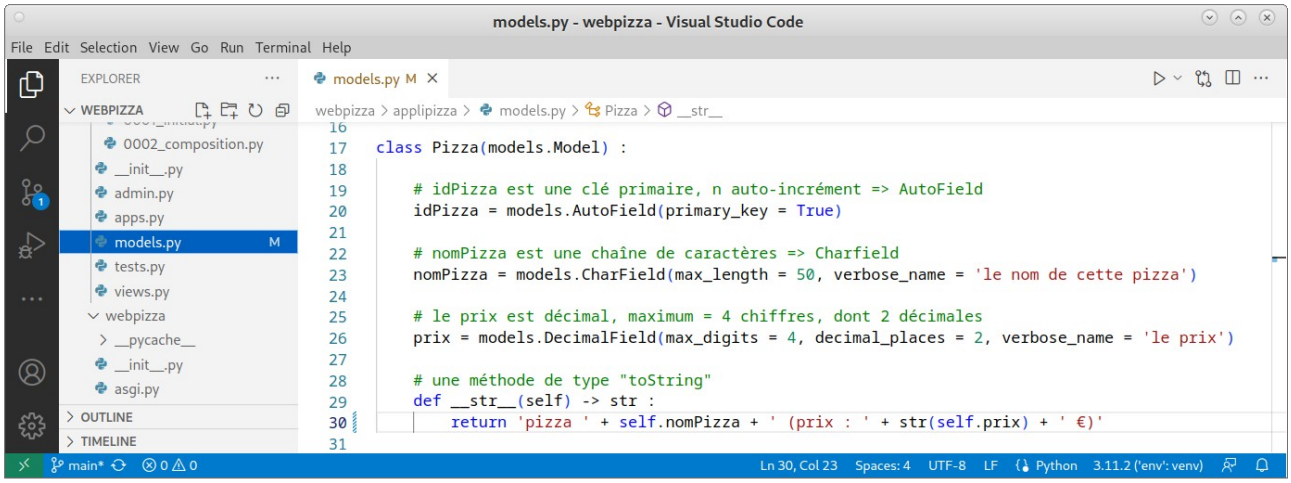
   a. changez le fichier models.py du répertoire applipizza pour qu'il définisse la classe Ingredient. Cette classe héritera de la classe Model de la bibliothèque models. On indique les champs idIngredient et nomIngredient, et on définit la méthode __str__ , équivalent de toString() en java.

      Le code suivant est donné en modèle :



voyez cette page pour plus de documentation : https://docs.djangoproject.com/fr/4.1/topics/db/models/

b. Créez de même la classe Pizza (dans le même fichier, à la suite de la classe Ingredient). Vous pourrez utiliser le type DecimalField pour le prix.



2. Migrez les modèles vers la base de données

   a. Dans le répertoire webpizza (là où il y a le fichier qui gère les commandes, manage.py), lancez la commande

      python3 manage.py makemigrations

      qui prépare l'adaptation de la base de données en fonction des nouveaux modèles.

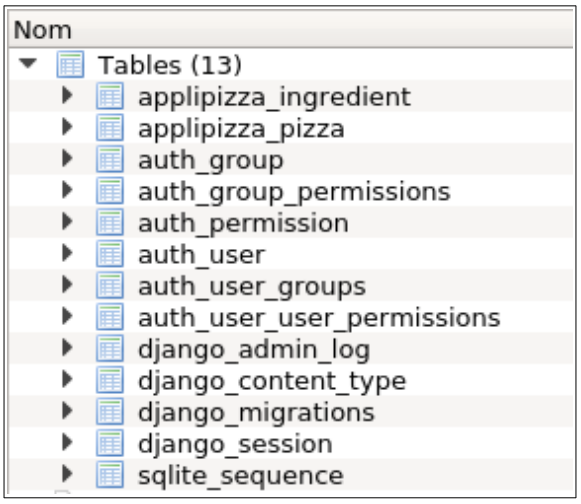
      Cela produira un affichage comme :

      Cet affichage montre les deux modèles créés. Un fichier de migration nommé 0001_initial.py a aussi été écrit, dans lequel vous trouverez les instructions qui permettront de créer la structure de données dans la base sqlite.

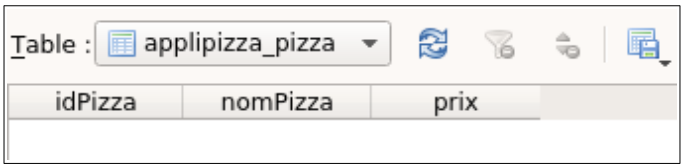
   b. Ensuite, validez la migration par la commande

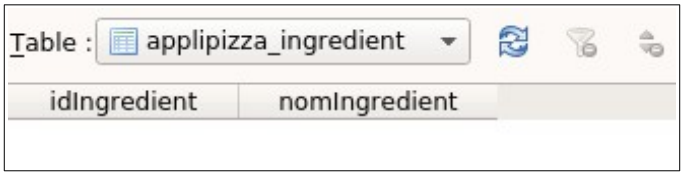
      python3 manage.py migrate

c. Constatez l'effet sur la base de données en ouvrant sqlitebrowser. Pour le moment vous ne voyez que la structure des tables Ingredient et Pizza.



Pour le moment, il n'y a pas encore de données, mais vous pouvez accéder à la vue des données dans l'onglet parcourir les données :



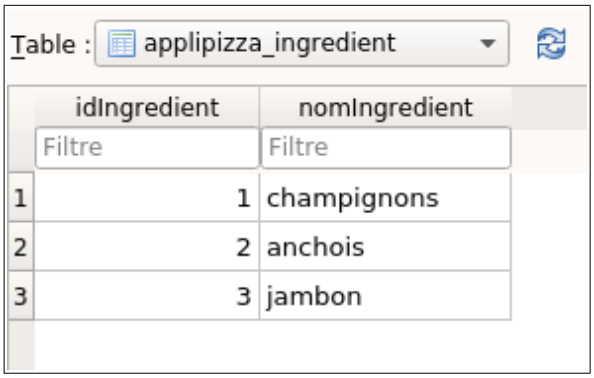


3. Enregistrer, consulter des données

a. Vous pouvez enregistrer ou consulter des données par l'interface, c'est le plus simple, grâce au bouton d'insertion d'un nouvel enregistrement.

N'oubliez pas d'enregistrer les modifications de la table (si vous ne le faites pas, le navigateur vous demandera de sauvegarder en quittant).

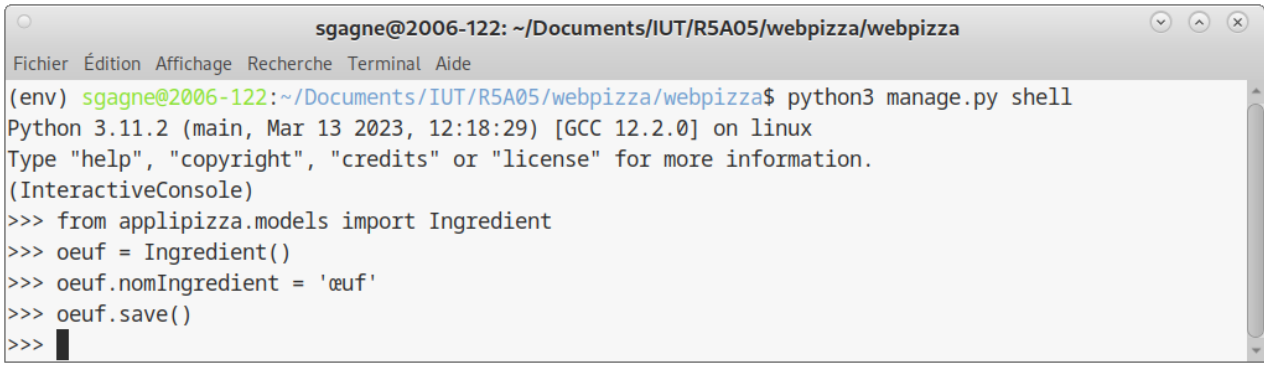
Enregistrez 3 nouveaux ingrédients : dans la capture suivante, champignons, anchois et jambon.



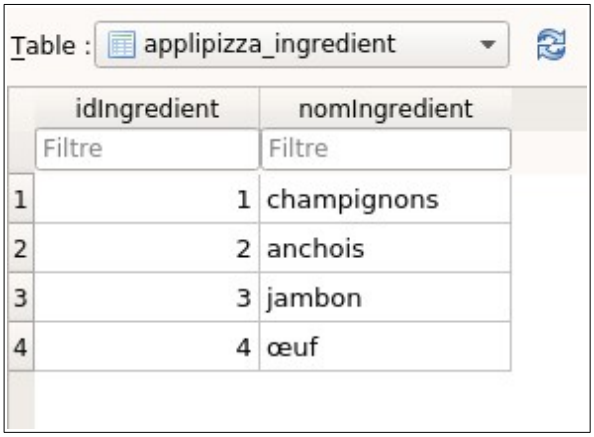
b. Vous pouvez aussi ajouter des enregistrements en ligne de commande grâce au shell python :

- **python3 manage.py shell** pour entrer dans le shell python (**CTRL + D** pour quitter)

- ```
  from applipizza.models import Ingredient
  oeuf = Ingredient()
  oeuf.nomIngredient = 'œuf'
  oeuf.save()
  ```



Ouvrez de nouveau sqlitebrowser pour constater qu'il y a bien une nouvelle entrée, puis quittez sqlitebrowser :


  ```

c. Vous pouvez consulter les données dans la console python (accessible par la commande **python3 manage.py shell**).

Ci-dessous, on a récupéré tous les ingrédients, puis parcouru la liste correspondante et appelé la méthode print sur chaque ingrédient (ceci a implicitement appelé la méthode **__str__** redéfinie précédemment).

Vous pouvez donc agir sur les données ou les consulter par le shell python ou par l'interface.

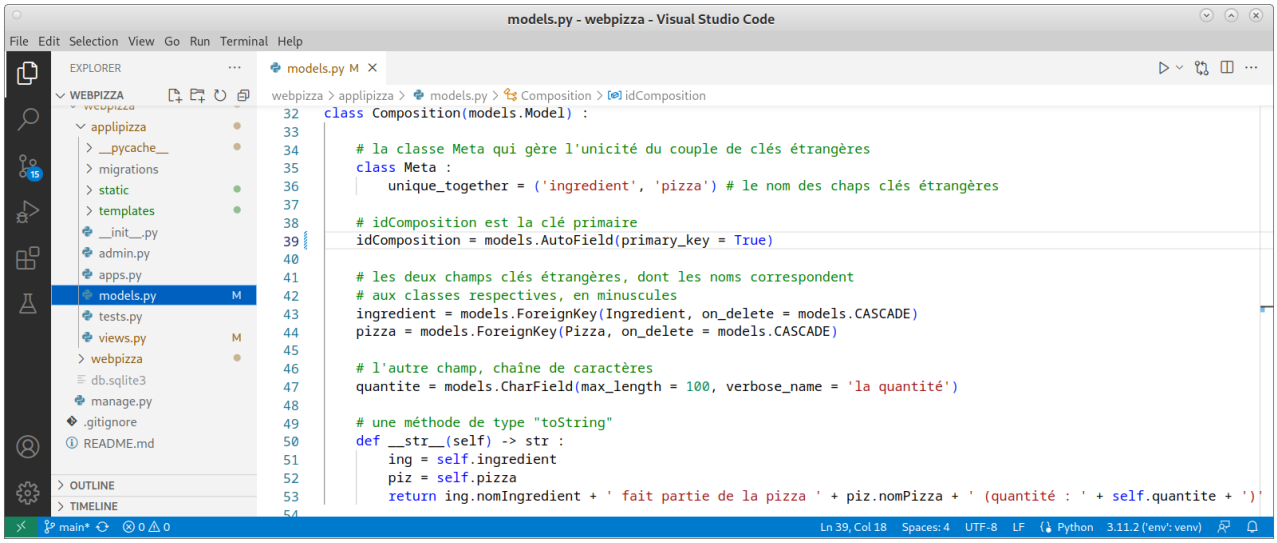
Remarques : en console vous devez comme dans l'éditeur de texte inclure des tabulations (boucle for) sinon l'instruction python n'est pas comprise. En python, les blocs sont délimités par la hiérarchie des indentations. La fonction python **len** renvoie la longueur de l'ensemble passé en paramètre : ici, 4 ingrédients en tout.

N'oubliez pas de mettre votre dépôt distant à jour.

4. Un modèle avec clés étrangères

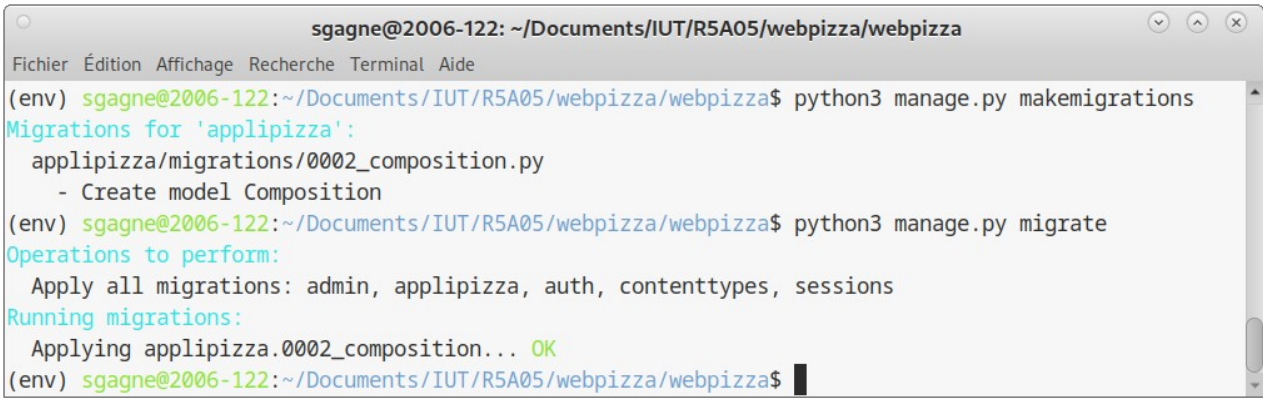
La classe Composition aura donc une clé primaire idComposition, et une contrainte d'unicité sur le couple (idIngredient, idPizza).

Voici le code correspondant. Vous noterez que la contrainte d'unicité est gérée dans une classe interne « Meta » qui gère tout ce qui est hors déclaration de champs.



a. créez la classe dans models.py

b. mettez en place la migration.



c. enregistrez des données via sqlitebrowser, pour avoir un jeu de données intéressant (vous devrez actualiser la base pour voir la nouvelle table composition) :

Table : applipizza_ingredient

	idIngredient	nomIngredient
	Filtre	Filtre
1	1	champignons
2	2	anchois
3	3	jambon
4	4	œuf
5	5	gorgonzola
6	6	fromage de chèvre
7	7	parmesan
8	8	mozzarella
9	9	crème fraîche
10	10	oignon
11	11	basilic
12	12	origan
13	13	coulis de tomate
14	14	tomate
15	15	pomme de terre
16	16	miel
17	17	roquette
18	18	saumon
19	19	thon
20	20	lardons
21	21	viande hâchée

Table : applipizza_pizza

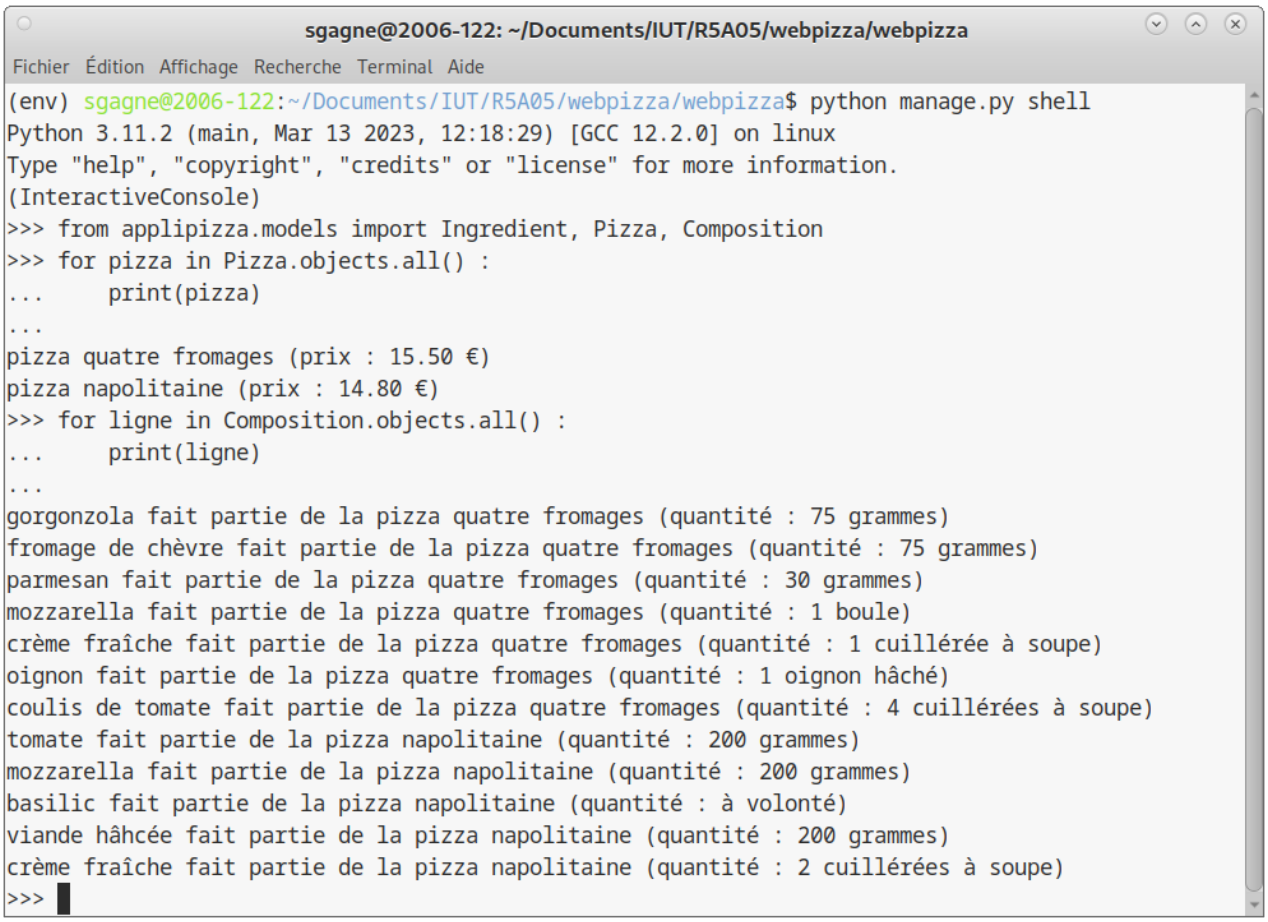
	idPizza	nomPizza	prix
	Filtre	Filtre	Filtre
1	1	quatre fromages	15.5
2	2	napolitaine	14.8

Table : applipizza_composition

	idComposition	quantite	ingredient_id	pizza_id
	Filtre	Filtre	Filtre	Filtre
1	1	75 grammes	5	1
2	2	75 grammes	6	1
3	3	30 grammes	7	1
4	4	1 boule	8	1
5	5	1 cuillérée à soupe	9	1
6	6	1 oignon hâché	10	1
7	7	4 cuillérées à soupe	13	1
8	8	200 grammes	14	2
9	9	200 grammes	8	2
10	10	à volonté	11	2
11	11	200 grammes	21	2
12	12	2 cuillérées à soupe	9	2

N'oubliez pas d'enregistrer les modifications de la base, et n'oubliez pas de mettre à jour le dépôt distant.

d. Entraînez-vous aussi à consulter les tables dans le shell python pour vous habituer à la manœuvre.


sgagne@2006-122: ~/Documents/IUT/R5A05/webpizza/webpizza
Fichier Édition Affichage Recherche Terminal Aide
(env) sgagne@2006-122:~/Documents/IUT/R5A05/webpizza/webpizza$ python manage.py shell
Python 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from applipizza.models import Ingredient, Pizza, Composition
>>> for pizza in Pizza.objects.all() :
...   print(pizza)
...
pizza quatre fromages (prix : 15.50 €)
pizza napolitaine (prix : 14.80 €)
>>> for ligne in Composition.objects.all() :
...   print(ligne)
...
gorgonzola fait partie de la pizza quatre fromages (quantité : 75 grammes)
fromage de chèvre fait partie de la pizza quatre fromages (quantité : 75 grammes)
parmesan fait partie de la pizza quatre fromages (quantité : 30 grammes)
mozzarella fait partie de la pizza quatre fromages (quantité : 1 boule)
crème fraîche fait partie de la pizza quatre fromages (quantité : 1 cuillérée à soupe)
oignon fait partie de la pizza quatre fromages (quantité : 1 oignon hâché)
coulis de tomate fait partie de la pizza quatre fromages (quantité : 4 cuillérées à soupe)
tomate fait partie de la pizza napolitaine (quantité : 200 grammes)
mozzarella fait partie de la pizza napolitaine (quantité : 200 grammes)
basilic fait partie de la pizza napolitaine (quantité : à volonté)
viande hâhcée fait partie de la pizza napolitaine (quantité : 200 grammes)
crème fraîche fait partie de la pizza napolitaine (quantité : 2 cuillérées à soupe)
>>>