LA BASE DE DONNÉES

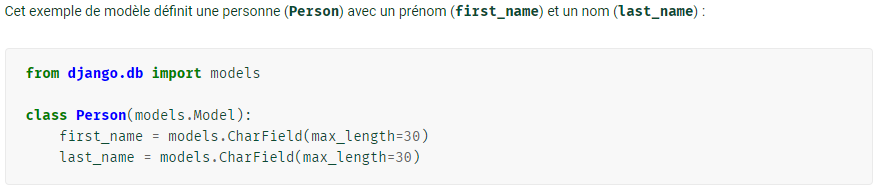
Lexique :

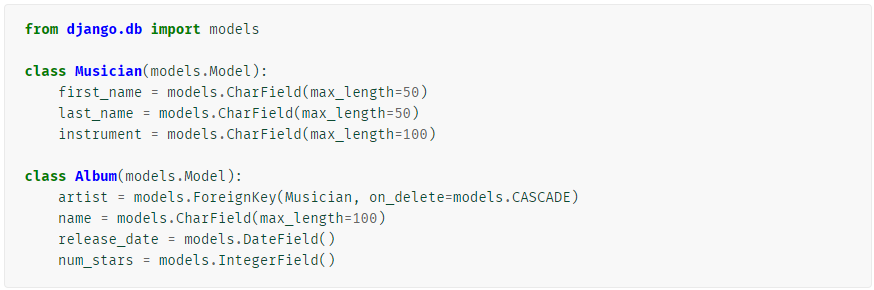
Modèle = table

Champ du modèle (ou du formulaire) = colonne de la table

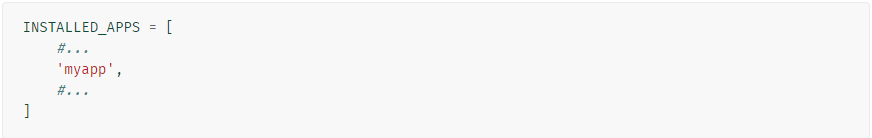
Relations = cardinalité (mcd, mld)

1. On créé les tables dans le fichier **models.py**.



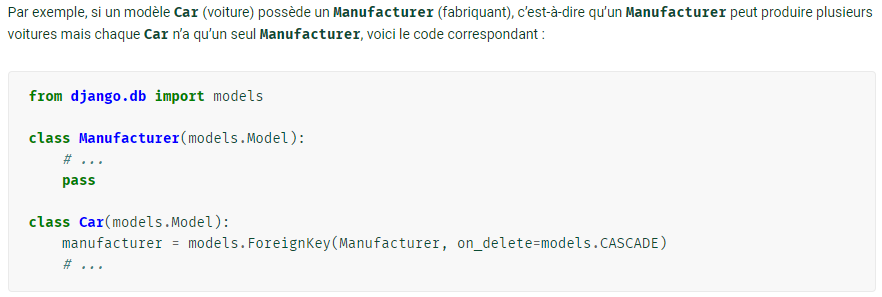


1. Puis on configure ces tables dans le fichier **settings.py =>** [**INSTALLED\_APPS**](https://docs.djangoproject.com/fr/4.0/ref/settings/#std-setting-INSTALLED_APPS)



1. Faire des relations (des liaisons) entre les tables

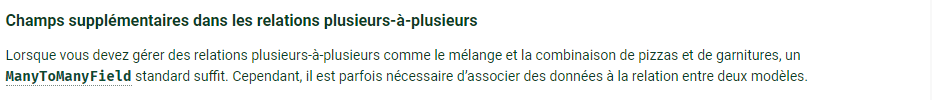
Relation *1, n* ou *0,n* (plusieurs à un)



Relation plusieurs à plusieurs





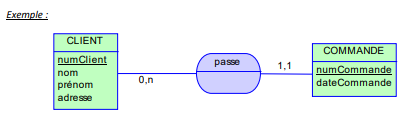




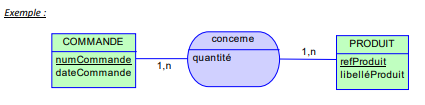
Questionnaire :

1. Dans quel fichier créé-t-on les tables (les modèles) d’une base de données ? : **models.py**
2. Comment créé-t-on ces tables (décrire la syntaxe) ?
3. Donner quelques exemples des types de champs : [**CharField**](https://docs.djangoproject.com/fr/4.0/ref/models/fields/#django.db.models.CharField) , **BooleanField**, [**DateField**](https://docs.djangoproject.com/fr/4.0/ref/models/fields/#django.db.models.DateField)
4. Donner quelques exemples de paramètres d’un champ :  [**max\_length**](https://docs.djangoproject.com/fr/4.0/ref/models/fields/#django.db.models.CharField.max_length) , [**null**](https://docs.djangoproject.com/fr/4.0/ref/models/fields/#django.db.models.Field.null) , [**blank**](https://docs.djangoproject.com/fr/4.0/ref/models/fields/#django.db.models.Field.blank) , [**choices**](https://docs.djangoproject.com/fr/4.0/ref/models/fields/#django.db.models.Field.choices) , [**default**](https://docs.djangoproject.com/fr/4.0/ref/models/fields/#django.db.models.Field.default)
5. Comment indiquer qu’un champ est obligatoire :  [**blank=True**](https://docs.djangoproject.com/fr/4.0/ref/models/fields/#django.db.models.Field.blank)
6. Comment indiquer qu’un champ peut être vide :  [**blank=False**](https://docs.djangoproject.com/fr/4.0/ref/models/fields/#django.db.models.Field.blank)
7. Quelle sont les types de liaisons (de relations, de cardinalités) entre les tables (relations) les plus courantes :

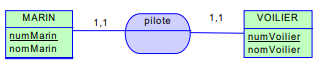
* **plusieurs-à-un (*1, n* ou *0, n*) :** [**ForeignKey**](https://docs.djangoproject.com/fr/4.0/ref/models/fields/#django.db.models.ForeignKey)

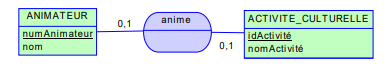


* **plusieurs-à-plusieurs (*n, n*) :** [**ManyToManyField**](https://docs.djangoproject.com/fr/4.0/ref/models/fields/#django.db.models.ManyToManyField)



* **un-à-un (*0,1* ou *1,1*) :** [**OneToOneField**](https://docs.djangoproject.com/fr/4.0/ref/models/fields/#django.db.models.OneToOneField)





1. Que faire après avoir configuré le champ  [**ManyToManyField**](https://docs.djangoproject.com/fr/4.0/ref/models/fields/#django.db.models.ManyToManyField) pour qu’il utilise la table intermédiaire (**Membership**, dans ce cas) ? : il faut créer des relations (les variables (attributs) d’instance) de cette table intermédiaire
2. Comment se passe la création de ces (relations) variables (attributs) d’instance  (syntaxe) ?

Ouvrir l’interpréteur python :

**>>>** ringo = Person.objects.create(name="Ringo Starr")

**>>>** paul = Person.objects.create(name="Paul McCartney")

**>>>** beatles = Group.objects.create(name="The Beatles")

**>>>** m1 = Membership(person=ringo, group=beatles,

**...**  date\_joined=date(1962, 8, 16),

**...**  invite\_reason="Needed a new drummer.")

**>>>** m1.save()

**>>>** beatles.members.all()

<QuerySet [<Person: Ringo Starr>]>

**>>>** ringo.group\_set.all()

<QuerySet [<Group: The Beatles>]>

OU BIEN

**>>>** m2 = Membership.objects.create(person=paul, group=beatles,

**...**  date\_joined=date(1960, 8, 1),

**...**  invite\_reason="Wanted to form a band.")

**>>>** beatles.members.all()

<QuerySet [<Person: Ringo Starr>, <Person: Paul McCartney>]>

OU BIEN (mieux)

NB : Il peut être préférable de créer directement des instances de la table intermédiaire.

**>>>** Membership.objects.create(person=ringo, group=beatles,

**...**  date\_joined=date(1968, 9, 4),

**...**  invite\_reason="You've been gone for a month and we miss you.")

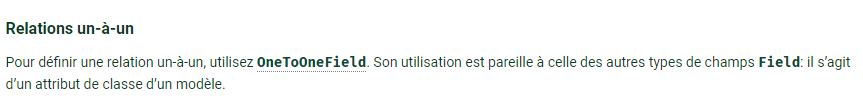
1. Quel méthode permet de supprimer les variables (les colonnes) des tables intermédiaires ([**ManyToManyField**](https://docs.djangoproject.com/fr/4.0/ref/models/fields/#django.db.models.ManyToManyField))

[**remove()**](https://docs.djangoproject.com/fr/4.0/ref/models/relations/#django.db.models.fields.related.RelatedManager.remove) : **>>>** beatles.members.remove(ringo)

1. Quelle méthode permet de supprimer toutes les relations plusieurs à plusieurs d’une table ?

[**clear()**](https://docs.djangoproject.com/fr/4.0/ref/models/relations/#django.db.models.fields.related.RelatedManager.clear) : **>>>** beatles.members.clear()

Relations un à un



1. Fff,,