REPUBLIQUE TUNISIENNE

MINISTERE DE L'ENSEIGNEMENT

SUPERIEUR, DE LA RECHERCHE SCIENTIFIQUE

ET DES TECHNOLOGIES DE L'INFORMATION ET DE LA

COMMUNICATION

DIRECTION GENERALE DES ETUDES TECHNOLOGIQUES

INSTITUT SUPÉRIEUR DES ETUDES TECHNOLOGIQUES DE SFAX

DÉPARTEMENT TECHNOLOGIE DE L'INFORMATIQUE

Travaux Pratiques n°3

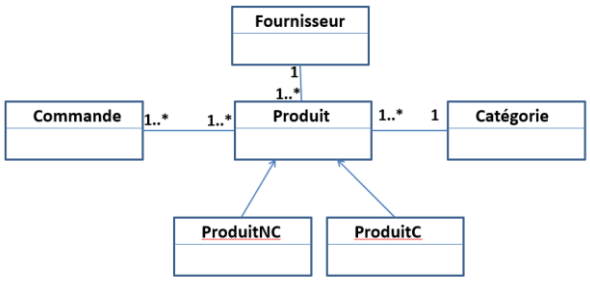
Framework Django

Les modèles (Les associations)

Nous allons travailler dans ces travaux pratiques sur le même projet '~~myproject'~~mysite' et la même application '**magasin'** créés dans le tp2.

a mis en forme : Police :Gras

Le diagramme UML suivant décrit le modèle que nous allons créer pour notre application Django magasin en ligne.



La classe produit garde la même structure définie dans le tp2.

1- Créer dans le modèle de l'application magasin la classe **Categorie** qui a la structure suivante :

Attribut	Type	Paramètres
name	**CharField**	**max_length**=50, **default**='Alimentaire' Sachant que : **TYPE_CHOICES=[** **('Al','Alimentaire'), ('Mb','Meuble'),** **('Sn','Sanitaire'), ('Vs','Vaisselle'),** **('Vt','Vêtement'),('Jx','Jouets'),** **('Lg','Linge de Maison'),('Bj','Bijoux'),('Dc','Décor')]**

Pour implémenter l'association **Many-to-one** entre le produit et sa catégorie, on ajoute un champ de type **models.ForeignKey** dans la classe Produit.

Attribut	Type	Paramètres
catégorie	`models.ForeignKey(Categorie,)`	`on_delete=models.CASCADE, null=True`

NB : ***Chaque classe du modèle doit être munie d'une méthode __str__() qui retourne les valeurs des attributs pour une instance donnée.***

- Enregistrer le modèle ainsi modifié puis ajouter la classe **Cathegorie** dans le fichier **admin.py** de l'application.
- Exécuter **makemigrations** et **migrate**.
- Lancer le serveur
- Accéder à l'interface d'administration et ajouter quelques enregistrements dans **Catégorie**.

Catégorie	
Immobilier	Meuble
Vêtement	Linge
ParaPharmacie	Electroménager
Tapis	Frais

- Mettre à jours les enregistrements déjà existants dans la table Produit en leurs affectant les emballages correspondants.

2- Nous allons implémenter la classe Fournisseur qui admet la structure suivante :

Attribut	**Type**	**Paramètres**
nom	**CharField**	**max_length**=100
adresse	**TextField**	
email	**EmailField**	
telephone	**CharField**	**max_length**=8

- Pour implémenter l'association « plusieurs à un » entre le produit et son fournisseur, on ajoute un champ de type **ForeignKey** (→ ManyToOne) dans la classe **Produit**.

Attribut	**Type**	**Paramètres**
Fournisseur	`ForeignKey('Fournisseur',…)`	**on_delete=models.CASCADE, null=True**

- Enregistrer le modèle ainsi modifié puis ajouter la classe **Fournisseur** dans le fichier **admin.py** de l'application.
- Mettre à jour le modèle et la base de données ensuite relancer le serveur
- Accéder à l'interface d'administration et ajouter quelques enregistrements dans la table Fournisseur.

Nom	**Adresse**	**Email**	**téléphone**
Sicam	Tunis	sican@topnet.tn	71254321
SOFAB	Sousse	sofab@atti.tn	73520200
SAID	Monastir	SocSaid@dist.tn	73200100
Farah	Sfax	farahdisstribution@des.tn	74800500

- Mettre à jours les enregistrements déjà existants dans la table Produit en leurs affectant les fournisseurs -correspondants.

3- Une liaison d'héritage existe entre la classe **Produit** et ses deux sous classes **ProduitC** (produit consommable) et **ProduitNC** (produit non consommable).
Dans notre modèle nous allons créer la classe **ProduitNC** qui hérite de Produit et qui admet un seul champ :

Attribut	Type	Paramètres
Duree_garantie	**CharField**	**max_length**=100

- Enregistrer le modèle ainsi modifié puis ajouter la classe **ProduitNC** dans le fichier admin.py de l'application.
- Mettre à jour le modèle et la base de données
- Accéder à l'interface d'administration et ajouter quelques enregistrements dans **ProduitNC** .

Libellé	Description	prix	Type	Duree_garantie
Micro-onde	Four à ondes	365.500	'em'	2 ans
veilleuse	Lampe led	28.000	'em'	3 mois
Lave vaisselle	15 couverts	1350.500	'em'	2 ans

4- Ajouter la classe **Commande** au modèle **models.py** de l'application « **magasin** ». La classe commande admet la structure suivante :

Attribut	Type	Paramètres
dateCde	**DateField**	**null**=True, **default**=date.today NB : ajouter ***from datetime import date***
totalCde	**DecimalField**	`max_digits=10,decimal_places=3`

- Pour implémenter l'association « plusieurs à plusieurs » entre Produit et Commande, on ajouter un champ **ManyToMany** dans la classe **Commande** ainsi :

Attribut	Type	Paramètres
produits	**ManyToManyField('Produit')**	

- Enregistrer le modèle ainsi modifié puis ajouter la classe Commande dans le fichier admin.py de l'application.
- Accéder à l'interface d'administration et ajouter quelques enregistrements dans Commande.

NB : Pour pouvoir consulter les différentes tables de base de données telles que créées par Django, on vous recommande d'installer en local l'outil « **DB browser pour SQLite** ».