

TP2 – Manipulation de l'ORM dans Odoo

19

Introduction

Dans Odoo, toutes les données sont stockées dans une base PostgreSQL. Pour les manipuler, Odoo utilise l'**ORM (Object-Relational Mapping)**.

L'ORM transforme chaque table en **objet Python**, et chaque ligne en **instance de cet objet**. Grâce à l'ORM :

- On peut **créer, lire, modifier et supprimer** des enregistrements directement avec Python.
- Les opérations respectent automatiquement les **règles métiers et la sécurité**.
- Le même code fonctionne pour **l'interface utilisateur, les scripts et les modules**.

Exemples simples :

Action ORM	Description	SQL équivalent
<code>env['product.template'].create({...})</code>	Créer un produit	INSERT INTO product_template ...
<code>env['product.template'].search([...])</code>	Chercher des produits	SELECT * FROM product_template WHERE ...
<code>record.write({...})</code>	Modifier un produit	UPDATE product_template SET ...
<code>record.unlink()</code>	Supprimer un produit	DELETE FROM product_template WHERE ...

Objectif du TP

Ce TP a pour objectif de vous faire **découvrir l'ORM d'Odoo**, en manipulant des produits depuis le **shell Python d'Odoo**. Vous allez :

- Créer des produits avec différents attributs.
- Lire et filtrer les produits selon certaines conditions.
- Modifier des produits existants selon des critères.
- Supprimer des produits de manière sécurisée.
- Réaliser des statistiques simples et créer un catalogue promotionnel.

Tout le travail se fait directement via **Python**. Le code fourni sert d'exemple ou de point de départ, vous devez l'exécuter et observer les effets sur vos données.

les commandes shell sont :

```
>>>>>>cd "C:\Program Files\Odoo 19.0.20251004\server"
```

```
>>>>>>"C:\Program Files\Odoo 19.0.20251004\python\python.exe" odoo-bin shell -d  
<nom_de_la_base>
```

Exercice 1 – Création de produits variés

Consigne : Créez 10 produits avec des prix différents et un champ personnalisé `x_is_promo` qui indique si le produit est en promotion.

Exemple de code :

```
1 products_data = [  
2     {'name': 'Clavier AZERTY', 'list_price': 120, 'x_is_promo': True},  
3     {'name': 'Souris Optical', 'list_price': 80, 'x_is_promo': False},  
4     {'name': 'Ecran 24"', 'list_price': 200, 'x_is_promo': True},  
5     {'name': 'Enceinte Bluetooth', 'list_price': 150, 'x_is_promo': False},  
6     {'name': 'Webcam HD', 'list_price': 60, 'x_is_promo': True},  
7     {'name': 'Clavier Gamer', 'list_price': 180, 'x_is_promo': False},  
8     {'name': 'Souris Gamer', 'list_price': 100, 'x_is_promo': True},  
9     {'name': 'Casque Audio', 'list_price': 130, 'x_is_promo': False},  
10    {'name': 'Disque Dur 1TB', 'list_price': 90, 'x_is_promo': True},  
11    {'name': 'Clé USB 64GB', 'list_price': 30, 'x_is_promo': False},  
12 ]  
13  
14 for data in products_data:  
15     env['product.template'].create(data)  
16  
17 print("10 produits créés avec succès !")  
18
```

À faire :

- Exécuter le code.
- Vérifier dans l'interface Odoo que les produits ont été créés.

Exercice 2 – Lecture et filtrage

Consigne : Recherchez tous les produits dont le prix est supérieur à 100, dont le nom contient "Clavier", et qui sont en promotion.

Exemple de code :

```
1 products = env['product.template'].search([
2     ('list_price', '>', 100),
3     ('name', 'ilike', 'Clavier'),
4     ('x_is_promo', '=', True)
5 ])
6
7 print("Produits filtrés :")
8 for p in products:
9     print(p.name, p.list_price, p.x_is_promo)|
```

À faire :

- Exécuter le code et observer le résultat.
- Identifier quels produits correspondent aux critères.

Exercice 3 – Modification conditionnelle

Consigne : Augmentez de 10% le prix de tous les produits en promotion et ajoutez “-Promo” au nom du produit.

Exemple de code :

```
1 promo_products = env['product.template'].search([('x_is_promo', '=', True)])
2 for p in promo_products:
3     p.write({
4         'list_price': p.list_price * 1.10,
5         'name': p.name + '-Promo'
6     })
7
8 print("Prix et noms des produits en promo mis à jour !")|
```

À faire :

- Vérifiez les changements dans l'interface Odoo.
- Notez le nouveau prix et le nouveau nom des produits en promotion.

Exercice 4 – Suppression sécurisée

Consigne : Supprimez tous les produits dont le prix est inférieur à 50.

Exemple de code :

```
1 low_price_products = env['product.template'].search([('list_price', '<', 50)])
2 for p in low_price_products:
3     p.unlink()
4
5 print(f"{len(low_price_products)} produits supprimés")
```

À faire :

- Vérifier dans l'interface que ces produits ont bien été supprimés.

Exercice 5 – Recherche et statistiques

Consigne : Comptez le nombre de produits par tranche de prix et calculez le prix moyen de tous les produits.

Exemple de code :

```
1 ranges = [(0,50),(51,100),(101,150),(151,1000)]
2 for r in ranges:
3     count = env['product.template'].search_count([
4         ('list_price','>=',r[0]), ('list_price','<=',r[1])
5     ])
6     print(f"Produits entre {r[0]} et {r[1]} : {count}")
7
8 products_all = env['product.template'].search([])
9 if products_all:
10     avg_price = sum(p.list_price for p in products_all) / len(products_all)
11     print("Prix moyen de tous les produits :", round(avg_price,2))
```

À faire :

- Observer la répartition des produits.
- Noter le prix moyen calculé.

Exercice 6 – Défi final : Catalogue promotionnel

Consigne : Créez un catalogue promotionnel des produits en promo avec le prix actuel et le prix après une remise de 10%, trié par prix décroissant.

Exemple de code :

```
1 promo_products = env['product.template'].search([('x_is_promo', '=', True)])
2 catalogue = []
3
4 for p in promo_products:
5     catalogue.append({
6         'name': p.name,
7         'price': p.list_price,
8         'price_after_10': round(p.list_price * 0.9, 2)
9     })
10
11 catalogue_sorted = sorted(catalogue, key=lambda x: x['price'], reverse=True)
12
13 print("Catalogue promotionnel trié par prix décroissant :")
14 for item in catalogue_sorted:
15     print(item)
```

À faire :

- Vérifier que le tri est correct.
- Vérifier que le calcul de la remise est correct.

Livrables à rendre

Pour ce TP, vous devez remettre un dossier compressé contenant :

1. **Captures d'écran** de l'interface Odoo :
 - Produits créés après l'Exercice 1.
 - Résultats des filtres et recherches (Exercice 2 et 5).
 - Produits modifiés (Exercice 3).
 - Produits supprimés (Exercice 4).
 - Catalogue promotionnel final (Exercice 6).
2. **Fichier texte ou PDF** contenant vos réponses et observations :
 - Les produits trouvés dans chaque filtre.
 - Le nombre de produits créés, modifiés et supprimés.
 - Le prix moyen calculé et la répartition par tranche.
 - Vos commentaires sur le fonctionnement de l'ORM et les résultats obtenus.