La couche ORM

La couche **ORM** (Object-Relational Mapping) d'Odoo évite d'écrire des requêtes SQL. Les objets métiers qui accèdent à la base de données héitent de la **classe Model** qui intègre les mécanismes de persistance des objets.

Déclaration d'un modèle de données

Les informations à fournir sont :

- **_name** : le nom du modèle
- les champs de données, attributs du modèle
  - les champs **simples** : Boolean, Date, Char ;
  - les champs gérant les **relations** entre enregistrements

Exemples :

- déclaration du champ **name** de type **suite de caractères** :

name = fields.Char()

La méthode Char() pourra accepter des paramètres qui précisent les caractéristiques du champ (voir ci-après)

- déclaration du champ **dateachat** de type **Date** :

dateachat = fields.Date()

On peut préciser les caractéristiques des champs :

- **string** : par défaut c'est le nom du champ et c'est ce qui est affiché comme label dans les pages
- **required** : booléen et par défaut est False ; Si True le champ doit être renseigné par l'utilisateur ou par défaut lors de la création d'un enregistrement
- **help** : Unicode et par default '' ; fournit une aide à la saisie
- **index** : booléen par défault False ; un index sera créé sur le champ par Odoo

Exemple :

- déclaration du champ **name** de type **suite de caractères** avec un label précis et une saisie obligatoire:

name = fields.Char(string="Titre", required=True)

Les champs réservés

Odoo crée des champs réservés qui ne peuvent être modifiés par l'utilisateur. Ils peuvent être lus si nécessaire pour des traitements.

- **id** (Id) : identifiant unique ;
- **create_date** (Datetime) : date de création de l'enregistrement ;
- **create_uid** (Many2one) : identifiant de l'utilisateur qui a créé l'enregistrement
- **write_date** (Datetime) : date de la dernière modification
- **write_uid** (Many2one) : identifiant du dernier utilisateur qui a modifié l'enregistrement.

Les champs spéciaux

Odoo **exige** l'existence d'un champ **name** pour tous les modèles à des fins d'affichage et de recherches.

A faire :

- Définissez le modèle suivant pour un cours avec un titre (obligatoire) et une description.
- Editez le fichier **models.py** et ajouter la classe **Cours**.

```
from openerp import models, fields, api

class Cours(models.Model):
    _name = 'openacademy.cours'

    name = fields.Char(string="Titre", required=True)
    description = fields.Text()
```

Les fichiers de données de démonstration

Si un module de données de démonstration a été défini, il est automatiquement chargé.

Un module de données est déclaré dans un fichier au format **XML** avec des éléments **<record>**. Chaque élément **<record>** crée ou met à jour un enregistrement de la base de données.

```
<openerp>
    <data>
        <record model="{model name}" id="{record identifier}">
            <field name="{a field name}">{a value}</field>
        </record>
    </data>
```

```
<openerp>
```

- **model** : est le nom du modèle Odoo dans l'enregistrement ;
- **id** : est un identifiant externe pour faire référence à l'enregistrement ;
- les éléments **<field>** ont le même nom que celui du champ dans le modèle. Ils contiennent les valeurs affectées aux champs.

Les fichiers de données Data doivent être déclarés dans le fichier manifest **__openerp__.py** :

- soit dans la section **data** s'ils doivent être systématiquement chargés ;
- soit dans la section **demo** s'ils ne doivent être chargés que si la base est créée en mode **démonstration**.

Exercices : définir des donnés de démonstration pour quelques cours :

- Modifiez le fichier **demo.xml** avec les informations suivantes .. ou d'autres.

```
<openerp>
    <data>
        <record model="openacademy.cours" id="cours0">
            <field name="name">Français</field>
            <field name="description">Activités sur la rédaction
des mémoires
Activités sur les grands classiques de la science-fiction (Stars
Wars, etc.)
et bien d'autres choses encore ...
Plusieurs lignes peuvent être utilisées
            </field>
        </record>
        <record model="openacademy.cours" id="cours1">
            <field name="name">Mathématiques</field>
            <!-- pas de description pour ce cours -->
        </record>
        <record model="openacademy.cours" id="cours2">
            <field name="name">Anglais</field>
            <field name="description">Préparation TOEIC</field>
        </record>
    </data>
</openerp>
```

Attention au copier-coller pour les caractères accentués.

Mettez à jour le module et visualisez sous PostgreSQL la création de la table **openacademy_cours** qui contient :

- **7** colonnes,
- et vos **données de démonstration**.

Si la table table **openacademy_cours** n'est pas créée :

- **relancez** le service Odoo :

```
$ sudo service odoo restart
```

- et **actualisez** le module.

Retour au sommaire de création d'un module

Développer un module dans Odoo (version 8)