2025/09/29 21:30 1/5 Relations entre modèles

Relations entre modèles

Création d'un nouveau modèle pour gérer les session de cours

Un enregistrement dans un modèle (une table) peut être **relié** à un enregistrement d'un autre modèle. Par exemple, un ordre de vente est relié à un enregistrement client qui contient les informations sur ce client.

Exercice : créer un modèle de session :

Pour ce module Open Academy, un modèle va permettre de gérer les **sessions** de cours. Une session est un cours planifié à une date pour des personnes.

Pour créer un modèle de sessions il faut préciser :

- le **nom** de la session,
- une date de début,
- une durée,
- le nombre de place.

Une action doit être ajoutée et associé à un nouvel élément de menu dans le menu pour afficher les sessions **Démarche à suivre :**

- 1. Créez la classe Session dans le fichier openacademy/models.py.
- 2. Ajoutez un accès à l'objet Session dasn le fichier openacademy/view/openacademy.xml.
- Modifiez le fichier openacademy/models.py

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

class Session(models.Model):
    _name = 'openacademy.session'

nom = fields.Char(required=True)
date_deb = fields.Date()
duree = fields.Float(digits=(6, 2), help="Duree en jours")
places = fields.Integer(string="Nombre de place")
```

• Modifiez le fichier openacademy/view/openacademy.xml

```
<!-- Full id location:
     action="openacademy.cours_list_action"
     It is not required when it is the same module -->
<!-- session form view -->
<record model="ir.ui.view" id="session form view">
    <field name="name">session.form</field>
    <field name="model">openacademy.session</field>
    <field name="arch" type="xml">
        <form string="Formulaire Session">
            <sheet>
                <group>
                    <field name="nom"/>
                    <field name="date deb"/>
                    <field name="duree"/>
                    <field name="places"/>
                </group>
            </sheet>
        </form>
    </field>
</record>
<record model="ir.actions.act_window" id="session_list_action">
    <field name="name">Sessions</field>
    <field name="res_model">openacademy.session</field>
    <field name="view_type">form</field>
```

Information : digits=(6, 2) indique la précision du réel :

- 6 est le nombre total de chiffres,
- 2 est le nombre de chiffres après la virgule (au maximum 4.

Les champs permettant de gérer les relations

Des champs permettent de gèrer les relations entre enregistrements :

- d'un même modèle → gérer des hiérarchies,
- ou entre des modèles différents.

Voici les différebnets types de champs relationnels :

- Many2one(othermodel, ondelete='set null'): * c'est un simple lien vers un autre objet * One2many(othermodel, relatedfield) * c'est l'inverse Many2one. Cela peut renvoyer une liste d'enregistrement qui seront à traiter au sein d'une boucle. * Many2many(othermodel):
 - Tout comme One2many, cela peut renvoyer une liste d'enregistrement qui seront à traiter au sein d'une boucle.

Exercice : gérer des relations de type Many2one :

Vous allez modifier les modèles de cours et de session pour prendre en compte leurs relations avec d'autres modèles :

- un cours à un responsable : la valeur de ce champ permettra de lier l'enregistrement au modèle existant res.users ;
- une session a un formateur : la valeur de ce champ permettra de lier l'enregistrement au modèle existant resusers :
- une session est liée à un cours : la valeur de ce champ permettra de lier l'enregistrement au modèle openacademy.course que vous avez créé;
- et pour finir, il faudra **adapter les vues**.

Travail à faire :

- 1. ajouter les champs Many2one aux modèles,
- 2. et les ajouter aux vues ;
- Modifiez le fichier openacademy/models.py

Printed on 2025/09/29 21:30

2025/09/29 21:30 3/5 Relations entre modèles

• Modifiez le fichier openacademy/views/openacademy.xml <code xml> <!- modifier la vue list des cours-<record model="ir.ui.view" id="cours_tree_view"> <field name="name">course.search</field> <field name="model">openacademy.cours</field> <field name="arch" type="xml"> <tree string="Liste des cours"> <field name="name"/> <field name="description"/> <field name="responsable id"/> </tree> </field> </record> <!-- session form view --> <record model="ir.ui.view" id="session_form_view"> <field name="name">session.form</field> <field name="model">openacademy.session</field> <field name="arch" type="xml"> <form string="Formulaire Session"> <sheet> <group> <group string="General"> <field name="cours id"/> <field name="nom"/> <field name="formateur_id"/> </group> <group string="Planification"> <field name="date_deb"/> <field name="duree"/> <field name="places"/> </group> </group> </sheet> </form> </field> </record> <!-- session tree/list view --> <record model="ir.ui.view" id="session_tree_view"> <field name="name">session.tree</field> <field name="model">openacademy.session</field> <field name="arch" type="xml"> <tree string="Liste des sessions"> <field name="nom"/> <field name="cours_id"/> </tree> </field> </record> < /code> Modifiez la vue coursformview pour permettre la saisie du responsable du cours dans uin nouvel onglet. <code xml> <notebook> <page string="Responsable" > <field name="responsable_id"/> <page string="Description"> <field name="description"/> </page>

<

/code> </WRAP>

Exercice : gérer des relations de type un à plusieurs One2many :

Vous allez modifier les modèles de **cours** afin de pouvoir gérer la relation d'**un cours avec plusieurs sessions** de famation :

- Modification de la classe Cours, et
- ajout du champ dans le formulaire de cours.
- Modifiez le fichier openacademy/models.py :

 Modifiez le fichier openacademy/views/openacademy.xml pour compléter le formulaire coursformview : <code xml> <notebook> <page string="Responsable"

>

```
<field name="responsable_id"/>
    </page>
    <page string="Description">
        <field name="description"/>
   </page>
    <page string="Sessions">
        <field name="session_ids">
            <tree string="Sessions enregistrees">
                <field name="nom"/>
                <field name="formateur_id"/>
            </tree>
        </field>
   </page>
    <page string="A propos">
        Ceci est un exemple de notebooks
    </page>
</notebook>
```

</code> </WRAP>

Retour au sommaire de création d'un module

Développer un module dans Odoo (version 8)

/ Printed on 2025/09/29 21:30

2025/09/29 21:30 5/5 Relations entre modèles

From:

/ - Les cours du BTS SIO

Permanent link: /doku.php/dev/odoo/creermodule/relationmodel

Last update: 2017/12/17 21:26

