

**Unité d'Enseignement en Informatique  
Année 2012-2013**

**Master M1 EFREI – ASI – ISI  
Devoir Écrit de ERP – Première Session du 12 Avril 2013  
(durée 2h00, aucun document autorisé)**

*Consigne: Réponses en Français ou en Anglais, terminologie Anglophone autorisée dans les deux cas.*

**Exercice 1: (5 points)**

- 1) Selon l'analyse systémique de l'entreprise, quels sont les trois sous-systèmes de l'entreprise ?
- 2) Que signifie le sigle ERP ?
- 3) Que signifie le sigle CRM ?
- 4) Que signifie le sigle MRP ?
- 5) Quelles sont les trois particularités techniques qui caractérisent les logiciels ERP ?
- 6) Le passage de l'an 2000 et le passage à l'Euro ont poussé de nombreuses entreprises à migrer vers un ERP. Expliquez pourquoi ?
- 7) Quels coûts doivent être pris en compte dans le budget d'un projet ERP ?

**Exercice 2: (4 points)**

Vous travaillez au service DSI d'une entreprise. Votre entreprise s'interroge sur l'installation d'un ERP. Afin d'alimenter le débat, pourriez-vous donner:

- 1) Trois arguments forts en faveur de l'utilisation d'un logiciel ERP ?
- 2) Trois arguments forts contre l'utilisation d'un logiciel ERP ?

**Exercice 3: (2 points)**

- 1) Quels sont les trois parties logicielles nécessaires pour faire fonctionner OpenERP ?
- 2) Avec quel langage de programmation OpenERP est-il écrit ? Avec quel langage de programmation les modules d'OpenERP sont-ils écrits ?
- 3) Quel logiciel héberge les données lorsqu'on utilise OpenERP ?

**Exercice 4: (9 points)**

Le module "Idea" d'OpenERP permet de collecter, au fil du temps, les idées nouvelles des employés, de manière informelle. Ainsi, ces idées ne sont pas oubliées et peuvent servir un jour à faire progresser l'entreprise. Dans la suite, vous trouverez des parties du code source de ce module : le fichier `idea.py` en entier et un extrait du fichier `idea_view.xml`.

Dans le module `idea`, l'évolution d'une idée commence par être un document brouillon (`draft`) et termine en étant close ou annulée (`closed` ou `cancelled`). Afin d'améliorer ce petit module de gestion d'idées, nous voudrions enregistrer la date de création et de fin (fermeture ou annulation) des idées proposées. Pour ce faire, deux champs doivent être ajoutés à la classe `idea`. Ces champs ne seront pas entrés par l'utilisateur, mais seront affectés avec la date courante durant la création d'une idée and durant la fermeture ou annulation d'une idée.

Pour décrire votre travail, lors des trois questions à suivre, vous procéderez ainsi :

- Donner le nom du fichier et le numéro de la ligne que vous souhaitez modifier:  
"Dans le fichier `xxx.xx`, je remplace la ligne `n` par ....."
- Donner le nom du fichier et les numéros des lignes où vous souhaitez insérer du code :  
"Dans le fichier `xxx.xx`, entre les lignes `n` et `n+1`, j'ajoute les lignes ....."

N'oubliez pas de lire attentivement la documentation fournie dans la suite du présent document.

1) Ajouter deux nouveaux champs `date_start` et `date_end` au modèle `idea`, puis de faire les modifications nécessaires pour qu'ils apparaissent lors de l'affichage d'une idée.

2) Les méthodes `idea_draft()`, `idea_close()` et `idea_cancel()` sont appelées quand une idée entre respectivement dans les états brouillon (`draft`), close (`closed`) et annulée (`cancelled`). Modifiez ces fonctions pour que les champs `date_start` et `date_end` y soient initialisés.

3) Définissez les trois filtres pour :

- Filtrer les idées créées le jour même
- Filtrer les idées créées moins de 1 an auparavant
- Filtrer les idées créées plus de 1 an auparavant

### Documentation:

a) Type de champ: `date`

```
fields.date('Field Name' [, Optional Parameters]),
```

b) Obtenir la date courante en Python :

```
from datetime import datetime
datetime.now()
```

c) La méthode `write` des objets OpenERP, qui permet d'affecter les valeurs des champs :

(par exemple, elle est appelée lignes 48, 51, 54 et 57 de `idea.py` pour donner sa valeur au champ `state`)

```
write(cr, user, ids, vals, context=None)
```

Update records with given ids with the given field values

**Parameters:**

- **cr** – database cursor
- **user** (integer) – current user id
- **ids** – object id or list of object ids to update according to vals
- **vals** (dictionary) – field values to update, e.g. {'field\_name': new\_field\_value, ...}
- **context** (dictionary) – (optional) context arguments, e.g. {'lang': 'en\_us', 'tz': 'UTC', ...}

**Returns:** True

**Raises:**

- **AccessError** –
  - if user has no write rights on the requested object
  - if user tries to bypass access rules for write on the requested object
- **ValidationError** – if user tries to enter invalid value for a field that is not in selection
- **UserError** – if a loop would be created in a hierarchy of objects a result of the operation (such as setting an object as its own parent)

d) Opérateurs de comparaison dans les fichiers XML :

opérateur	xml
=	=
≠	!=

opérateur	xml
<	&lt;
≤	&lt;=

opérateur	xml
>	&gt;
≥	&gt;=

e) Pour le filtrage des dates :

- Obtenir la date courante :  
`context_today().strftime('%Y-%m-%d')`
- Pour connaître la date 1 an auparavant :  
`(context_today() - datetime.timedelta(weeks=52)).strftime('%Y-%m-%d')`

## idea.py

```

1 from openerp.osv import osv
2 from openerp.osv import fields
3 from openerp.tools.translate import _
4 import time
5
6 VoteValues = [(-1, 'Not Voted'), (0, 'Very Bad'), (25, 'Bad'), \
7               ('50', 'Normal'), (75, 'Good'), (100, 'Very Good') ]
8 DefaultVoteValue = '50'
9
10 class idea_category(osv.osv):
11     """ Category of Idea """
12     _name = "idea.category"
13     _description = "Idea Category"
14     _columns = {
15         'name': fields.char('Category Name', size=64, required=True),
16     }
17     _sql_constraints = [
18         ('name', 'unique(name)', 'The name of the category must be unique')
19     ]
20     _order = 'name asc'
21
22
23 class idea_idea(osv.osv):
24     """ Idea """
25     _name = 'idea.idea'
26     _inherit = ['mail.thread']
27     _columns = {
28         'create_uid': fields.many2one('res.users', 'Creator', required=True, readonly=True),
29         'name': fields.char('Idea Summary', size=64, required=True, readonly=True,
30 oldname='title', states={'draft': [('readonly', False)]}),
31         'description': fields.text('Description', help='Content of the idea', readonly=True,
32 states={'draft': [('readonly', False)]}),
33         'category_ids': fields.many2many('idea.category', string='Tags', readonly=True,
34 states={'draft': [('readonly', False)]}),
35         'state': fields.selection([('draft', 'New'),
36 ('open', 'Accepted'),
37 ('cancel', 'Refused'),
38 ('close', 'Done')],
39 'Status', readonly=True, track_visibility='onchange',
40 )
41     }
42     _sql_constraints = [
43         ('name', 'unique(name)', 'The name of the idea must be unique')
44     ]
45     _defaults = {
46         'state': lambda *a: 'draft',
47     }
48     _order = 'name asc'
49
50 def idea_cancel(self, cr, uid, ids, context=None):
51     return self.write(cr, uid, ids, {'state': 'cancel'}, context=context)
52
53 def idea_open(self, cr, uid, ids, context={}):
54     return self.write(cr, uid, ids, {'state': 'open'}, context=context)
55
56 def idea_close(self, cr, uid, ids, context={}):
57     return self.write(cr, uid, ids, {'state': 'close'}, context=context)
58
59 def idea_draft(self, cr, uid, ids, context={}):
60     return self.write(cr, uid, ids, {'state': 'draft'}, context=context)

```

## idea\_view.xml

```

60 <!-- New Idea Form View -->
61
62 <record model="ir.ui.view" id="view_idea_idea_form">
63   <field name="name">idea.idea.form</field>
64   <field name="model">idea.idea</field>
65   <field name="arch" type="xml">
66     <form string="Idea" version="7.0">
67       <header>
68         <button name="idea_open" string="Open" states="draft" class="oe_highlight"/>
69         <button name="idea_close" string="Accept" states="open" class="oe_highlight"/>
70         <button name="idea_cancel" string="Refuse" states="open" class="oe_highlight"/>
71         <field name="state" widget="statusbar" statusbar_visible="draft,open,close"/>
72       </header>
73       <sheet>
74         <label for="name" class="oe_edit_only"/>
75         <h1><field name="name"/></h1>
76         <label for="category_ids" class="oe_edit_only"/>
77         <field name="category_ids" widget="many2many_tags"/>
78         <label for="description"/><newline/>
79         <field name="description"/>
80       </sheet>
81       <div class="oe_chatter">
82         <field name="message_follower_ids" widget="mail_followers"/>
83         <field name="message_ids" widget="mail_thread"/>
84       </div>
85     </form>
86   </field>
87 </record>
88
89 <!-- New Idea Tree View -->
90
91 <record model="ir.ui.view" id="view_idea_idea_tree">
92   <field name="name">idea.idea.tree</field>
93   <field name="model">idea.idea</field>
94   <field name="arch" type="xml">
95     <tree colors="blue:state == 'draft';black:state in ('open','close');gray:state == 'cancel'"
96     string="Ideas">
97       <field name="name"/>
98       <field name="create_uid"/>
99       <field name="state"/>
100     </tree>
101   </field>
102 </record>
103
104 <!-- Search Idea -->
105
106 <record model="ir.ui.view" id="view_idea_idea_search">
107   <field name="name">idea.idea.search</field>
108   <field name="model">idea.idea</field>
109   <field name="arch" type="xml">
110     <search string="Ideas">
111       <field name="name" string="Idea"/>
112       <filter icon="terp-document-new" string="New" domain="['state','=','draft']" help="New
113       Ideas" />
114       <filter icon="terp-camera_test" string="In Progress" domain="['state','=','open']"
115       help="Open Ideas" />
116       <filter icon="terp-check" string="Accepted" domain="['state','=','close']"
117       help="Accepted Ideas" />
118       <field name="category_ids"/>
119       <group expand="0" string="Group By...">
120         <filter icon="terp-personal" string="Creator" help="By Creators"
121         context="{ 'group_by': 'create_uid' }"/>
122         <filter icon="terp-stock_symbol-selection" string="Category" help="By Idea Category"
123         context="{ 'group_by': 'category_ids' }"/>
124         <filter icon="terp-stock_effects-object-colorize" string="Status" help="By States"
125         context="{ 'group_by': 'state' }"/>
126       </group>
127     </search>
128   </field>
129 </record>

```