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

**Master M1 EFREI – ASI – BI**
**Devoir Écrit de ERP – First Session April 12, 2013**
(2h00 duration, no document allowed)

***Rule:*** *All answers must be written in English.*

**<u>Exercise 1</u>: (5 points)**

1) According to the system analysis of enterprises, what are the three sub-systems of an enterprise?
2) What do the letters ERP stand for?
3) What do the letters CRM stand for?
4) What do the letters MRP stand for?
5) What are the 3 main technical particularities of ERP which characterize this software?
6) The year 2000 problem and the Euro changeover made lot of companies migrating to ERP. Why?
7) What costs must be included in the budget of an ERP project?

**<u>Exercise 2</u>: (4 points)**

You are working in the IT department of a company. Your company wonders about installing an ERP. In order to debate, could you give:

1) Three strong arguments in favor of the use of ERP software?
2) Three strong arguments against it?

**<u>Exercise 3</u>: (2 points)**

1) What are the 3 main software parts to run in order to make OpenERP work?
2) What programming language is OpenERP written with? What programming language are the modules of OpenERP written with?
3) What software stores data when using OpenERP?

**<u>Exercise 4</u>: (9 points)**

The module named "Idea" in OpenERP allows collecting, across time, new ideas of employees, thought an informal way. Thus, these ideas are not forgotten and can help the progression of the company one day. In the following, you will find a part of the source code of this module: the full `idea.py` file and an extract of the `idea_view.xml` file.

In the idea module, the evolution of an idea starts from being a draft document, and ends by being closed or cancelled. In order to improve this small module for the management of ideas, we would like to record the creation date and the end date (close or cancellation) of the proposed ideas. To do so, two fields have to be added to the class idea. These fields will not be input by the user, but will be affected with the current date during the creation of the idea and while closing or cancelling it.

To describe your work, for the next three questions, you are going to:

- Give the name of the file and the number of the line you want to edit:

  "In the file *xxx.xx*, I replace the line *n* by ….."

- Give the name of the file and the number of the lines where you want to add new code:

  "In the file *xxx.xx*, between lines *n* and *n+1*, I add the lines ….."

Don't forget to read carefully the documentation provided in this document.

1) Add the two new fields `date_start` and `date_end` to the idea model, and then do the necessary modifications in order to make it appear when an idea is displayed.

2) The methods `idea_draft()`, `idea_close()` and `idea_cancel()` are called when an idea enters the states draft, closed and cancel, respectively. Modify these functions in order to initialize the two fields `date_start` and `date_end` in there.

3) Define three new filters in order to:

- Filter ideas created on the current day
- Filter ideas created less than one year ago
- Filter ideas created more than one year ago

**Documentation:**

a) Type of fields: date

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

b) Get the current date in Python:

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

c) The write method of OpenERP objects, which allows to affect values of fields:
(for example, it is called on lines 48, 51, 54 and 57 of `idea.py` in order to set the value of the field `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
- **ValidateError** – 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) Comparison operators in XML files:

| operator | xml | operator | xml | operator | xml |
|---|---|---|---|---|---|
| = | = | < | &lt; | > | &gt; |
| ≠ | != | ≤ | &lt;= | ≥ | &gt;= |

e) For the filtering of dates:

- Getting the current date:
  `context_today().strftime('%%Y-%%m-%%d')`
- Getting the date 1 year ago from current date:
  `(context_today() - datetime.timedelta(weeks=52)).strftime('%%Y-%%m-%%d')`

### idea.py

```python
from openerp.osv import osv
from openerp.osv import fields
from openerp.tools.translate import _
import time

VoteValues = [('-1', 'Not Voted'), ('0', 'Very Bad'), ('25', 'Bad'), \
              ('50', 'Normal'), ('75', 'Good'), ('100', 'Very Good') ]
DefaultVoteValue = '50'

class idea_category(osv.osv):
    """ Category of Idea """
    _name = "idea.category"
    _description = "Idea Category"
    _columns = {
        'name': fields.char('Category Name', size=64, required=True),
    }
    _sql_constraints = [
        ('name', 'unique(name)', 'The name of the category must be unique')
    ]
    _order = 'name asc'


class idea_idea(osv.osv):
    """ Idea """
    _name = 'idea.idea'
    _inherit = ['mail.thread']
    _columns = {
        'create_uid': fields.many2one('res.users', 'Creator', required=True, readonly=True),
        'name': fields.char('Idea Summary', size=64, required=True, readonly=True, oldname='title', states={'draft': [('readonly', False)]}),
        'description': fields.text('Description', help='Content of the idea', readonly=True, states={'draft': [('readonly', False)]}),
        'category_ids': fields.many2many('idea.category', string='Tags', readonly=True, states={'draft': [('readonly', False)]}),
        'state': fields.selection([('draft', 'New'),
            ('open', 'Accepted'),
            ('cancel', 'Refused'),
            ('close', 'Done')],
            'Status', readonly=True, track_visibility='onchange',
        )
    }
    _sql_constraints = [
        ('name', 'unique(name)', 'The name of the idea must be unique')
    ]
    _defaults = {
        'state': lambda *a: 'draft',
    }
    _order = 'name asc'

    def idea_cancel(self, cr, uid, ids, context=None):
        return self.write(cr, uid, ids, {'state': 'cancel'}, context=context)

    def idea_open(self, cr, uid, ids, context={}):
        return self.write(cr, uid, ids, {'state': 'open'}, context=context)

    def idea_close(self, cr, uid, ids, context={}):
        return self.write(cr, uid, ids, {'state': 'close'}, context=context)

    def idea_draft(self, cr, uid, ids, context={}):
        return self.write(cr, uid, ids, {'state': 'draft'}, context=context)
```

## idea_view.xml

```
<!-- New Idea Form View -->

<record model="ir.ui.view" id="view_idea_idea_form">
  <field name="name">idea.idea.form</field>
  <field name="model">idea.idea</field>
  <field name="arch" type="xml">
    <form string="Idea" version="7.0">
      <header>
        <button name="idea_open" string="Open" states="draft" class="oe_highlight"/>
        <button name="idea_close" string="Accept" states="open" class="oe_highlight"/>
        <button name="idea_cancel" string="Refuse" states="open"  class="oe_highlight"/>
        <field name="state" widget="statusbar" statusbar_visible="draft,open,close"/>
      </header>
      <sheet>
        <label for="name" class="oe_edit_only"/>
        <h1><field name="name"/></h1>
        <label for="category_ids" class="oe_edit_only"/>
        <field name="category_ids" widget="many2many_tags"/>
        <label for="description"/><newline/>
        <field name="description"/>
      </sheet>
      <div class="oe_chatter">
        <field name="message_follower_ids" widget="mail_followers"/>
        <field name="message_ids" widget="mail_thread"/>
      </div>
    </form>
  </field>
</record>

<!-- New Idea Tree View -->

<record model="ir.ui.view" id="view_idea_idea_tree">
  <field name="name">idea.idea.tree</field>
  <field name="model">idea.idea</field>
  <field name="arch" type="xml">
  <tree colors="blue:state == 'draft';black:state in ('open','close');gray:state == 'cancel'" string="Ideas">
    <field name="name"/>
    <field name="create_uid"/>
    <field name="state"/>
  </tree>
  </field>
</record>

<!-- Search Idea  -->

<record model="ir.ui.view" id="view_idea_idea_search">
  <field name="name">idea.idea.search</field>
  <field name="model">idea.idea</field>
  <field name="arch" type="xml">
    <search string="Ideas">
      <field name="name" string="Idea"/>
      <filter icon="terp-document-new" string="New" domain="[('state','=', 'draft')]" help="New Ideas"/>
      <filter icon="terp-camera_test" string="In Progress" domain="[('state','=', 'open')]" help="Open Ideas"/>
      <filter icon="terp-check" string="Accepted" domain="[('state','=','close')]" help="Accepted Ideas" />
      <field name="category_ids"/>
      <group expand="0" string="Group By...">
        <filter icon="terp-personal" string="Creator" help="By Creators" context="{'group_by':'create_uid'}"/>
        <filter icon="terp-stock_symbol-selection" string="Category" help="By Idea Category" context="{'group_by':'category_ids'}"/>
        <filter icon="terp-stock_effects-object-colorize" string="Status" help="By States" context="{'group_by':'state'}"/>
      </group>
    </search>
  </field>
</record>
```