

**Unité d'Enseignement en Informatique  
Année 2015-2016**

**Master M1 EFREI – ASI – ISI**  
**Devoir Écrit de ERP – Première Session du 5 Avril 2016**  
 (durée 1h45, aucun document autorisé, aucun ustensile de calcul et/ou de communication)

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

**Exercice 1: (5 points)**

- 1) Quelles sont les trois particularités techniques qui caractérisent les logiciels ERP ?
- 2) Quels coûts doivent être pris en compte dans le budget d'un projet ERP ?
- 3) Quel est le but des entités organisationnelles dans SAP ? (2 réponses)
  - a) Elles représentent une entité légale
  - b) Les données sont saisies selon les entités organisationnelles
  - c) Les données sont tracées selon les entités organisationnelles
  - d) Elles représentent le serveur applicatif
- 4) Quelles propositions décrivent les données et les transactions dans SAP ? (2 réponses)
  - a) Un document est créé pour chaque transaction dans le système
  - b) Les données de référence (données maîtres) peuvent être supprimée à tout moment
  - c) Les documents sont créés en utilisant les données de référence (données maîtres) et les entités organisationnelles
  - d) Les données de référence (données maîtres) saisies au niveau client peuvent seulement être utilisées par un seul code entreprise
- 5) SAP ERP inclut : (2 réponses)
  - a) SAP Customer Relationship Management
  - b) SAP Supply Chain Management
  - c) SAP Netweaver
  - d) SAP ECC Core
- 6) Quel est le plus grand élément organisationnel dans SAP ?

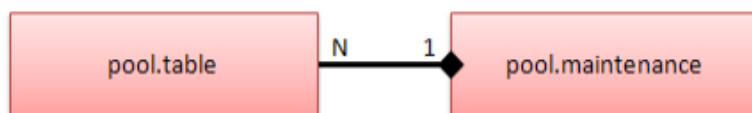
**Exercice 2: (3 points)**

- 1) Qu'est que le mode SaaS ?
- 2) Pouvez-vous donner des avantages à utiliser un ERP en mode SaaS ?
- 3) Pouvez-vous donner des inconvénients ?

**Exercice 3: (12 points)**



Un centre de billard (*pool*) possède de tables de billard, situées sur différents étages. Le centre de billard propose quatre genres de tables : Snooker (3,6×1,8 m), Américain (2,5×1,3 m), Français (3,1×1,7 m) et Anglais (2,1×1,0 m). Ces tables sont louées aux joueurs et les joueurs payent en fonction du temps passé sur les tables. Afin de gérer le centre billard, nous avons commencé le développement d'un nouveau module OpenERP. Le but de ce module est de gérer l'occupation des tables et les billets de maintenance des tables. Le modèle de donnée de ce nouveau module est décrit ci-dessous. Chaque billet de maintenance concerne une table, et chaque table peut être maintenue plusieurs fois.



L'écriture du code source de `pool.category` a déjà commencé. Voir les quatre fichiers `pool.py`, `pool_data.xml`, `pool_view.xml` et `pool_workflow.xml` ci-après. Afin de poursuivre ce développement, vous devez répondre aux questions suivantes.

1) Que font les boutons dans la vue `form` de `pool.table` ? Que se passera-t'il quand ils seront cliqués par un utilisateur ?

2) Définir l'objet métier `pool.maintenance` avec 3 colonnes, définissant notamment la relation avec l'autre autre objet métier et aussi :

- Le nom du billet de maintenance (mais pas étiqueté « Name » mais « Title »)
- La description du billet de maintenance (qui doit pouvoir être écrit comme un texte sur plusieurs lignes)

3) Créer les vues `form` et `tree` de l'objet métier `pool.maintenance`.

4) Un flux de travail (ou *workflow*) a déjà été défini pour `pool.table` (voir le fichier `opendata_workflow.xml`) et est illustré figure 2. Afin d'implanter le flux de travail de `pool.maintenance`, illustré figure 3, faire ce qui suit :

- a) Donner les modifications de code à apporter au modèle `pool.maintenance`
- b) Donner la description XML de ce second flux de travail dans le fichier `opendata_workflow.xml`

5) Créer la vue `search` de l'objet métier `pool.maintenance` et définir les filtrages suivant :

- Filtrer les billets de maintenance restant à faire
- Filtrer les billets de maintenance terminés
- Regrouper les billets de maintenance par état
- Regrouper les billets de maintenance par table de billard

## pool.py

```

1 from openerp.osv import osv
2 from openerp.osv import fields
3 from openerp.tools.translate import _
4 import time
5
6 listKND = [('sn', 'Snooker'), ('am', 'American'), \
7           ('fr', 'French'), ('en', 'English')]
8
9 class pool_table(osv.osv):
10     """ A pool table """
11     _name = "pool.table"
12     _description = "Pool table "
13     _columns = {
14         'name': fields.char('Table number', size=64, required=True),
15         'kind': fields.selection(listKND, 'Kind of table', required=True),
16         'floor': fields.integer('Location of the table'),
17         'state': fields.selection(
18             [
19                 ('ready', 'Ready'),
20                 ('inuse', 'In use'),
21                 ('maint', 'In maintenance'),
22                 ('close', 'Closed'),
23             ],
24             'Status', readonly=True, track_visibility='onchange',
25         )
26     }
27     _sql_constraints = [
28         ('name', 'unique(name)', 'The name of a pool table must be unique')
29     ]
30     _order = 'name asc'
31
32     def table_ready (self, cr, uid, ids, context={}):
33         return self.write(cr, uid, ids, {'state': 'ready'}, context=context)
34
35     def table_inuse (self, cr, uid, ids, context={}):
36         return self.write(cr, uid, ids, {'state': 'inuse'}, context=context)
37
38     def table_maint (self, cr, uid, ids, context={}):
39         return self.write(cr, uid, ids, {'state': 'maint'}, context=context)
40
41     def table_close (self, cr, uid, ids, context={}):
42         return self.write(cr, uid, ids, {'state': 'close'}, context=context)

```

## pool\_data.xml

```

1 <?xml version="1.0"?>
2 <openerp>
3     <data>
4         <!-- Table -->
5         <record model="pool.table" id="pool_table_0">
6             <field name="name">001</field>
7             <field name="kind">am</field>
8             <field name="floor">0</field>
9         </record>
10        <record model="pool.table" id="pool_table_1">
11            <field name="name">003</field>
12            <field name="kind">fr</field>
13            <field name="floor">0</field>
14        </record>
15        <record model="pool.table" id="pool_table_2">
16            <field name="name">004</field>
17            <field name="kind">en</field>
18            <field name="floor">0</field>
19        </record>
20        <record model="pool.table" id="pool_table_3">
21            <field name="name">101</field>
22            <field name="kind">en</field>
23            <field name="floor">1</field>
24        </record>
25        <record model="pool.table" id="pool_table_4">
26            <field name="name">103</field>
27            <field name="kind">en</field>
28            <field name="floor">1</field>
29        </record>
30        <record model="pool.table" id="pool_table_5">

```

31	<field name="name">104</field>
32	<field name="kind">en</field>
33	<field name="floor">1</field>
34	</record>
35	<record model="pool.table" id="pool_table_6">
36	<field name="name">105</field>
37	<field name="kind">am</field>
38	<field name="floor">1</field>
39	</record>
40	<record model="pool.table" id="pool_table_7">
41	<field name="name">301</field>
42	<field name="kind">sn</field>
43	<field name="floor">3</field>
44	</record>
45	</data>
46	</openerp>

pool_view.xml	
1	<?xml version="1.0"?>
2	<openerp>
3	<data>
4	
5	<!-- Pool Table: Form View -->
6	<record model="ir.ui.view" id="view_pool_table_form">
7	<field name="name">pool.table.form</field>
8	<field name="model">pool.table</field>
9	<field name="arch" type="xml">
10	<form string="Table of Pool" version="7.0">
11	<header>
12	<button name="signal_table_inuse" string="In use" states="ready" class="oe_highlight" />
13	<button name="signal_table_ready" string="Ready" states="inuse" class="oe_highlight" />
14	<button name="signal_table_maint" string="Maintain" states="ready" class="oe_highlight" />
15	<button name="signal_table_ready" string="Ready" states="maint" class="oe_highlight" />
16	<button name="signal_table_close" string="Close" states="ready" class="oe_highlight" />
17	<field name="state" widget="statusbar" statusbar_visible="ready,inuse,close" />
18	</header>
19	<sheet>
20	<label for="name"/><field name="name"/>
21	<label for="kind"/><field name="kind"/>
22	<label for="floor"/><field name="floor"/>
23	</sheet>
24	</form>
25	</field>
26	</record>
27	
28	<!-- Pool Table: Tree View -->
29	<record model="ir.ui.view" id="view_pool_table_tree">
30	<field name="name">pool.table.tree</field>
31	<field name="model">pool.table</field>
32	<field name="field_parent"></field>
33	<field name="arch" type="xml">
34	<tree string="Table of Open Data">
35	<field name="name"/>
36	<field name="kind"/>
37	<field name="floor"/>
38	<field name="state"/>
39	</tree>
40	</field>
41	</record>
42	
43	<!-- Pool Table: Search View -->
44	<record model="ir.ui.view" id="view_pool_table_search">
45	<field name="name">pool.table.search</field>
46	<field name="model">pool.table</field>
47	<field name="arch" type="xml">
48	<search string="Models of Open Data">
49	<filter string="Snooker" domain="[('kind','=', 'sn')]" help="Snooker tables"/>
50	<filter string="American" domain="[('kind','=', 'am')]" help="American tables"/>
51	<filter string="French" domain="[('kind','=', 'fr')]" help="French tables"/>
52	<filter string="English" domain="[('kind','=', 'en')]" help="English tables"/>
53	<filter string="Free tables" domain="[('state','=', 'ready')]" help="Free tables"/>
54	<filter string="Not free tables" domain="[('state','!=', 'ready')]" help="Not free tables"/>
55	<group expand="0" string="Group By...">
56	<filter string="floor" help="Floor location" context="{ 'group_by': 'floor' }"/>
57	<filter string="kind" help="Kind of table" context="{ 'group_by': 'kind' }"/>
58	</group>

```

59     </search>
60     </field>
61 </record>
62
63 <!-- Pool Table: Action -->
64 <record model="ir.actions.act_window" id="action_pool_table">
65     <field name="name">Tables</field>
66     <field name="res_model">pool.table</field>
67     <field name="view_type">form</field>
68     <field name="view_mode">tree,form</field>
69     <field name="search_view_id" ref="view_pool_table_search"/>
70 </record>
71
72 <!-- Top menu item -->
73 <menuitem name="Pools" id="base.menu_pool_root" sequence="120" groups="base.group_user"/>
74
75 <!-- Menus sections -->
76 <menuitem name="Pools" id="menu_pool_pools" parent="base.menu_pool_root" sequence="2"/>
77
78 <!-- Menus items -->
79 <menuitem name="Tables" id="menu_pool_tables" parent="menu_pool_pools"
80 action="action_pool_table" sequence="1"/>
81 </data>
82 </openerp>

```

## pool\_workflow.xml

```

1 <?xml version="1.0"?>
2 <openerp>
3   <data>
4     <!-- WORKFLOW -->
5     <record model="workflow" id="wkf_table">
6       <field name="name">table.wkf</field>
7       <field name="osv">pool.table</field>
8       <field name="on_create">True</field>
9     </record>
10
11     <!-- STATES -->
12     <record model="workflow.activity" id="act_table_ready">
13       <field name="wkf_id" ref="wkf_table" />
14       <field name="name">ready</field>
15       <field name="kind">function</field>
16       <field name="action">table_ready</field>
17       <field name="flow_start">True</field>
18     </record>
19     <record model="workflow.activity" id="act_table_inuse">
20       <field name="wkf_id" ref="wkf_table" />
21       <field name="name">inuse</field>
22       <field name="kind">function</field>
23       <field name="action">table_inuse</field>
24     </record>
25     <record model="workflow.activity" id="act_table_maint">
26       <field name="wkf_id" ref="wkf_table" />
27       <field name="name">maint</field>
28       <field name="kind">function</field>
29       <field name="action">table_maint</field>
30     </record>
31     <record model="workflow.activity" id="act_table_close">
32       <field name="wkf_id" ref="wkf_table" />
33       <field name="name">close</field>
34       <field name="kind">function</field>
35       <field name="action">table_close</field>
36       <field name="flow_stop">True</field>
37     </record>
38
39     <!-- TRANSITIONS -->
40     <record model="workflow.transition" id="trans_table_ready_inuse">
41       <field name="act_from" ref="act_table_ready" />
42       <field name="act_to" ref="act_table_inuse" />
43       <field name="signal">signal_table_inuse</field>
44     </record>
45     <record model="workflow.transition" id="trans_table_inuse_ready">
46       <field name="act_from" ref="act_table_inuse" />
47       <field name="act_to" ref="act_table_ready" />
48       <field name="signal">signal_table_ready</field>
49     </record>
50     <record model="workflow.transition" id="trans_table_ready_maint">

```

```

51     <field name="act_from" ref="act_table_ready" />
52     <field name="act_to" ref="act_table_maint" />
53     <field name="signal">signal_table_maint</field>
54 </record>
55 <record model="workflow.transition" id="trans_table_maint_ready">
56     <field name="act_from" ref="act_table_maint" />
57     <field name="act_to" ref="act_table_ready" />
58     <field name="signal">signal_table_ready</field>
59 </record>
60 <record model="workflow.transition" id="trans_table_ready_close">
61     <field name="act_from" ref="act_table_ready" />
62     <field name="act_to" ref="act_table_close" />
63     <field name="signal">signal_table_close</field>
64 </record>
65
66 </data>
67 </openerp>

```

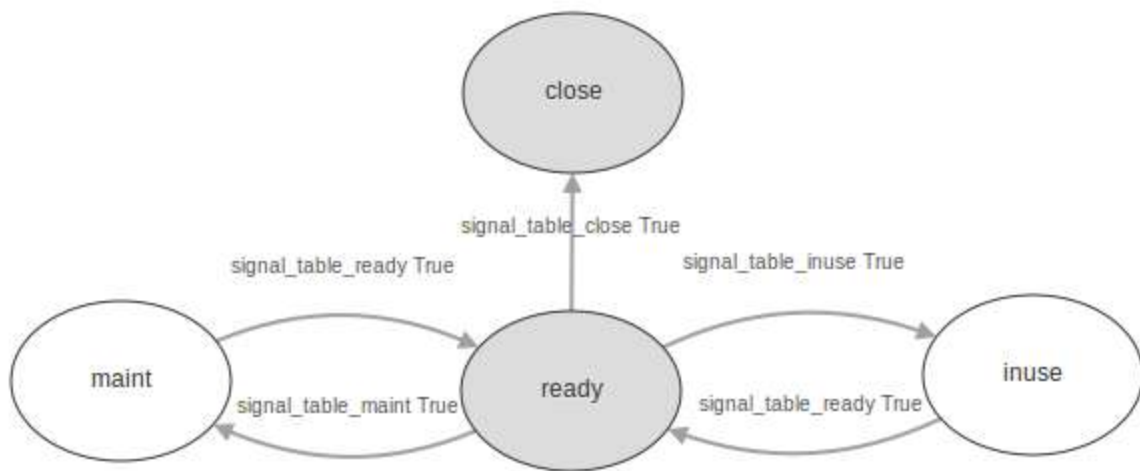


Figure 2: the workflow of a table.

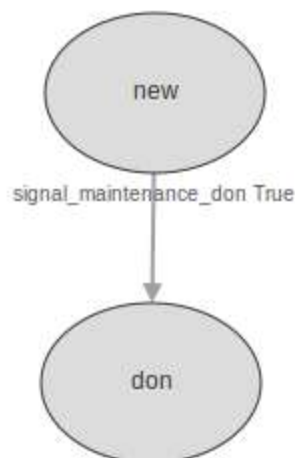


Figure 3: the workflow of maintenance to implement.