

**Unité d'Enseignement en Informatique
Année 2013-2014**

**Master M1 EFREI – ASI – ISI
Devoir Écrit de ERP – Deuxième Session de Juin 2014
(durée 2h00, aucun document autorisé)**

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

Exercice 1: (2 points)

Dans les années 1960 et le début des années 1970, avant que n'apparaissent les premiers éditeurs de logiciels et les premiers progiciels, les logiciels étaient développés en spécifiques par les entreprises elle-même.

- 1) Pour une entreprise, quels sont les avantages et inconvénients de développer ses logiciels elle-même ?
- 2) Pour une entreprise, quels sont les avantages et inconvénients d'utiliser le progiciel d'un éditeur ? Des développements supplémentaires sont souvent nécessaires pour combler l'écart entre la couverture fonctionnelle de l'ERP et les besoins de l'entreprise.
- 3) Quel est le langage de développement des modules pour l'ERP de SAP ?
- 4) Quel est le langage de développement des modules pour OpenERP ?

Exercice 2: (6 points)

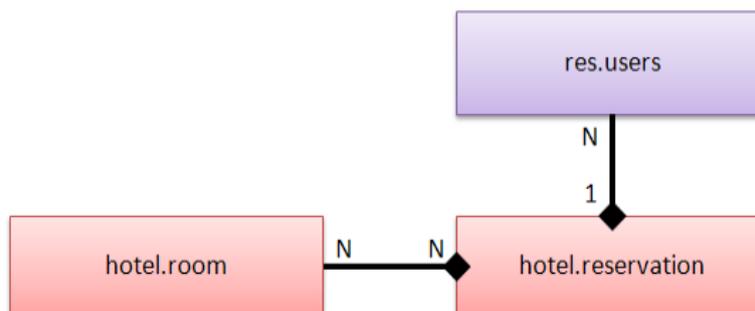
Une analyse de trois ERP, montrant quelles applications sont intégrées (I) au composant central de l'ERP et lesquelles sont installables en tant que module (M), est présentée dans le tableau suivant :

Application	ERP1	ERP2	ERP3
Sales	I	M	I
Purchasing		M	I
Warehouse Management	I	M	I
CRM		M	I
MRP	I	M	I
Accounting		M	M
Human Resources		M	M
CMMS	I	M	

- 1) Lequel de ces ERP est le plus facile et rapide à installer ? Pourquoi ?
- 2) Lequel de ces ERP est le plus difficile et long à installer ? Pourquoi ?
- 3) Lequel recommanderiez-vous pour une grande entreprise multinationale ? Pourquoi ?

Exercice 3: (12 points)

Nous voulons verticaliser OpenERP pour la gestion d'un hôtel. Même si une verticalisation d'OpenERP pour la gestion d'hôtels existe déjà, nous voulons en écrire une nouvelle mieux adaptée à nos besoins. Le modèle de données est décrit ci-dessous :



L'écriture du code source de `hotel.room` a déjà commencé. Voir les deux fichiers `hotel.py` and `hotel_view.xml` ci-après. Vous devez poursuivre ce développement.

Écrivez le modèle et la vue pour le modèle `hotel.reservation`, sachant que :

- Chaque réservation possède une date d'entrée (*checkin*) et de sortie (*checkout*) ;
- Une réservation appartient à un (et un seul) client, mais un client peut faire plusieurs réservations ;
- Une réservation peut concerner plusieurs chambres ;
- Chaque réservation doit informer sur le nombre d'enfants et d'adultes qui dormiront à l'hôtel ;
- Le système devra indiquer si la reservation a déjà été payée ou non.
- Nous voulons pouvoir :
 - Filtrer les réservations payées et pas payées,
 - Filtrer les réservations avec plus de 2 enfants,
 - Filtrer les réservations avec adultes mais pas d'enfants,
 - Filtrer les réservations avec adultes et enfants,
 - Filtrer les reservations dont la date d'entrée est celle du jour courant,
 - Filtrer les reservations dont la date de sortie est celle du jour courant,
 - Grouper les reservations par client.

Documentation :

Récupérer la date du jour dans les fichiers XML :

```
context_today().strftime('%Y-%m-%d')
```

hotel.py	
1	<code>from openerp.osv import osv</code>
2	<code>from openerp.osv import fields</code>
3	<code>from openerp.tools.translate import _</code>
4	<code>import time</code>
5	
6	<code>class hotel_room(osv.osv):</code>
7	<code> """ Room """</code>
8	<code> _name = 'hotel.room'</code>
9	<code> _description = "Hotel Room"</code>
10	<code> _columns = {</code>
11	<code> 'name': fields.integer('Room number', required=True),</code>
12	<code> 'single_bed': fields.integer('Number of single beds', required=True),</code>
13	<code> 'double_bed': fields.integer('Number of double beds', required=True),</code>
14	<code> 'shower': fields.boolean('Presence of a shower', required=True),</code>
15	<code> 'bath': fields.boolean('Presence of a bath', required=True),</code>
16	<code> 'balcony': fields.boolean('Presence of a balcony', required=True),</code>
17	<code> 'table': fields.boolean('Presence of a table', required=True),</code>
18	<code> 'couch': fields.boolean('Presence of a couch', required=True),</code>
19	<code> 'smoking': fields.boolean('Smoking allowed', required=True),</code>
20	<code> }</code>
21	<code> _sql_constraints = [</code>
22	<code> ('name', 'UNIQUE(name)', 'The number of a room must be unique')</code>
23	<code>]</code>
24	<code> _order = 'name ASC'</code>
25	

hotel_view.xml

```

1 <?xml version="1.0"?>
2 <openerp>
3   <data>
4
5     <!-- Top menu item -->
6     <menuitem name="Hotel"
7               id="base.menu_hotel_root"
8               sequence="120"
9               groups="base.group_user"/>
10
11    <!-- Configuration menu item -->
12    <menuitem name="Configuration"
13             parent="base.menu_hotel_root"
14             id="menu_hotel_configuration"
15             sequence="20" />
16
17    <!-- New Room Form View -->
18    <record model="ir.ui.view" id="view_hotel_room_form">
19      <field name="name">hotel.room.form</field>
20      <field name="model">hotel.room</field>
21      <field name="arch" type="xml">
22        <form string="Hotel" version="7.0">
23          <sheet>
24            <label for="name"/> <field name="name"/>
25            <label for="single_bed"/> <field name="single_bed" widget="integer"/>
26            <label for="double_bed"/> <field name="double_bed" widget="integer"/>
27            <label for="shower"/> <field name="shower"/>
28            <label for="bath"/> <field name="bath"/>
29            <label for="balcony"/> <field name="balcony"/>
30            <label for="table"/> <field name="table"/>
31            <label for="couch"/> <field name="couch"/>
32            <label for="smoking"/> <field name="smoking"/>
33          </sheet>
34        </form>
35      </field>
36    </record>
37
38    <!-- Room Tree View -->
39    <record model="ir.ui.view" id="view_hotel_room_tree">
40      <field name="name">hotel.room.tree</field>
41      <field name="model">hotel.room</field>
42      <field name="arch" type="xml">
43        <tree string="Hotels">
44          <field name="name"/>
45          <field name="single_bed"/>
46          <field name="double_bed"/>
47          <field name="shower"/>
48          <field name="bath"/>
49          <field name="balcony"/>
50          <field name="table"/>
51          <field name="couch"/>
52          <field name="smoking"/>
53        </tree>
54      </field>
55    </record>
56
57    <!-- Room Search -->
58    <record model="ir.ui.view" id="view_hotel_room_search">
59      <field name="name">hotel.room.search</field>
60      <field name="model">hotel.room</field>
61      <field name="arch" type="xml">
62        <search string="Hotels">
63          <filter string="Smoking allowed" domain="[('smoking','=', True)]" help="Smoking
64 allowed in room"/>
65          <filter string="Smoking not allowed" domain="[('smoking','=', False)]" help="Smoking
66 not allowed in room"/>
67          <filter string="With double bed" domain="[('double_bed','>', 0)]" help="Having at
68 least one double bed"/>
69          <filter string="With bath" domain="[('bath','=', True)]" help="With a shower"/>
70          <filter string="With shower" domain="[('shower','=', True)]" help="With a bath"/>
71          <group expand="0" string="Group By...">
72            <filter string="Smoking" help="By smoking" context="{ 'group_by': 'smoking' }"/>
73            <filter string="Shower" help="By shower" context="{ 'group_by': 'shower' }"/>
74            <filter string="Bath" help="By bath" context="{ 'group_by': 'bath' }"/>
75          </group>
76        </search>
77      </field>
78    </record>

```

```
76
77 <!-- Hotel Room Action -->
78
79 <record model="ir.actions.act_window" id="action_hotel_room">
80   <field name="name">Hotels</field>
81   <field name="res_model">hotel.room</field>
82   <field name="view_type">form</field>
83   <field name="view_mode">tree,form</field>
84   <field name="search_view_id" ref="view_hotel_room_search"/>
85 </record>
86
87 <menuitem name="Rooms"
88   parent="menu_hotel_configuration"
89   id="menu_hotel_room"
90   action="action_hotel_room"
91   sequence="1"/>
92
93 </data>
94 </openerp>
```