

Office de la Formation Professionnelle  
et de la Promotion du Travail

Direction de la Recherche et Ingénierie de la Formation

**Examen de fin de formation  
session Juillet 2015**

Filière : **Techniques de Développement Informatique**  
Niveau : **TS**  
Durée : **4 heures**

Epreuve : **Théorique**  
Barème : **/40 pts**

**Dossier 1 : (10 pts)**

Une entreprise désire développer une application permettant la création d'agendas électroniques partagés de ses employés. Chaque employé peut utiliser son agenda pour saisir ses activités personnelles et aussi pour modifier ou supprimer ces activités. L'agenda d'un employé peut être consulté par tous les employés. Les chefs de services peuvent utiliser l'application pour inscrire des activités personnelles dans l'agenda des employés qui travaillent dans leur service. Le directeur des ressources humaines utilise l'application pour inscrire des activités à exécuter dans l'agenda des chefs de service. Un administrateur utilise l'application pour créer ou supprimer les agendas.

Un employé est identifié par un matricule, le nom, le prénom et adresse email. Chaque employé travaille dans un service unique qui est caractérisé par un code et un nom. Un service possède un chef de service. Chaque employé a un seul agenda qui est identifié par le code et la date de création. Une activité est caractérisée par un type d'activité (exemple « **exposé** », « **réunion** », « **séminaire** »...), par l'heure de début, l'heure de fin de réalisation de l'activité et par une description (exemple « exposé sur les nouveaux produits »). Une activité peut être unique ou périodique (chaque semaine). Une activité unique possède en plus la date de réalisation. Une activité périodique est caractérisée par les jours de la semaine concernés par cette activité (exemple chaque 'lundi' et 'jeudi'). Un employé peut inscrire une activité privée (non professionnelle comme « rendez-vous médecin » ...) dans son agenda ; dans ce cas cette activité est marquée par l'information « non visible » (les autres employés ne peuvent pas la consulter).

1. Etablir le diagramme de cas d'utilisation. **(3 pts)**
2. Etablir le diagramme de séquence système du cas d'utilisation « inscrire activités pour chefs de services ». **(3 pts)**
3. Etablir le diagramme de classes. **(4 pts)**

| Filière | Epreuve                    | Session      | 1/5 |
|---------|----------------------------|--------------|-----|
| DI      | Théorique Fin de Formation | Juillet 2015 |     |



## Dossier 2 : (13pts)

Pour gérer les élections des représentants des employés d'une entreprise, une application utilise une base de données composée des tables Electeurs, Candidats et Votes :

- **Electeurs** (idElecteur , nomElecteur, prenomElecteur, aVoté)  
Le champ aVoté prend la valeur 0 quand l'électeur n'a pas encore voté et 1 quand il a voté.
- **Candidats** (idCandidat , nomCandidat, prenomCandidat, dateNaissance, nombreVoix)  
nombreVoix est le nombre de voix obtenus par le candidat.
- **Votes** (#idElecteur , #idCandidat)  
Cette table enregistre les votes des électeurs.

**N.B :** Les champs en gras et soulignés sont des clés primaires, les champs précédés par # sont des clés étrangères.

Un électeur ne peut voter qu'une seule fois pour choisir 1, 2 ou 3 candidats.

1. Ecrire une requête qui affiche la liste des électeurs qui ont choisi un seul candidat pendant le vote. **(2 pts)**
2. Ecrire une requête qui affiche les trois premiers candidats qui ont gagné les élections (ceux qui ont obtenu le plus grand nombre de voix). En cas d'égalité des nombres de voix, on retient le candidat le plus âgé. **(3 pts)**
3. Ecrire une fonction qui retourne dans une table la liste des électeurs ayant votés pour un candidat donné. **(2 pts)**
4. Ecrire un trigger qui permet d'incrémenter de 1, le champ nombreVoix d'un candidat à chaque ajout d'une ligne à la table **Votes** qui concerne ce candidat. Le trigger doit également mettre le champ aVoté à 1 pour l'électeur qui vient de voter. **(3 pts)**
5. Ecrire une procédure stockée qui permet d'enregistrer le vote d'un électeur ; elle a les paramètres : **(3 pts)**
  - @idElect : identifiant de l'électeur.
  - @idCandidat1, @idCandidat2 et @idCandidat3 : identifiants des 3 candidats choisis par l'électeur (si l'électeur choisit moins de 3 candidats, les valeurs non choisies restent NULL).
  - La procédure ajoute 1 à 3 lignes à la table Votes selon les valeurs non NULL des paramètres @idCandidat1, @idCandidat2 et @idCandidat3.

Si ces paramètres sont tous NULL, la procédure affiche un message d'erreur.

|         |                            |              |     |
|---------|----------------------------|--------------|-----|
| Filière | Epreuve                    | Session      | 2/5 |
| DI      | Théorique Fin de Formation | Juillet 2015 |     |



### Dossier 3 : (6pts)

Une Application de gestion des emplois du temps a généré le fichier XML suivant :

```
<?xml version="1.0"?>
<seances>
  <seance jour="lundi" debut="08:00" fin="10:30">
    <salle>S1</salle>
    <module>SGBDR I</module>
    <formateur>CHAKIR Said</formateur>
    <groupe>TDI201</groupe>
  </seance>
  <seance jour="lundi" debut="10:30" fin="13:00">
    <salle>S2</salle>
    <module>Reseaux </module>
    <formateur>TAHIR Nadia</formateur>
    <groupe>TDI201</groupe>
  </seance>
  <seance jour="mardi" debut="08:00" fin="10:30">
    <salle>S1</salle>
    <module>SGBDR I</module>
    <formateur>CHAKIR Said</formateur>
    <groupe>TDI202</groupe>
  </seance>
  .....
</seances>
```

- 1) Donner les expressions XPath à appliquer à ce document XML pour extraire :
- a) Toutes les séances du formateur CHAKIR Said. **(1 pt)**
  - b) Toutes les salles utilisées par le groupe TDI201. **(1 pt)**
  - c) Le nombre de séances qui commencent à 08 :00 heure. **(1 pt)**

|                |                            |                |     |
|----------------|----------------------------|----------------|-----|
| <i>Filière</i> | <i>Epreuve</i>             | <i>Session</i> | 3/5 |
| DI             | Théorique Fin de Formation | Juillet 2015   |     |

2) Soit le fichier XSL suivant :

```
< ?xml version='1.0' ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<html>
<body>
<table border="1">
<tr>
<td>Formateurs</td>
<td>Groupes</td>
<td>Modules</td>
</tr>
<xsl:for-each select=''seances/seance''>
<tr>
<td> <xsl:value-of select=''formateur''/> </td>
<td> <xsl:value-of select=''groupe''/> </td>
<td> <xsl:value-of select=''module''/> </td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

- Expliquer la syntaxe : `<xsl:for-each select="seances/seance">`. (1 pt)
- Donner l'affichage obtenu si on applique ce fichier XSL au document XML précédent. (1 pt)
- Ajouter une ligne à ce fichier XSL pour inclure l'affichage des jours des séances. (1 pt)

#### **Dossier 4 : (6 pts)**

On veut réaliser une page Web statique contenant un formulaire permettant de calculer la valeur énergétique d'un repas en kilo calorie. L'utilisateur sélectionne, dans des cases à cocher, les aliments qui composent le repas avec les quantités, puis clique sur le bouton « **Calculer** » pour afficher le total des calories du repas sachant que :

- 100 grammes de pain donne 275 kilo calories
- 100 grammes de viandes donne 180 kilo calories
- 100 grammes de légumes donne 40 kilo calories
- 1 fruit de banane donne 116 kilo calories
- 1 fruit de pomme donne 80 kilo calories
- 1 yaourt donne 140 kilo calories

|                |                            |                |     |
|----------------|----------------------------|----------------|-----|
| <i>Filière</i> | <i>Epreuve</i>             | <i>Session</i> | 4/5 |
| DI             | Théorique Fin de Formation | Juillet 2015   |     |



# Calcul valeur énergétique d'un repas

Choisir les aliments qui composent votre repas :

- ☒ Pain :  grammes  
☒ Viande :  grammes  
☒ Légumes :  grammes  
☐ Banane :  unités  
☒ Pomme :  unités  
☐ Yaourt :  unités

Calculer

La valeur énergétique de votre repas est :  Kilo calories

- 1) Ecrire le code HTML de cette page. (2 pts)
- 2) Ecrire le code JavaScript ou jQuery associé au bouton « **Calculer** » ; tenir compte de la validation suivante : Au moins une case à cocher doit être sélectionnée sinon afficher un message d'erreur. (4 pts)

## Dossier 5 : (5pts)

- 1) Donner la classe d'adresse de chacune des adresses IP suivantes : (2pts)  
131.107.2.89 - 3.3.57.1 - 196.179.110.15 - 120.55.55.55
- 2) Quelles classes d'adresses permettent d'avoir plus de 1000 hôtes par réseau ? (1pt)
- 3) Donner le nom et une description de chacune des couches 1, 2, 4 et 5 du modèle OSI en remplissant le tableau suivant : (2pts)

| Numéro de la couche | Nom | Description |
|---------------------|-----|-------------|
| 1                   |     |             |
| 2                   |     |             |
| 4                   |     |             |
| 5                   |     |             |

|         |                            |              |     |
|---------|----------------------------|--------------|-----|
| Filière | Epreuve                    | Session      | 5/5 |
| DI      | Théorique Fin de Formation | Juillet 2015 |     |