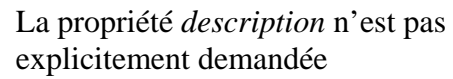


E4D : ÉTUDE DE CAS

**CAS THALI
ÉLÉMENTS DE
CORRIGÉ**

Proposition de corrigé



- L'entité spécialisée WEEK-END n'est pas indispensable.
- L'entité SAISON-ANNÉE peut être appelée SAISON et être identifiée par son libellé.
- Les périodes constitutives d'une saison peuvent être représentées de la façon suivante :



1.2	Proposer un schéma relationnel issu du sous-schéma relatif aux soins.
-----	---

ETAPE (codeSoin, numéro, description, durée)
codeSoin, numéro : clé primaire
codeSoin : clé étrangère en référence à code de SOIN

Plusieurs solutions sont possibles pour la représentation de l'héritage au niveau du schéma relationnel :

- 1) Créer deux relations correspondant à l'entité mère et l'entité fille.

SOIN (code, libellé)
code : clé primaire

SOIN_COLLECTIF (codeSoinCollectif, effectifMaxi)
codeSoinCollectif : clé primaire
codeSoinCollectif : clé étrangère en référence à code de SOIN

Cette solution est acceptable car elle traduit fidèlement le schéma conceptuel mais on peut critiquer son opérationnalité : exploiter les deux relations va entraîner des traitements de jointure, un coût qui eput paraître élevé au regard de ce qu'apporte la relation SOIN_COLLECTIF qui possède un seul attribut spécifique.

- 2) Créer une seule relation avec un attribut discriminant les types de soin.

SOIN (code, libellé, typeSoin, effectifMaxi)
code : clé primaire

L'attribut *typeSoin* permet de discriminer les prestations de soins collectifs des autres. On peut également opter pour un booléen *Collectif*. Dans les deux cas, la valorisation ou le renseignement de l'attribut *effectifMaxi* sont conditionnés par la valorisation de l'attribut précédent (contraintes d'intégrité).

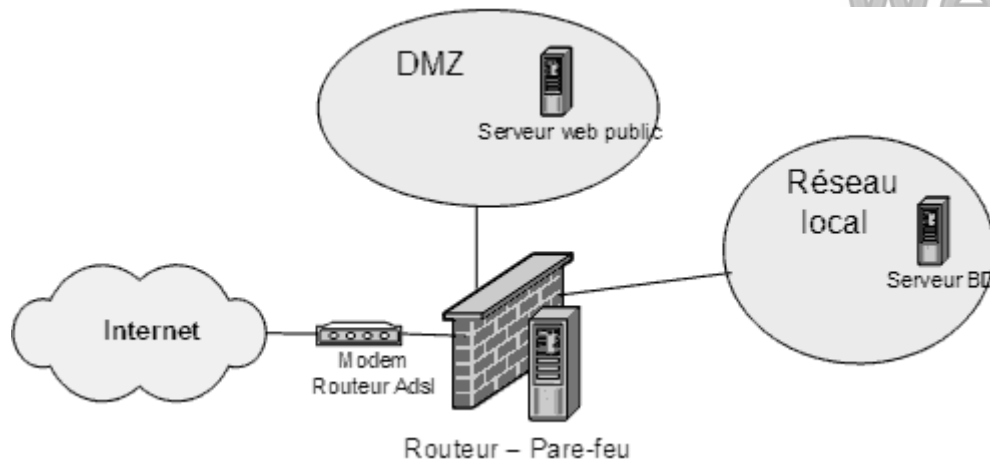
On peut enfin utiliser l'attribut *effectifMaxi* comme attribut discriminant en le valorisant à 1 pour les soins individuels. Ce genre de considération aurait pu être pris en compte dès le niveau conceptuel.

Dossier 2	La mise à disposition de la documentation
------------------	--

2.1	Indiquer les précautions à prendre pour protéger au mieux le réseau local (en précisant clairement la localisation du serveur <i>web</i> et du serveur de bases de données dans la nouvelle architecture) et illustrer le propos à l'aide d'un schéma.
-----	--

Pour protéger au maximum le réseau local, la a solution pour protéger au maximum le réseau local est la mise en place d'une DMZ et la localisation du serveur *web* en DMZ ; ainsi l'accès au réseau local est interdit depuis l'extérieur.

La base de données utilisée pour les applications internet pourra être hébergée sur le serveur de données existant qui doit bien sûr être situé dans le réseau local de THALI.



Autre architecture possible : mettre le pare-feu juste derrière le modem-routeur ADSL.

2.2	À l'aide des fonctions de script fournies en annexe 2C et en utilisant la classe <i>GenPDF</i> décrite en annexe 2D , écrire le script <i>genDoc.php</i> permettant la génération et l'affichage d'un document au format PDF personnalisé. Le code du script pourra être écrit en pseudo-langage ou en PHP, au choix du candidat.
-----	---

Corrigé du script en pseudo-langage :

```

doc      : Objet de type GenPDF
datj     : Date
txt      : Chaîne
i        : Entier
tabChoixDoc[0 : 9] : tableau de Chaîne
Début
doc.init()
datj ← dateJour()
doc.ajoutTexte("Documentation éditée le : "+datj)
doc.sautLigne(1)
tabChoixDoc ← extParamVersTab()
// Ajout des documents, le dernier est ajouté à part
Pour i de 0 à (compte(tabChoixDoc) - 2)
    txt ← genTxt(tabChoixDoc[i])
    doc.ajoutTexte(txt)
    doc.sautPage()
Fpour
// Génération du dernier document sans saut de page après : Non exigé
txt ← genTxt(tabChoixDoc[compte(tabChoixDoc)-1])
doc.ajoutTexte(txt)
// Affichage du PDF
doc.edite()
Fin

```

Corrigé du script en PHP :

```

<?
require('fonctionsPerso.php');

$doc = new GenPDF();
$datj = dateJour();
$doc->cell("Documentation éditée le : ".$datj)
$doc->Ln(1);
$tab= extParamCheck();

```

```
For ($i=0; $i<(count($tab)-1); $i++)
{
    $txt = genTxt($tab[$i])
    $doc->Cell($txt);
    $doc->AddPage();
}
$txt = genTxt($tab[count($tab)-1])
$doc->Cell($txt);
$doc->Output();
?>
```

Remarque concernant le sujet : *erreur à rectifier dans l'annexe 2B*

Annexe 2B – Extrait du code source du formulaire de la page HTML et exemple d'URL transmise

```
<form name="choixDoc" method="get" action="genDoc.php">
.
.
<input type="checkbox" name="choixBE" value="Bien_etre.txt">Bien-être
<input type="checkbox" name="choixRE" value="Remise_en_forme.txt">Remise en forme
<input type="checkbox" name="choixSA" value="Sante.txt">Santé
.
.
<input type="Submit" value="Envoyer">
</form>
```

Lors de la soumission du formulaire, un script vérifie qu'une ou plusieurs cases sont cochées puis l'URL est transmise au script *genDoc.php*. Dans l'exemple ci-dessus, l'URL transmise lors de la soumission du formulaire sera :

http://www.thali.fr/genDoc.php?choixBE=Bien_etre.txt&choixSA=Sante.txt

Dossier 3

Les mini-excursions

- 3.1 A. Présenter la déclaration de la classe *MiniExcursionPayante* (la procédure *init()* paramétrée et les méthodes *get()* et *set()* de cette classe sont à déclarer mais leur code n'est pas à écrire).

Classe *MiniExcursionPayante* hérite de *MiniExcursion*

// Attributs privés

tarif : Réel

// Constructeur

procédure *init*(pNum : Entier, pLib : Chaîne, pNombrePlaces : Entier, pTarif : Réel)

// Méthodes publiques

fonction *getTarif*() : Réel

procédure *setTarif*(pTarif : Réel)

Fin Classe *MiniExcursionPayante*

- B. Écrire le corps de la méthode *estComplexe()* de la classe *MiniExcursionPlanifiée*

fonction *MiniExcursionPlanifiée* : *estComplexe*() : Booléen

Variables

nombrePlaces : Entier
bool : Booléen

Début

nombrePlaces \leftarrow laMiniExcursion.getNombrePlaces()

Si nombreInscrits = nombrePlaces Alors

 bool \leftarrow VRAI

Sinon

 bool \leftarrow FAUX

Fsi

retourner(bool)

Fin

Remarques :

- On peut également écrire la fonction en une seule instruction :
 retourner(nombreInscrits = laMiniExcursion.getNombrePlaces())
- D'autres solutions intermédiaires sont envisageables.

C. Écrire le corps de la méthode *ajouteEtape()* de la classe *MiniExcursion*

```

fonction MiniExcursion : ajouteEtape(e unNumEtape : Entier, uneDescription : Chaîne, uneDuréePrévue : Entier) : Booléen
Variables
    bool : Booléen
    i : Entier
    uneEtape : Etape
Début
    bool ← .VRAI ;    i ← 1
    Tantque (i <= lesEtapes.cardinal()) ET (bool)
        Si lesEtapes.extraire(i).getNumEtape() = unNumEtape
            Alors
                bool ← .FAUX
            Finsi
        FinTantQue
    Si bool
        Alors
            uneEtape.init(unNumEtape, uneDescription, uneDuréePrévue)
            lesEtapes.ajouter(uneEtape)
        FSi
    retourner(bool)
Fin

```

Pour cette question, on acceptera (avec ou sans usage d'une variable *bool*) :

```

Début
    uneEtape.init(unNumEtape, uneDescription, uneDuréePrévue)
    si lesEtapes.existe(uneEtape) alors
        retourner FAUX
    sinon
        lesEtapes.ajouter(uneEtape)
        retourner VRAI
    FSi
Fin

```

Même si cette solution ne répond pas exactement à la demande (le test ne portant pas uniquement sur la présence d'un numéro d'étape).

D. Écrire le corps de la méthode *donneDuréePrévue()* de la classe *MiniExcursion*.

```

fonction MiniExcursion : donneDuréePrévue() : Entier
Variables
    uneEtape : Etape
    duréeTot : Entier
    i : Entier
Début
    duréeTot ← 0
    Pour i ← 1 à lesEtapes.cardinal()
        uneEtape ← lesEtapes.extraire(i)
        duréeTot ← duréeTot + uneEtape.getDuréePrévue()
    FinPour
    retourner(duréeTot)
Fin

```

E. Écrire le corps de la méthode *donneDuréePrévuehhmm()* de la classe *MiniExcursion*

```

fonction MiniExcursion : donneDuréePrévuehhmm() : Chaîne
Début
    retourner(Temps.conversion_hhmm(donneDuréePrévue()))
Fin

```

3.2	<p>D'après le résultat de cette commande :</p> <ul style="list-style-type: none"> • indiquer la classe d'adresses (en précisant la façon de la déterminer) et l'adresse de réseau, • expliquer le rôle du masque, • préciser s'il s'agit d'une adresse publique ou privée.
-----	---

L'adresse IP 192.168.1.15 appartient à la classe C. L'écriture en base 2 du 1^{er} octet nous donne la valeur **1100 0000**. On constate que les 3 premiers bits sont 110 ce qui caractérise une adresse de classe C.

Le masque de réseau permet de savoir si une machine dont on connaît l'adresse IP est dans le réseau ou non.

Cette adresse est privée car elle appartient à la plage des adresses de classe C utilisables uniquement en interne ; cette plage d'adresses (qui va de 192.168.0.0 à 192.168.255.255 pour la classe C) est dite plage d'adresses privées.

Aucun paquet provenant de ces réseaux ou à destination de ces réseaux, ne sera routé sur internet.

Il existe également des plages d'adresses privées dans la classe A et dans la classe B.

3.3	<p>Proposer brièvement une solution matérielle permettant de répondre à la demande du centre. <i>La solution proposée devra également préciser l'équipement matériel dont devra disposer l'ordinateur portable ou l'assistant personnel d'un curiste.</i></p>
-----	---

La meilleure solution est d'offrir aux clients un accès *WiFi* à Internet et à l'intranet du centre.

Au niveau matériel, il faudra doter chacun des trois salons d'un ou de deux points d'accès *WiFi* (le nombre de points d'accès nécessaires dépend de la géométrie des salons).

L'ordinateur portable ou l'assistant personnel doivent être équipés d'une carte *WiFi*. Il est possible que la carte soit déjà intégrée.

On admettra une proposition de solution basée sur la technologie « courant porteur ».

Dossier 4	La boutique en ligne
------------------	-----------------------------

4.1	Écrire en langage SQL les requêtes permettant de :
-----	--

A. Obtenir la liste des produits (libellé de gamme, code produit, désignation, conditionnement, contenance) de la catégorie *Visage*.

```
Select G.libellé, P.code, désignation, conditionnement, contenance
From CATEGORIE C, GAMME G, PRODUIT P
Where C.libellé = 'Visage'
And C.code = G.codeCatég
And G.codeCatég = P.codeCatég
And G.numéro = P.numéroGamme
```

B. Obtenir les informations de la fiche produit A044 présentée en **annexe 4**.

```
Select désignation, descriptifComposition, descriptifProduit, conditionUtilisation, PrixEnVigueur('A044',
SysDate())
From PRODUIT
Where code = 'A044'
```

C. Déterminer le nombre de produits pour chaque gamme de la catégorie *Visage* (préciser le numéro et le libellé de la gamme). La liste est à présenter par nombres de produits décroissants

```
Select numéro, G.libellé, count(*)
From CATEGORIE C, GAMME G, PRODUIT P
Where C.libellé = 'Visage'
And C.code = G.codeCatég
And G.codeCatég = P.codeCatég
And numéro = numéroGamme
Group by numéro, G.libellé
```


D. Modifier le conditionnement du produit *B036* qui est maintenant commercialisé en *tube* et non plus en *pot*. Le prix est inchangé.

Update PRODUIT
 Set conditionnement = 'tube '
 Where code = 'B036'

4.2 Expliquer clairement le mécanisme à mettre en place. *Le code correspondant n'est pas à écrire.*

Il faut écrire un déclencheur (*trigger*) qui se déclenche sur insertion dans la table PRIXPRODUIT et effectue le travail demandé.

Principe de ce trigger (non demandé) : recherche de la dernière ligne (avec select count(*)) de la table PRIXPRODUIT pour le produit en question (donc avec code = :new.code si Oracle ou jointure avec la table inserted si SQLServer). Si le nombre retourné est différent de 0 et que la date de fin est à NULL alors on effectue un UPDATE pour alimenter la date de fin d'application avec SysDate().

4.3 Sur la base de 30 000 unités annuelles produites et vendues annuellement, déterminer le prix à partir duquel, l'entreprise commence à faire du bénéfice.

RÉSULTAT DIFFÉRENTIEL 2008			
	Calcul	Montant	Taux
Chiffre d'affaires	30000x	30000x	1
Charges variables	Coût unitaire		
Fucus	0,07	Soit	
Eau purifiée	0,03	0,5 *30 000	
Concentré aqueux de laminaires	0,1	Soit 15 000	
Sel Marin	0,3		
Amortissement	(30 000/150 000)*50 000	10 000	
Total charges variables		15 000+10 000	(15 000+10 000)/ 30 000x
MARGE SUR COÛT VARIABLE	30000x-15 000-10000	30000x-15000-10000	(30 000x-15 000-10 000)/30 000x
Charges fixes			
Charges de personnel	(35*150*12)*2	126000	
Charges administratives		14 000	
Total charges fixes		140 000	
Résultat		0	

$$30\,000x - 15\,000 - 10\,000 - 140\,000 = 0$$

$$30\,000x = 165\,000$$

$$x = 5,5$$

THALI doit vendre le produit à plus de 5,5 €/la boîte pour faire du bénéfice compte tenu des charges fixes et variables. Autrement dit le seuil de rentabilité en valeur est de 5,5 €/l'unité vendue.

4.4	Décrire le déroulement de la transaction.
-----	---

Le serveur web du Crédit Mutuel est sécurisé par SSL et possède une URL commençant par *https://*, où le "s" signifie *secured* (*sécurisé*).

Le serveur, à réception de la requête, envoie un certificat au client, contenant la clé publique du serveur, signée par une autorité de certification (CA).

Le client vérifie la validité du certificat (donc l'authenticité du marchand), puis crée une clé secrète aléatoire (plus exactement un bloc prétendument aléatoire), chiffre cette clé à l'aide de la clé publique du serveur, puis lui envoie le résultat (la clé de session).

Le serveur est en mesure de déchiffrer la clé de session avec sa clé privée. Ainsi, les deux entités sont en possession d'une clé commune dont ils ont seuls la connaissance. Le reste des transactions peut se faire à l'aide de la clé de session, garantissant l'intégrité et la confidentialité des données échangées.