

<b>E5 : Production et fourniture de services informatiques</b>
--

## SESSION 2017

<b>Cas AHM-23</b>
-------------------

### Proposition de corrigé

Mission 1	Évolution de l'architecture logicielle	20 points
	1.1	6
	1.2	4
	1.3	5
	1.4	5
	1.5	

Mission 2	Mise en place d'une gestion de stock centralisée	30 points
	2.1	7
	2.2	3
	2.3	20

Mission 3	Finalisation de l'application de saisie des livraisons de bois	20 points
	3.1	5
	3.2	8
	3.3	7

Mission 4	Valorisation des déchets	30 points
	4.1	7
	4.2	7
	4.3	8
	4.4	8

Total		100 points
-------	--	------------

**Mission 1 : Évolution de l'architecture logicielle**

(20 points)

**D1.1 Analyse de la demande**  
**D2.3 Gestion des problèmes et des changements**  
**D4.1 Conception et réalisation d'une solution applicative**  
• Conception ou adaptation d'une base de données

**Question 1.1 (6 points)**

Énumérer les composants logiciels type devant être installés pour la mise en production de l'application *web*.

La question amenait le candidat à ne pas confondre environnement de développement et environnement de production.

La mise en production d'une application *web* nécessite à minima:

- un logiciel serveur (serveur *web*) et éventuellement associé à un interpréteur de script ;
- un système de gestion de bases de données.

Un candidat qui cite :

- un logiciel de type FTP sera accepté
- un IDE sera pénalisé

On n'acceptera pas WampServer qui n'est pas un environnement de production.

Le navigateur web n'est pas exigé.

**Question 1.2 (4 points)**

Présenter les étapes nécessaires à la récupération des données depuis l'application tableur pour alimenter et valider le contenu de la nouvelle base de données.

La migration des données d'un tableur vers la base de données cible nécessite plusieurs étapes. On pourra accepter la solution suivante :

Étape 1 : Dans le tableur, exporter les données dans un format d'échange accepté par la base de données cible. Par exemple : format XML, CSV, JSON ou encore TXT.

Étape 2 : Dans le SGBD cible, créer une base de données ainsi que les tables et les contraintes de clé primaire et étrangères (étape 2 non exigée on peut interpréter que la base de données est déjà créé)

Étape 3 : Importer les données exportées précédemment dans le bon ordre afin de respecter les contraintes de clés étrangères.

Étape 4 : Vérifier si les données ont été importées correctement (par le contrôle du nombre d'enregistrements par exemple). Cette étape de vérification fonctionnelle est à valoriser.

*NB : On pourra aussi trouver comme solution le recours aux outils d'ETL (Extract, Transform, Load). Toute autre solution cohérente de type programme d'import/export, script/dump de base de données sera acceptée.*

**Question 1.3 (5 points)**

Proposer une solution à mettre en œuvre pour s'assurer qu'un chef d'atelier puisse modifier les informations de son site et que le comptable puisse consulter l'ensemble des informations des deux Esat.

Le candidat doit proposer une solution technique permettant de gérer :

- l'identification
- l'authentification
- l'habilitation

**Questions 1.4 et 1.5 à globaliser sur 5 points :****Question 1.4**

Présenter au comptable les arguments en faveur du maintien de cette solution de sauvegarde.

La sauvegarde incrémentielle se base sur la sauvegarde incrémentielle précédente alors que la sauvegarde différentielle se base sur la dernière sauvegarde complète.

La sauvegarde incrémentielle a donc l'avantage de **nécessiter moins d'espace de stockage** et d'être **plus rapide** car seuls sont sauvegardés les fichiers modifiés ou ajoutés depuis la dernière sauvegarde. Son inconvénient est de nécessiter un temps de restauration plus long.

**Question 1.5**

Indiquer comment utiliser les sauvegardes incrémentielles pour restaurer l'ensemble des données, en cas de perte des données contenues dans la base de données.

Pour restaurer la totalité des données, on procèdera en 2 temps :

1. restauration de la dernière sauvegarde complète ;
2. restauration de tous les jeux suivants de sauvegardes incrémentielles réalisés depuis la dernière sauvegarde complète en respectant l'ordre chronologique.

## Mission 2 : Mise en place d'une gestion de stock centralisée

(30 points)

### D1.1 Analyse de la demande

#### D4.1 Conception et réalisation d'une solution applicative

- Conception ou adaptation d'une base de données

#### D1.2 Choix d'une solution

- Définition des niveaux d'habilitation associés à un service
- Élaboration et présentation d'un dossier de choix de solution technique
- Évaluation des risques liés à l'utilisation d'un service

### Question 2.1 (7 points)

Écrire la vue demandée par le chef de projet permettant de répondre à ce besoin.

```
CREATE VIEW stockPalettes
```

```
AS
```

```
  SELECT SUM(quantiteCommandee) nbPalettes, SUM(quantiteCommandee*coutRevient)
  coutRevientTotal
  FROM Commande
  JOIN ligneCommande ON Commande.id = LigneCommande.idCommande
  JOIN ModelePalette ON ligneCommande.idModele = ModelePalette.id
  WHERE estFabrique = 1
  AND dateLivraisonReelle IS NULL
```

*Dans ce contexte c'est bien une vue SQL que l'on attend. On n'acceptera pas comme solution une vue HTML.*

### Question 2.2 (3 points)

Expliquer en quoi la création d'une vue est une solution pour répondre à ce besoin.

Bien que critiquable à certains égards, la création d'une vue va permettre, dans ce contexte d'urgence, de masquer l'aspect technique au profit de l'aspect métier de la base, la fonctionnalité utilisée par le comptable utilisera la vue comme s'il s'agissait d'une table :

```
SELECT *
FROM stockPalettes
```

Par ailleurs pour répondre à l'aspect de la sécurité il faudra restreindre l'utilisation de cette vue uniquement à la fonctionnalité utilisée par le comptable. Cette restriction pourra se faire par la création de plusieurs comptes d'accès, un seul ayant les droits de lecture sur cette vue.

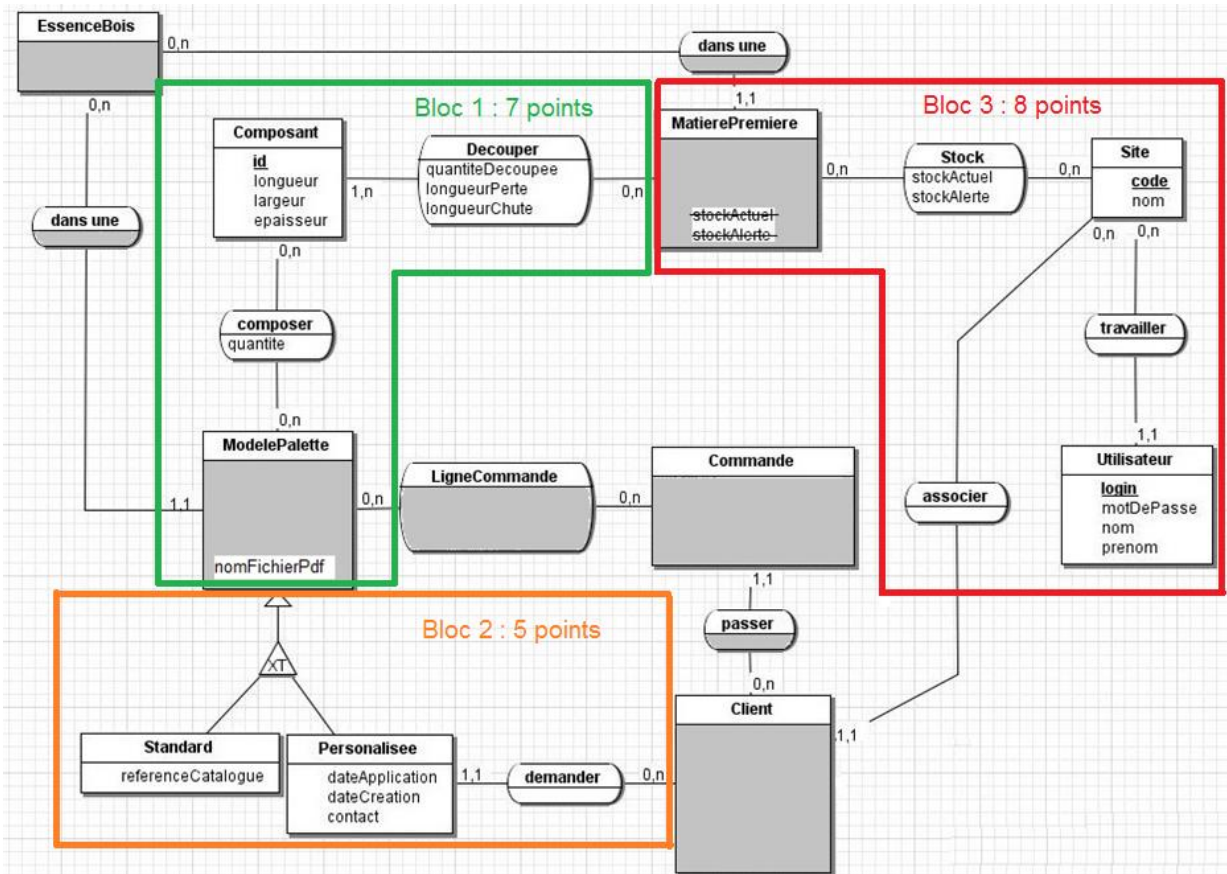
On pourra aussi trouver la solution qui consiste à donner le droit en lecture (SELECT) seulement à l'utilisateur comptable. Par exemple si le comptable s'appelle Dupond :

```
GRANT SELECT
ON stockPalettes
TO Dupond
```

**Question 2.3 (20 points)**

Proposer une modélisation de l'évolution de la structure de la base de données existante en intégrant la structure existante et les nouveaux besoins en information. Seuls les éléments nécessaires du schéma existant seront repris dans le schéma proposé

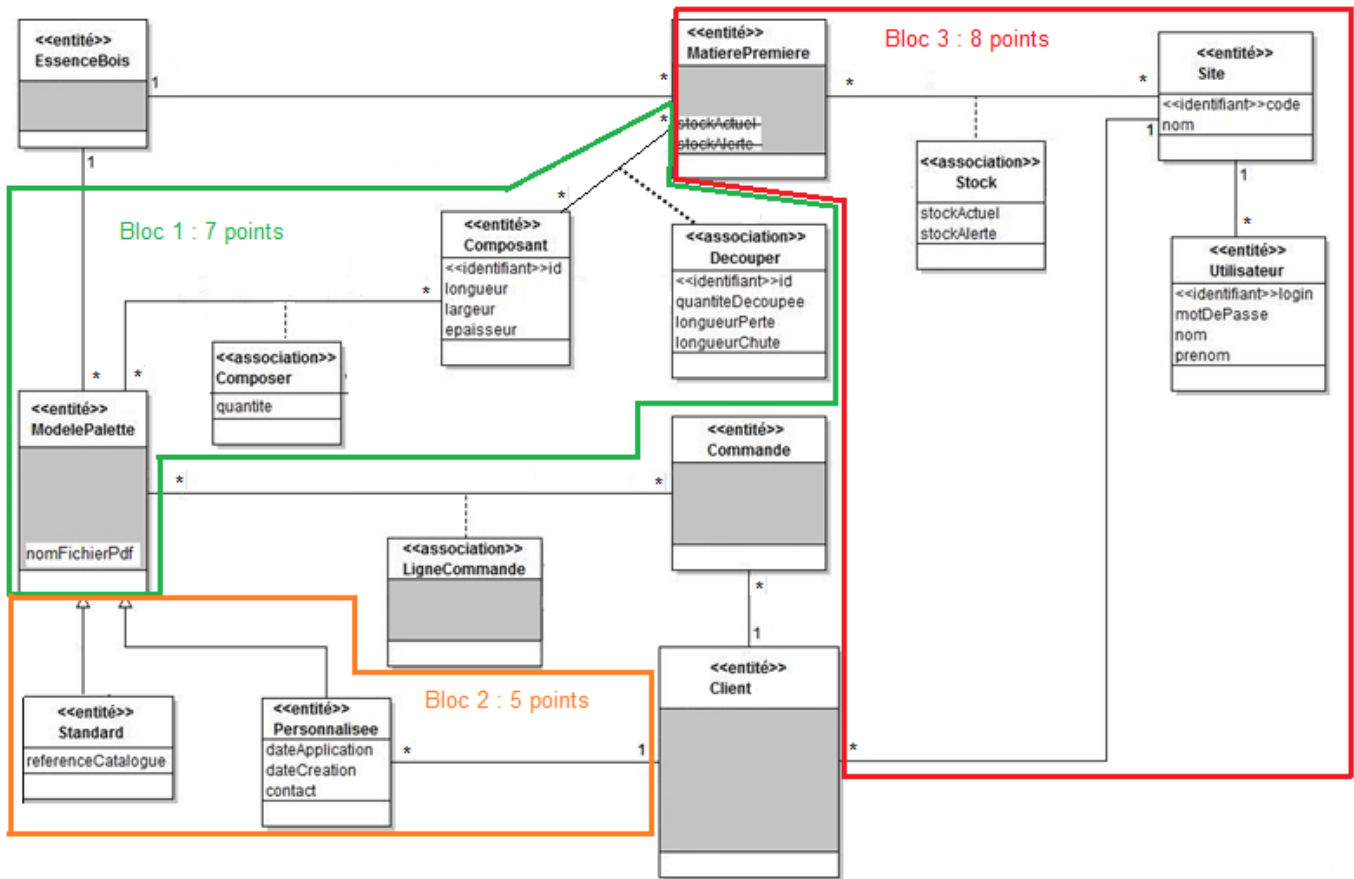
**Sous la forme d'un modèle entité-association**



Remarques :

- La quantitéDecoupee (donnée portée de l'association Decouper) pourra ne pas apparaître sur le schéma, cette donnée correspond au volume découpé (exprimé en m3) et peut donc être considérée comme une donnée calculée par certains candidats (à partir de la longueur, largeur et épaisseur du composant)
- A propos du contact :
  - L'attribut contact est normalement placé dans l'entité Personnalisee mais on acceptera son placement dans l'entité Client,
  - On pourra aussi trouver une entité Contact liée aux entités Personnalisee et Client (l'association demander disparaît alors)
- On pourra trouver la notion de chef d'atelier comme une spécialisation de l'entité Utilisateur ou comme attribut booléen dans l'entité Utilisateur.

Sous la forme d'un diagramme de classes



**Sous la forme d'un modèle relationnel :**

(Seules les nouvelles tables sont présentées)

**Site**(code, nom)

*Clé primaire : code*

**Utilisateur**(login, motDePasse, nom, prenom, codeSite)

*Clé primaire : login*

*Clé étrangère : codeSite en référence à code de Site*

**MatierePremiere**(id, libelle, longueur, largeur, epaisseur, idEssence)

*Clé primaire : id*

*Clé étrangère : idEssence en référence à id de EssenceBois*

**ModelePalette**(id, designation, longueur, largeur, poidsCharge, prixRevient, idEssence, type, referenceCatalogue, nomFichierPdf, dateApplication, dateCreation, contact, numeroClient)

*Clé primaire : id*

*Clé étrangère : idEssence en référence à id de EssenceBois*

*Clé étrangère : numeroClient en référence à numero de Client*

*Le champ type sera valorisé du caractère 'S' pour un modèle de palette standard et 'P' pour un modèle de palette personnalisé ;*

*Pour un modèle de palette standard les champs dateApplication, dateCreation, contact et numeroClient seront valorisés à NULL ;*

*Pour un modèle de palette personnalisé le champ referenceCatalogue sera valorisé à NULL.*

**Composant**(id, longueur, largeur, epaisseur)

*Clé primaire : id*

**Composer**(idModelePalette, idComposant, quantite)

*Clé primaire : idModelePalette, idComposant*

*Clés étrangères : idModelePalette en référence à id de ModelePalette  
idComposant en référence à id de Composant*

**Decouper**(idMatiere, idComposant, quantiteDecoupee, longueurPerte, longueurChute)

*Clé primaire : idMatiere, idComposant*

*Clés étrangères : idMatiere en référence à id de MatierePremiere*

*idComposant en référence à id de Composant*

**Stock**(codeSite, idMatiere, stockActuel, stockAlerte)

*Clé primaire : codeSite, idMatiere*

*Clés étrangères : codeSite en référence à code de Site*

*idMatiere en référence à id de Matiere*

**Client**(numero, raisonSociale, adresseRue, codePostal, ville, telephone, courriel, imageLogo, codeSite)

*Clé primaire : numero*

*Clé étrangère : codeSite en référence à code de Site*

## Mission 3 : Finalisation de l'application de saisie des livraisons de bois

(20 points)

### D4.1 Conception et réalisation d'une solution applicative

- Développement, utilisation ou adaptation de composants logiciels

#### Question 3.1 (5 points)

Justifier le choix de la société ALM-TIC pour l'application native, par rapport à d'autres types d'applications.

La zone de stockage de bois n'est pas couverte par le réseau Wi-Fi de l'association. Cette contrainte induit le développement d'une application qui ne pourra fonctionner qu'en mode déconnecté d'où le choix d'une application native.

Une autre solution aurait été de développer une application *web*, accessible depuis un navigateur, en HTML5 toujours en mode déconnecté ou alors en mode connecté ce qui aurait nécessité un accès internet et donc d'un abonnement pour la tablette (3G, 4G, connexion LoRa,...) générant une contrainte financière (solution non exigée du candidat)

#### Question 3.2 (8 points)

En vous inspirant du code implémentant le premier écran de l'application (affichage des commandes livrées), compléter à partir de la ligne 5 le code permettant d'alimenter le deuxième écran (affichage des lignes de la commande sélectionnée sur l'écran précédent).

```
// Parcours de la collection "lesLignes"
// Pour chaque objet LigneCommande de cette collection, stockage de sa description
// (type de la matière première et dimensions) dans la collection "lesDescriptions"
for (LigneCommande uneLigne : lesLignes)
{
    lesDescriptions.add(uneLigne.getLaMatierePremiere().toString());
}

// Création d'un objet "Adaptateur de données" qui va adapter les données de la
// collection des descriptions de commandes pour une ListView
// (paramètre "android.R.layout.simple_list_item_1")
ArrayAdapter<String> monAdaptateur = new ArrayAdapter<String> (this,
    android.R.layout.simple_list_item_1, lesDescriptions);

// Affectation, à la ListView (de nom "listeLignes"), d'une entête
// contenant le texte "Contenu commande" suivi du numéro de la commande
ListView listViewLignes = (ListView) findViewById(R.id.listeLignes);
TextView textView = new TextView(this);
textView.setText("Contenu commande n°" + numeroCommande);
listViewLignes.addHeaderView(textView);

// Affectation de l'objet Adaptateur à la ListView (de nom "listeLignes")
listViewLignes.setAdapter(monAdaptateur);
```



**Question 3.3 (7 points)**

Compléter, à partir de la ligne 4, le code exécuté au clic du bouton "+" du troisième écran, en suivant les consignes données par M. Laplace.

```
// Ajout de la quantité saisie à l'objet uneLigneCommande
uneLigneCommande.ajoutQuantiteMesuree(quantiteMesureeSaisie);

// Affectation à l'objet TextView tvTotalLivre du texte contenant la quantité totale livrée
TextView tvTotalLivre;
tvTotalLivre = (TextView) findViewById(R.id.totalLivre);
tvTotalLivre.setText("Quantite totale livrée : " +
    String.valueOf(uneLigneCommande.getQuantiteReelleLivree( ) + " m3");

// Mise à jour de la base de données par l'utilisation de la méthode statique miseAJour
// de la classe LigneCommandeDAO
LigneCommandeDAO.miseAJour(uneLigneCommande);
```

Remarque : on pénalisera de 1 point si on trouve l'instanciation de l'objet LigneCommandeDAO alors que la méthode miseAJour est une méthode statique.

**Mission 4 : Valorisation des déchets (30 points)****D4.1 Conception et réalisation d'une solution applicative**

- Développement, utilisation ou adaptation de composants logiciels

*Rappel : le candidat peut répondre en pseudo code ou dans le langage de son choix*

**Question 4.1 (7 points)**

Ecrire le constructeur de la classe AllumeFeu.

AllumeFeu (unId : entier, unLibellé : chaîne, unNbKilosParSac : entier, unPrixUnitaire : réel,  
unLibelEssence : Chaîne, uneValeurEnergétique : réel)

Début

```
Super(unId , unLibellé , unNbKilosParSac, unPrixUnitaire )
libelEssence ← unLibelEssence
valeurEnergétique ← uneValeurEnergétique
```

Fin

**Question 4.2 (7 points)**

Ecrire la méthode *totalCA* de la classe Période.

Fonction totalCA() :réel

Variables

```
uneCde : Commande
total : réel
```

Début

```
total ← 0
Pour chaque uneCde dans lesCommandes
    total ← total + uneCde.getMontantTotal()
FinPour
Retourner total
```

Fin

Remarque : on ne pénalisera pas un test complémentaire sur la date.

**Question 4.3 (8 points)**

Expliquer ce que retourne la méthode *ventesAllumeFeu* de la classe Période.

La méthode retourne un dictionnaire contenant comme clé le libellé de l'essence de bois et comme valeur le nombre de sacs d'allume-feux vendus.

Remarque : le terme « dictionnaire » n'est pas forcément attendu s'il est explicité.

**Question 4.4 (8 points)**

Ecrire la méthode *getPoidsTotal* de la classe Commande.

Fonction getPoidsTotal() : entier

Variables

```
lesTypesDéchet : Collection de TypeDéchet
unTypeDéchet : TypeDéchet
totalPoids : entier
```

Début

```
totalPoids ← 0
// parcours du dictionnaire lesLignes
lesTypesDéchet ← lesLignes.donnerLesClés()
Pour chaque unTypeDéchet dans lesTypesDéchet
    totalPoids ← totalPoids + unTypeDéchet.getNbKilosParSac() *
        lesLignes.donnerValeur(unTypeDéchet).getNbSacs()
// on multiplie par le conditionnement du sac pour avoir le poids
```

FinPour

Retourner totalPoids

Fin