# CAS WebCaisse

## Proposition de corrigé

Mission 1 : 30 points

Mission 2 : 25 points

Mission 3 : 30 points

Mission 4 : 15 points

### Mission 1 – Partie 1

**Question 1.1**

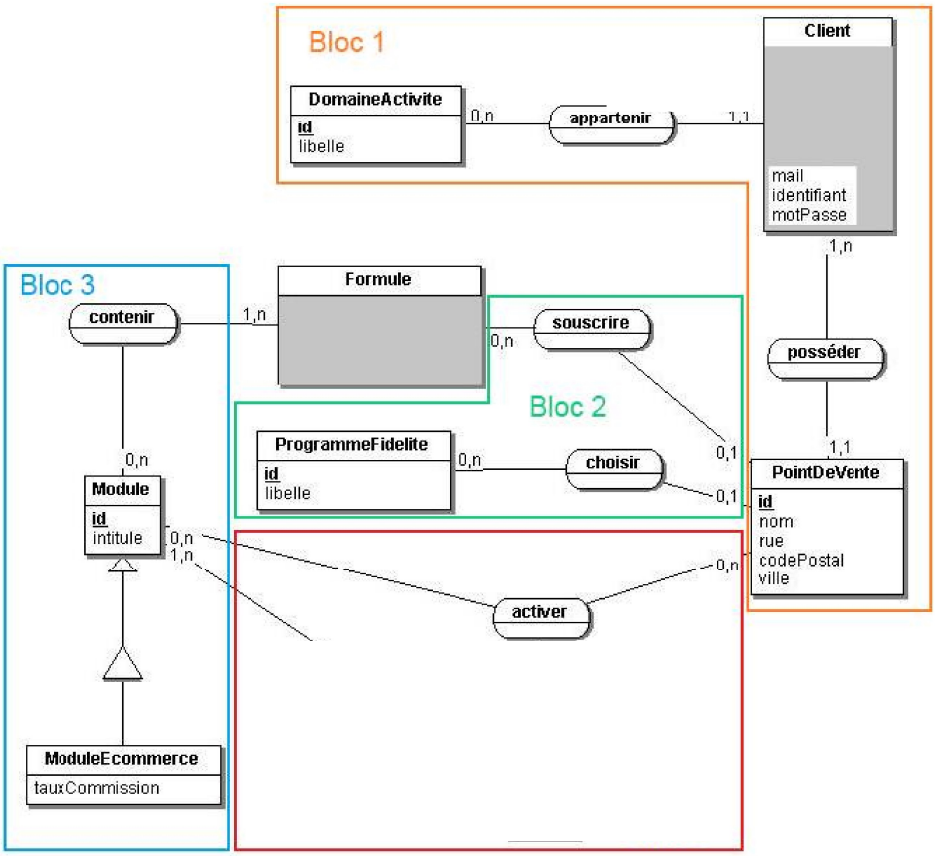
Quelles sont les risques de transmettre les identifiants de cette façon ? Proposez des solutions pour limiter les risques.

La première règle de base est de ne pas transmettre le mot de passe et le login en même temps. En effet, en cas d'interception le travail du hacker n'en est que simplifié. Il pourra alors utiliser le login et le mot de passe directement.

Il est préférable d'envoyer ces 2 informations par 2 système de communication différents (email et courrier …).
**(3 points)**

**Question 1.2**

Modifier la structure de la base de données utilisée par l'application ***AchatWebCaisse*** afin de permettre la souscription en ligne du logiciel *WebCaisse*.



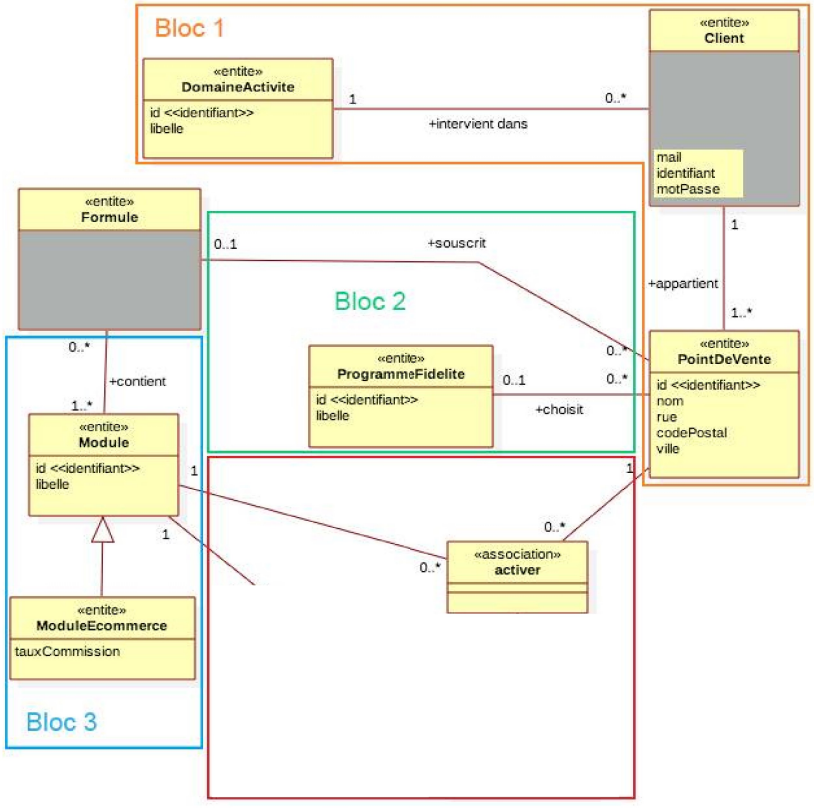



**Schéma relationnel**

Formule(id, libelle, prix, nbVendeursSimultanes)
*clé primaire : id*

Module(id, intitule)
*clé primaire : id*

Contenir(idFormule, idModule)
*clé primaire : idFormule, idModule*
*clés étrangères : idFormule en référence à id de Formule*
*idModule en référence à id de Module*

ModuleECommerce(id, tauxCommission)
*clé primaire : id*
*clé étrangère : id en référence à id de Module*

ProgrammeFidelite(id, libelle)
*clé primaire : id*

DomaineActivite(id, libelle)
*clé primaire : id*

Client(id, raisonSociale,rue, codePostal, ville, telephone, mail, identifiant, motPasse, idActivite)
*clé primaire : id*
*clé étrangère : idActivite en référence à id de DomaineActivite*

PointDeVente(id, nom, rue, codePostal, ville, idClient, idFormule, idProgrammeFidelite)
*clé primaire : id*
*clés étrangères : idClient en référence à id de Client*
*idFormule en référence à id de Formule*
*idProgrammeFidelite en référence à id de ProgrammeFidelite*

Activer(idPointDeVente, idModule)
*clé primaire : idPointDeVente, idModule*
*clé étrangères : idPointDeVente en référence à id de PointDeVente*
*idModule en référence à id de Module*

**Bloc 1** : 5 points
**Bloc 2** : 3 points
**Bloc 3** : 4 points
**Bloc 4** : 1 points **(Total 13 points)**

## Mission 1 – Partie 2

**Question 1.3**

Indiquez si le code d'interrogation de la base de données est sûr ou alors expliquez quelles seraient les modifications à apporter dans le cas contraire.

Il n'y a aucun contrôle sur les données utilisées et saisies par l'utilisateur. Il faut absolument contrôler ces informations pour éviter les tentatives d'injection de code javascript ou bien d'injection sql.
Il faut mettre en place un contrôle en utilisant des expressions rationnelles.
L'interrogation de la base de données est réalisée par une requête SQL simple. Il est alors possible par injection SQL de récupérer des données non prévues ou d'altérer les données.
Il faudrait utiliser une requête paramétrée qui bloquera l'utilisation d'injection SQL.
**(4 points)**

**Question 1.4**

Concluez sur la façon de stocker les mots de passe au regard du code observé.

Utiliser la méthode statique Hash.SHA256("mot") produit un hash de "mot" en utilisant l'algorithme cryptographique SHA256. Le mot de passe cair ne peut pas être retrouvé à partir de l'empreinte SHA256 stockée en base. Par contre comme aucun salage n'est effectué on note 2 inconvénients majeurs :
2 comptes avec le même mot de passe clair ont la même empreinte donc en connaissant l'un on connaît l'autre.
Un hackeur, s'il récupère les empreintes peut utiliser des "rainbow tables" pour retrouver les mots de passe en clair.
Il faut donc ajouter un salage avant hachage SHA256 pour plus de sécurité.
**(4 points)**

**Question 1.5**

Expliquer en quoi la structure de la table ne permettra pas de gérer l'historique des formules souscrites, indispensable à la détermination du montant mensuel à régler par le client.

La table ne contient pas la date à laquelle le changement de formule a eu lieu.

**(2 points)**

**Question 1.6**

Proposer une correction de la structure de la table qui réponde au besoin exprimé.

**FormuleSouscrite(idPointDevente, dateSouscription, idFormule)**
clé primaire : idPointDeVente, dateSouscription
clé étrangère : idFormule en référence à id de Formule
idPointDeVente en référence à id de PointDeVente

On acceptera également :

## Mission 4 – Partie 1

**Question 4.1 :**

Où devrait être placé le SGBD pour un maximum de sécurité ? Justifiez.

La DMZ est accessible depuis l'internet. Le SGBD est donc accessible à d'éventuels hackeurs. Une solution est de placer le SGBD dans le réseau LAN de l'entreprise et permettre au serveur Web d'accéder à cette seule machine dans le LAN.
Une meilleure solution serait de créer un deuxième réseau DMZ (DMZ privée) et d'y placer le SGBD. Cette solution évite d'ouvrir un port du pare-feu vers le LAN, risquant de provoquer une brèche.

**(5 points)**

**Question 4.2 :**

Critiquez le code et proposez les modifications qui permettraient d'améliorer la sécurité.

La directive ALL donne toutes les permissions au compte acces_webcaisse à la base webcaisse_db.

Ce qui inclut les droits de création de base, modification de la structure, sauvegarde, etc.

Alors que seuls les permissions correspondant aux requêtes SQL SELECT, INSERT, UPDATE et DELETE devraient être autorisées.

Donc on pourrait plutôt écrire :

GRANT SELECT, UPDATE, DELETE, INSERT ON webcaisse_db TO acces_webcaisse;

**(5 points)**

## Mission 4 – Partie 2

**Question 4.3**

Rédiger une courte note à destination du chef de projet en justifiant le choix d'un hébergeur parmi les trois propositions.

Je vous transmets le comparatif des solutions d'hébergement et je conseille de choisir l'hébergeur ***A*** qui répond aux trois critères de *Nasdy*, à savoir :

- une sauvegarde en temps réel et une sauvegarde effectuée tous les lundis soirs
- une disponibilité de 99,9 %
- un chiffrement des échanges avec un certificat SSL

L'hébergeur ***B*** propose une sauvegarde journalière et la disponibilité mais pas de chiffrement, mais le client peut le mettre en place car son serveur est dédié.

L'hébergeur ***C*** ne propose que la possibilité de faire une sauvegarde manuelle et la disponibilité avec une garantie du temps de disponibilité mais pas de chiffrement.

**(5 points)**