

Documents Autorisés

Exercice 1

Une base de données est utilisée pour représenter les clients d'une très grande entreprise. Chaque client est décrit par les attributs suivants :

NomC : Identification du client **NomF** : Nom de famille du client **Adr1** : 1^{er} adresse du client
Adr2 : 2^{ème} adresse du client **VilleC** : Ville du client **Cp** : code postal du client

Soit F1 l'ensemble de dépendances fonctionnelles suivant:

1. **NomC** → **NomF**, **Adr1**, **Adr2**, **VilleC**, **Cp**
2. **Adr1**, **Adr2** → **VilleC**
3. **Cp** → **VilleC**

Q1.1. Interpréter les dépendances (2) et (3)

Q1.2. Construire un schéma en troisième forme normale correspondant à l'ensemble des dépendances fonctionnelles de F1.

Q1.3. Illustrer les anomalies de mises à jour dans la relation initiale **Client**. Le schéma normalisé élimine-t-il ces anomalies ? Justifier votre réponse.

Soit F2 un ensemble de dépendances fonctionnelles obtenu en supprimant la dépendance fonctionnelle 2 de l'ensemble initial F1. On obtient

1. **NomC** → **NomF**, **Adr1**, **Adr2**, **VilleC**, **Cp**
3. **Cp** → **VilleC**

Q1.4. Interpréter cette suppression.

Exercice 2

Considérons un système de gestion de bases de données acceptant des transactions. Ces transactions sont réalisées par des utilisateurs qui consultent et modifient une base de données comprenant les tables **R**, **S**, **T** et **W**.

On se propose de définir une politique de transactions dans la situation d'utilisation suivante :

- les utilisateurs modifient les **N-tuplets** de **R**, **S** et **T**
- les utilisateurs accèdent (lisent) les **n-tuplets** de **R**, **S**, **T** et **W**.

De plus :

- aucune modification de **R** ne peut intervenir lorsque **T** est en cours de modification.
- la modification de **R** entraîne une modification de **W** car certaines valeurs des attributs de **W** dépendent de valeurs des attributs de **R**

Questions

Q2.1. Identifier les problèmes engendrés par les lectures et modifications ainsi définies.

Q2.2. Définir les types de verrou puis les verrous à poser pour que cette politique de transactions soit mise en œuvre et conserve la cohérence de la base de données.

Q2.3. Définir deux transactions différentes possibles ainsi que leur déroulement. Vous donner une transaction de lecture sur les relations ainsi qu'une transaction de modification. Pour décrire ces transactions, on utilisera la notation (**update**, **read**, **write**, **commit**, etc.) vues en cours.

Exercice 3

Q3.1. Définir un diagramme entité-association répondant au cahier des charges suivant :

- Le conservateur d'une grande ville souhaite gérer les musées et leurs œuvres. Chaque musée est caractérisé par un nom et une adresse. Un musée possède plusieurs œuvres (art contemporain, antiquités, etc.) ;

- Chaque œuvre du musée a un titre, un ou plusieurs auteurs, une date d'acquisition et un identifiant,

- Une œuvre est exposée dans l'une des salles du musée (caractérisée par un numéro, son nom, le nombre d'œuvres, sol, éclairage) ;

- Certaines œuvres exposées dans le musée peuvent avoir été empruntées à un autre musée. Dans ce cas, on connaît son titre, son (ou ses) auteur(s), la date et la durée de l'emprunt.

- De plus, l'œuvre doit alors être assurée. On veut savoir la valeur pour laquelle l'œuvre est assurée, le nom et l'adresse de la compagnie qui l'assure;

- Le conservateur souhaite avoir des informations sur les musées qui ont prêté ou qui sont susceptibles de prêter des œuvres.

Remarque

Vous pouvez compléter le texte partout où vous pensez apporter une précision. Il faudra le noter explicitement sur la copie.

Exercice 4

Considérons le modèle logique suivant permettant de stocker les utilisateurs d'un forum et les messages qu'ils ont écrits. Nous supposons qu'un message peut être écrit par plusieurs utilisateurs.

```
Utilisateur (
    id : integer,
    name : varchar )
```

```
Message (
    id : integer,
    titre : varchar )
```

```
Auteurs (
    idUtilisateur integer,
    idMessage integer,
    Foreign key idUtilisateur references Utilisateur(id),
    Foreign key idMessage references Message(id) )
```

Q4.1. Donner un modèle orienté objets associé à ce modèle logique. On pourra utiliser pour cela un diagramme de classes UML sans préciser les méthodes associées aux classes.

Q4.2. Décrire une interface de programmation (API) associée au modèle orienté objets obtenu en question 1.

Q4.3. Définir une table de correspondances entre les tables et/ou attributs du modèle logique précédent et les classes et/ou attributs du modèle orienté objets obtenus en question 1.

Q4.4. Considérons la requête suivante « *Donner la liste des identifiants de messages rédigés par l'auteur de name 'TOTO'.* ». On suppose que cette requête est implantée dans le modèle orienté objets obtenu en question 1. En utilisant la table de correspondances précédente, décrire les accès aux tables du modèle logique réalisés par cette méthode.