Base de Données Avancées B
Enseignant : Pierre Parrend
Licence Professionnelle CE-Stat
06.03.2006

TD4 - Autoformation 2

1 Formes normales

1.1 Exercice 1

Considérez la relation suivante :

VenteVoiture(NumVoiture, DateVente, NumVendeur, Commission, Réduction).

On suppose que chaque voiture (type de voiture) peut être vendue par plusieurs vendeurs, et que par conséquent {NumVoiture, NumVendeur} soit la clé primaire.

Les autres dépendances sont :
DateVente -> Réduction,
NumVendeur -> Commission.

Compte tenu de la clé primaire donnée, la relation est-elle en 1NF, en 2NF, ou en 3NF ? Expliquez.

Comment procéder pour la normaliser complètement (c'est à dire la passer en BCNF) ?

Réponse La seule clé candidate possible est (numVoiture, numVendeur).
La relation est en 1 NF (forme normale 1).
En effet :

1. Les attributs ont des valeurs simples (critères de 1NF respectés),
2. Une DF (dépendance fonctionnelle) existe entre numVendeur et Commision, c'est à dire entre une sous-partie de la clé et un attribut (critères de 2FN non respecté),
3. De plus, une DF existe entre DateVente et Commission, c'est à dire entre deux (ensembles d') attributs ne faisant pas partie de la clé candidate (critère de 3FN non respecté).

Le passage en BCNF (Forme Normale de Boyce-Codd) consiste à isoler chaque dépendance fonctionnelle dans un schéma relationnel :

- reduction(DateVente, Réduction),
- vendeur(NumVendeur, Commission),
- vente(NumVoiture, NumVendeur, DateVente).

1.2 Exercice 2

Considérez la relation suivante :
Livre (Titre, Auteur, CategorieLivre, Prix, AffiliationAuteur, Editeur).
Les dépendances suivantes sont valables :
Titre -> Editeur, CategorieLivre
CategorieLivre -> Prix
Auteur -> AffiliationAuteur

1. Quelle est la forme normale dans laquelle se trouve cette relation ? Expliquez.
2. Effectuez la normalisation, jusqu'à ce que vous ne puissiez plus décomposer les relations. Donnez les raisons de chaque décomposition.
3. A quelle forme normale se conforme maintenant les relations ?

Réponses La seule clé candidate possible est l'attribut Titre.
Forme normale de la relation :
2NF. En effet :
En effet :

1. Les attributs ont des valeurs simples (critères de 1NF respectés),
2. La clé candidate est composée d'un seul attribut. Par conséquent, aucun sous-ensemble de la clé candidate ne peut être source de DF. Pat conséquent, la relation est en 2NF,
3. De plus, une DF existe entre CatégorieLivre et Prix, entre Auteur et AffiliationAuteur, c'est à dire entre deux (ensembles d') attributs ne faisant pas partie de la clé candidate (critère de 3FN non respecté).

Le passage en 3NF consiste à isoler les DF ayant pour source des attributs hors de la clé candidate :

- livre(Titre, Auteur, Editeur, CatégorieLivre),
- prixLivre(CategorieLivre, Prix),
- auteur(Auteur, AffiliationAuteur).

Les schémas relationnels résultants respectent également les critères de la BCNF (Car une seule clé candidate existe).

2 Etude de Cas

Vous proposerez une solution à la situation suivante, en veillant à ce que vos tables de données soient conformes à la forme normale de Boyce-Codd (BCNF).

2.1 Enoncé

Réalisez la conception de la Base de Données correspondant à la situation suivante, à l'aide de la méthodologie UML:

- Diagrammes de Cas d'Utilisation,
- Diagrammes de Classes,
- Schéma relationnel des tables de données
- Requêtes SQL de création de ces tables.

Vous devez réaliser une Base de Données d'utilisateurs pour une bibliothèque. Les informations de cette Base de Données sont :

- les fiches d'identité des utilisateurs (Nom, Prénom, Age, Adresse, numéro de téléphone, date de premier emprunt),
- les données concernant les livres (Auteur, Titre, Edition, Identifiant dans le catalogue de la bibliothèque),
- les emprunts (Utilisateur, Livres empruntés).

2.2 Questions

1. Analysez maintenant vos relations afin de vérifier la forme normale dans laquelle elles se trouvent.
2. Quelle requête SQL permet de trouver l'ensemble des livres empruntés (actuellement et auparavant) pour un utilisateur ?
3. Quelle requête SQL permet pour un livre de savoir s'il est disponible ou s'il est emprunté (attention il peut exister plusieurs exemplaires d'un livre dans le catalogue de la bibliothèque)?

2.3 Réponses

2.3.1 Diagramme de Cas d'utilisation : Analyse

Acteurs Les acteurs de ce scénario sont :

- Les bibliothécaires, qui gèrent les emprunts d'ouvrages,
- Les utilisateurs de la bibliothèques, qui peuvent consulter le catalogue pour savoir si les ouvrages qu'ils cherchent sont disponibles.

Systèmes Nous nous situons dans une architecture d'application web classique. Les sytèmes sont donc :

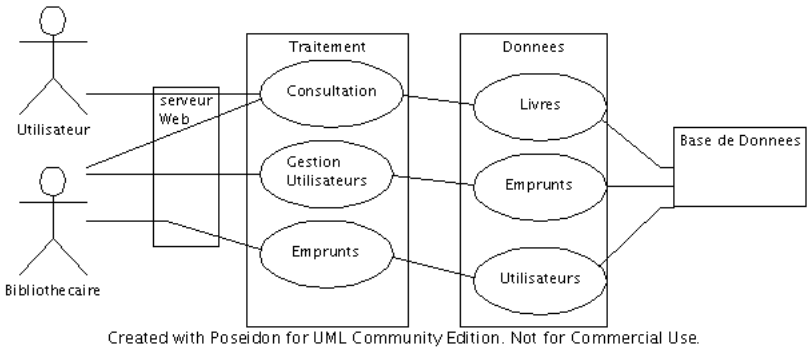
- le serveur web,
- le serveur de traitement,
- le serveur de gestion des données,
- la base de données.

Cas d'utilisation Les cas d'utilisation sont les suivants :

- un utilisateur consulte le système, pour chercher un ouvrage donné,
- un(e) bibliothécaire ajoute un utilisateur,
- un(e) bibliothécaire réalise l'emprunt d'un ouvrage pour un utilisateur.

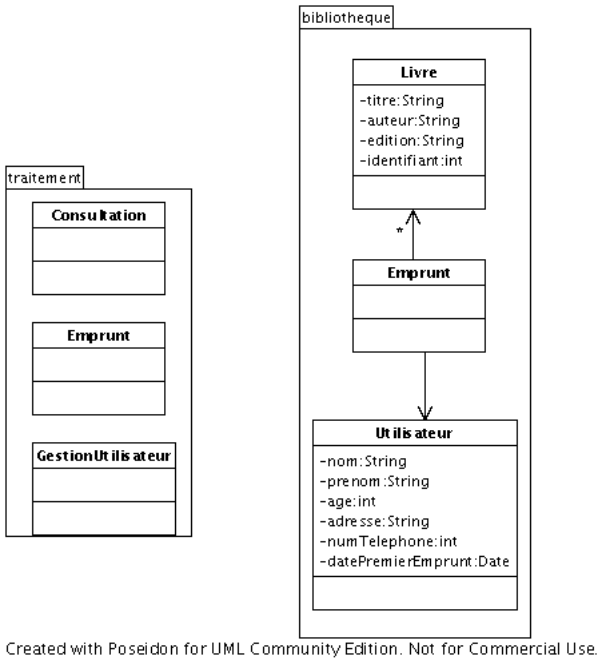
2.3.2 Diagramme de Cas d'utilisation

Le diagramme suivant est le diagramme de Cas d'Utilisation correspondant à l'analyse que nous venons de réaliser :



2.3.3 Diagramme de Classes

Le diagramme suivant est le diagramme de Classes correspondant au diagramme de Cas d'Utilisation précédent, complété par les informations données par l'énoncé concernant les caractéristiques des différents types de données.



2.3.4 Schémas relationnels

Les schémas relationnels correspondant à ce diagramme de Classe sont les suivants :

- Livre(titre, auteur, edition, identifiant),
- Emprunt(utilisateur, emprunt),
- Utilisateur(nom, prenom, age, adresse, numTelephone, dataPremierEmprunt).

2.3.5 Analyse des formes normales

Schéma relationnel 'Livre'. Les clés candidates sont :

- le titre et l'auteur (on considère qu'un auteur n'écrit qu'un seul ouvrage avec un titre donné),
- l'identifiant.

Les dépendances fonctionnels sont :

1. identifiant → (titre, auteur),
2. titre, auteur→ edition.

La situation la plus simple est d'avoir une clé primaire composée d'un seul attribut. On choisit donc 'identifiant'.

- Validation des critères de 1NF : pas d'atributs à valeur multiple. La table est donc en 1NF.
- Validation des critères de 2 NF : un seul attribut compose la clé. Il ne peut donc pas exister de DF entre un sous-ensemble de la clé et un (ou plusieurs) attributs. La table est donc en 2NF.
- Validation des critères de 3NF : il existe une DF entre (titre, auteur) et édition. La table n'est donc pas en 3NF.

Afin d'obtenir des schémas relationnels conformes à la 3NF, on sépare les DF dans des tables de données distinctes :
livre(identifiant, #titre, #auteur),
edition(titre, auteur, edition).
Comme il existe maintenant une seule DF par table, les schémas relationnels proposés sont également conforme à la BCNF.

Schéma relationnel 'Emprunt'. La clé candidate est utilisateur.
La dépendance fonctionnelle est :
utilisateur→ livre.
Il existe une seule clé candidate. Par conséquent, le choix de la clé primaire est direct.

- Validation des critères de 1NF : pas de champs à valeur multiple.La table est donc en 1NF.
- Validation des critères de 2 NF : un seul attribut compose la clé. Il ne peut donc pas exister de DF entre un sous-ensemble de la clé et un (ou plusieurs) attributs. La table est donc en 2NF.
- Validation des critères de 3NF : il existe un seul attribut hors de la clé. Il ne peut donc y avoir de DF entre deux attributs hors de la clé. Le schéma relationnel respecte donc la 3FN.

De plus, comme une seule clé candidate existe, la BCNF est également respectée.

Schéma relationnel 'Utilisateur'. La seule clé candidate possible est (Nom, Prenom). En effet, les autres attributs à valeur fixe, comme le numéro de téléphone, peuvent être amenés à évoluer.
Les DF sont :

- Nom, Prenom → adresse, datePremierEmprunt,
- adresse → numeroDeTelephone.

Analyse du type de normalisation de la Base de Données

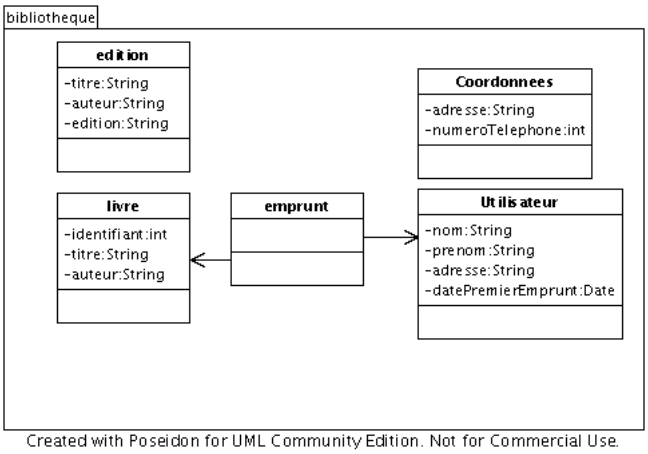
- Validation des critères de 1NF : pas de champs à valeur multiple.La table est donc en 1NF.
- Validation des critères de 2 NF : aucun sous-ensemble de la clé n'est source de dépendance fonctionnelle. Par conséquent, la table est donc en 2NF.
- Validation des critères de 3NF : il existe une DF entre deux attributs hors de la clé. Par conséquent, le schéma relationnel ne respecte pas la 3eme Forme Normale.

Afin d'obtenir des schémas relationnels conformes à la 3NF, on sépare les DF dans des tables de données distinctes :

- Utilisateur (Nom, Prenom, adresse, datePremierEmprunt),
- Coordonnées (adresse, numeroDeTelephone).

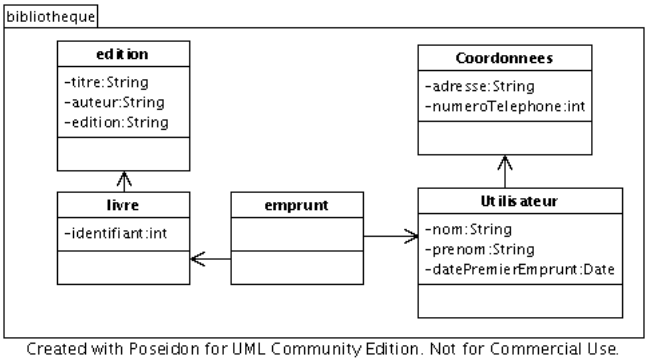
2.3.6 Diagramme de Classes modifié

Les analyses réalisées dans la question précédente conduisent à la réécriture du Diagramme de Classe représentant le système. Le système modifié est la bibliothèque.



On observe dans ce Diagramme de Classes des redondances de données, qui peuvent être simplifiées par le biais d'associations entre classes.

La modification du Diagramme de Classes conduit aux diagramme suivant :



Attention : Ces modifications concernent les classes, et donc le programme. Les relations de données restent inchangées.

2.3.7 Requêtes SQL de création des tables

Les requêtes SQL de création des tables de données sont, d'après les schémas précédents :

Création de la table Edition

```
CREATE TABLE Edition

(

 titre varchar(50) NOT NULL,

 auteur varchar(50) NOT NULL,

 edition varchar(50),

 Primary Key(titre, auteur)

);
```

Création de la table Livre

```
CREATE TABLE Livre

(

 identifiant int(8) NOT NULL,
```

```
 titre varchar(50) NOT NULL,
 auteur varchar(50) NOT NULL,
 Primary Key(identifiant),
 Foreign Key(titre,auteur)
);
```

Création de la table Emprunt

```
CREATE TABLE Emprunt
(
 idLivre int(8) NOT NULL,
 nomUtilisateur varchar(50) NOT NULL,
 prenomUtilisateur varchar(50) NOT NULL,
Primary Key(ideLivre, nomUtilisateur, prenomUtilisateur),
Foreign Key(idLivre),
Foreign Key (nomUtilisateur, prenomUtilisateur)
);
```

Création de la table Utilisateur

```
CREATE TABLE Utilisateur
(
 nom varchar(50) NOT NULL,
 prenom varchar(50) NOT NULL,
 adresse varchar(50),
 datePremierEmprunt int(8),
```

```
 Primary Key (nom, prenom)
);
```

Création de la table Coordonnees

```
CREATE TABLE Coordonnees
(
 adresse varchar(50) NOT NULL,
 numeroDeTelephone int(10),
 Primary Key (adresse)
);
```

2.3.8 Requêtes SQL d'utilisation des tables

Ensemble des libres empruntés par un utilisateur On pose 'nomUtil' la variable permettant d'identifier le nom de l'utilisateur recherché, dans la requête paramétrique.

```
Select Livre.identifiant, Livre.titre, Livre.auteur
From Livre, Emprunt, Utilisateur, Coordonnees
Where Utilisateur.nom = 'nomUtil'
      and Emprunt.idLivre = Livre.IDLivre
      and Emprunt.nomUtilisateur = Utilisateur.nom
      and Emprunt.prenomUtilisateur = Utilisateur.prenom
      and Utilisateur.adresse = Coordonnees.adresse
```

Disponibilité d'un livre On pose 'titre' la variable permettant d'identifier le nom du livre recherché, 'auteur' la variable identifiant le nom de son auteur.

```
Select Livre.identifiant, Emprunt.identifiant
from Livre, Emprunt
where Livre.titre = 'titre'
      and Livre.auteur = 'auteur'
      and Emprunt.titre = Livre.titre
```